3 Direkte Lösung Linearer Gleichungssysteme

Wir schreiben lineare Gleichungssysteme in der Form

$$\boldsymbol{A}x = b; \tag{3.0.1}$$

hier ist $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ eine reguläre Matrix, $b \in \mathbb{R}^n$ ist gegeben, und $x \in \mathbb{R}^n$ ist die gesuchte Lösung. Die Matrix $\boldsymbol{A}$ und die Vektoren x, b haben die Komponenten

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{nn} \end{pmatrix}, \qquad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix}, \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}.$$

3.1 Gauss'sche Eliminationsmethode (GEM)

3.1.1 Dreiecksmatrizen

Wir betrachten zuerst einmal zwei Spezialfälle von Matrizen $\boldsymbol{A}$. Wir sagen, dass $\boldsymbol{A}$ eine *linke Dreiecksmatrix* (oft auch: *untere Dreiecksmatrix*) ist, falls

$$a_{ij} = 0 \qquad \text{für alle } i,\, j \text{ mit } i < j.$$

Analog sprechen wir von $\boldsymbol{A}$ als einer *rechten Dreiecksmatrix* (oft auch: *obere Dreiecksmatrix*), falls

$$a_{ij} = 0 \qquad \text{für alle } i,\, j \text{ mit } i > j.$$

Linke Dreiecksmatrizen werden meist mit $\boldsymbol{L}$ bezeichnet und rechte Dreiecksmatrizen mit $\boldsymbol{R}$. Die Namensgebung ist aus der Struktur der Matrix leicht verständlich:

$$\boldsymbol{L} = \begin{pmatrix} a_{11} & 0 & \cdots & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & \\ a_{31} & a_{32} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ a_{n1} & a_{n2} & \cdots & a_{nn-1} & a_{nn} \end{pmatrix}, \qquad \boldsymbol{R} = \begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & \cdots & a_{2n} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix}.$$

(Im englischsprachigen Raum werden Matrizen dieses Typs typischerweise mit $\boldsymbol{L}$ und $\boldsymbol{U}$ für *lower* und *upper* bezeichnet.) Hat die Matrix $\boldsymbol{A}$ in (4.1.1) linke oder rechte Dreiecksgestalt, dann ist das Lösen des Gleichungssystems besonders einfach. Bei Lösen von

$$\boldsymbol{L}x = b$$

spricht man von *Vorwärtssubstitution* und beim Lösen von

$$\boldsymbol{R}x = b$$

spricht man von *Rückwärtssubstitution*. Die Namensgebung erfolgt aus der Tatsache, dass man beim Lösen von $\boldsymbol{L}x = b$ die Unbekannten x_i sukzessive "vorwärts" bestimmt d.h. zuerst

$x_1 = b_1/a_{11}$, mit dessen Hilfe man x_2 bestimmt, dann x_3 u.s.w. Bei Lösen von $\boldsymbol{R}x = b$ werden die Unbekannten x_i sukzessive "rückwärts" bestimmt, d.h. zuerst $x_n = b_n/a_{nn}$, dann damit x_{n-1}, dann x_{n-2} u.s.w. Dieses Vorwärts- und Rückwärtseinsetzen formalisieren wir in den folgenden zwei Algorithmen

Algorithmus 3.1.1 (Vorwärtssubstitution) *Sei $\boldsymbol{A}$ eine Linksdreiecksmatrix mit $a_{ii} \neq 0$ für $i = 1, \ldots, n$. Dann kann die Lösung x von $\boldsymbol{A}x = b$ wie folgt berechnet werden:*
`for` i `from` 1 `to` n `do` {

$$x_i := \frac{1}{a_{ii}} \left(b_i - \sum_{k=1}^{i-1} a_{ik} x_k \right) \qquad \texttt{(Konvention: leere Summe} = 0\texttt{)}$$

}

Algorithmus 3.1.2 (Rückwärtssubstitution) *Sei $\boldsymbol{A}$ eine Rechtsdreiecksmatrix mit $a_{ii} \neq 0$ für $i = 1, \ldots, n$. Dann kann die Lösung x von $Ax = b$ wie folgt berechnet werden:*
`for` i `from n to` 1 `by` -1 `do` {

$$x_i := \frac{1}{a_{ii}} \left(b_i - \sum_{k=i+1}^{n} a_{ik} x_k \right) \qquad \texttt{(Konvention: leere Summe} = 0\texttt{)}$$

}

Man überzeugt sich leicht davon, dass in beiden Algorithmen für jedes i auf der rechten Seite der Zuweisung Objekte stehen, die bereits in einem vorangehenden Schritt bestimmt wurden.

Theorem 3.1.3 *Seien $\boldsymbol{L}_k \in \mathbb{R}^{n \times n}$, $k = 1, \ldots, n-1$ Linksdreiecksmatrizen von der Form*

$$\boldsymbol{L}_k = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & l_{k+1\,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & l_{n\,k} & & & 1 \end{pmatrix}.$$

Dann ist

$$\boldsymbol{L}_k^{-1} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1\,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{n\,k} & & & 1 \end{pmatrix}.$$

Ferner hat das Produkt $\boldsymbol{L}_1\boldsymbol{L}_2\cdots\boldsymbol{L}_{n-1}$ die Darstellung

$$\boldsymbol{L}_1\boldsymbol{L}_2\cdots\boldsymbol{L}_{n-1} = \begin{pmatrix} 1 & & & & & \\ l_{21} & \ddots & & & & \\ l_{31} & \ddots & 1 & & & \\ \vdots & & l_{k+1\,k} & 1 & & \\ \vdots & & \vdots & \ddots & \ddots & \\ l_{n1} & l_{n2}\cdots & l_{n\,k} & \cdots & l_{nn-1} & 1 \end{pmatrix}.$$

Beweis: Übung (vgl. Aufgabe 2). Satz 1.6.4 zeigt bereits, dass die Matrizen $\boldsymbol{L}_k$ invertierbar sind und dass das Produkt eine Linksdreiecksmatrix sein muss, dessen Diagonaleinträge 1 sind. □

3.1.2 Gauss'scher Algorithmus und $\boldsymbol{LR}$-Zerlegung

Im vorhergehenden Abschnitt haben wir gesehen, dass gestaffelte Gleichungssysteme (d.h. solche, bei denen die Matrix $\boldsymbol{A}$ linke oder rechte Dreiecksgestalt hat) besonders einfach aufzulösen sind. Der Gauss'sche Algorithmus führt nun den allgemeinen Fall auf diese beiden Fälle zurück, indem er eine Matrix $\boldsymbol{A}$ in ein Produkt aus einer Linksdreiecksmatrix und einer Rechtsdreiecksmatrix zerlegt:

$$\boldsymbol{A} = \boldsymbol{LR}. \tag{3.1.2}$$

Ist eine solche Zerlegung bekannt, dann kann das Gleichungssystem (4.1.1) mithilfe einer Vorwärts- und einer Rückwärtssubstitution gelöst werden. Führt man nämlich den Hilfsvektor $y = \boldsymbol{R}x$ ein, so ergibt sich $b = \boldsymbol{A}x = \boldsymbol{LR}x = \boldsymbol{L}(\boldsymbol{R}x) = \boldsymbol{L}y$; dies führt auf folgende Vorgehensweise:

1. löse das Gleichungssystem $\boldsymbol{L}y = b$ für y mithilfe von Algorithmus 3.1.1;
2. löse das Gleichungssystem $\boldsymbol{R}x = y$ für x mithilfe von Algorithmus 3.1.2.

Definition 3.1.4 *Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. Dann besitzt $\boldsymbol{A}$ eine $\boldsymbol{LR}$-Zerlegung, falls es eine Rechtsdreiecksmatrix $\boldsymbol{R}$ und eine Linksdreiecksmatrix $\boldsymbol{L} \in \mathcal{L}_N^1$ gibt, so dass $\boldsymbol{A} = \boldsymbol{LR}$.*

Hat eine reguläre Matrix $\boldsymbol{A}$ eine $\boldsymbol{LR}$-Zerlegung, so ist diese eindeutig:

Theorem 3.1.5 *Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ regulär und habe eine $\boldsymbol{LR}$-Zerlegung $\boldsymbol{LR} = \boldsymbol{A}$. Dann ist $\boldsymbol{R}_{ii} \neq 0$ für $i = 1, \ldots, n$, und die Zerlegung ist eindeutig.*

Beweis: Wegen $0 \neq \det \boldsymbol{A} = \det \boldsymbol{L} \cdot \det \boldsymbol{R} = \prod_{i=1}^n \boldsymbol{R}_{ii}$ folgt die erste Behauptung. Seien $\boldsymbol{LR} = \boldsymbol{A} = \boldsymbol{L}'\boldsymbol{R}'$ zwei $\boldsymbol{LR}$-Zerlegungen von $\boldsymbol{A}$. Dann sind nach obiger Überlegung $\boldsymbol{R}$ und $\boldsymbol{R}'$ invertierbar (ihre Determinanten verschwinden nicht). Somit gilt

$$\boldsymbol{L}'\boldsymbol{R}' = \boldsymbol{LR} \quad \Longrightarrow \quad L^{-1}\boldsymbol{L}' = \boldsymbol{R}(\boldsymbol{R}')^{-1}.$$

Nach Satz 1.6.4 ist $\boldsymbol{L}^{-1}\boldsymbol{L}' \in \mathcal{L}_n^1$; ebenfalls nach Satz 1.6.4 ist $\boldsymbol{R}(\boldsymbol{R}')^{-1}$ eine Rechtsdreiecksmatrix. Die einzige Matrix, die zugleich Linksdreiecksmatrix und Rechtsdreiecksmatrix ist und ein Element von $\mathcal{L}_n^1$ ist, ist die Identität. Also ist $\boldsymbol{R} = \boldsymbol{R}'$ und $\boldsymbol{L} = \boldsymbol{L}'$.

Die Voraussetzung der Regularität von $\boldsymbol{A}$ ist wesentlich für die Eindeutigkeitsaussage, wie das Beispiel

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$$

zeigt. □

Die $\boldsymbol{LR}$-Zerlegung einer Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ geschieht in $n-1$ Schritten. Zur Motivation des Algorithmus schreiben wir das Gleichungssystem (4.1.1) aus:

$$\begin{array}{ccccccccc}
a_{11}\,x_1 & + & a_{12}\,x_2 & + & \cdots & + & a_{1n}\,x_n & = & b_1 \\
a_{21}\,x_1 & + & a_{22}\,x_2 & + & \cdots & + & a_{2n}\,x_n & = & b_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
a_{n1}\,x_1 & + & a_{n2}\,x_2 & + & \cdots & + & a_{nn}\,x_n & = & b_n
\end{array}$$

Es wird nun von der zweiten, dritten, etc. Zeile ein geeignetes Vielfaches der ersten Zeile subtrahiert, um die Variable x_1 in Zeilen 2 bis n zu eliminieren. Wir definieren also (für $a_{11} \neq 0$)

$$l_{i1} := \frac{a_{i1}}{a_{11}}, \qquad i = 2, \ldots, n$$

und ziehen von der i-ten Zeile das l_{i1}-fache der ersten Zeile ab. Wir erhalten damit ein Gleichungssystem von der folgenden Form:

$$\begin{array}{ccccccccc}
a_{11}\,x_1 & + & a_{12}\,x_2 & + & \cdots & + & a_{1n}\,x_n & = & b_1 \\
 & & a_{22}^{(1)}\,x_2 & + & \cdots & + & a_{2n}^{(1)}\,x_n & = & b_2^{(1)} \\
 & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & & a_{n2}^{(1)}\,x_2 & + & \cdots & + & a_{nn}^{(1)}\,x_n & = & b_n^{(1)}
\end{array}$$

wobei die neuen Koeffizienten $a_{ij}^{(1)}$ gegeben sind durch

$$\begin{aligned}
a_{ij}^{(1)} &= a_{ij} - a_{1j}\,l_{i1}, \qquad i, j = 2, \ldots, n, \\
b_i^{(1)} &= b_i - b_1\,l_{i1}, \qquad i = 2, \ldots, n.
\end{aligned}$$

Offenbar kann man (falls $a_{22}^{(1)} \neq 0$) nun ähnlich weitermachen, um in den Zeilen 3 bis n die Variable x_2 zu eliminieren. Dies geschieht, indem man

$$l_{i2} := \frac{a_{i2}^{(1)}}{a_{22}^{(1)}}, \qquad i = 3, \ldots, n$$

setzt und dann von der i-ten Zeile ($i \geq 3$) das l_{i2}-fache der 2-ten Zeile abzieht. Auf diese Weise erhält man dann ein System der Form

$$\begin{array}{ccccccccccccc}
a_{11}\,x_1 & + & a_{12}\,x_2 & + & a_{13}\,x_3 & + & \cdots & + & a_{1n}\,x_n & = & b_1 \\
 & & a_{22}^{(1)}\,x_2 & + & a_{23}^{(1)}\,x_3 & + & \cdots & + & a_{2n}^{(1)}\,x_n & = & b_2^{(1)} \\
 & & & & a_{33}^{(2)}\,x_3 & + & \cdots & + & a_{3n}^{(2)}\,x_n & = & b_3^{(2)} \\
 & & & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & & & & a_{n3}^{(2)}\,x_3 & + & \cdots & + & a_{nn}^{(2)}\,x_n & = & b_n^{(2)}
\end{array}$$

Nach $n-1$ Schritten erhält man dann schliesslich ein System von Gleichungen, das Rechtsdreiecksgestalt hat:

$$\begin{array}{ccccccccccc}
a_{11}\,x_1 & + & a_{12}\,x_2 & + & a_{13}\,x_3 & + & \cdots & + & a_{1n}\,x_n & = & b_1 \\
 & & a_{22}^{(1)}\,x_2 & + & a_{23}^{(1)}\,x_3 & + & \cdots & + & a_{2n}^{(1)}\,x_n & = & b_2^{(1)} \\
 & & & & a_{33}^{(2)}\,x_3 & + & \cdots & + & a_{3n}^{(2)}\,x_n & = & b_3^{(2)} \\
 & & & & & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\
 & & & & & & \ddots & \ddots & \vdots & \vdots & \vdots \\
 & & & & & & & & a_{nn}^{(n-1)}\,x_n & = & b_n^{(n-1)}
\end{array}$$

Die Zahlen $a_{kk}^{(k-1)}$, $k = 1, \ldots, n-1$, die während der Eliminationschritte auftreten, heissen *Pivots*[1]. Offensichtlich müssen wir verlangen, dass die Pivots nicht verschwinden, d.h. $a_{kk}^{(k-1)} \neq 0$ für $k = 1, \ldots, n-1$. Die berechneten Koeffizienten l_{ij} und das Endschema ergeben dann die gesuchte $\boldsymbol{LR}$-Zerlegung: Wir setzen

$$\boldsymbol{L} := \begin{pmatrix}
1 & & & & & \\
l_{21} & \ddots & & & & \\
l_{31} & \ddots & 1 & & & \\
\vdots & & l_{k+1,k} & 1 & & \\
\vdots & & \vdots & \ddots & \ddots & \\
l_{n1} & l_{n2}\cdots & l_{n\,k} & \cdots & l_{n,n-1} & 1
\end{pmatrix}, \quad \boldsymbol{R} := \begin{pmatrix}
a_{11} & a_{12} & \cdots & & & \\
0 & a_{22}^{(1)} & a_{21}^{(1)} & \cdots & & \\
\vdots & 0 & a_{33}^{(2)} & a_{34}^{(2)} & & \\
\vdots & & \ddots & \ddots & & \\
\vdots & & & \ddots & \ddots & \\
0 & \cdots & \cdots & \cdots & 0 & a_{nn}^{(n-1)}
\end{pmatrix}$$

und behaupten, dass $\boldsymbol{A} = \boldsymbol{LR}$ gilt. Um dies einzusehen, schreiben wir die entstandenen Gleichungssysteme in Matrixschreibweise. Im k-ten Eliminiationschritt hat das Gleichungssystem die Form

$$\boldsymbol{A}^{(k)} x = b^{(k)}$$

wobei $\boldsymbol{A}^{(k)}$, $b^{(k)}$ folgende Form haben:

$$\boldsymbol{A}^{(k)} = \begin{pmatrix}
a_{11} & a_{12} & a_{13} & \cdots & \cdots & \cdots & a_{1n} \\
 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & \cdots & \cdots & a_{2n}^{(1)} \\
 & & \ddots & \vdots & \vdots & \vdots & \vdots \\
 & & & a_{k+1\,k+1}^{(k)} & a_{k+1\,k+2}^{(k)} & \cdots & a_{k+1\,n}^{(k)} \\
 & & & \vdots & \vdots & \vdots & \vdots \\
 & & & a_{n\,k+1}^{(k)} & a_{n\,k+2}^{(k)} & \cdots & a_{nn}^{(k)}
\end{pmatrix}, \qquad b^{(k)} = \begin{pmatrix}
b_1 \\ b_2^{(1)} \\ \vdots \\ b_{k+1}^{(k)} \\ b_{k+2}^{(k)} \\ \vdots \\ b_n^{(k)}
\end{pmatrix}. \tag{3.1.3}$$

Wir setzen aus Notationsgründen

$$\boldsymbol{A}^{(0)} := \boldsymbol{A}, \qquad b^{(0)} := b.$$

Für die Ausführung des k-ten Schrittes werden die Faktoren

$$l_{ik} := \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}, \qquad i = k+1, \ldots, n,$$

[1] Bonaparte Pivot, 1.4.1769–1.4.1821, mit dem Spitznamen “der gute Teiler”

benötigt. Die Verbindung unseres Vorgehens mit der gesuchten $\boldsymbol{LR}$-Zerlegung von $\boldsymbol{A}$ ist nun, dass $\boldsymbol{A}^{(k)}$ aus $\boldsymbol{A}^{(k-1)}$ durch Multiplikation mit eine speziellen Linksdreiecksmatrix ergibt: Setzt man

$$\boldsymbol{L}_k := \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{n\,k} & & & 1 \end{pmatrix}, \tag{3.1.4}$$

so kann man nachrechnen (Übung!), dass gilt:

$$\boldsymbol{A}^{(k)} = \boldsymbol{L}_k \boldsymbol{A}^{(k-1)} \quad \text{und} \qquad b^{(k)} = \boldsymbol{L}_k b^{(k-1)}, \qquad k = 1, \ldots, n-1. \tag{3.1.5}$$

Man erhält also

$$\boldsymbol{A}^{(n-1)} = \boldsymbol{L}_{n-1}\boldsymbol{L}_{n-2}\cdots\boldsymbol{L}_1\boldsymbol{A}^{(0)} = \boldsymbol{L}_{n-1}\boldsymbol{L}_{n-2}\cdots\boldsymbol{L}_1\boldsymbol{A}.$$

Da alle auftretenden Linksdreiecksmatrizen $\boldsymbol{L}_k$ regulär sind (vgl. Sätze 1.6.4, 3.1.3), können wir dies umschreiben als

$$\boldsymbol{LR} = \boldsymbol{A},$$

wobei

$$\boldsymbol{L} := \boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdots\boldsymbol{L}_{n-1}^{-1}, \qquad \boldsymbol{R} = \boldsymbol{A}^{(n-1)}.$$

Hier ist $\boldsymbol{R}$ eine Rechtsdreiecksmatrix nach Konstruktion (vgl. (3.1.3)), und $\boldsymbol{L}$ ist eine Linksdreiecksmatrix nach Satz 1.6.4. Aus Satz 3.1.3 erhalten wir

$$\boldsymbol{L}_k^{-1} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & l_{n\,k} & & & 1 \end{pmatrix} \tag{3.1.6}$$

und damit—wiederum aus Satz 3.1.3—für die Einträge in $\boldsymbol{L}$ ganz explizit:

$$\boldsymbol{L} = \boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdots\boldsymbol{L}_{n-1}^{-1} = \begin{pmatrix} 1 & & & & & & \\ l_{21} & \ddots & & & & & \\ l_{31} & \ddots & 1 & & & & \\ \vdots & & l_{k+1,k} & 1 & & & \\ \vdots & & \vdots & \ddots & \ddots & & \\ l_{n1} & l_{n2}\cdots & l_{n\,k} & \cdots & l_{n,n-1} & 1 \end{pmatrix} \tag{3.1.7}$$

Wir haben also eine explizite Konstruktion einer $\boldsymbol{LR}$-Zerlegung von $\boldsymbol{A}$ gefunden: Die Einträge der Linksdreiecksmatrix $\boldsymbol{L}$ sind die Faktoren l_{ik}, die im Laufe des Algorithmus bestimmt werden und die Rechtsdreiecksmatrix $\boldsymbol{R}$ ist gerade das Endschema des Gauss'schen Algorithm, die Matrix $\boldsymbol{A}^{(n-1)}$. Wir können unser Vorgehen in dem folgenden abstrakten Algorithmus, der *Gauss'schen Elimination ohne Pivotsuche* festhalten:

Algorithmus 3.1.6 (Rohfassung der $\boldsymbol{LR}$-Zerlegung ohne Pivotsuche)
$\boldsymbol{A}^{(0)} := \boldsymbol{A}$
`for` k `from` 1 `to` $n-1$ `do` {

1. `bestimme Matrix` L_k `(vgl. (3.1.4)) durch Berechnung der Faktoren`

$$l_{ik} = \frac{A_{ik}^{(k-1)}}{A_{kk}^{(k-1)}}, \qquad i = k+1, \ldots, n$$

2. `setze` $\boldsymbol{A}^{(k)} := \boldsymbol{L}_k \boldsymbol{A}^{(k-1)}$

}
$\boldsymbol{LR}$`-Zerlegung von` $\boldsymbol{A}$ `ist` $\boldsymbol{A} = \boldsymbol{LR}$ `mit` $\boldsymbol{R} = \boldsymbol{A}^{(n-1)}$ `und` $\boldsymbol{L}$ `gegeben durch (3.1.7).`

Für eine Computerimplementierung von Algorithmus 3.1.6 müssen die Matrixmultiplikationen noch explizit ausgeschrieben werden. In tatsächlichen Implementierungen wird man direkt "auf" der Matrix $\boldsymbol{A}$ operieren, d.h. sie während des Algorithmus verändern. Dies geschieht aus Speicherplatzgründen, weil man nicht Speicher für die n Matrizen $\boldsymbol{A}^{(0)}$, $\boldsymbol{A}^{(1)}, \ldots$ bereitstellen kann/will. In dieser Form erhält man dann

Algorithmus 3.1.7 ($\boldsymbol{LR}$-Zerlegung ohne Pivotsuche)
`input:` Matrix $\boldsymbol{A}$
`output:` Linksdreiecksmatrix $\boldsymbol{L}$ und Rechtsdreiecksmatrix $\boldsymbol{R}$ mit $\boldsymbol{LR} = \boldsymbol{A}$
$\boldsymbol{L} := \boldsymbol{I}d_n =$ `Identitätsmatrix der Grösse` $n \times n$
`for` k `from` 1 `to` $n-1$ `do` {
  `for` i `from` $k+1$ `to` n `do` {
    $L_{ik} := \dfrac{A_{ik}}{A_{kk}}$ `%` k-te Spalte von $\boldsymbol{L}$
    `for` j `from` i+1 `to` n `do` { `%` k-te Zeile von $\boldsymbol{R}$ und updaten der Zeilen $k+1, \ldots, n$ von $\boldsymbol{A}$
      $A_{ij} := A_{ij} - L_{ik} A_{kj}$
    }
  }
 }
`setze` $\boldsymbol{R} :=$ Rechtsdreiecksanteil von $\boldsymbol{A}$
`return` $(\boldsymbol{L}, \boldsymbol{R})$

In der formulierten Form geht die Matrix $\boldsymbol{A}$ in Algorithmus 3.1.7 verloren, da sie mit der Rechtsdreiecksmatrix $\boldsymbol{R}$ überschrieben wird. In der rechentechnischen Praxis wird zudem weiter Speicher gespart: Nach Beendigung von Algorithmus 3.1.7 enthält die Matrix im oberen Teil die gesuchte Rechtsdreiecksmatrix $\boldsymbol{R}$. Der untere Teil enthält noch den unteren Teil der Originalmatrix $\boldsymbol{A}$ (man überzeuge sich davon, dass Algorithmus 3.1.7 den unteren Teil von $\boldsymbol{A}$ nicht verändert). Der untere Teil der Matrix $\boldsymbol{A}$ hat genausoviele Einträge wie zum Abspeichern der Linksdreiecksmatrix $\boldsymbol{L}$ genötigt werden (die Diagonaleinträge von $\boldsymbol{L}$ sind alle 1 und müssen daher nicht gesondert abgespeichert werden). In den meisten Implementierungen von $\boldsymbol{LR}$-Zerlegungen wird deshalb einfach nur die Matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ übergeben, und zurückgegeben

$\boldsymbol{LR}$-Zerlegung (Algorithmus 3.1.9)	$\frac{1}{3}n(n-1)(n+1)$
Vorwärtssubst. (Algorithmus 3.1.1 unter Ausnutzung von $L_{ii}=1$)	$\frac{1}{2}n(n-1)$
Rückwärtssubst. (Algorithmus 3.1.2)	$\frac{1}{2}n(n+1)$
Gesamtkosten	$\frac{1}{3}n^3+n^2-\frac{1}{3}n \approx \frac{1}{3}n^3$

Tabelle 3.1: Kosten für das Lösen eines linearen Gleichungssystems mit Algorithmus 3.1.9.

wird wieder eine Matrix $\tilde{\boldsymbol{A}} \in \mathbb{R}^{n\times n}$, in der die wesentliche Information über die Faktoren $\boldsymbol{L}$ und $\boldsymbol{R}$ gespeichert ist:

$$\tilde{A}_{ij} = \begin{cases} R_{ij} & \text{falls } j \geq i \\ L_{ij} & \text{falls } j < i \end{cases} \tag{3.1.8}$$

Eine Implementierung dieses Algorithmus ist dann wie folgt:

Algorithmus 3.1.8 ($\boldsymbol{LR}$-Zerlegung ohne Pivotsuche: klassische Implementierung)

`input:` Matrix $\boldsymbol{A}$
`output:` überschreibt die Matrix $\boldsymbol{A}$ mit ihrer $\boldsymbol{LR}$-Zerlegung (vgl. (3.1.8))

```
for k from 1 to n-1 do {
   for i from k+1 to n do {
      A_ik := A_ik / A_kk                                   % k-te Spalte von L
      for j from i+1 to n do { %k-te Zeile von R und updaten der Zeilen k+1...,n von A
       A_ij := A_ij - A_ik A_kj
      }
   }
 }
return (A)
```

Das Lösen eines linearen Gleichungssystems (4.1.1) wird deshalb wie folgt durchgeführt:

Algorithmus 3.1.9 (Gauss-Algorithmus ohne Pivotsuche)

1. *Bestimme $\boldsymbol{LR}$-Zerlegung von $\boldsymbol{A}$ mithilfe von Algorithmus 3.1.8.*
2. *Löse $\boldsymbol{L}y = b$ mithilfe der Vorwärtssubstitution Algorithmus 3.1.1. Dabei beachtet man, dass die Diagonalelemente von $\boldsymbol{L}$ gilt: $L_{ii} = 1$.*
3. *Löse $\boldsymbol{R}x = y$ mithilfe der Rückwärtssubstitution Algorithmus 3.1.2.*

In Tabelle 3.1 sind die Kosten beim Lösen eines linearen Gleichungssystems mithilfe von Algorithmus 3.1.9 zusammengestellt. Wir haben nur die Multiplikationen/Divisionen gezählt und die Additionen vernachlässigt. Wie man sieht, sind die Kosten (für grosse n) dominiert durch die $\boldsymbol{LR}$-Zerlegung. Bereits für $n = 100$ machen die Vorwärts- und Rückwärtssubstitionen zusammen nur 3% der Gesamtkosten aus. Ein positiver Nebeneffekt ist, dass, falls eine $\boldsymbol{LR}$-Zerlegung erst einmal aufgestellt ist, das lineare Gleichungssystem (4.1.1) für viele verschiedene rechte Seiten b billig gelöst werden kann.

Bemerkung 3.1.10 In der $\boldsymbol{LR}$-Zerlegung in Algorithmus 3.1.8 haben wir nicht den Fall abgefangen, dass ein sog. *Pivot* $A_{kk}^{(k-1)} = 0$ sein könnte. Algorithmus 3.1.8 versagt deshalb bereits bei dem trivialen Beipiel

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

Der Behandlung solcher Fälle werden wir uns im Abschnitt 3.3 zuwenden. ∎

Abschliessend stellen wir einen zu Algorithmus 3.1.8 äquivalenten Algorithmus zur Bestimmung der $\boldsymbol{LR}$-Zerlegung vor.

Algorithmus 3.1.11 (Doolittle Variante der $\boldsymbol{LR}$-Zerlegung)

`for` k `from` 1 `to` n `do` {
  $L_{kk} = 1$
  `for` j `from` k `to` n `do` { % Berechne k-te Zeile von $\boldsymbol{R}$
    $R_{kj} := A_{kj} - \sum_{l=1}^{k-1} L_{kl} R_{lj}$ %Konvention: leere Summe = 0
  }
  `for` i `from` $k+1$ `to` n `do` { % Berechne k-te Spalte von $\boldsymbol{L}$
    $L_{ik} := \dfrac{1}{R_{kk}} \left(A_{ik} - \displaystyle\sum_{l=1}^{k-1} L_{il} R_{lk} \right)$ %Konvention: leere Summe = 0
  }
}

Man beachte, dass der Algorithmus wohldefiniert ist, da die Rechtsdreiecksmatrix $\boldsymbol{R}$ zeilenweise und die Linksdreiecksmatrix $\boldsymbol{L}$ spaltenweise aufgebaut wird. Für jedes k werden von $\boldsymbol{L}$ nur die Spalten 1 bis $k-1$ und von $\boldsymbol{R}$ nur die Zeilen 1 bis $k-1$ benötigt, die bereits berechnet wurden. Algorithmus 3.1.11 stellt die Matrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ in genau derselben Reihenfolge auf wie Algorithmus 3.1.8. Von Interesse ist jedoch, dass er aus folgenden Überlegungen hergeleitet werden kann: Für jedes i, j gilt für die $\boldsymbol{LR}$-Zerlegung von $\boldsymbol{A}$:

$$A_{ij} = \sum_{l=1}^{n} L_{il} R_{lj}.$$

Aus der Tatsache, dass $\boldsymbol{L}$ Linksdreiecksmatrix, $\boldsymbol{R}$ Rechtsdreiecksmatrix und $L_{ii} = 1$, folgt damit

$$\begin{aligned} A_{ij} &= \sum_{l=1}^{i-1} L_{il} R_{lj} + R_{ij} \quad \text{für } j \geq i\,, \\ A_{ij} &= \sum_{l=1}^{j-1} L_{il} R_{lj} + R_{jj} L_{ij} \quad \text{für } j < i\,. \end{aligned}$$

Auflösen dieser beiden Gleichungen nach R_{ij} und L_{ij} ergibt dann die Ausdrücke in Algorithmus 3.1.11.

3.1.3 $\boldsymbol{LR}$-Zerlegung für schwach besetzte Matrizen

Die $\boldsymbol{LR}$-Zerlegung in Algorithmus 3.1.8 geht von einer *vollbesetzten* Matrix $\boldsymbol{A}$ aus. In der Praxis (z.B. in der Strukturmechanik und bei der Diskretiersierung von partiellen Differentialgleichungen) sind die auftretenden Matrizen oft *schwach besetzt* (engl. *sparse*), d.h. viele Einträge von $\boldsymbol{A}$ sind gleich Null. Dies kann in zweierlei Hinsicht ausgenutzt werden:

1. Speicherersparnis: Man speichert nicht die gesamte Matrix $\boldsymbol{A}$ ab, sondern nur die wesentliche Information, d.h. welche Einträge von Null verschieden sind und was ihre Werte sind.

2. Die Matrizen $\boldsymbol{L}$, $\boldsymbol{R}$ der $\boldsymbol{LR}$-Zerlegung von $\boldsymbol{A}$ sind ebenfalls schwach besetzt. Auch hier kann Speicher und Rechenzeit eingespart werden, indem nur die nicht-trivialen Einträge von $\boldsymbol{L}$ und $\boldsymbol{R}$ berechnet werden.

Im folgenden stellen wir zwei Typen von schwach besetzten Matrizen vor: Bandmatrizen und *Skyline*-Matrizen. Selbstverständlich decken diese beiden Typen nicht alle in der Praxis auftretenden Fälle von schwach besetzten Matrizen ab.

Bandmatrizen

Definition 3.1.12 *Eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ heisst eine* Bandmatrix *mit* Bandbreite $p+q+1$, *falls es p, $q \in \mathbb{N}_0$ mit*

$$a_{ij} = 0 \qquad \text{für } j > i+p \text{ oder } i > j+q.$$

Die Zahl p heisst die oberere Bandbreite *und q die* untere Bandbreite.

Bandmatrizen haben also nichtverschwindende Einträge höchstens auf den p Nebendiagonalen über der Hauptdiagonalen und auf den q Nebendiagonalen unter der Hauptdiagonalen:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1,p+1} & 0 & \cdots & \cdots & 0 \\ a_{21} & a_{22} & \cdots & \cdots & a_{2,p+2} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & & \ddots & & 0 \\ a_{q+1,1} & a_{q+1,2} & & \ddots & & & \ddots & 0 \\ 0 & a_{q+2,2} & & & \ddots & & & a_{n-p,n} \\ \vdots & 0 & \ddots & & & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{n,n-q} & \cdots & a_{n,n-1} & a_{nn} \end{pmatrix} \tag{3.1.9}$$

Um diese Matrix darzustellen, brauchen wir nur $(p+q+1)n - \frac{p(p+1)}{2} - \frac{q(q+1)}{2}$ reelle Zahlen abzuspeichern. Auch die $\boldsymbol{LR}$-Zerlegung einer Bandmatrix erbt die spezielle Struktur:

Theorem 3.1.13 *Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ eine Bandmatrix mit oberer Bandbreite p und unterer Bandbreite q und $\boldsymbol{LR}$-Zerlegung $\boldsymbol{LR} = \boldsymbol{A}$. Dann haben $\boldsymbol{L}$, $\boldsymbol{R}$ Bandstruktur, und es gilt:*

$$\begin{aligned} L_{ij} &= 0 \qquad \text{falls } j > i \text{ oder } j < i-q \\ R_{ij} &= 0 \qquad \text{falls } j < i \text{ oder } j > i+p. \end{aligned}$$

Beweis: Die Ausssage des Satzes folgt durch sorgfältige Untersuchung von Algorithmus 3.1.11. Man sieht recht einfach, dass die Aussage richtig ist für die erste Zeile von $\boldsymbol{R}$ und die erste Spalte von $\boldsymbol{L}$. Dann schliesst man induktiv für die weiteren Zeilen/Spalten mithilfe von Algorithmus 3.1.11. □

Satz 3.1.13 sagt aus, dass die Matrizen $\boldsymbol{L}$, $\boldsymbol{R}$ der $\boldsymbol{LR}$-Zerlegung der Bandmatrix $\boldsymbol{A}$ aus (3.1.9) folgende Struktur haben:

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & & & & 0 \\ l_{q+1,1} & l_{q+1,2} & & \ddots & \ddots & & & 0 \\ 0 & l_{q+2,2} & & & \ddots & \ddots & & \vdots \\ 0 & 0 & \ddots & & & \ddots & \ddots & \vdots \\ 0 & 0 & & \ddots & & & \ddots & 0 \\ 0 & 0 & \cdots & 0 & l_{n,n-q} & \cdots & l_{n,n-1} & 1 \end{pmatrix}$$

$$\boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1,p+1} & 0 & \cdots & \cdots & 0 \\ 0 & r_{22} & \cdots & \cdots & r_{2,p+2} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & & & \ddots & & 0 \\ \vdots & \vdots & \ddots & \ddots & & & \ddots & 0 \\ \vdots & \vdots & & \ddots & \ddots & & & r_{n-p,n} \\ \vdots & \vdots & & & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & & \cdots & 0 & r_{nn} \end{pmatrix}$$

Die Tatsache, dass die Matrizen $\boldsymbol{L}$ und $\boldsymbol{R}$ auch wieder schwach besetzt sind, wird bei Algorithmen zur Bestimmung von $\boldsymbol{LR}$-Zerlegungen von Bandmatrizen ausgenutzt. Es brauchen insbesondere nur die nicht-trivialen Einträge von $\boldsymbol{L}$, $\boldsymbol{R}$ berechnet zu werden. Dies führt auf die folgende Variante von Algorithmus 3.1.8, bei dem die beiden inneren Schleifen verkürzt werden können.

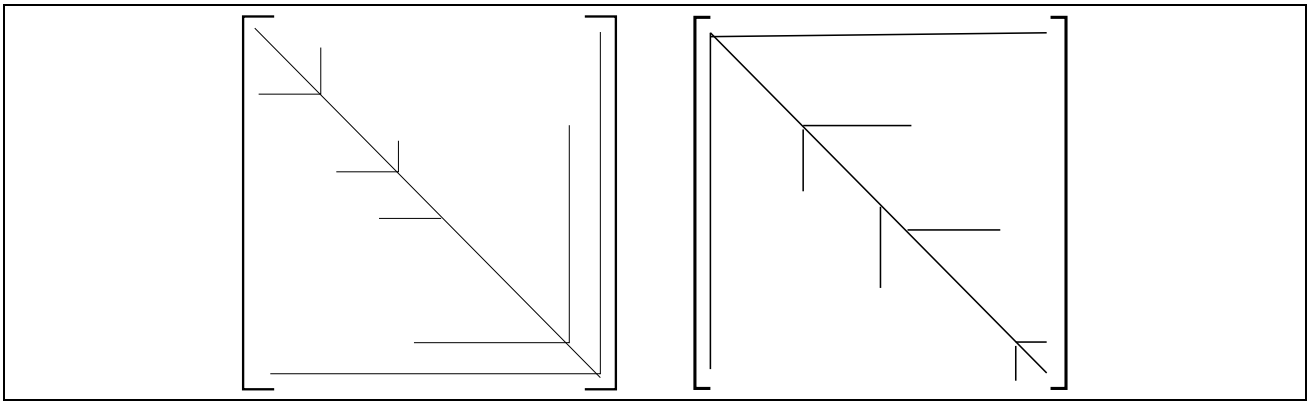
Algorithmus 3.1.14 ($\boldsymbol{LR}$-Zerlegung für Bandmatrizen)

`input:` Matrix $\boldsymbol{A}$ mit oberer Bandbreite p und unterer Bandbreite q
`output:` überschreibt die Matrix $\boldsymbol{A}$ mit ihrer $\boldsymbol{LR}$-Zerlegung

```
for k from 1 to n − 1 do {
   for i from k + 1 to min{n, k + q} do {
      A_ik := A_ik / A_kk
      for j from k + 1 to min{n, k + p} do {
       A_ij := A_ij − A_ik A_kj
      }
   }
 }
return (A)
```

$\boldsymbol{LR}$-Zerlegung (Algorithmus 3.1.14)	$\approx nqp$
Vorwärtssubst. (Alg. 3.1.1; Ausnutzen der Bandstruktur von $\boldsymbol{L}$)	$\approx nq$
Rückwärtssubst. (Alg. 3.1.2; Ausnutzen der Bandstruktur von $\boldsymbol{R}$)	$\approx np$
Gesamtkosten Multiplikationen/Divisionen	$\approx n(pq+p+q)$

Tabelle 3.2: Kosten beim Lösen von Gleichungssystemen mit Bandmatrizen



Figur 3.1: Striche deuten von Null verschiedene Einträge an. Links: *Skyline*-Matrix, bei der die Besetzungsstruktur bei $\boldsymbol{LR}$-Zerlegung erhalten bleibt. Rechts: Keine *Skyline*-Matrix und die $\boldsymbol{LR}$-Zerlegung erhält *nicht* die Besetzungsstruktur.

Die Bandstruktur von $\boldsymbol{L}$ und $\boldsymbol{R}$ wird ebenfalls bei der Vorwärts- und Rückwärtssubstitution ausgenutzt (Übung: Man formuliere die entsprechenden Varianten von Algorithmen 3.1.1, 3.1.2). Die Kosten der Algorithmen sind in Tabelle 3.2 zusammengestellt. Es werden nur Multiplikationen und Divisionen gezählt und vereinfachend $n \gg \max\{p,q\}$ angenommen.

Skyline-Matrizen

Ein wichtiger weiterer Spezialfall der schwach besetzten Matrizen sind die sog. *Skyline-Matrizen* (engl.: *skyline matrices*). Dies sind Matrizen, wie auf der linken Seite in Fig. 3.1 illustriert. Eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ heisst eine *Skyline-Matrix*, falls es für $i = 1,\dots,n$ Zahlen p_i, $q_i \in \mathbb{N}_0$ gibt, so dass

$$A_{ij} = 0 \quad \text{falls } j < i - p_i \text{ oder } i < j - q_j. \tag{3.1.10}$$

$$\boldsymbol{A} = \begin{pmatrix} 1 & & & 1 & & & 1 \\ & 1 & & 2 & & & 2 \\ & & 1 & 3 & & & 3 \\ 1 & 2 & 3 & 5 & & & 18 \\ & & & & 1 & & 5 \\ & & & & & 1 & 6 \\ 1 & 2 & 3 & 18 & 5 & 6 & 92 \end{pmatrix} \qquad \boldsymbol{L} = \boldsymbol{R}^\top = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ 1 & 2 & 3 & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ 1 & 2 & 3 & 4 & 5 & 6 & 1 \end{pmatrix}$$

Figur 3.2: $\boldsymbol{A} \in \mathbb{R}^{7\times 7}$ und ihre $\boldsymbol{LR}$-Zerlegung.

Es gilt dann

Theorem 3.1.15 *Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ eine Skyline-Matrix, d.h. es gebe p_i, q_i mit (3.1.10). Möge $\boldsymbol{A}$ die $\boldsymbol{LR}$-Zerlegung $\boldsymbol{A} = \boldsymbol{LR}$ haben. Dann gilt für die Einträge von $\boldsymbol{L}$ und $\boldsymbol{R}$:*

$$L_{ij} = 0 \quad \text{für } j < i - p_i, \qquad R_{ij} = 0 \quad \text{für } i < j - q_j.$$

Beweis: Wie in Satz 3.1.13 kann die Aussage mithilfe von Algorithmus 3.1.11 eingesehen werden (Übung). □

Satz 3.1.15 besagt, dass die Faktoren $\boldsymbol{L}$ und $\boldsymbol{R}$ der $\boldsymbol{LR}$-Zerlegung einer Skyline-Matrix $\boldsymbol{A}$ dieselbe Besetzungsstruktur haben wie $\boldsymbol{A}$. Figur 3.2 zeigt dies für ein einfaches Beispiel. Das Erhalten der Besetzungsstruktur kann natürlich algorithmisch ausgenutzt werden, sowohl was Speicher angeht als auch bzgl. Rechenzeit, indem nur die nicht-trivialen Einträge von $\boldsymbol{L}$ und $\boldsymbol{R}$ ausgerechnet und abgespeichert werden. Dies führt auf Varianten von Algorithmus 3.1.11, die analog zum Fall der Bandmatrizen sind. Man beachte, dass man die Matrizen in Fig. 3.1 nicht als Bandmatrix behandeln will, da die Bandbreiten p, q je gleich n wären. Das rechte Beispiel in Fig. 3.1 ist keine Skyline-Matrix im Sinne obiger Definition, und die Besetzungsstruktur geht bei der $\boldsymbol{LR}$-Zerlegung verloren: $\boldsymbol{L}$ ist i.a. eine volle Linksdreiecksmatrix und $\boldsymbol{R}$ eine volle Rechtsdreiecksmatrix (Man spricht bei Zerstörung der Besetzungsstruktur von *fill-in*).

3.2 SPD Matrizen. Cholesky Zerlegung.

In vielen Anwendungen treten symmetrische, positiv definite (SPD) Matrizen auf, die oft zudem schwach besetzt sind. Der Grund hierfür ist, dass diese Matrizen meist von physikalischen Modellen stammen, bei denen der Ausdruck $x^\top \boldsymbol{A} x$ eine Energie darstellt.

Definition 3.2.1 *Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. Die* transponierte Matrix $\boldsymbol{A}^\top$ *ist gegeben durch*

$$(\boldsymbol{A}^\top)_{ij} := A_{ji} \qquad \forall i, j \in \{1, \dots, n\}.$$

Eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ heisst

1. symmetrisch, *falls $\boldsymbol{A} = \boldsymbol{A}^\top$;*
2. positiv definit, *falls $x^\top \boldsymbol{A} x > 0 \qquad \forall 0 \neq x \in \mathbb{R}^n$;*
3. positiv semi-definit, *falls $x^\top \boldsymbol{A} x \geq 0 \qquad \forall 0 \neq x \in \mathbb{R}^n$.*

Eine symmetrische, positiv definite Matrix $A \in \mathbb{R}^{n\times n}$ heisst kurz SPD.

Theorem 3.2.2 *Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ SPD. Dann gilt:*

1. *$\boldsymbol{A}$ ist regulär (invertierbar);*
2. *$A_{ii} > 0$ für alle $i \in \{1, \dots, n\}$;*
3. *$|A_{ij}| < \frac{1}{2}(A_{ii} + A_{jj})$ für $i \neq j$ und damit $\max_{ij} |A_{ij}| = \max_i A_{ii}$.*

Beweis: Übung. □

Theorem 3.2.3 *Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ SPD. Dann existiert ein $\boldsymbol{L} \in \mathcal{L}_n^1$ und eine Diagonalmatrix $\boldsymbol{D}$ mit $D_{ii} > 0$, $i = 1, \ldots, n$, so dass $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^\top$. Die Matrizen $\boldsymbol{L}$ und $\boldsymbol{D}$ sind eindeutig. Zudem ist $\boldsymbol{L}$, $\boldsymbol{R} := \boldsymbol{D}\boldsymbol{L}^\top$ die $\boldsymbol{LR}$-Zerlegung von $\boldsymbol{A}$, die mithilfe von Algorithmus 3.1.8 bestimmt werden kann.*

Beweis: Wir partitionieren die Matrix $\boldsymbol{A}$ wie folgt:

$$\boldsymbol{A}^{(0)} = \boldsymbol{A} = \left(\begin{array}{c|c} A_{11} & z^\top \\ \hline z & B \end{array}\right),$$

wobei $z^\top = (A_{12}, \ldots, A_{1n})$. Im ersten Schritt der Gauss-Elimination erhält man

$$\boldsymbol{A}^{(1)} = \boldsymbol{L}_1\boldsymbol{A}^{(0)} = \left(\begin{array}{c|c} A_{11} & z^\top \\ \hline 0 & \\ \vdots & \boldsymbol{B}' \\ 0 & \end{array}\right), \qquad \boldsymbol{L}_1 = \begin{pmatrix} 1 & & & \\ -l_{21} & 1 & & \\ \vdots & & \ddots & \\ -l_{n1} & & & 1 \end{pmatrix}, \tag{3.2.11}$$

wobei $l_{i1} = A_{i1}/A_{11}$. Man beachte, dass nach Satz 3.2.2 $A_{11} > 0$. Eine Rechnung zeigt nun, dass

$$\boldsymbol{L}_1\boldsymbol{A}\boldsymbol{L}_1^\top = \boldsymbol{A}^{(1)}\boldsymbol{L}_1^\top = \left(\begin{array}{c|ccc} A_{11} & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \boldsymbol{B}' & \\ 0 & & & \end{array}\right).$$

Hier ist insbesondere die Matrix B' unverändert aus (3.2.11) übernommen. Man rechnet nun nach (oder verwendet den Trägheitssatz von Sylvester[2]), dass B' wieder SPD ist, denn die Matrix $\boldsymbol{L}_1\boldsymbol{A}\boldsymbol{L}_1^\top$ ist wieder SPD. Mithin kann man dieselbe Argumentation für $\boldsymbol{B}'$ wiederholen. Induktiv schliesst man dann, dass

$$\boldsymbol{L}_{n-1}\cdots\boldsymbol{L}_1\boldsymbol{A}\boldsymbol{L}_1^\top\boldsymbol{L}_2^\top\cdots\boldsymbol{L}_{n-1}^\top = \begin{pmatrix} A_{11} & & & \\ & d_{22} & & \\ & & \ddots & \\ & & & d_{nn} \end{pmatrix} =: \boldsymbol{D}.$$

Nach Konstruktion sind die Matrizen $\boldsymbol{L}_k$, $k = 1, \ldots, n-1$ gerade die Linksdreiecksmatrizen, die in Algorithmus 3.1.9 berechnet werden. Das hier vorgestellte Induktionsargument zeigt, dass der Algorithmus nicht abbricht, weil in jedem Eliminationsschritt das Pivotelement nicht verschwindet (die Diagonalelemente einer SPD-Matrix sind nach Satz 3.2.2 strikt positiv!). Die Eindeutigkeit von $\boldsymbol{L}$ (die dann die Eindeutigkeit von $\boldsymbol{D}$ nach sich zieht) folgt aus Satz 3.1.5, weil $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^\top = \boldsymbol{L}(\boldsymbol{D}\boldsymbol{L}^\top)$ eine $\boldsymbol{LR}$-Zerlegung von $\boldsymbol{A}$ ist. □

Satz 3.2.3 ist die Basis für die *Cholesky*[3]-Zerlegung einer SPD-Matrix $\boldsymbol{A}$.

[2]Sylvester, James Joseph 1814–1897
[3]Cholesky, André-Louis, 1875-1918

Korollar 3.2.4 *Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ SPD. Dann existiert eine eindeutige Linksdreiecksmatrix $\overline{\boldsymbol{L}} \in \mathcal{L}_n$ mit $\overline{\boldsymbol{L}}_{ii} > 0$, $i = 1, \ldots, n$, so dass $\boldsymbol{A} = \overline{\boldsymbol{L}} \cdot \overline{\boldsymbol{L}}^\top$. Die Matrix $\overline{\boldsymbol{L}}$ heisst der* Cholesky-Faktor *von $\boldsymbol{A}$. Umgekehrt gilt: Sei $\overline{\boldsymbol{L}} \in \mathcal{L}_n$ regulär. Dann ist $\boldsymbol{A} := \overline{\boldsymbol{L}} \cdot \overline{\boldsymbol{L}}^\top$ SPD.*

Beweis: Nach Satz 3.2.3 existiert ein $L \in \mathcal{L}_n^1$ und eine Diagonalmatrix $\boldsymbol{D}$ mit $D_{ii} > 0$, so dass $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^\top$. Wir setzen nun

$$\overline{\boldsymbol{L}} := \boldsymbol{L}\boldsymbol{D}^{1/2}, \qquad \left(\boldsymbol{D}^{1/2}\right)_{ij} := \delta_{ij} D_{ii}^{1/2}.$$

Dann gilt offensichtlich $\overline{\boldsymbol{L}} \cdot \overline{\boldsymbol{L}}^\top = \boldsymbol{A}$. Eindeutigkeit des Cholesky-Faktors: Sei $\overline{\boldsymbol{L}}$ eine Linksdreiecksmatrix mit $\overline{L}_{ii} > 0$, $i = 1, \ldots, n$ und $\overline{\boldsymbol{L}} \cdot \overline{\boldsymbol{L}}^\top = \boldsymbol{A}$. Definiere die Diagonalmatrix $\boldsymbol{D}$ durch $D_{ij} = \delta_{ij}\overline{L}_{ii}$. Dann ist $\boldsymbol{D}$ invertierbar und

$$\boldsymbol{L} := \overline{\boldsymbol{L}}\boldsymbol{D}^{-1}$$

ist ein Element von $\mathcal{L}_n^1$. Wir haben also

$$\boldsymbol{A} = \overline{\boldsymbol{L}} \cdot \overline{\boldsymbol{L}}^\top = \boldsymbol{L}\boldsymbol{D}\boldsymbol{D}^\top\boldsymbol{L}^\top = \boldsymbol{L}(\boldsymbol{D}\boldsymbol{D}^\top)\boldsymbol{L}^\top.$$

Dies ist eine Zerlegung wie in Satz 3.2.3. Aus der Eindeutigkeitsaussage von Satz 3.2.3 folgt damit, dass $\boldsymbol{L}$ und $\boldsymbol{D}\boldsymbol{D}^\top$ eindeutig bestimmt sind. Weil die Diagonalmatrix $\boldsymbol{D}$ positive Diagonaleinträge hat, ist damit auch $\boldsymbol{D}$ eindeutig bestimmt.

Der Beweis der Aussage, dass für eine reguläre Linksdreiecksmatrix $\overline{\boldsymbol{L}}$ die Matrix $\overline{\boldsymbol{L}} \cdot \overline{\boldsymbol{L}}^\top$ SPD ist: Übung. □

Für SPD-Matrizen benutzt man anstelle der $\boldsymbol{LR}$-Zerlegung in der Praxis die Cholesky-Zerlegung. Algorithmisch bestimmt man sie mithilfe einer Variante von Algorithmus 3.1.11:

Algorithmus 3.2.5 (Cholesky-Zerlegung)
input: SPD-Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$
output: Cholesky-Faktor $\overline{\boldsymbol{L}}$ von $\boldsymbol{A}$

`for` k `from` 1 `to` n `do` {
  $\overline{L}_{kk} := \left(A_{kk} - \sum_{j=1}^{k-1} \overline{L}_{kj}^2\right)^{1/2}$ %Konvention: leere Summe = 0
  `for` i `from` $k+1$ `to` n `do` { % Berechne k-te Spalte von $\boldsymbol{L}$
    $\overline{L}_{ik} := \frac{1}{\overline{L}_{kk}}\left(A_{ik} - \sum_{j=1}^{k-1} \overline{L}_{ij}\overline{L}_{kj}\right)$ %Konvention: leere Summe = 0
  }
}

dass der Algorithmus das Gewünschte leistet, sieht man in ähnlicher Weise wie bei Algorithmus 3.1.11, indem man den Anzatz $\boldsymbol{A} = \overline{\boldsymbol{L}}\overline{\boldsymbol{L}}^\top$ macht und dann Bestimmungsgleichungen für die Einträge von $\overline{\boldsymbol{L}}$ herleitet:

$$A_{ik} = \sum_{j=1}^{n} \overline{L}_{ij}\overline{L}_{kj}.$$

Wegen der Symmetrie von $\boldsymbol{A}$ reicht es, $k \leq i$ zu betrachten. Nutzt man die Tatsache aus, dass $\overline{\boldsymbol{L}}$ eine untere Dreiecksmatrix ist, dann folgt:

$$\text{für } k = i: \qquad A_{kk} = \sum_{j=1}^{i-1} \overline{L}_{kj}^2 + \overline{L}_{kk}^2,$$

$$\text{für } k < i: \qquad A_{ik} = \sum_{j=1}^{k-1} \overline{L}_{ij}\overline{L}_{kj} + \overline{L}_{ik}\overline{L}_{kk}.$$

Auflösen nach $\overline{L}_{ij}$ und $\overline{L}_{ii}$ ergibt dann die Ausdrücke, die in Algorithmus 3.2.5 auftreten.

Betrachten wir die Kosten der Cholesky-Zerlegung. Aus Algorithmus 3.2.5 geht hervor, dass die Cholesky-Zerlegung einer SPD-Matrix $\boldsymbol{A}$ mit

$$\sim \frac{1}{6}n^3 \quad \text{Multiplikationen/Divisionen und} \quad n \quad \text{Quadratwurzeln}$$

berechnet wird. Vernachlässigt man die Kosten für das Wurzelziehen, dann ist die Cholesky-Zerlegung ungefähr halb so teuer wie die $\boldsymbol{LR}$-Zerlegung. Die Reduktion um den Faktor 2 liegt daran, dass wegen der Symmetrie der Matrix und der Zerlegung nur ein Faktor der Zerlegung berechnet werden muss.

Da viele in der Praxis auftretenden SPD-Matrizen Bandstruktur haben, formulieren wir noch die Variante der Cholesky-Zerlegung, die die Bandstruktur ausnutzt.

Algorithmus 3.2.6 (Cholesky-Zerlegung für SPD Bandmatrizen)
input: SPD-Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$; oberere Bandbreite p = untere Bandbreite q
output: Cholesky-Faktor $\overline{\boldsymbol{L}}$ von $\boldsymbol{A}$

`for` k `from` 1 `to` n `do` {

$$\overline{L}_{kk} := \left(A_{kk} - \sum_{j=\max\{1,k-p\}}^{k-1} \overline{L}_{kj}^2 \right)^{1/2}$$

  `for` i `from` $k+1$ `to` $\min\{n, k+p\}$ `do` {

$$\overline{L}_{ik} := \frac{1}{\overline{L}_{kk}} \left(A_{ik} - \sum_{j=\max\{1,k-p\}}^{k-1} \overline{L}_{ij}\overline{L}_{kj} \right)$$

  }

}

Bemerkung 3.2.7 Algorithmus 3.2.5 zur Bestimmung der Cholesky-Zerlegung kann auch dazu benutzt werden, zu prüfen, ob eine gegebene symmetrische Matrix positiv definit ist. Man führt Algorithmus 3.2.5 durch; bricht er ab, weil eine Division durch Null auftritt oder weil eine Wurzel aus einer negativen Zahl gezogen werden soll, dann war die Matrix nicht SPD. Andernfalls ist sie SPD (vgl. die zweite Aussage aus Korollar 3.2.4). ∎

3.3 Pivotstrategien & Nachiteration

3.3.1 Spaltenpivotstrategie

Bei unserer Herleitung der $\boldsymbol{LR}$-Zerlegung nahmen wir stets an, dass die Pivots $A_{kk}^{(k-1)} \neq 0$. Dies muss nicht immer der Fall sein, wie die Matrix

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

zeigt. Die Matrix $\boldsymbol{A}$ ist jedoch regulär, und man kann Gleichungssysteme von der Form $\boldsymbol{A}x = b$ lösen, indem man zuerst die Zeilen 1 und 2 vertauscht. Die nach Vertauschung dieser Zeilen erhaltene Matrix $\boldsymbol{A}_{per}$ hat dann ein von Null verschiedenes Pivot $(\boldsymbol{A}_{per})_{11}$; in diesem Fall hat A_{per} sogar bereits Rechtsdreiecksgestalt.

Man erwartet, dass auch bei kleinen Pivots numerische Schwierigkeiten auftauchen. Folgendes Beispiel erfüllt diese Erwartung:

Beispiel 3.3.1 Wir bestimmen in 4-stelliger Gleitkommaarithmetik $\mathbb{F}$ (d.h. $\beta = 10$, $t = 4$) die Lösung x des folgenden linearen Gleichungssystems:

$$\boldsymbol{A} = \begin{pmatrix} 3.1 \cdot 10^{-4} & 1 \\ 1 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} -3 \\ -7 \end{pmatrix}.$$

Es ist dann $l_{21} = 1/(3.1 \cdot 10^{-4}) \approx 3.226 \cdot 10^3$ und die $\boldsymbol{LR}$-Zerlegung von $\boldsymbol{A}$ ist

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 \\ 3.226 \cdot 10^3 & 1 \end{pmatrix}, \qquad \boldsymbol{R} = \begin{pmatrix} 3.1 \cdot 10^{-4} & 1 \\ 0 & 1 - l_{21} \end{pmatrix} \approx \begin{pmatrix} 3.1 \cdot 10^{-4} & 1 \\ 0 & -3.225 \cdot 10^3 \end{pmatrix}.$$

Die Lösung von $\boldsymbol{L}y = b$ führt dann auf

$$y = \boldsymbol{L}^{-1}b = \begin{pmatrix} -3 \\ 9.671 \cdot 10^3 \end{pmatrix}$$

und damit ist die Lösung x von $\boldsymbol{R}x = y$

$$x = \boldsymbol{R}^{-1}y = \begin{pmatrix} \frac{1}{3.1 \cdot 10^{-4}}(-3 - (-2.999)) \\ -2.999 \end{pmatrix} = \begin{pmatrix} -3.226 \\ -2.999 \end{pmatrix}.$$

Das "exakte" Ergebnis ist $x = (-4.001246\ldots, -2.99875\ldots)^\top$. Hier sind beim Rückwärtseinsetzen in der x_1-Komponente durch Auslöschung alle Ziffern verloren gegangen. Der Grund ist die schlechte Pivotwahl. Wir starteten mit einem sehr kleinen Pivot und erhielten dadurch sehr grosse (und auch sehr kleine) Einträge in der $\boldsymbol{LR}$-Zerlegung. Dies führt dann zu Auslöschung bei der Vorwärts- und Rückwärtssubstitution. ∎

Beispiel 3.3.1 zeigt, dass kleine Pivots zu numerischen Instabilitäten bei Vorwärts- und Rückwärtssubstitution führen können. dass kleine Pivots gemieden werden sollen, legt folgende Betrachtung nahe: Wir nehmen an, dass die rechte Seite b und die gesuchte Lösung Einträge haben, die von vergleichbarer Grössenordnung sind (wie in Beispiel 3.3.1). Wird bei der Vorwärts- oder Rückwärtssubstitution mit grossen Zahlen multipliziert, so entstehen Zwischenergebnisse, die gross sind (wie in Beispiel 3.3.1). Da das Endergebnis wieder moderat ist, erwartet man,

dass dies durch Subtraktion vergleichbarer Zahlen erreicht wird—bei diesen Subtraktionen tritt dann die Gefahr der Auslöschung auf. Dies ist im obigen Beispiel 3.3.1 eingetreten.

Um das Problem des kleinen Pivots in den Griff zu bekommen, wird deshalb nicht eine $\boldsymbol{LR}$-Zerlegung der Matrix $\boldsymbol{A}$ gesucht, sondern die $\boldsymbol{LR}$-Zerlegung einer Matrix $\boldsymbol{A}_{per}$, die durch geeignetes Vertauschen von Zeilen von $\boldsymbol{A}$ entstanden ist. Man beachte, dass für das Lösen von Gleichungssystemen das Vertauschen von Zeilen keine Rolle spielt (wenn man beim Vektor auf der rechten Seite die entsprechende Vertauschung ebenfalls durchführt). dass Zeilenvertauschen numerisch vorteilhaft sein kann, zeigt folgende Fortsetzung von Beispiel 3.3.1:

Beispiel 3.3.2 Wir lösen das lineare Gleichungssystem aus Beispiel 3.3.1, indem wir die beiden Zeilen von $\boldsymbol{A}x = b$ vertauschen, d.h. wir betrachten

$$\boldsymbol{A}_{per} = \begin{pmatrix} 1 & 1 \\ 3.1 \cdot 10^{-4} & 1 \end{pmatrix}, \qquad b_{per} = \begin{pmatrix} -7 \\ -3 \end{pmatrix}.$$

Nun ist $l_{21} = 3.1 \cdot 10^{-4}$ und

$$\boldsymbol{L}_{per} = \begin{pmatrix} 1 & 0 \\ 3.1 \cdot 10^{-4} & 1 \end{pmatrix}, \qquad \boldsymbol{R}_{per} = \begin{pmatrix} 1 & 1 \\ 0 & 1 - l_{21} \end{pmatrix} \approx \begin{pmatrix} 1 & 1 \\ 0 & 0.9997 \end{pmatrix}.$$

Für die Lösungen y, x von $\boldsymbol{L}y = b$, $\boldsymbol{R}x = y$ erhalten wir damit

$$y = \begin{pmatrix} -7 \\ -2.998 \end{pmatrix}, \qquad x = \begin{pmatrix} -4.001 \\ -2.999 \end{pmatrix}.$$

Wir erhalten also das korrekte Ergebnis bis auf Rundungsgenauigkeit. Da die Permutationsmatrix

$$\boldsymbol{P} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

bei Multiplikation von links an die Matrix $\boldsymbol{A}$ die Zeilen 1 und 2 vertauscht, haben wir also folgende Zerlegung erhalten:

$$\boldsymbol{L}_{per}\boldsymbol{R}_{per} = \boldsymbol{A}_{per} = \boldsymbol{PA}.$$

■

In Beispiel 3.3.2 konnte das lineare Gleichungssystem numerisch stabil gelöst werden, indem die beiden Zeilen des Gleichungssystems vertauscht wurden. Für eine allgemeine Matrix $\boldsymbol{A}$ ist die richtige Zeilenanordnung natürlich nicht im Voraus bekannt. Sie muss also während des Algorithmus mitbestimmt werden. Man geht deshalb algorithmisch wie folgt vor (siehe Algorithmus 3.3.5 unten): In jedem Eliminationsschritt wird nicht einfach $A_{kk}^{(k-1)}$ als Pivot benutzt, sondern es wird in der Spalte k das betragsgrösste Element $A_{ik}^{(k-1)}$ mit Zeilenindex $i \geq k$ gesucht; anschliessend werden die Zeilen k und i vertauscht und dann der Eliminationsschritt durchgeführt.

Das Vertauschen von Zeilen in einer Matrix beschreibt man formal am besten mit *Permutationsmatrizen*:

Definition 3.3.3 (Permutationsmatrizen) *Sei $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ eine Permutation der Zahlen $1, \ldots, n$ (d.h. π ist eine bijektive Abbildung) und bezeiche $e_1, \ldots, e_n$ die n Einheitsvektoren: $(e_i)_j = \delta_{ij}$. Dann heisst*

$$\boldsymbol{P}_\pi := \left(e_{\pi(1)}, e_{\pi(2)}, \ldots, e_{\pi(n)}\right)$$

die zu π gehörige Permutationsmatrix.

Lemma 3.3.4 (Eigenschaften von Permutationsmatrizen) *Sei* $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ *eine Permutation und* P_π *die zugehörige Permutationsmatrix. Dann gilt:*

(i) $\boldsymbol{P}_\pi e_i = e_{\pi(i)}$ *für* $i \in \{1, \ldots, n\}$.

(ii) $\boldsymbol{P}_\pi^{-1} = \boldsymbol{P}_\pi^\top$.

(iii) Die Matrix $\boldsymbol{P}_\pi A$ *entsteht aus* $\boldsymbol{A}$ *durch Zeilenpermutation, d.h. die* i*-te Zeile von* $\boldsymbol{A}$ *ist die* $\pi(i)$*-te Zeile von* $\boldsymbol{P}_\pi \boldsymbol{A}$, $i = 1, \ldots, n$.

Beweis: Übung. □

Der Algorithmus zur $\boldsymbol{LR}$-Zerlegung mit Pivotsuche ist damit wie folgt:

Algorithmus 3.3.5 ($\boldsymbol{LR}$-Zerlegung mit Spaltenpivotsuche)
$\boldsymbol{A}^{(0)} := \boldsymbol{A}$
`for` k `from 1 to` $n-1$ `do {`

1. `suche` $i \in \{k, \ldots, n\}$ `mit` $|A_{ik}^{(k-1)}| \geq |A_{i'k}^{(k-1)}|$ `für alle` $i' \in \{k, \ldots, n\}$
2. `setze` $\pi_k : \{1, \ldots, n\} \to \{1, \ldots, n\}$ `die Permutation, die` i `und` k `vertauscht:`
$$\pi_k(j) = \begin{cases} i & \texttt{falls } j = k, \\ k & \texttt{falls } j = i, \\ j & \texttt{sonst} \end{cases}$$
3. $\boldsymbol{A}^{(k-1)} := \boldsymbol{P}_{\pi_k} \boldsymbol{A}^{(k-1)}$ `% vertausche Zeilen` k `und` i `in` $\boldsymbol{A}^{(k-1)}$
4. `bestimme Matrix` $\boldsymbol{L}_k$ `(vgl. (3.1.4)) durch Berechnung der Faktoren`
$$l_{ik} = \frac{A_{ik}^{(k-1)}}{A_{kk}^{(k-1)}}, \qquad i = k+1, \ldots, n$$
5. `setze` $\boldsymbol{A}^{(k)} := L_k \boldsymbol{A}^{(k-1)}$

`}`
`setze` $\pi := \pi_{n-1} \circ \pi_{n-2} \circ \cdots \circ \pi_1$
`setze` $\boldsymbol{P} := \boldsymbol{P}_\pi$
`setze` $\boldsymbol{R} := \boldsymbol{A}^{(n-1)}$,
$\boldsymbol{L}$ `gegeben durch (3.1.7)`
`return(`$\boldsymbol{P}, \boldsymbol{L}, \boldsymbol{R}$`)` % $\boldsymbol{LR}$-Zerlegung von $\boldsymbol{PA}$ ist $\boldsymbol{PA} = \boldsymbol{LR}$.

Bemerkung 3.3.6 Der Algorithmus wird in der Praxis etwas anders realisiert: wie beim Fall ohne Pivotsuche, Algorithmus 3.1.6, operiert man direkt auf der Matrix $\boldsymbol{A}$ und erhält am Schluss das Endschema $\boldsymbol{R}$ anstelle von $\boldsymbol{A}$. Ausserdem wird wie in Algorithmus 3.1.8 der Linksdreiecksfaktor $\boldsymbol{L}$ ebenfalls im unteren Teil von $\boldsymbol{A}$ abgespeichert. Die Permutationsmatrix $\boldsymbol{P}$ wird nicht explizit aufgestellt, sondern es wird nur ein Vektor mit natürlichen Zahlen zurückgegeben, der angibt, wie die Zeilen von $\boldsymbol{A}$ permutiert werden. ■

Wir führen das Vorgehen an einem einfachen Beispiel vor.

Beispiel 3.3.7

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 2 \\ 2 & -7 & 2 \\ 1 & 24 & 0 \end{pmatrix}$$

Bei der Pivotsuche in der 1. Spalte, stossen wir auf die 2. Zeile. Man vertauscht also die 1. und 2. Zeile:

$$\boldsymbol{A}^{(0)} = \begin{pmatrix} 2 & -7 & 2 \\ 1 & 2 & 2 \\ 1 & 24 & 0 \end{pmatrix}$$

und führt dann den Eliminationsschritt durch. Es ist $l_{21} = 0.5$, $l_{31} = 0.5$ und damit

$$\boldsymbol{A}^{(1)} = \begin{pmatrix} 2 & -7 & 2 \\ 0 & 5.5 & 1 \\ 0 & 27.5 & -1 \end{pmatrix}.$$

Bei der Pivotsuche in der 2. Spalte müssen wir nun nur die Elemente $\boldsymbol{A}_{22}^{(1)}$ und $\boldsymbol{A}_{32}^{(1)}$ vergleichen. Da $\boldsymbol{A}_{32}^{(1)}$ betragsmässig grösser ist als $\boldsymbol{A}_{22}^{(1)}$, vertauschen wir die 2. und die 3. Zeile:

$$\boldsymbol{A}^{(1)} = \begin{pmatrix} 2 & -7 & 2 \\ 0 & 27.5 & -1 \\ 0 & 5.5 & 1 \end{pmatrix}.$$

Beim nächsten Eliminationsschritt entsteht $l_{32} = \frac{5.5}{27.5} = 0.2$. Wir erhalten als Endschema und als Matrix $\boldsymbol{L}$

$$\boldsymbol{R} = \boldsymbol{A}^{(2)} = \begin{pmatrix} 2 & -7 & 2 \\ 0 & 27.5 & -1 \\ 0 & 0 & 1.2 \end{pmatrix}, \qquad \boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.5 & 0.2 & 1 \end{pmatrix}.$$

Es bleibt, die Permutationsmatrix $\boldsymbol{P}$ zu bestimmen. In Algorithmus 3.3.5 wurde die Permutationen π_1, π_2 aufgestellt mit

$$\begin{aligned} \pi_1(1) &= 2, \qquad \pi_1(2) = 1, \qquad \pi_1(3) = 3 \\ \pi_2(1) &= 1, \qquad \pi_2(2) = 3, \qquad \pi_2(3) = 2. \end{aligned}$$

Damit ergibt sich für $\pi = \pi_2 \circ \pi_1$:

$$\pi(1) = 3, \qquad \pi(2) = 1, \qquad \pi(3) = 2$$

und somit für die Permutationsmatrix $\boldsymbol{P}_\pi$

$$\boldsymbol{P} = \boldsymbol{P}_\pi = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

Man hätte die Permutation π auch weniger formal durch Verfolgen der Zeilenvertauschungen erhalten können: die ursprünglich 1. Zeile ist zur 3. Zeile geworden (im ersten Schritt wurden Zeilen 1 und 2 vertauscht, in zweiten Schritt Zeilen 2 und 3), die ursprüngliche 2. Zeile ist zur 1. Zeile geworden, und die ursprünglich 3. Zeile ist am Schluss die 2. Zeile. Die Permutation π ist damit $\pi(1) = 3$, $\pi(2) = 1$, $\pi(3) = 2$. Man überzeugt sich davon, dass in der Tat $LR = PA$. ■

Das Lösen von Gleichungssystemen erfolgt dann so:

Algorithmus 3.3.8
input: reguläre Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$,
output: Lösung x von $\boldsymbol{A}x = b$

1. `Bestimme` P, $\boldsymbol{L}$, $\boldsymbol{R}$ `mithilfe von Algorithmus 3.3.5 so, dass` $\boldsymbol{PA} = \boldsymbol{LR}$.

2. `Setze` $b' := \boldsymbol{P}b$ `und löse` $\boldsymbol{L}y = b'$ `mithilfe von Algorithmus 3.1.1.`

3. `Löse` $\boldsymbol{R}x = y$ `mihilfe von Algorithmus 3.1.2.`

Bemerkung 3.3.9 Im 2. Schritt von Algorithmus 3.3.8 wird die Multiplikation $\boldsymbol{P}b$ nicht als Matrix-Vektor Multiplikation ausgeführt, sondern es werden natürlich nur die entsprechenden Zeilen in b vertauscht. ∎

dass Algorithmus 3.3.5 tatsächlich die $\boldsymbol{LR}$-Zerlegung einer Zeilenpermutation von $\boldsymbol{A}$ liefert, garantiert der folgende Satz.

Theorem 3.3.10 *Zu jeder regulären Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ existiert eine Permutationsmatrix $\boldsymbol{P}_\pi$, so dass eine Dreieckszerlegung*

$$\boldsymbol{LR} = \boldsymbol{P}_\pi \boldsymbol{A}$$

möglich ist. Hier ist $\boldsymbol{L} \in \mathcal{L}_n^1$ und $\boldsymbol{R}$ ist eine Rechtsdreiecksmatrix. Zudem gilt

$$|L_{ij}| \leq 1 \qquad \forall i, j \in \{1, \ldots, n\}.$$

Beweis: Die $\boldsymbol{LR}$-Zerlegung und die Permutationsmatrix, deren Existenz im Satz behauptet wird, konstruieren wir mithilfe von Algorithmus 3.3.5. Im ersten Schritt von Algorithmus 3.3.5 wird (falls nötig) eine Zeilenvertauschung von zwei Zeilen durchgeführt, so dass die neue Matrix

$$\boldsymbol{A}^{(0)} := \boldsymbol{P}_{\pi_1} \boldsymbol{A},$$

so dass $A_{11}^{(0)}$ das betragsmässig grösste Element der ersten Spalte von $\boldsymbol{A}^{(0)}$ ist. Die Permutationsmatrix P_{π_1} vermittelt dabei die Vertauschung der beiden Zeilen ($P_{\pi_1} = \boldsymbol{I}d$ falls keine Vertauschung nötig ist). Zudem ist $A_{11}^{(0)} \neq 0$, denn sonst wäre die erste Spalte identisch Null im Widerspruch zur Annahme, dass $\boldsymbol{A}$ regulär ist. Weil $A_{11}^{(0)}$ das betragsgrösste Element in der ersten Spalte von $\boldsymbol{A}^{(0)}$ ist, gilt für die Einträge $l_{i1} = A_{i1}^{(0)}/A_{11}^{(1)}$, dass $|l_{i1}| \leq 1$. Wir erhalten also nach dem ersten Eliminationsschritt mit $\boldsymbol{L}_1$ gegeben durch (3.1.4):

$$\boldsymbol{A}^{(1)} = \boldsymbol{L}_1 \boldsymbol{A}^{(0)} = \boldsymbol{L}_1 \boldsymbol{P}_{\pi_1} \boldsymbol{A} = \left(\begin{array}{c|ccc} A_{11}^{(1)} & * & \cdots & * \\ \hline 0 & & & \\ \vdots & & \boldsymbol{B}^{(1)} & \\ 0 & & & \end{array}\right).$$

Aus der Regularität von $\boldsymbol{L}_1$, $\boldsymbol{P}_{\pi_1}$ und $\boldsymbol{A}$ folgt also $0 \neq \det \boldsymbol{A}^{(1)} = A_{11}^{(1)} \det \boldsymbol{B}^{(1)}$. Mithin ist die Untermatrix $B^{(1)}$ regulär. Wir können also mit Algorithmus 3.3.5 fortfahren und erhalten schliesslich

$$\boldsymbol{R} = \boldsymbol{A}^{(n-1)} = \boldsymbol{L}_{n-1} \boldsymbol{P}_{\pi_{n-1}} \boldsymbol{L}_{n-2} \boldsymbol{P}_{\pi_{n-2}} \cdots \boldsymbol{L}_1 \boldsymbol{P}_{\pi_1} \boldsymbol{A}, \tag{3.3.12}$$

wobei die Matrizen $\boldsymbol{L}_k$ alle Einträge haben, die betragsmässig durch 1 beschränkt sind. Um die Frobeniusmatrizen $\boldsymbol{L}_k$ von den Permutationsmatrizen $\boldsymbol{P}_{\pi_k}$ zu separieren, schieben wir in der Darstellung von $\boldsymbol{R}$ aus (3.3.12) zwischen die Faktoren $\boldsymbol{L}_k$ und $\boldsymbol{P}_{\pi_k}$ die Identität $\boldsymbol{P}_k^{-1} \boldsymbol{P}_k$, wobei

die Permutationsmatrix $\boldsymbol{P}_k$ gegeben ist durch: $\boldsymbol{P}_k := \boldsymbol{P}_{\pi_{n-1}}\boldsymbol{P}_{\pi_{n-2}}\cdots\boldsymbol{P}_{\pi_{k+1}}$ ($\boldsymbol{P}_{n-1} = \boldsymbol{I}d_n$). Wir erhalten damit

$$\begin{aligned}\boldsymbol{R} &= \boldsymbol{L}_{n-1}\boldsymbol{P}_{n-1}^{-1}\boldsymbol{P}_{n-1}\boldsymbol{P}_{\pi_{n-1}}\boldsymbol{L}_{n-2}\boldsymbol{P}_{n-2}^{-1}\boldsymbol{P}_{n-2}\boldsymbol{P}_{\pi_{n-2}}\boldsymbol{L}_{n-3}\boldsymbol{P}_{n-3}^{-1}\boldsymbol{P}_{n-3}\boldsymbol{P}_{\pi_{n-3}}\\ &\boldsymbol{L}_{n-4}\boldsymbol{P}_{n-4}^{-1}\boldsymbol{P}_{n-4}\cdots\boldsymbol{L}_1\boldsymbol{P}_1^{-1}\boldsymbol{P}_1\boldsymbol{P}_{\pi_1}\boldsymbol{A}.\end{aligned}$$

Weil $\boldsymbol{P}_k\boldsymbol{P}_{\pi_k} = \boldsymbol{P}_{k-1}$, ergibt sich mit der Abkürzung

$$\hat{\boldsymbol{L}}_k := \boldsymbol{P}_k\boldsymbol{L}_k\boldsymbol{P}_k^{-1} \tag{3.3.13}$$

für $\boldsymbol{R}$:

$$\boldsymbol{R} = \hat{\boldsymbol{L}}_{n-1}\hat{\boldsymbol{L}}_{n-2}\cdots\hat{\boldsymbol{L}}_1\boldsymbol{P}_0\boldsymbol{A}.$$

Da $\boldsymbol{P}_0$ als Verkettung von Permutationsmatrizen eine Permutationsmatrix ist, bleibt zu zeigen, dass das Produkt der Matrizen $\hat{\boldsymbol{L}}_k$ tatsächlich eine Linksdreiecksmatrix ist. Sei $\pi : \{1,\ldots,n\} \to \{1,\ldots,n\}$ eine beliebige Permutation, die nur die Zahlen $\geq k+1$ permutiert (d.h. $\pi(j) = j$ für $j \leq k$). Dann überzeugt man sich davon, dass für eine Frobeniusmatrix L_k von der Form (3.1.4) gilt:

$$\hat{\boldsymbol{L}}_k = \boldsymbol{P}_\pi\boldsymbol{L}_k\boldsymbol{P}_\pi^{-1} = \boldsymbol{P}_\pi\boldsymbol{L}_k\boldsymbol{P}_\pi^\top = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{\pi_k(k+1),k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{\pi_k(n),k} & & & 1 \end{pmatrix}. \tag{3.3.14}$$

Die oben eingeführten Permutationen $\boldsymbol{P}_k$ sind genau von der Art, dass bei den zugehörige Permutationen der Zahlen 1 bis n nur die Zahlen $\geq k+1$ permutiert werden. Also haben die Matrizen $\hat{\boldsymbol{L}}_k$ aus (3.3.13) die Darstellung (3.3.14). Aus Satz 3.1.3 folgt damit, dass

$$\boldsymbol{L} := \hat{\boldsymbol{L}}_1^{-1}\hat{\boldsymbol{L}}_2^{-1}\cdots\hat{\boldsymbol{L}}_{n-1}^{-1}$$

tatsächlich ein Element von $\mathcal{L}_n^1$ ist. Zusätzlich liefert Satz 3.1.3 die explizite Darstellung

$$\boldsymbol{L} = \begin{pmatrix} 1 & & & & \\ l_{\pi_1(2),1} & 1 & & & \\ l_{\pi_1(3),1} & l_{\pi_2(3),2} & 1 & & \\ \vdots & \vdots & & \ddots & \\ l_{\pi_1(n),1} & l_{\pi_2(n),2} & \cdots & l_{\pi_{n-1}(n),n-1} & 1 \end{pmatrix}$$

Die Permutationsmatrix P_0 gehört nach Definition zu einer Permutation der Zahlen 1 bis n, die gerade die Vertauschung der Zeilenindizes während Algorithmus 3.3.5 angibt. □

Bemerkung 3.3.11 Satz 3.3.10 zusammen mit Algorithmus 3.3.5 kann (zumindest bei der Handrechnung) dazu benutzt werden, festzustellen, ob eine gegebene Matrix $\boldsymbol{A}$ regulär ist, da Satz 3.3.10 zeigt, dass Algorithmus 3.3.5 nur abbricht, wenn die gegebene Matrix nicht regulär ist. ■

Bemerkung 3.3.12 (Rechtfertigung der Spaltenpivotsuche) Beispiel 3.3.2 zeigte, dass Spaltenpivotsuche die numerische Stabilität von Algorithmen erhöhen kann. Wie unsere Plausibilitätsbetrachtungen im Anschluss an Beispiel 3.3.1 gezeigt haben, ist die Spaltenpivotsuche eine sinnvolle Strategie unter einer Annahme an die Skalierung des Problems; wir nahmen nämlich an, dass die Einträge der Lösung und der rechte Seite des linearen Gleichungssystems von vergleichbarer Grösse sind. Diese Art der Skalierung kann man erreichen, indem man nicht das lineare Gleichungssystem $\boldsymbol{A}x = b$, sondern das System

$$(\boldsymbol{D}_1\boldsymbol{A}\boldsymbol{D}_2)x' = b' := \boldsymbol{D}_2 b, \qquad x = \boldsymbol{D}_2 x'$$

betrachtet, wobei $\boldsymbol{D}_1$, $\boldsymbol{D}_2$ geeignete Diagonalmatrizen sind. Algorithmisch ist es jedoch sehr schwer, $\boldsymbol{D}_1$, $\boldsymbol{D}_2$ zu finden; Programme zum Lösen linearer Gleichungssysteme überlassen deshalb die Skalierung meist dem Benutzer und verwenden höchstens die Spaltenpivotsuche.

■

Bemerkung 3.3.13 (Vollpivotsuche) Die Spaltenpivotsuche bei der $\boldsymbol{LR}$-Zerlegung verursacht zusätzliche Kosten $O(n^2)$, die im Verhältnis zu den Gesamtkosten $O(n^3)$ relativ klein sind. (Übung: Geben Sie genau an, wieviele Vergleiche bei der Spaltenpivotsuche gemacht werden müssen.)

Alternativ zur Spaltenpivotsuche kann auch eine Vollpivotsuche durchgeführt werden. Dabei wird in jedem Schritt des Gauss-Algorithmus das betragsmässig grösste Element der Restmatrix gesucht und durch Zeilen- und Spaltenvertauschungen zum Pivot gemacht. Man erhält auf diese Weise eine Zerlegung

$$\boldsymbol{P}_\pi \boldsymbol{A} \boldsymbol{P}_{\pi'} = \boldsymbol{LR}$$

der Matrix $\boldsymbol{A}$, wobei die Permutationen π, π' die Zeilen- und Spaltenvertauschungen repräsentieren. Die Einträge der berechneten Faktoren $\tilde{\boldsymbol{L}}$, $\tilde{\boldsymbol{R}}$ sind betragsmässig kleiner als bei Spaltenpivotsuche, so dass man erwartet, dass Gauss-Elimination mit Vollpivotsuche numerisch stabiler ist. Sie wird jedoch aus Kostengründen nur sehr selten eingesetzt, denn der zusätzliche Aufwand ist nun $O(n^3)$. ■

3.3.2 Existenz von $\boldsymbol{LR}$-Zerlegungen ohne Spaltenpivotsuche

Pivotstrategien sind oft nötig, um numerische Stabilität beim Lösen von Gleichungssystemen zu gewährleisten. Nachteile sind:

1. *Mehraufwand:* Für die Pivotsuche werden zusätzliche $O(n^2)$ Operationen benötigt.

2. *Verlust der Besetzungsstruktur:* Bei Matrizen mit bestimmten Besetzungsstrukturen (z.B. *Skyline-Matrizen*) kann die Besetzungsstruktur durch die Zeilenvertauschungen verloren gehen. Damit erhöhen sich die Kosten für die Berechnung und die Speicherung der Faktoren $\boldsymbol{L}$ und $\boldsymbol{R}$.

Es ist deshalb von Interesse, Klassen von Matrizen auszumachen, bei denen die $\boldsymbol{LR}$-Zerlegung auch ohne Pivotsuche durchgeführt werden kann.

Der folgende Satz charakterisiert die Menge der Matrizen, die eine $\boldsymbol{LR}$-Zerlegung besitzen:

Theorem 3.3.14 *Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ regulär. Dann besitzt $\boldsymbol{A}$ eine $\boldsymbol{LR}$-Zerlegung genau dann wenn alle Hauptminoren $\boldsymbol{A}_k$, $k = 1, \ldots, n$ regulär sind. Hierbei ist $\boldsymbol{A}_k \in \mathbb{R}^{k\times k}$ gegeben durch $(\boldsymbol{A}_k)_{ij} = A_{ij}$, i, $j = 1, \ldots, k$.*

Beweis: Übung. □

In der Praxis ist die Regularität der Hauptminoren nur schwer überprüfbar. Für die folgenden beiden Klassen von Matrizen ist jedoch die die Existenz einer $\boldsymbol{LR}$-Zerlegung gesichert:

1. *Symmetrisch positiv definite Matrizen:* Satz 3.2.3 garantiert, dass die Cholesky-Faktorisierung ohne Pivotsuche durchgeführt werden kann.

2. *Diagonal dominanten Matrizen*: der folgende Satz 3.3.15 zeigt, dass diagonal dominante Matrizen eine $\boldsymbol{LR}$-Zerlegung besitzen.

Wir rekapitulieren: Eine Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ heisst *Zeilen diagonaldominant*, falls

$$|A_{ii}| > \sum_{j\neq i} |A_{ij}| \qquad \forall i \in \{1, \ldots, n\}.$$

Theorem 3.3.15 *Es gilt: eine diagonal dominante Matrix hat eine $\boldsymbol{LR}$-Zerlegung, die mithilfe der Gauss-Elimination* ohne *Pivotsuche, d.h. Algorithmus 3.1.7, bestimmt werden kann.*

Beweis: Übung (vgl. Aufgabe 13). □

3.3.3 Nachiteration

Bei schlecht konditionierten Problemen kann es passieren, dass die Lösung, die man mit Algorithmus 3.3.8 erhält, stark fehlerbehaftet ist. Durch Rundungsfehler hat man z.B. nicht die exakte $\boldsymbol{LR}$-Zerlegung von $\boldsymbol{PA}$ sondern nur eine Approximation $\widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{R}}$. Anstatt nun die approximative $\boldsymbol{LR}$-Zerlegung zu verwerfen und mit erhöhter Rechengenauigkeit die gesamte Rechnung erneut durchzuführen, wendet man *Nachiteration* (engl.: *iterative refinement/improvement*) an.

Es soll $\boldsymbol{A}x = b$ berechnet werden, und es existiere eine (approximative) $\boldsymbol{LR}$-Zerlegung $\widetilde{L}\widetilde{R}$ von $\boldsymbol{A}$. Es wird zunächst eine Lösung x von $\widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{R}}x = b$ mithilfe der Algorithmen 3.1.1, 3.1.2 bestimmt. Durch Rundungsungenauigkeiten (auch schon beim Aufstellen von $\widetilde{\boldsymbol{L}}$ und $\widetilde{\boldsymbol{R}}$) kann es sein, dass die berechnete Lösung x zu ungenau ist. Um eine Verbesserung der Lösung zu erhalten, wird nun eine Korrektur Δx der berechneten Lösung x gesucht, d.h. wir verlangen, dass $x + \Delta x$ die Gleichung $\boldsymbol{A}(x + \Delta x) = b$ erfüllt. Somit erhalten für die gesuchte Korrektur Δx die sog. *Residualgleichung*

$$\boldsymbol{A}\Delta x = r := b - \boldsymbol{A}x. \tag{3.3.15}$$

Man beachte, dass das Residuum $r \neq 0$, weil x nicht die exakte Lösung unseres ursprünglichen Problems ist. Da für $\boldsymbol{A}$ bereits eine (approximative) $\boldsymbol{LR}$-Zerlegung vorliegt, erfolgt das Lösen der Residualgleichung (3.3.15) wieder einfach nur durch eine Vorwärts- und eine Rückwärtssubstitution. Der folgende Algorithmus formalisiert dieses Vorgehen.

Algorithmus 3.3.16 (Nachiteration)
input : (approximative) $\boldsymbol{LR}$-Zerlegung einer Matrix $\boldsymbol{A}$, rechte Seite b
output: Lösung x von $\boldsymbol{A}x = b$

1. `Bestimme Lösung` x `von` $\boldsymbol{LR}x = b$ `mithilfe von Algorithmen 3.1.1, 3.1.2`
2. `Bestimme das` ***`Residuum`*** $r = b - \boldsymbol{A}x$ `mit erhöhter Rechengenauigkeit`
3. `Falls das Residuum` r `zu gross ist (in einer geeigneten Norm), finde Korrektur` Δx `als Lösung von` $\boldsymbol{LR}\Delta x = r$ `mithilfe von Algorithmen 3.1.1, 3.1.2.`
4. `setze` $x := x + \Delta x$ `und gehe zu 2.`

Wesentlich in der Praxis ist dabei, dass das Residuum im 2. Schritt mit erhöhter Genauigkeit ausgewertet wird; wurde die $\boldsymbol{LR}$-Zerlegung z.B. nur in einfacher Genauigkeit bestimmt, so wird das Residuum in doppelter Genauigkeit ausgewertet. Da wir bereits gesehen hatten, dass die Vorwärts- und Rückwärtssubstitution relativ billig im Vergleich zur $\boldsymbol{LR}$-Zerlegung sind, ist die Nachiteration vom Standpunkt des Aufwandes eine attraktive Möglichkeit, die Genauigkeit einer rundungsfehlerbehafteten Lösung zu verbessern.

3.4 Störungssätze

3.4.1 Störungen und Kondition von Funktionen

In der Gleitkommaauswertung von Funktionen $f(x) \in V_2$ an der Stelle x in einem Vektorraum V_1 (x kann eine Matrix sein; dann wäre $V_1 = \mathbb{C}^{n\times n}$) muss, schon aufgrund von Rundungsfehlern bei der Darstellung von x in der Gleitkommaarithmetik $\mathbb{F}$, mit einem Fehler im Funktionswert $f(x)$ gerechnet werden. Es stellt sich die Frage, *wie sich ein Fehler $\delta \in V_1$ (z.B. Rundungsfehler bei Gleitkommadarstellung) im Argument x von $f(\cdot)$ bei der Auswertung von $f(\cdot)$ verstärkt.*

Hierzu nehmen wir an, dass $f : V_1 \longmapsto V_2$ an der Stelle $x \in V_1$ zweimal stetig (Frechet) differenzierbar ist, d.h. die Differentiale f', f'' existieren in einer hinreichend grossen Umgebung von x.

Dann gilt formal nach Taylor:

$$x \longmapsto f(x), \quad x + \delta \longmapsto f(x+\delta) \simeq f(x) + \delta f'(x) + O(\delta^2).$$

Die Grösse des Differentials f' an der Stelle x, d.h. $\|f'\|$, ist ein Mass für **absolute Sensitivität** von $f(x)$ gegen eine (kleine) Störung δ von x.

Für die Analyse der Fehlerfortpflanzung in Gleitkommaarithmetik $\mathbb{F}$ ist es besser, den sog. **relativen Fehler** zu betrachten, denn er ist unabhängig von der Skalierung von $f(x)$: für

$f(x) \neq 0$, $x \neq 0$, $\delta \to 0$ ist für $f(\cdot)$ (die zweimal Frechet differenzierbar sind an der Stelle x),

$$\begin{aligned}
\frac{\|f(x+\delta)-f(x)\|_{V_2}}{\|f(x)\|_{V_2}} &= \frac{\|f'(x)\delta + O(\delta^2)\|_{V_2}}{\|f(x)\|_{V_2}} \quad \Longrightarrow \\
\underbrace{\frac{\|f(x+\delta)-f(x)\|_{V_2}}{\|f(x)\|_{V_2}}}_{\text{rel. Fehler in } f} &\leq \left(\frac{\|f'(x)\|_{\mathcal{L}(V_1,V_2)}}{\|f(x)\|_{V_2}}\,\|x\|_{V_1}\right) \underbrace{\frac{\delta}{\|x\|_{V_1}}}_{\text{rel. Störung in } x} + O(\delta^2)\,.
\end{aligned}$$

Definition 3.4.1 *Die* ***Kondition der Funktion*** $x \longmapsto f(x)$ *an der Stelle* x *ist*

$$\operatorname{cond}\big(f(x)\big) := \frac{\|f'(x)\|_{\mathcal{L}(V_1,V_2)}}{\|f(x)\|_{V_2}}\,\|x\|_{V_1}\,.$$

Bemerkung 3.4.2 x kann in V_1, $f(x) \in V_2$ und $f'(x) \in \mathcal{L}(V_1, V_2)$ sein mit verschiedenen Normen.

3.4.2 Kondition einer Matrix

Im Rest dieses Kapitels nehmen wir durchweg an, dass $\|\circ\|_M$ **submultiplikative Matrixnorm** und $\|\circ\|_V$ dazu **kompatible Vektornorm** ist. Wir lassen zur Vereinfachung der Notation die Subskripten durchweg fallen – es sollte aus dem Zusammenhang klar sein, was gemeint ist.

Sei $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ nicht singulär und $f(\boldsymbol{A}) = \boldsymbol{A}^{-1}$. Wir betrachten die Kondition dieser Matrixfunktion an der "Stelle" $\boldsymbol{A}$.

Sei dazu weiter $\boldsymbol{\Delta} \in \mathbb{C}^{n\times n}$ "kleine" Störung von $\boldsymbol{A}$:

$$\|\boldsymbol{\Delta}\| \ll \|\boldsymbol{A}\|\,.$$

Sei zunächst $n = 1$. Dann ist $0 \neq \boldsymbol{A} \in \mathbb{C}$, $\boldsymbol{\Delta} = \delta \in \mathbb{C}$ und

$$\begin{aligned}
f(\boldsymbol{A}+\delta) = f(\boldsymbol{A}) + \delta\, f'(\boldsymbol{A}) + O(\delta^2) \quad &\Longrightarrow \\
(\boldsymbol{A}+\delta)^{-1} - \boldsymbol{A}^{-1} = \delta(-\boldsymbol{A}^{-2}) + O(\delta^2) \quad &\Longrightarrow \\
\frac{|(\boldsymbol{A}+\delta)^{-1} - \boldsymbol{A}^{-1}|}{|\boldsymbol{A}^{-1}|} &= \big|\boldsymbol{A}\big[(\boldsymbol{A}+\delta)^{-1} - \boldsymbol{A}^{-1}\big]\big| \\
&= |\boldsymbol{A}\delta(-\boldsymbol{A}^{-2}) + O(A\delta^2)| \\
&\leq |\boldsymbol{A}||\boldsymbol{A}^{-1}|\,\frac{|\delta|}{|\boldsymbol{A}|} + O(\delta^2)\,.
\end{aligned}$$

Für $n > 1$ rechnen wir aus:

$$\big[(\boldsymbol{A}+\boldsymbol{\Delta})^{-1} - \boldsymbol{A}^{-1}\big](\boldsymbol{A}+\boldsymbol{\Delta}) = \boldsymbol{1} - \boldsymbol{A}^{-1}(\boldsymbol{A}+\boldsymbol{\Delta}) = -\boldsymbol{A}^{-1}\boldsymbol{\Delta}\,,$$

also gilt

$$(\boldsymbol{A}+\boldsymbol{\Delta})^{-1} - \boldsymbol{A}^{-1} = -\boldsymbol{A}^{-1}\boldsymbol{\Delta}(\boldsymbol{A}+\boldsymbol{\Delta})^{-1}$$

und

$$\begin{aligned}\|(\boldsymbol{A}+\boldsymbol{\Delta})^{-1}-\boldsymbol{A}^{-1}\| &\le \|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{\Delta}\|\,\|(\boldsymbol{A}+\boldsymbol{\Delta})^{-1}-\boldsymbol{A}^{-1}+\boldsymbol{A}^{-1}\| \\ &\le \|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{\Delta}\|\big(\|(\boldsymbol{A}+\boldsymbol{\Delta})^{-1}-\boldsymbol{A}^{-1}\|+\|\boldsymbol{A}^{-1}\|\big)\end{aligned}$$

$\Longrightarrow$

$$\|(\boldsymbol{A}+\boldsymbol{\Delta})^{-1}-\boldsymbol{A}^{-1}\|\big(1-\|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{\Delta}\|\big)\le\|\boldsymbol{A}^{-1}\|^2\|\boldsymbol{\Delta}\|$$

$\Longrightarrow$

$$\begin{aligned}\frac{\|(\boldsymbol{A}+\boldsymbol{\Delta})^{-1}-\boldsymbol{A}^{-1}\|}{\|\boldsymbol{A}^{-1}\|} &\le \|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{\Delta}\|\big(1-\|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{\Delta}\|\big)^{-1} \\ &= \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|\,\frac{\|\boldsymbol{\Delta}\|}{\|\boldsymbol{A}\|}\,\big(1-\|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{\Delta}\|\big)^{-1} \\ &= \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|\,\frac{\|\boldsymbol{\Delta}\|}{\|\boldsymbol{A}\|}\,\sum_{i=0}^{\infty}\big(\|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{\Delta}\|\big)^i .\end{aligned}$$

Proposition 3.4.3 *(Störungsschranke bei Matrixinversion)*
Sei $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ nicht singulär und $\boldsymbol{\Delta}\in\mathbb{C}^{n\times n}$ Störung von $\boldsymbol{A}$ mit $\|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{\Delta}\|<1$. Dann gilt

$$\underbrace{\frac{\|(\boldsymbol{A}+\boldsymbol{\Delta})^{-1}-\boldsymbol{A}^{-1}\|}{\|\boldsymbol{A}^{-1}\|}}_{\textit{Rel. Fehler in } \boldsymbol{A}^{-1}}\le\|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|\,\underbrace{\frac{\|\boldsymbol{\Delta}\|}{\|\boldsymbol{A}\|}}_{\textit{Rel. Fehler in } \boldsymbol{A}}\,\Big(1+O(\|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{\Delta}\|)\Big).$$

Wir sehen, dass der Ausdruck $\|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|$ die *Kondition der Funktion $f(\boldsymbol{X})=\boldsymbol{X}^{-1}$ an der Stelle $\boldsymbol{X}=\boldsymbol{A}$* ist. Wegen ihrer Wichtigkeit bezeichnet man sie kurz als *Kondition* von $\boldsymbol{A}$, d.h.

$$\kappa(\boldsymbol{A}):=\|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\| \tag{3.4.1}$$

Beachte, dass die Kondition von der verwendeten Norm $\|\circ\|$ abhängt. Andere Schreibweisen für die Kondition von $\boldsymbol{A}$, die dies ausdrücken, sind:

$$\operatorname{cond}_{\|\circ\|}(\boldsymbol{A}),\qquad \boldsymbol{K}_{\|\circ\|}(\boldsymbol{A}),\qquad \kappa_{\|\circ\|}(\boldsymbol{A}),$$

oder, für den Fall, dass die Matrixnorm $\|\circ\|$ die p-Matrixnorm in Beispiel 1.12.4 ist,

$$\operatorname{cond}_p(\boldsymbol{A}),\qquad \boldsymbol{K}_p(\boldsymbol{A}),\qquad \kappa_p(\boldsymbol{A}).$$

3.4.3 Störungsanalyse linearer Gleichungssysteme

Betrachte für $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ nichtsingulär, $\underline{b}\in\mathbb{C}^n$:

$$\boldsymbol{A}\,\underline{x}=b\,. \tag{3.4.1}$$

Sei $\widetilde{\underline{b}} \neq \underline{b}$ Störung von $\underline{b}$ und $\widetilde{\underline{x}} = \boldsymbol{A}^{-1}\widetilde{\underline{b}}$ Lösung von

$$\boldsymbol{A}\,\widetilde{\underline{x}} = \widetilde{\underline{b}}\,. \tag{3.4.2}$$

Dann gilt

$$\underline{x} - \widetilde{\underline{x}} = \boldsymbol{A}^{-1}(\underline{b} - \widetilde{\underline{b}}) \qquad \Longrightarrow$$

$$\|\underline{x} - \widetilde{\underline{x}}\| \leq \|\boldsymbol{A}^{-1}\|\;\|\underline{b} - \widetilde{\underline{b}}\|\,. \tag{3.4.3}$$

Der absolute Fehler in $\underline{x}$ ist gross, falls $\|\boldsymbol{A}^{-1}\|$ gross ist. Der absolute Fehler ist insbesondere nicht invariant unter Skalierung der Matrix.

Ein besseres Fehlermass ist daher der **relative Fehler**

$$\frac{\|\underline{x} - \widetilde{\underline{x}}\|}{\|\underline{x}\|}\,.$$

Dazu: (3.4.1) $\Longrightarrow \|\underline{x}\| \geq \|\underline{b}\|/\|\boldsymbol{A}\|$. Aus (3.4.3) folgt dann,

$$\frac{\|\underline{x} - \widetilde{\underline{x}}\|}{\|\underline{x}\|} \leq \underbrace{\|\boldsymbol{A}\|\;\|\boldsymbol{A}^{-1}\|}_{\mathrm{cond}_{\|\circ\|}(\boldsymbol{A})}\;\frac{\|\underline{b} - \widetilde{\underline{b}}\|}{\|\underline{b}\|}\,. \tag{3.4.4}$$

Was passiert wenn in (3.4.2) auch $\boldsymbol{A}$ gestört wird? Sei dazu $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{E}$ mit Störung $\boldsymbol{E}$ so, dass

$$\|\boldsymbol{A}^{-1}\,\boldsymbol{E}\| < 1 \tag{3.4.5}$$

und $\widetilde{\underline{x}}$ Lösung (falls sie existiert!) von

$$\widetilde{\boldsymbol{A}}\,\widetilde{\underline{x}} = \widetilde{\underline{b}}\,. \tag{3.4.6}$$

Dann gilt

Theorem 3.4.4 *Für $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ invertierbar und Störung $\boldsymbol{E} \in \mathbb{C}^{n\times n}$ mit (3.4.5) ist $\widetilde{\boldsymbol{A}} = \boldsymbol{A}+\boldsymbol{E}$ invertierbar und (3.4.6) lösbar. Es gilt*

$$\frac{\|\underline{x} - \widetilde{\underline{x}}\|}{\|\underline{x}\|} \leq \frac{\|\boldsymbol{A}^{-1}\|\;\|\boldsymbol{A}\|}{1 - \|\boldsymbol{A}^{-1}\,\boldsymbol{E}\|}\left(\frac{\|\boldsymbol{E}\|}{\|\boldsymbol{A}\|} + \frac{\|\underline{b} - \widetilde{\underline{b}}\|}{\|\underline{b}\|}\right). \tag{3.4.7}$$

→ *N.B. (3.4.7) mit $\boldsymbol{E} = \boldsymbol{0} \Longrightarrow$ (3.4.4).*

Beweis von Theorem 3.4.4: wird unten gegeben.

Bemerkung 3.4.5 Beachte: (3.4.7) ist (wie jede Abschätzung) eine obere Schranke für die Norm des Fehlers Δx im Lösungsvektor; sie ist zwar scharf (d.h., es gibt Beispiele, in denen die obere Schranke (3.4.7) angenommen wird), aber dennoch in der Praxis oft pessimistisch - die **Richtung der Störung** $\Delta\underline{b}$ ist wichtig: falls nach Annahme $\boldsymbol{A}^{-1}$ existiert, so folgt $\sigma_n > 0$ und die SVD von $\boldsymbol{A}$ in Korollar 1.15.5 ergibt:

$$\boldsymbol{A} = \sum_{k=1}^{n} \sigma_k\, v_k\, u_k^H = V\boldsymbol{\Sigma}\boldsymbol{U}^H$$

$$\Longrightarrow \qquad \boldsymbol{A}^{-1} = \sum_{k=1}^{n} \sigma_k^{-1}\, u_k\, v_k^H \qquad \Longrightarrow$$

$$\underline{x} = \boldsymbol{A}^{-1}\,\underline{b} = \sum_{k=1}^{n} \frac{1}{\sigma_k}\,(\underline{v}_k^H\,\underline{b})\,\underline{u}_k\,, \tag{3.4.8}$$

und $\underline{\widetilde{x}} = \boldsymbol{A}^{-1}\underline{\widetilde{b}} \implies$

$$\underline{x} - \underline{\widetilde{x}} = \boldsymbol{A}^{-1}(\underline{b} - \underline{\widetilde{b}}) = \sum_{k=1}^{n} \frac{1}{\sigma_k} \left(\underline{v}_k^H \, (\underline{b} - \underline{\widetilde{b}})\right) \underline{u}_k \,, \tag{3.4.9}$$

$$\iff \qquad \Delta x = \sum_{k=1}^{n} \frac{1}{\sigma_k} \left(\underline{v}_k^H \, \Delta b\right) \underline{u}_k \,.$$

$\underline{u}_1, \ldots, \underline{u}_n$ ONB von $\mathbb{C}^n \implies$

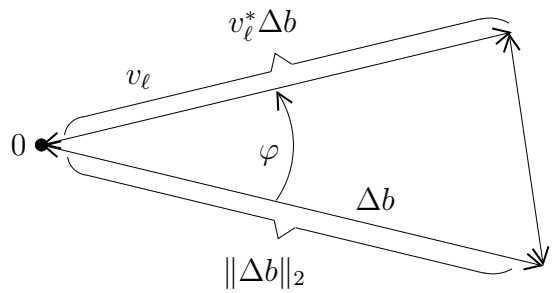
$$\begin{aligned}
\|\Delta x\|_2^2 &= \sum_{\ell=1}^{n} |\underline{u}_\ell^H \, \Delta x|^2 \\
&= \sum_{\ell=1}^{n} \left| \underline{u}_\ell^H \Big(\sum_{k=1}^{n} \sigma_k^{-1} (v_k^H \, \Delta b) \, \underline{u}_k \Big) \right|^2 \\
&= \sum_{\ell=1}^{n} \Big| \sum_{k=1}^{n} \sigma_k^{-1} \, (v_k^H \, \Delta b) \, \underbrace{\underline{u}_\ell^H \, \underline{u}_k}_{\delta_{\ell k}} \Big|^2 \qquad \implies
\end{aligned}$$

$$\boxed{\|\Delta x\|_2^2 = \sum_{\ell=1}^{n} \sigma_\ell^{-1} \, |v_\ell^H \, \Delta b|^2 \,.} \tag{3.4.10}$$

Grösse der Störung $\|\Delta x\|_2$ hängt entscheidend von Richtung der Störung Δb ab:
$v_1, \ldots, v_n$ ONB von $\mathbb{C}^n \implies$ Störung Δb:

$$\Delta b = \sum_{\ell=1}^{n} (v_\ell^H \, \Delta b) \, v_\ell \,.$$

Datenfehler in Richtung der Singulärvektoren v_ℓ **zu kleinen Singulärwerten σ_ℓ's** werden dramatisch **verstärkt**, da z.B.



$$\cos\varphi = \frac{v_\ell^H \, \Delta b}{\|v_\ell\|_2 \, \|\Delta b\|_2} = \frac{v_\ell^H \, \Delta b}{\|\Delta b\|_2} \qquad \implies \qquad v_\ell^H \, \Delta b = \|\Delta b\|_2 \cos\varphi \,.$$

Bemerkung 3.4.6 Falls

$$\Delta b \perp \operatorname{span}\{v_{r+1}, \ldots, v_n\} \qquad \implies \qquad \frac{\|\Delta x\|_2}{\|x\|_2} \le \frac{\sigma_1}{\sigma_r} \, \frac{\|\Delta b\|_2}{\|b\|_2} \,.$$

3.5 Konvergente Reihen von Matrizen. Neumann'sche Reihe.

Zum Beweis von Satz 3.4.4 sowie später für die Konvergenz iterativer Verfahren zur Gleichungslösung brauchen wir die Verallgemeinerung geometrischer Reihen auf Matrizen. Wir erinnern daran, dass im Folgenden $\|\circ\|$ immer eine **submultiplikative Matrixnorm** darstellt.

Motivation: sei $q \in \mathbb{C}$. Dann gilt für alle $m \in \mathbb{N}$

$$(1-q)(1+q+q^2+\cdots+q^m) = 1-q^{m+1}\,.$$

Falls $|q| < 1$, gilt $|q|^{m+1} \to 0$ für $m \to \infty$ und

$$1 = \lim_{m\to\infty}(1-q^{m+1}) = (1-q)\lim_{m\to\infty}\sum_{i=0}^{m} q^i = (1-q)\sum_{i=0}^{\infty} q^i\,,$$

d.h. es gilt die wohlbekannte Summationsformel für die geometrische Reihe:

$$(1-q)^{-1} = \sum_{i=0}^{\infty} q^i\,.$$

Sei nun $\boldsymbol{B} \in \mathbb{C}^{n\times n}$ eine Matrix. Dann gilt für alle $m \in \mathbb{N}$

$$(\mathbf{1}-\boldsymbol{B})(\mathbf{1}+\boldsymbol{B}+\boldsymbol{B}^2+\cdots+\boldsymbol{B}^m) = \mathbf{1}-\boldsymbol{B}^{m+1}\,.$$

Falls $\|\boldsymbol{B}\| < 1$ in einer submultiplikativen Matrixnorm $\|\circ\|$ ist, folgt für $m \to \infty$

$$\|\boldsymbol{B}^{m+1}\| = \|\boldsymbol{B}^m\boldsymbol{B}\| \le \|\boldsymbol{B}^m\|\,\|\boldsymbol{B}\| \le \cdots \le \|\boldsymbol{B}\|^{m+1} \to 0\,.$$

Somit existiert der Grenzwert der **Neumann'schen Reihe**

$$\lim_{m\to\infty}\sum_{i=0}^{m}\boldsymbol{B}^i = \sum_{i=0}^{\infty}\boldsymbol{B}^i \tag{3.5.1}$$

und es gilt

$$(\mathbf{1}-\boldsymbol{B})^{-1} = \sum_{i=0}^{\infty}\boldsymbol{B}^i.$$

Insbesondere ist in diesem Fall $\mathbf{1}-\boldsymbol{B}$ invertierbar.

Problem: das obige Argument zeigt, dass $\|\boldsymbol{B}\| < 1$ hinreichend ist für die Konvergenz der Neumann'schen Reihe *in dieser Norm*. Damit ist das Konvergenzkriterium (nicht aber die Konvergenztopologie! Warum?) abhängig von der Wahl der Norm $\|\circ\|$.

Gibt es eine notwendige und hinreichende Bedingung für die Konvergenz der Neumann'schen Reihe?

Für die Formulierung einer notwendigen und hinreichenden Bedingung benutzt man den Spektralradius $\rho(\boldsymbol{B})$ einer Matrix $\boldsymbol{B}$ – wir erinnern daran, dass $\rho(\cdot)$ *keine Matrixnorm* darstellt (Übung! zeigen Sie, dass $\rho(\cdot)$ die Normeigenschaft (N1) nicht erfüllt). Der Beweis des folgenden Satzes zeigt aber, dass es für jedes $\varepsilon > 0$ eine (von ε abhängige) Matrixnorm $\|\circ\|_\varepsilon$ gibt derart, dass für alle Matrizen gilt:

$$\forall \boldsymbol{B} \in \mathbb{C}^{n\times n}: \quad \|\boldsymbol{B}\|_\varepsilon \le \rho(\boldsymbol{B}) + \varepsilon$$

Theorem 3.5.1 *Sei $\boldsymbol{B} \in \mathbb{C}^{n\times n}$. Dann gilt*

$$\boldsymbol{B}^m \to \boldsymbol{0} \text{ für } m \to \infty \Longleftrightarrow \rho(\boldsymbol{B}) < 1\,.$$

Beweis: "$\Longleftarrow$: Wir zeigen: ist $\rho(\boldsymbol{B}) < 1 \Longrightarrow$ existiert eine submultiplikative Matrixnorm $\|\circ\|_*$ mit $\|\boldsymbol{B}\|_* < 1$. Daraus folgt wegen $\|\boldsymbol{B}^m\|_* \le \|\boldsymbol{B}\|_*^m \to 0$, dass $\boldsymbol{B}^m \to \boldsymbol{0}$ für $m \to \infty$.

Sei $\boldsymbol{B} \in \mathbb{C}^{n\times n}$ und $\rho(\boldsymbol{B}) < 1$. Der Satz 1.9.1 von Schur impliziert: es existiert eine unitäre Matrix $\boldsymbol{U} \in \mathbb{C}^{n\times n}$ mit

$$\boldsymbol{U}^H \boldsymbol{B}\boldsymbol{U} = \begin{pmatrix} \lambda_1 & & * \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} =: \boldsymbol{T} = (t_{ij})_{i,j=1}^n\,.$$

Sei $\varepsilon > 0$ und $\boldsymbol{D}_\varepsilon := \operatorname{diag}\{1, \varepsilon, \ldots, \varepsilon^{n-1}\}$. Dann

$$\boldsymbol{D}_\varepsilon^{-1}\,\boldsymbol{U}^H\,\boldsymbol{B}\boldsymbol{U}\,\boldsymbol{D}_\varepsilon = (t_{ij}\,\varepsilon^{j-i})_{i,j=1}^n$$

$$= \begin{pmatrix} \lambda_1 & t_{12}\,\varepsilon & & \ldots & t_{1n}\,\varepsilon^{n-1} \\ & \ddots & t_{23}\,\varepsilon & & \\ & & \ddots & \ddots & \\ & 0 & & \ddots & t_{n-1n}\,\varepsilon \\ & & & \ddots & \\ & & & & \lambda_n \end{pmatrix}.$$

Wähle $\varepsilon > 0$ hinreichend klein und definiere

$$\|\boldsymbol{B}\|_* := \|\boldsymbol{D}_\varepsilon^{-1}\,\boldsymbol{U}^H\,\boldsymbol{B}\boldsymbol{U}\,\boldsymbol{D}_\varepsilon\|_\infty\,.$$

$\|\circ\|_*$ ist submultiplikative Matrixnorm und es gilt $\|\boldsymbol{B}\|_* < 1$, denn

$$\rho(\boldsymbol{B}) < 1 \Longrightarrow |\lambda_1| \le |\lambda_2| \le \cdots \le |\lambda_n| = \rho(\boldsymbol{B}) < 1 \Longrightarrow$$

$$\begin{aligned}
\|\boldsymbol{B}\|_* &= \max_{1\le i\le n} \Big(\sum_{j=1}^n |t_{ij}|\,\varepsilon^{j-i}\Big) \\
&= \max_{1\le i\le n} \Big(|\lambda_i| + \sum_{j=i+1}^n |t_{ij}|\,\varepsilon^{j-i}\Big) \\
&\le \rho(\boldsymbol{B}) + \max_{1\le i\le n-1} \sum_{k=1}^{n-i} |t_{i,i+k}|\,\varepsilon^k \\
&\le \rho(\boldsymbol{B}) + \varepsilon \underbrace{\max_{1\le i\le n-1} \sum_{k=1}^{n-i} |t_{i,i+k}|}_{\boldsymbol{A}} \\
&< 1 - \delta
\end{aligned}$$

falls $\varepsilon < \big(1 - \delta - \rho(\boldsymbol{B})\big)/\boldsymbol{A}$.

$\| \circ \|_*$ ist submultiplikative Matrixnorm. Also gilt $\|\boldsymbol{B}^m\|_* \leq \|\boldsymbol{B}\|_*^m \to 0$ für $m \to \infty$, somit $\boldsymbol{B}^m \to \boldsymbol{0}$ für $m \to \infty$.

"$\Longrightarrow$" per Widerspruch: es gelte $\boldsymbol{B}^m \to \boldsymbol{0}$ und $\rho(\boldsymbol{B}) \geq 1$, obdA sei $|\lambda_1| = \rho(\boldsymbol{B}) \geq 1$.
Der Satz 1.9.1 von Schur impliziert, dass

$$\boldsymbol{U}^H \boldsymbol{B} \boldsymbol{U} = \boldsymbol{T} := \begin{pmatrix} \lambda_1 & & * \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}, \quad \text{mit } \boldsymbol{U} \text{ unitär}.$$

$\Longrightarrow$

$$\boldsymbol{B}^m = \boldsymbol{U}^H \underbrace{\boldsymbol{B} \underbrace{\boldsymbol{U}\boldsymbol{U}^H}_{\boldsymbol{1}} \boldsymbol{B}\boldsymbol{U} \ldots \boldsymbol{U}^H \boldsymbol{B}\boldsymbol{U}}_{m\text{-mal}}$$

$$= \begin{pmatrix} \lambda_1 & & * \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}^m = \begin{pmatrix} \lambda_1^m & & * \\ & \ddots & \\ 0 & & \lambda_n^m \end{pmatrix}.$$

Falls $|\lambda_1| \geq 1 \Longrightarrow |\lambda_1|^m \geq 1 \Longrightarrow \boldsymbol{B}^m \not\to \boldsymbol{0}$ für $m \to \infty$, Widerspruch. Also muss gelten:

$$\boldsymbol{B}^m \to \boldsymbol{0} \Longrightarrow \rho(\boldsymbol{B}) < 1.$$

□

Korollar 3.5.2 *Sei $\boldsymbol{B} \in \mathbb{C}^{n \times n}$ mit $\rho(\boldsymbol{B}) < 1$.*

Dann gilt:

i) $(\boldsymbol{1} - \boldsymbol{B})^{-1}$ existiert,

ii) die Neumann-Reihe $\sum_{j=0}^{\infty} \boldsymbol{B}^j$ konvergiert, und

$$(\boldsymbol{1} - \boldsymbol{B})^{-1} = \sum_{j=0}^{\infty} \boldsymbol{B}^j. \tag{3.5.2}$$

Beweis von Theorem 3.4.4: Es ist

$$\boldsymbol{A}\,\underline{x} = \underline{b}$$

und

$$\underbrace{(\boldsymbol{A} + \boldsymbol{E})}_{\widetilde{\boldsymbol{A}}}\,\underline{\widetilde{x}} = \underbrace{\underline{b} + \underline{e}}_{\underline{\widetilde{b}}}$$

$$\underline{\widetilde{x}} = \underline{\widetilde{b}},$$

sowie $\|\boldsymbol{A}^{-1}\,\boldsymbol{E}\| < 1$. $\boldsymbol{A}$ invertierbar $\Longrightarrow$

$$\begin{aligned}(\boldsymbol{A}+\boldsymbol{E})^{-1} &= \big(\boldsymbol{A}(\boldsymbol{1}+\boldsymbol{A}^{-1}\boldsymbol{E})\big)^{-1}\\ &= (\boldsymbol{1}-\boldsymbol{B})^{-1}\,\boldsymbol{A}^{-1},\ \text{ wobei } \boldsymbol{B} := -\boldsymbol{A}^{-1}\boldsymbol{E}\end{aligned}$$

Wegen $\|\boldsymbol{B}\| = \|\boldsymbol{A}^{-1}\,\boldsymbol{E}\| < 1$ folgt aus Korollar 3.5.2, dass $(\boldsymbol{1}-\boldsymbol{B})^{-1}$ existiert, also auch

$$\widetilde{\boldsymbol{A}}^{-1} = (\boldsymbol{A}+\boldsymbol{E})^{-1} = (\boldsymbol{1}-\boldsymbol{B})^{-1}\,\boldsymbol{A}^{-1}\,.$$

Damit folgt

$$\begin{aligned}\underline{\widetilde{x}} - \underline{x} &= (\boldsymbol{A}+\boldsymbol{E})^{-1}(\underline{b}+\underline{e}) - \boldsymbol{A}^{-1}\,\underline{b}\\ &= \big[(\boldsymbol{A}+\boldsymbol{E})^{-1} - \boldsymbol{A}^{-1}\big]\,\underline{b} + (\boldsymbol{A}+\boldsymbol{E})^{-1}\,\underline{e}\\ &= \big[(\boldsymbol{1}+\underbrace{\boldsymbol{A}^{-1}\,\boldsymbol{E}}_{-\boldsymbol{B}})^{-1}\,\boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\big]\,\underline{b} + (\boldsymbol{1}+\boldsymbol{A}^{-1}\,\boldsymbol{E})^{-1}\,\boldsymbol{A}^{-1}\,\underline{e}\\ &= \Big[\Big(\sum_{k=0}^{\infty}\boldsymbol{B}^k\Big) - \boldsymbol{1}\Big]\,\boldsymbol{A}^{-1}\,\underline{b} + \Big[\sum_{k=0}^{\infty}(-\boldsymbol{A}^{-1}\,\boldsymbol{E})^k\Big]\,\boldsymbol{A}^{-1}\,\underline{e}\\ &= \Big[\Big(\sum_{k=1}^{\infty}(-\boldsymbol{A}^{-1}\,\boldsymbol{E})^k\Big)\ \underbrace{\boldsymbol{A}^{-1}\,\underline{b}}_{\underline{x}} + \Big[\sum_{k=0}^{\infty}(-\boldsymbol{A}^{-1}\,\boldsymbol{E})^k\Big]\,\boldsymbol{A}^{-1}\,\underline{e}\,,\end{aligned}$$

und somit

$$\|\underline{x}-\underline{\widetilde{x}}\| \le \Big[\sum_{k=1}^{\infty}\|\boldsymbol{A}^{-1}\,\boldsymbol{E}\|^k\Big]\,\|\underline{x}\| + \Big[\sum_{k=0}^{\infty}\|\boldsymbol{A}^{-1}\,\boldsymbol{E}\|^k\Big]\,\|\boldsymbol{A}^{-1}\|\,\|\underline{e}\|\,.$$

Mit $\|\underline{b}\| = \|\boldsymbol{A}\underline{x}\| \le \|\boldsymbol{A}\|\,\|\underline{x}\| \Longrightarrow \|\underline{x}\|^{-1} \le \|\boldsymbol{A}\|/\|\underline{b}\|$ folgt

$$\frac{\|\underline{x}-\underline{\widetilde{x}}\|}{\|\underline{x}\|} \le \|\boldsymbol{A}^{-1}\,\boldsymbol{E}\|\,\Big[\sum_{k=0}^{\infty}\|\boldsymbol{A}^{-1}\,\boldsymbol{E}\|^k\Big] + \Big[\sum_{k=0}^{\infty}\|\boldsymbol{A}^{-1}\,\boldsymbol{E}\|^k\Big]\,\|\boldsymbol{A}^{-1}\|\,\|\underline{e}\|\,\|\boldsymbol{A}\|/\|b\|\,,$$

woraus folgt

$$\begin{aligned}\frac{\|\underline{x}-\underline{\widetilde{x}}\|}{\|\underline{x}\|} &\le \frac{1}{1-\|\boldsymbol{A}^{-1}\,\boldsymbol{E}\|}\Bigg[\underbrace{\|\boldsymbol{A}^{-1}\,\boldsymbol{E}\|}_{\le\|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{A}\|\,\|\boldsymbol{E}\|/\|\boldsymbol{A}\|} + \underbrace{\|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|}_{\mathrm{cond}_{\|\circ\|}(\boldsymbol{A})}\,\frac{\|\underline{e}\|}{\|\underline{b}\|}\Bigg]\\ &\le \frac{\|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{A}\|}{1-\|\boldsymbol{A}^{-1}\,\boldsymbol{E}\|}\,\Big[\frac{\|\boldsymbol{E}\|}{\|\boldsymbol{A}\|} + \frac{\|\underline{e}\|}{\|\underline{b}\|}\Big]\,.\end{aligned} \tag{3.5.3}$$

□

3.6 Systeme mit Rechteckmatrizen

Bisher untersuchten wir Verfahren für Lösung linearer Gleichungssteme $\boldsymbol{A}\underline{x} = \underline{b}$ mit quadratischer Matrix $\boldsymbol{A}$. In vielen Anwendungen tritt das Problem auf, lineare Gleichungssysteme

$$\boldsymbol{A}\underline{x} = \underline{b} \tag{3.6.1}$$

zu "lösen", wobei

$$\boldsymbol{A} \in \mathbb{C}^{m\times n} \quad \text{ist mit} \quad r = \operatorname{rank}(\boldsymbol{A}) \le n \le m, \quad \underline{x} \in \mathbb{C}^n, \quad \underline{b} \in \mathbb{C}^m, \tag{3.6.2}$$

Einfache Beispiele zeigen, dass (3.6.1) entweder genau eine, keine oder unendliche viele Lösungen besitzen kann. Es ist daher zuerst nach einem Lösungsbegriff fuer (3.6.1) zu fragen: was bedeutet es, (3.6.1) zu lösen? In der Praxis verlangt man, dass eine Lösung $\underline{x}$ von (3.6.1) das *Residuum*

$$\underline{r}(\underline{x}) := \underline{b} - \boldsymbol{A}\underline{x}$$

möglichst klein macht.

3.6.1 Motivation: Gauss'sche Methode der kleinsten Quadrate

Bei Ausgleichsproblemen handelt es sich um das Problem, m Messdaten (t_i, y_i), $i = 1, ..., m$, mit einem mathematischen Gesetz eines *vorgegebenen Typs* in Einklang zu bringen. Dabei ist die Form des mathematischen Gesetzes bis auf einige Parameter x_j, $j = 1, ..., n$, festgelegt, welche durch die Messungen bestimmt werden sollen:

$$y(t) = f(t) := \sum_{j=1}^{n} x_j \Phi_j(t) \tag{3.6.3}$$

mit vorgegebenen Funktionen $\Phi_j(t)$.

Typischerweise tritt dann das Problem auf, dass mehr Messungen gemacht wurden als unbekannte Parameter vorliegen, d.h.

$$m > n,$$

und die Messungen selbst durch *Messfehler* nicht exakt dem vorgegebenen physikalischen Gesetz entsprechen. Man erhält damit ein *überbestimmtes* Gleichungssystem, das nicht exakt gelöst werden kann.

Bei Ausgleichsproblemen handelt es sich darum, die Parameter x_j in (3.1.3) so zu finden, dass die mathematischen Gesetze von den Messwerten y_i "möglichst gut" erfüllt werden.

Im vorliegenden Kapitel wird die Gauss'sche Methode der kleinsten Fehlerquadrate vorgestellt. Diese ist anwendbar bei Problemen, bei denen die zu bestimmenden Parameter x_j *linear* in das mathematische Gesetz $\Phi(t)$ eingehen.

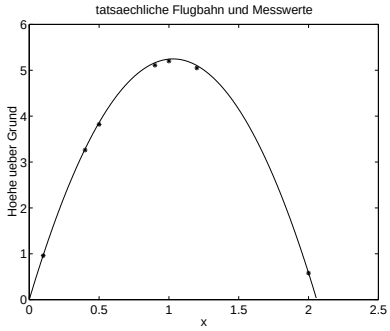
Beispiel 3.6.1 Unter Einfluss der Schwerkraft fliegen geworfene Körper auf Parabeln. Hat der Körper die Anfangsgeschwindigkeit $v = (v_x, v_y)$ zum Zeitpunkt $t = 0$ am Punkt $(0, 0)$ und fliegt er anschliessend nur unter Einfluss der Schwerkraft, so ist er zum Zeitpunkt $t > 0$ am Ort

$$x = v_x t, \qquad y = v_y t - \frac{1}{2} g t^2, \qquad \text{wobei } g \text{ die Erdbeschleunigung ist.}$$

Es sei die Anfangsgeschwindigkeit v_y und die Erdbeschleunigung g unbekannt; diese sollen aus Messungen bestimmt werden. Hierzu wurde die Höhe über Grund des Körpers zu folgenden Zeiten gemessen:

Es ergeben sich damit $m = 7$ Gleichungen für die $n = 2$ unbekannten Parameter v_y und g:

$$y_i = t_i v_y - \frac{1}{2} t_i^2 g, \qquad i = 1, \ldots, 7.$$



i	1	2	3	4	5	6	7
t_i[s]	0.1	0.4	0.5	0.9	1.0	1.2	2.0
y_i[m]	0.96	3.26	3.82	5.11	5.2	5.05	0.58

Figur 3.3: Flugbahn eines Körpers und gemessene Positionen.

Führt man die Matrix $\boldsymbol{A} \in \mathbb{R}^{7\times 2}$ mit

$$A_{i1} = \Phi_1(t_i) = t_i, \qquad \boldsymbol{A}\, A_{i2} = \Phi_2(t_i) = -\frac{1}{2}t_i^2, \qquad i = 1, \ldots, 7,$$

ein, so ergibt sich

$$\boldsymbol{A}\begin{pmatrix} v_y \\ g \end{pmatrix} = y,$$

d.h.

$$\begin{pmatrix} 0.1 & -0.005 \\ 0.4 & -0.08 \\ 0.5 & -0.125 \\ 0.9 & -0.405 \\ 1.0 & -0.5 \\ 1.2 & -0.72 \\ 2.0 & -2.0 \end{pmatrix} \cdot \begin{pmatrix} v_y \\ g \end{pmatrix} = \begin{pmatrix} 0.96 \\ 3.26 \\ 3.82 \\ 5.11 \\ 5.2 \\ 5.05 \\ 0.58 \end{pmatrix} \tag{3.6.4}$$

■

In Beispiel 3.6.1 erhielten wir ein überbestimmtes Gleichungssystem, das keine klassische Lösung hat. Die Ausgleichsrechnung liefert nun eine Methode, sinnvolle Lösungen von überbestimmten Gleichungssystemen zu definieren.

Für $m \geq n$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ werde eine "Lösung" $x \in \mathbb{R}^n$ des folgenden überbestimmten Gleichungssystems gesucht:

$$\sum_{j=1}^{n} A_{ij}x_j = b_i \qquad i = 1, \ldots, m. \tag{3.6.5}$$

Da wir für $m > n$ diese m Gleichungen i.a. nicht alle exakt erfüllen können, führen wir die sogenanten *Residuen* r_i ein, die messen, inwieweit die i-te Gleichung verletzt ist:

$$r_i := b_i - \sum_{j=1}^{n} A_{ij}x_j, \qquad i = 1, \ldots, m.$$

In Matrixschreibweise ist dies

$$r = b - \boldsymbol{A}x, \tag{3.6.6}$$

wobei der Residuumsvektor $r \in \mathbb{R}^m$ die Komponenten r_i hat. Wir suchen nach Vektoren $x \in \mathbb{R}^n$, für die das Residuum r möglichst "klein" (in einer zu wählenden Vektornorm!) ist.

Ein $x \in \mathbb{R}^n$, das das Residuum minimiert, wird *Ausgleichslösung* von (3.6.1) genannt.

Definition 3.6.2 (Ausgleichsproblem) *Sei $\|\cdot\|$ eine Norm auf $\mathbb{R}^m$. Für eine Matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ und ein $b \in \mathbb{R}^m$ heisst das Problem*

$$\text{Finde } x \in \mathbb{R}^n \text{ so, dass} \quad \|\boldsymbol{A}x - b\| \leq \|\boldsymbol{A}y - b\| \qquad \forall y \in \mathbb{R}^n \tag{3.6.7}$$

Ausgleichsproblem.

Eine Lösung $x \in \mathbb{R}^n$ eines Ausgleichsproblems heisst Ausgleichslösung.

Bemerkung 3.6.3 Der Lösungsbegriff in (3.6.7) ist eine Verallgemeinerung der klassischen Lösung. Ist nämlich $x \in \mathbb{R}^n$ eine klassische Lösung, d.h. gilt $\boldsymbol{A}x = b$, dann ist offensichtlich x ebenfalls eine Lösung von (3.6.7). ∎

Bemerkung 3.6.4 Offensichtlich hängt die Ausgleichslösung von der Wahl der Norm $\|\cdot\|$ ab, bzgl. der das Residuum minimiert wird. Im folgenden werden wir bei Ausgleichsproblemen die Norm $\|\cdot\|$ stets als die $\|\cdot\|_2$-Norm wählen. Ein wesentlicher Grund für der Wahl der $\|\cdot\|_2$-Norm, ist, dass ein relativ einfaches Rechenschema entsteht. Auch die Namensgebung, "Methode der kleinsten Fehlerquadrate" erklärt sich aus der Wahl der Norm: Für $\|\cdot\| = \|\cdot\|_2$ ist (3.6.7) äquivalent zum Minimieren von $\|r\|_2^2 = \sum_{i=1}^m r_i^2$, d.h. dem Minimieren der Summe der Quadrate der Residuen (Fehler). ∎

3.6.2 Normalengleichungen

Wir wenden uns nun der Frage zu, wie eine Lösung $x \in \mathbb{R}^n$ des Problems (3.6.7) bestimmt werden kann. Hierzu wollen wir eine Gleichung für das gesuchte $x \in \mathbb{R}^n$ herleiten.

Sei $x \in \mathbb{R}^n$ eine Lösung von (3.6.7). Dann gilt nach Definition

$$\Phi(x) := \|b - \boldsymbol{A}x\|_2^2 \leq \|b - \boldsymbol{A}y\|_2^2 \qquad \forall y \in \mathbb{R}^n. \tag{3.6.8}$$

Wir betrachten nun die *Richtungsableitung* des Funktionals Φ an einem Minimum $x \in \mathbb{R}^n$ in Richtung $0 \neq z \in \mathbb{R}^n$.

Wir betrachten dazu Variationen eines Lösungsvektors x von (3.6.8) der Form $y = x + tz$, wobei die Richtung $0 \neq z \in \mathbb{R}^n$ beliebig ist mit Norm $\|z\|_2 = 1$ und die "Schrittlänge" von x in Richtung z gegeben ist durch $0 < t \in \mathbb{R}$.

Aus (3.6.8) folgt dann, dass die Funktion

$$\begin{aligned} \Phi_z : \mathbb{R} &\to \mathbb{R}_0^+ \\ t &\mapsto \Phi_z(t) := \|b - \boldsymbol{A}(x + tz)\|_2^2 \end{aligned}$$

ein Minimum bei $t = 0$ hat.

Wir rechnen dazu nach, dass die Funktion Φ_z ein quadratisches Polynom in t ist: es gilt

$$\begin{aligned} \Phi_z(t) &= \|b - \boldsymbol{A}(x + tz)\|_2^2 = (b - \boldsymbol{A}(x + tz))^\top (b - \boldsymbol{A}(x + tz)) \\ &= (x + tz)^\top \boldsymbol{A}^\top \boldsymbol{A}(x + tz) - 2(x + tz)^\top \boldsymbol{A}^\top b + b^\top b \\ &= t^2 \left(z^\top \boldsymbol{A}^\top Az\right) + 2t \left(z^\top \boldsymbol{A}^\top \boldsymbol{A}x - z^\top \boldsymbol{A}^\top b\right) + \left(x^\top \boldsymbol{A}^\top \boldsymbol{A}x + b^\top b\right). \end{aligned}$$

Da Φ_z ein Minimum bei $t = 0$ hat, muss also gelten:

$$0 = \Phi_z'(0) = 2\left(z^\top \boldsymbol{A}^\top \boldsymbol{A}x - z^\top \boldsymbol{A}^\top b\right) = 2z^\top \left(\boldsymbol{A}^\top \boldsymbol{A}x - \boldsymbol{A}^\top b\right). \tag{3.6.9}$$

Da $\boldsymbol{A}^\top \boldsymbol{A}x - \boldsymbol{A}^\top b \in \mathbb{R}^m$ und (3.6.9) für *jedes* $z \in \mathbb{R}^m$ gilt, erhalten wir als notwendige Bedingung für die gesuchte Lösung $x \in \mathbb{R}^m$:

$$\boldsymbol{A}^\top \boldsymbol{A}x = \boldsymbol{A}^\top b. \tag{3.6.10}$$

Gleichung (3.6.10) wird als *Normalengleichung* bezeichnet. Wir halten dieses Ergebnis in dem folgenden Satz fest:

Theorem 3.6.5 *Sei $m \geq n$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ und $n = \operatorname{rank}(\boldsymbol{A}) \leq m$.*

Eine Lösung $x \in \mathbb{R}^n$ des Ausgleichsproblems (3.6.7) erfüllt die Normalengleichung (3.6.10).

Umgekehrt ist jede Lösung x der Normalengleichung (3.6.10) auch Lösung des Ausgleichsproblems (3.6.7).

Beweis: Obige Herleitung der Normalengleichnung zeigt die Notwendigkeit von (3.6.10). dass (3.6.10) auch hinreichend für Lösungen des Ausgleichsproblems (3.6.7) ist, erkennt man wie folgt.

Sei x Lösung von (3.6.10) und $z \in \mathbb{R}^n$ beliebig. Eigenschaft (3.6.10) impliziert, dass das quadratische Polynom $\Phi_z(t)$ ein Minimum bei $t = 0$ hat. Damit ist

$$\|\boldsymbol{A}x - b\|_2^2 = \Phi_z(0) \leq \Phi_z(t) = \|b - \boldsymbol{A}(x + tz)\|_2^2 \qquad \forall t \in \mathbb{R}.$$

Weil $z \in \mathbb{R}^n$ beliebig war, schliessen wir weiter

$$\|\boldsymbol{A}x - b\|_2^2 = \Phi_z(0) \leq \Phi_z(t) = \|b - \boldsymbol{A}(x + tz)\|_2^2 \qquad \forall t \in \mathbb{R}, \quad z \in \mathbb{R}^m.$$

Also löst x (3.6.7). □

In der Praxis sind die Spalten von $\boldsymbol{A}$ linear unabhängig und damit ist die Matrix $\boldsymbol{A}^\top A$ symmetrisch positiv definit (Beweisen!). Es gibt dann nach Satz 3.6.5 eine eindeutige Lösung von (3.6.7). Das Gleichungssystem (3.6.10) kann man z.B. mithilfe der Cholesky-Zerlegung der SPD-Matrix $\boldsymbol{A}^\top \boldsymbol{A}$ lösen. Es ergibt sich dann folgender Algorithmus:

Algorithmus 3.6.6 (Methode der Normalengleichungen)
input: `Matrix` $A \in \mathbb{R}^{m\times n}$ $m \geq n$ `(Annahme: Spalten von` $\boldsymbol{A}$ `linear unabhängig),`
`Vektor` $b \in \mathbb{R}^m$,
output: $x \in \mathbb{R}^n$ mit $\|b - \boldsymbol{A}x\|_2 \leq \|b - \boldsymbol{A}y\|_2$ für alle $y \in \mathbb{R}^n$

1. *$C := \boldsymbol{A}^\top A$,*
2. *Bestimme Cholesky-Faktor $\boldsymbol{L}$ von C mithilfe von Alg. 3.2.5 (d.h. $\boldsymbol{L}\boldsymbol{L}^\top = C$).*
3. *$b' := \boldsymbol{A}^\top b$.*
4. *Löse $\boldsymbol{L}y = b'$ mithilfe der Vorwärtssubstitution (Alg. 3.1.1).*
5. *Löse $\boldsymbol{L}^\top x = y$ mithilfe der Rückwärtssubstitution (Alg. 3.1.2).*

return (x)

Aufstellen von $\boldsymbol{A}^\top A$	$\frac{1}{2}n(n+1)m$
Cholesky-Zerlegung von $\boldsymbol{A}^\top \boldsymbol{A}$	$\frac{1}{6}n^3$
$\boldsymbol{A}^\top b$	mn
Vorwärtssubstitution	$\frac{1}{2}n^2$
Rückwärtssubstitution	$\frac{1}{2}n^2$

Tabelle 3.3: Wesentliche Operationen (= eine Addition plus eine Multiplikation) bei Lösung eines Ausgleichsproblems mithilfe der Normalengleichung.

Beispiel 3.6.7 Wir kommen nun auf das Einleitungsbeispiel 3.6.1 zurück. Für das überbestimmte System (3.6.4) ist die gesuchte Normalengleichung $\boldsymbol{A}^\top \boldsymbol{A}x = \boldsymbol{A}^\top y$

$$\begin{aligned}
\boldsymbol{A}^\top \boldsymbol{A} &= \begin{pmatrix} 0.1 & 0.4 & 0.5 & 0.9 & 1.0 & 1.2 & 2.0 \\ -0.005 & -0.08 & -0.125 & -0.405 & -0.5 & -0.72 & -2.0 \end{pmatrix} \begin{pmatrix} 0.1 & -0.005 \\ 0.4 & -0.08 \\ 0.5 & -0.125 \\ 0.9 & -0.405 \\ 1.0 & -0.5 \\ 1.2 & -0.72 \\ 2.0 & -2.0 \end{pmatrix} \\
&= \begin{pmatrix} 7.6700 & -5.8235 \\ -5.8235 & 4.954475 \end{pmatrix} \\
\boldsymbol{A}^\top y &= \begin{pmatrix} 0.1 & 0.4 & 0.5 & 0.9 & 1.0 & 1.2 & 2.0 \\ -0.005 & -0.08 & -0.125 & -0.405 & -0.5 & -0.72 & -2.0 \end{pmatrix} \begin{pmatrix} 0.96 \\ 3.26 \\ 3.82 \\ 5.11 \\ 5.2 \\ 5.05 \\ 0.58 \end{pmatrix} = \begin{pmatrix} 20.3290 \\ -10.20865 \end{pmatrix}
\end{aligned}$$

Die gesuchte Ausgleichslösung (v_y, g) des überbestimmten Systems (3.6.4) erfüllt nach (3.6.10)

$$\begin{pmatrix} 7.6700 & -5.8235 \\ -5.8235 & 4.954475 \end{pmatrix} \begin{pmatrix} v_y \\ g \end{pmatrix} = \begin{pmatrix} 20.3290 \\ -10.20865 \end{pmatrix}.$$

Die Matrix $\boldsymbol{A}^\top \boldsymbol{A}$ ist in der Tat symmetrisch positiv definit, und die damit eindeutige Lösung (v_y, g) ist auf 5 Stellen

$$\begin{pmatrix} v_y \\ g \end{pmatrix} = \begin{pmatrix} 10.096 \\ 9.8065 \end{pmatrix}.$$

■

Es ist von Interesse, die **Komplexität von Algorithmus** 3.6.6 zu betrachten. Tabelle 3.3 stellt sie unterteilt nach den einzelnen Schritten zusammen. Dabei wurde bei der Berechnung von $\boldsymbol{A}^\top A$ ausgenutzt, dass $\boldsymbol{A}^\top A$ symmetrisch ist. Man sieht, dass für den Fall $m \gg n$ das Aufstellen der Normalengleichungen (3.6.10) (d.h. die Berechnung von $\boldsymbol{A}^\top \boldsymbol{A}$ und $\boldsymbol{A}^\top b$) die Gesamtkosten dominiert. Dieser Fall tritt oft ein, weil in vielen Anwendungen die Anzahl n der zu bestimmenden Parameter relativ klein und die Anzahl m der Messungen gross ist, wie wir schon in Beispiel 3.6.7 beobachten.

Bemerkung 3.6.8 Die Normalengleichung (3.6.10) hat eine **geometrische Interpretation**: Aus

$$\boldsymbol{A}^\top (b - \boldsymbol{A}x) = 0$$

folgt, dass das Residuum $r = b - \boldsymbol{A}x$ senkrecht auf den Spalten von $\boldsymbol{A}$ steht, d.h., das Residuum r ist eine Normale zum von den Spalten der Matrix $\boldsymbol{A}$ aufgespannten Raum. Daher erklärt sich die Bezeichnung Normalengleichung. ∎

Bemerkung 3.6.9 In Anwendungen sind die Variablen x_i, die im Ausgleichsproblem (3.6.7) auftreten, gesuchte Parameter in einem mathematischen Modell. Offensichtlich müssen die Parameter *linear* in das Gesetz eingehen, damit ein Problem von der Form (3.6.7) ensteht. Manchmal kann bei Gesetzen, in denen die Parameter nicht linear auftreten, durch Umformulieren ein äquivalentes Gesetz hergeleitet werden, bei dem dies der Fall ist. Bei Zerfallsprozessen z.B. ist die Konzentration $c(t)$ von der Form

$$c(t) = c_0 e^{-bt},$$

wobei c_0 die Konzentration zum Zeitpunkt $t = 0$ ist und b die Zerfallsrate angibt. Die gesuchten Parameter c_0, b treten nicht linear auf. Durch Logarithmieren erhält man ein Gesetz

$$\ln c(t) = \ln c_0 - bt,$$

in dem der Parameter b und der neue Parameter $\tilde{c} := \ln c_0$ linear auftreten.

Auch bei Gesetzen von der Form

$$y(x) = c_0 x^\alpha$$

mit unbekannten Parametern c_0, $\alpha \in \mathbb{R}$ kann durch Logarithmieren ein neues Gesetz hergeleitet werden, in dem gesuchte Parameter linear auftreten. ∎

3.6.3 Methode der Orthogonalisierung

Für die Handrechnung bei kleinen Ausgleichsproblemen wendet man meist die Methode der Normalengleichungen an.

Für grosse, schlecht konditionierte Probleme jedoch werden oft Orthogonalisierungsverfahren eingesetzt. Ein wesentlicher Grund hierfür sind die guten Stabilitätseigenschaften von orthogonalen Transformationen.

Bei Ausgleichsproblemen ist man bestrebt, eine Gleichung für die gesuchte Lösung $x \in \mathbb{R}^n$ zu erhalten, die in der Form

$$\boldsymbol{B}x = b' \tag{3.6.11}$$

besitzt.

Hier sollte $\boldsymbol{B} \in \mathbb{R}^{n\times n}$ eine invertierbare Matrix sein und $b' \in \mathbb{R}^n$. Die Normalengleichung (3.6.10) ist ein solches Beispiel mit $\boldsymbol{B} = \boldsymbol{A}^\top \boldsymbol{A}$, $b' = \boldsymbol{A}^\top b$. Man muss damit rechnen, dass das Endergebnis $x \in \mathbb{R}^n$ fehlerbehaftet ist, weil (selbst wenn $\boldsymbol{A}$, b nicht fehlerbehaftet sind) zum einen beim Aufstellen von $\boldsymbol{B}$, b' Fehler eingeführt werden (z.B. Rundungsfehler), zum anderen Fehler beim anschliessenden Lösen von (3.6.11) verstärkt werden. Für das Lösen von Gleichungssystemen haben wir gesehen, dass die Konditionszahl $\kappa(\boldsymbol{B})$ ein Mass für die Verstärkung von Fehlern in b' ist. In Anwendungen ist nun oft $\kappa(\boldsymbol{A}^\top \boldsymbol{A}) \gg 1$. Es ist also von Interesse, Gleichungen von der Form (3.6.11) zu finden, bei denen $\kappa(\boldsymbol{B}) \ll \kappa(\boldsymbol{A}^\top \boldsymbol{A})$. Orthogonalisierungsverfahren leisten dies. Dort wird $\boldsymbol{A}^\top \boldsymbol{A}$ nicht aufgestellt, sondern es wird durch orthogonale Transformationen ein Gleichungssystem von der Form

$$\boldsymbol{R}x = b'$$

für die gesuchte Lösung $x \in \mathbb{R}^m$ hergeleitet. Hier ist $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ sogar eine Rechtsdreiecksmatrix; es gilt zudem (vgl. Satz 3.6.24):

$$\kappa(\boldsymbol{R}) = \sqrt{\kappa(\boldsymbol{A}^\top \boldsymbol{A})} < \kappa(\boldsymbol{A}^\top \boldsymbol{A}).$$

Wir betrachten zuerst den **Spezialfall, dass die Matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ Rechtsdreiecksstruktur hat** in dem Sinn, dass

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{R} \\ 0 \end{pmatrix} \tag{3.6.12}$$

für eine Rechtsdreiecksmatrix $\boldsymbol{R} \in \mathbb{R}^{n\times n}$. Der Vektor $b \in \mathbb{R}^m$ sei analog zerlegt:

$$b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \qquad b_1 \in \mathbb{R}^n, \quad b_2 \in \mathbb{R}^{m-n}.$$

Damit berechnen wir

$$\Phi(x) = \|b - \boldsymbol{A}x\|_2^2 = \left\| \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} - \begin{pmatrix} \boldsymbol{R} \\ 0 \end{pmatrix} x \right\|_2^2 = \left\| \begin{pmatrix} b_1 - \boldsymbol{R}x \\ b_2 - 0x \end{pmatrix} \right\|_2^2 = \|b_1 - \boldsymbol{R}x\|_2^2 + \|b_2\|_2^2.$$

Ist die Rechtsdreiecksmatrix $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ invertierbar, so ist die Lösung des Minimierungsproblems (3.6.7) offensichtlich gegeben durch die Lösung $x \in \mathbb{R}^n$ von $\boldsymbol{R}x = b_1$. D.h., die gesuchte Lösung x kann durch Rückwärtssubstitution (Algorithmus 3.1.2) des Gleichungssystems $\boldsymbol{R}x = b_1$ bestimmt werden.

Bei Orthogonalisierungsverfahren wird die Matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ so mithilfe orthogonaler Matrizen transformiert, dass sie die Gestalt (3.6.12) hat. Dabei nutzen wir wesentlich aus, dass orthogonale Matrizen die $\|\cdot\|_2$-Norm nach Proposition 1.13.4 invariant lassen:

Theorem 3.6.10 *Sei $m \geq n$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ mit linear unabhängigen Spalten, $b \in \mathbb{R}^m$. Sei $\boldsymbol{Q} \in O_m$ eine orthogonale Matrix und $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ Rechtsdreiecksmatrix, so dass gilt:*

$$\boldsymbol{Q}\boldsymbol{A} = \begin{pmatrix} \boldsymbol{R} \\ 0 \end{pmatrix}.$$

Sei ferner der Vektor $b' = Qb \in \mathbb{R}^m$ wie folgt partitioniert:

$$\boldsymbol{Q}b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \qquad b_1 \in \mathbb{R}^n, \quad b_2 \in \mathbb{R}^{m-n}.$$

Dann ist die Lösung $x \in \mathbb{R}^n$ des Ausgleichsproblems (3.6.7) die eindeutige Lösung des Dreieckssystems

$$\boldsymbol{R}x = b_1.$$

Beweis: Unter Ausnutzung von $\boldsymbol{Q}^\top = \boldsymbol{Q}^{-1}$ und der Tatsache, dass *Anwendung einer Unitären Matrix $\boldsymbol{Q}$ die Euklidische Länge eines Vektors invariant lässt* (vgl. Proposition 1.13.4) berechnen wir

$$\|b - \boldsymbol{A}x\|_2 = \|\boldsymbol{Q}^\top(\boldsymbol{Q}b - \boldsymbol{Q}\boldsymbol{A}x)\|_2 = \|\boldsymbol{Q}b - \boldsymbol{Q}\boldsymbol{A}x\|_2 = \left\| \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} - \begin{pmatrix} \boldsymbol{R} \\ 0 \end{pmatrix} x \right\|_2.$$

Also ist das ursprüngliche Ausgleichsproblem (3.6.7) äquivalent zu einem Ausgleichsproblem von der oben betrachteten Form. Da die Spalten von $\boldsymbol{A}$ linear unabhängig angenommen waren, ist die Rechtsdreiecksmatrix $\boldsymbol{R}$ invertierbar (Übung: Man überzeuge sich davon!). Die Aussage des Satzes folgt somit. □

Satz 3.6.10 legt somit eine weitere Methode nahe, wie das Ausgleichsproblem (3.6.7) gelöst werden kann. Man konstruiert eine orthogonale Matrix $\boldsymbol{Q} \in O_n$, so dass $\boldsymbol{QA}$ die Rechtsdreiecksgestalt (3.6.12) hat. Die $\boldsymbol{QR}$ Zerlegung einer Matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ mit vollem Spaltenrang n kann berechnet werden durch den Gram-Schmidt Algorithmus Algorithmus 1.14.1. Da dieser Algorithmus, wenn er einer Gleitkommaarithmetik $\mathbb{F}$ ausgeführt wird, anfällig ist für Rundungsfehler, berechnet man die $\boldsymbol{QR}$ Zerlegung einer Matrix in der Praxis dadurch, dass man $\boldsymbol{Q}$ ähnlich dem Faktor $\boldsymbol{L}$ in der $\boldsymbol{LR}$ Zerlegung quadratischer Matrizen, durch sukzessives akkumulieren, oder aufmultiplizieren, elementarer Eliminationsmatrizen aufbaut. Dies resultiert in der $\boldsymbol{QR}$ Faktorisierung der Matrix $\boldsymbol{A}$.

3.6.4 $\boldsymbol{QR}$-Faktorisierung

Die Grundidee bei der $\boldsymbol{LR}$-Zerlegung als Lösungsverfahren von linearen Gleichungssystemen war, die Matrix $\boldsymbol{A}$ so in zwei Faktoren $\boldsymbol{L}$ und $\boldsymbol{R}$ zu zerlegen, dass jeder der beiden Faktoren einfach zu invertieren ist. Neben der $\boldsymbol{LR}$-Zerlegung und ihren Varianten gibt es noch eine weitere grosse Klasse von Zerlegungen, die sog. $\boldsymbol{QR}$-Zerlegungen bei der eine Matrix $\boldsymbol{A}$ als $\boldsymbol{A} = \boldsymbol{QR}$ faktorisiert wird. Hier ist $\boldsymbol{R}$ wieder eine Rechtsdreiecksmatrix während $\boldsymbol{Q}$ eine *orthogonale Matrix* ist.

Orthogonale Matrizen

Definition 3.6.11 (orthogonale Matrizen) *Eine Matrix $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ heisst* orthogonal, *falls $\boldsymbol{Q}$ regulär ist und $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^\top$. Die Spalten von $\boldsymbol{Q}$ bilden somit eine Orthonormalbasis des $\mathbb{R}^m$. Die Menge der orthogonalen $m \times m$ Matrizen bezeichnet man mit O_m.*

Offensichtlich sind orthogonale Matrizen einfach zu invertieren. Ein Beispiel für orthogonale Matrizen haben wir bereits in Abschnitt 3.3 kennengelernt: Lemma 3.3.4, (ii) zeigt, dass Permutationsmatrizen orthogonale Matrizen sind.

Eine wichtige Eigenschaft von O_m ist, dass sie eine Gruppe bzgl. der Multiplikation bildet:

Theorem 3.6.12 *O_m ist eine Gruppe bzgl. der Multiplikation, d.h. Id_m ist eine orthogonale Matrix, zu jedem $\boldsymbol{Q} \in O_m$ existiert $\boldsymbol{Q}^{-1} \in O_m$ und für $\boldsymbol{Q}_1$, $\boldsymbol{Q}_2 \in O_m$ ist $\boldsymbol{Q}_1\boldsymbol{Q}_2 \in O_m$.*

Beweis: Übung. □

Orthogonale Matrizen kann man z.B. wie folgt erzeugen:

Theorem 3.6.13 *Sei $k \in \mathbb{N}$, $m \geq k+1$, $\boldsymbol{Q} \in O_{m-k}$. Dann ist die Matrix*

$$\left(\begin{array}{c|c} \mathrm{Id}_k & 0 \\ \hline 0 & Q \end{array}\right)$$

ein Element von O_m.

Beweis: Übung. □

Kann eine Matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ in der Form $\boldsymbol{A} = \boldsymbol{QR}$ zerlegt werden mit orthogonaler Matrix $Q \in O_m$ und Rechtsdreiecksmatrix $\boldsymbol{R} \in \mathcal{R}_n$, dann kann ein Gleichungssystem $\boldsymbol{A}x = b$ einfach aufgelöst werden:

1. Löse $\boldsymbol{Q}y = b$ durch Bilden von $y = \boldsymbol{Q}^{-1}b = \boldsymbol{Q}^\top b$.

2. Löse das Dreieckssystem $Rx = y$ mithilfe von Algorithmus 3.1.2.

Verfahren, die für eine Matrix eine $\boldsymbol{QR}$-Zerlegung finden, heissen auch Orthogonalisierungsverfahren. Das Vorgehen zur Lösung linearer Gleichungssysteme ist demnach ähnlich dem der Gauss-Elimination. Warum sollte man Orthogonalisierungsverfahren neben der Gauss'schen Elimination benutzen, um lineare Gleichungssysteme zu lösen? Die Antwort liegt in der numerischen Stabilität der $\boldsymbol{QR}$ Zerlegung: während für Linksdreiecksmatrix $\boldsymbol{L} \in \mathcal{L}_m$ in der GEM die Matrixnorm $\|L\|_2$ unter Umständen sehr gross werden kann, ist eine für die Numerik wesentliche Eigenschaft orthogonaler Matrizen gerade, dass sie die $\|\cdot\|_2$-Norm eines Vektors unverändert lassen: Proposition 1.13.4 zeigt, dass die Anwendung einer orthogonalen Matrix auf einen fehlerbehafteten Vektor x den Fehler nicht verstärkt:

$$\frac{\|\boldsymbol{Q}(x+\Delta x) - \boldsymbol{Q}x\|_2}{\|\boldsymbol{Q}x\|_2} = \frac{\|\boldsymbol{Q}\Delta x\|_2}{\|\boldsymbol{Q}x\|_2} = \frac{\|\Delta x\|_2}{\|x\|_2}.$$

Die Multiplikation mit einer orthogonalen Matrix ist somit numerisch stabil. Diese Stabilitätseigenschaft ist einer der Hauptgründe für den Einsatz von orthogonalen Matrizen in der Numerik.

Wie konstruiert man nun die Matrix $\boldsymbol{Q}$ in der $\boldsymbol{QR}$ Zerlegung von $\boldsymbol{A}$? Wie die Matrix $\boldsymbol{L}$ in der GEM baut man auch $\boldsymbol{Q}$ durch sukzessives Aufmultiplizieren elementarer orthogonaler Matrizen auf. Für die elementaren Matrizen gibt es zwei Konstruktionsmöglichkeiten: *Householder-Reflexionen* sowie *Givens-Rotationen*. Wir diskutieren im Folgenden die Householder-Reflexionen und behandeln die Givens Rotationen später, bei den Eigenwertproblemen.

Orthogonalisierung mit Householder-Reflexionen

Beim Orthogonalisieren mit Householderreflexionen[4] wird eine $\boldsymbol{QR}$-Zerlegung einer Matrix erzeugt, bei der die orthogonale Matrix $\boldsymbol{Q}$ das Produkt von sog. *Householder-Reflexionen* ist. Es gilt:

Theorem 3.6.14 (Eigenschaften von Householder-Reflexionen)
Sei $0 \neq v \in \mathbb{R}^n$. Dann hat die Householder-Reflexion

$$\boldsymbol{Q} := \boldsymbol{I}d_n - \frac{2}{v^\top v} vv^\top$$

folgende Eigenschaften:

1. *$\boldsymbol{Q}$ ist symmetrisch, d.h. $\boldsymbol{Q}^\top = \boldsymbol{Q}$.*

2. *$\boldsymbol{Q}$ ist orthogonal, d.h. $\boldsymbol{Q}^\top = \boldsymbol{Q}^{-1}$.*

3. *$\boldsymbol{Q}$ ist involutorisch, d.h. $\boldsymbol{Q}^2 = \boldsymbol{I}d_m$.*

Beweis: Übung. (vgl. Aufgabe 15). □

finis 1.DS

Bemerkung 3.6.15 Die Householder-Reflexionen haben eine geometrische Interpretation: Als lineare Abbildung gesehen, stellen sie eine Spiegelung ("Reflexion") an der Hyperebene $H := \{x \in \mathbb{R}^n \mid x^\top v = 0\}$ dar. In der Tat rechnet man nach: Für $x \in H$ gilt $\boldsymbol{Q}x = x$ und für x parallel zu v gilt $\boldsymbol{Q}x = -x$.

■

Eine $\boldsymbol{QR}$-Zerlegung einer quadratischen Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ wird mithilfe von Householder-Reflexionen erzeugt, indem man schrittweise die Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ durch Multiplikation mit orthogonalen Matrizen $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_{n-1}$ auf rechte Dreiecksgestalt bringt. Zu diesem Zweck nehmen wir an, dass man für jede Spalte $0 \neq a \in \mathbb{R}^k$, $k \leq n$, eine Householder-Reflexion $\boldsymbol{Q}$ finden kann, so dass $\boldsymbol{QA} = \alpha e_1$ ist, wobei $\alpha \in \mathbb{R}$ und $e_1 \in \mathbb{R}^k$ der erste Einheitsvektor ist (dies werden wir unten in Lemma 3.6.16 zeigen). Dann können wir wie folgt vorgehen: sei a die erste Spalte von $\boldsymbol{A}$. Nach Annahme können wir ein $\boldsymbol{Q}_1 \in O_n$ finden, so dass die Householder-Reflexion $\boldsymbol{Q}_1\boldsymbol{A} = \alpha^{(1)}e_1$ für ein $\alpha^{(1)} \in \mathbb{R}$. Das Produkt $\boldsymbol{Q}_1\boldsymbol{A}$ hat damit folgende Struktur:

$$\left(\begin{array}{c|c} \alpha^{(1)} & (b^{(1)})^\top \\ \hline 0 & \\ \vdots & \boldsymbol{A}^{(1)} \\ 0 & \end{array}\right) \quad \text{mit} \quad \boldsymbol{A}^{(1)} \in \mathbb{R}^{(n-1)\times(n-1)} \quad \text{und} \quad b^{(1)} \in \mathbb{R}^{n-1}. \tag{3.6.13}$$

Um weiter in Richtung Dreiecksgestalt zu transformieren, wiederholt man den Prozess für die Untermatrix $\boldsymbol{A}^{(1)}$. Dazu beobachten wir, dass für jede Matrix $\boldsymbol{Q}^{(1)} \in O_{n-1}$ die Matrix

$$\boldsymbol{Q}_2 := \left(\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \boldsymbol{Q}^{(1)} & \\ 0 & & & \end{array}\right),$$

nach Satz 3.6.13 orthogonal ist und zudem

$$\left(\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \boldsymbol{Q}^{(1)} & \\ 0 & & & \end{array}\right) \cdot \left(\begin{array}{c|c} \alpha^{(1)} & (b^{(1)})^\top \\ \hline 0 & \\ \vdots & \boldsymbol{A}^{(1)} \\ 0 & \end{array}\right) = \left(\begin{array}{c|c} \alpha^{(1)} & (b^{(1)})^\top \\ \hline 0 & \\ \vdots & \boldsymbol{Q}^{(1)}\boldsymbol{A}^{(1)} \\ 0 & \end{array}\right)$$

ist, wobei $b^{(1)}$ derselbe Vektor ist wie in (3.6.13). Man wählt also $\boldsymbol{Q}^{(1)} \in O_{n-1}$ wieder mithilfe obiger Annahme so als Householder-Reflexion, dass genau wie im ersten Schritt das Produkt $\boldsymbol{Q}^{(1)}\boldsymbol{A}^{(1)}$ in der ersten Spalte bis auf den Eintrag ganz links oben nur Nullen enthält. Wählt man $\boldsymbol{Q}_2$ mit diesem $\boldsymbol{Q}^{(1)}$, dann hat das Produkt $\boldsymbol{Q}_2\boldsymbol{Q}_1A$ die Form

$$\boldsymbol{Q}_2\boldsymbol{Q}_1\boldsymbol{A} = \left(\begin{array}{cc|ccc} \alpha^{(1)} & b_1^{(1)} & b_2^{(1)} & \cdots & b_{n-1}^{(1)} \\ 0 & \alpha^{(2)} & b_1^{(2)} & \cdots & b_{n-2}^{(2)} \\ \hline 0 & 0 & & & \\ \vdots & \vdots & & \boldsymbol{A}^{(2)} & \\ 0 & 0 & & & \end{array}\right),$$

[4]Householder, Alston Scott 1904–1993

wobei $(b_1^{(1)}, \ldots, b_{n-1}^{(1)}) = b^\top$ wieder der Vektor aus (3.6.13) ist. Man wird nun weiter verfahren, um die Matrix $\boldsymbol{A}^{(2)}$ zu behandeln. Dabei werden die Zeilen $(b^{(1)})^\top$, $(b^{(2)})^\top$ nicht mehr verändert. Sie sind somit bereits Zeilen der gesuchten Rechtsdreiecksmatrix. Man setzt diesen Prozess so lange fort, bis die Matrix $\boldsymbol{Q}_{n-1} \cdots \boldsymbol{Q}_1 \boldsymbol{A}$ Rechtsdreiecksgestalt hat.

Somit ist die gesuchte $\boldsymbol{QR}$-Zerlegung von $\boldsymbol{A} \in \mathbb{R}^{n \times n}$:

$$\boldsymbol{A} = \boldsymbol{Q}_1^\top \cdots \boldsymbol{Q}_{n-1}^\top \boldsymbol{R} = \boldsymbol{Q}_1 \cdots \boldsymbol{Q}_{n-1} \boldsymbol{R},$$

wobei $\boldsymbol{R}$ eine Rechtsdreiecksmatrix ist und die Matrizen $\boldsymbol{Q}_k$ folgende Bauart haben:

$$\boldsymbol{Q}_k = \left(\begin{array}{c|c} \boldsymbol{I}d_{k-1} & 0 \cdots 0 \\ \hline 0 & \\ \vdots & \boldsymbol{I}d_{n-k+1} - \dfrac{2}{v_k^\top v_k} v_k v_k^\top \\ 0 & \end{array}\right), \quad v_k \in \mathbb{R}^{n-k+1} \quad \text{\% Konvention: } \boldsymbol{Q}_1 = \mathrm{Id}_n - \frac{2}{v_1^\top v_1} v_1 v_1^\top. \tag{3.6.14}$$

Es bleibt zu zeigen, dass man in jedem der obigen Schritte die Matrizen $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_{n-1}$ wie gefordert als Householder-Reflexionen wählen kann.

Lemma 3.6.16 *Sei $0 \neq a \in \mathbb{R}^k$ und $e_1 \in \mathbb{R}^k$ der 1. Einheitsvektor. Sei*

$$\alpha = \|a\|_2 \quad \textit{oder} \quad \alpha = -\|a\|_2.$$

Im Falle $\boldsymbol{A} = e_1$ sei $\alpha = -\|a\|_2 = -1$ gewählt. Dann gilt für $v := a - \alpha e_1$:

$$\boldsymbol{QA} = \alpha e_1 \qquad \boldsymbol{Q} = \boldsymbol{I}d_k - \frac{2}{v^\top v} v v^\top. \tag{3.6.15}$$

Beweis: Die Wahl von α impliziert, dass $v \neq 0$. Die Behauptung $\boldsymbol{QA} = \alpha e_1$ kann dann direkt nachgerechnet werden (Übung: man rechne dies nach!), und das Lemma ist damit bewiesen. Zur Motivation der Formeln für α und v stellen wir jedoch folgende Betrachtungen an, welche zudem klären, inwieweit die Householder-Reflexion eindeutig ist, die $\boldsymbol{QA} = \alpha e_1$ leistet. Wir zeigen:

1. Die Householder-Reflexionen, die zu einem Vektor $v \neq 0$ und zu λv, $\lambda \in \mathbb{R} \setminus \{0\}$ gehören, stimmen überein.

2. Ist a nicht parallel zu e_1, dann muss die Householder-Reflexion $\boldsymbol{Q}$, die a auf ein Vielfaches von e_1 abbildet, von einem Householder-Vektor $v = \lambda(a - \alpha e_1)$, $\lambda \in \mathbb{R} \setminus \{0\}$, $\alpha = \pm\|a\|_2$, erzeugt werden.

3. Ist a parallel zu e_1, so muss der Householder-Vektor v von der Form $v = \lambda e_1$, $\lambda \in \mathbb{R} \setminus \{0\}$ sein, oder er muss $v^\top e_1 = 0 = v^\top a$ erfüllen.

Die erste Behauptung, dass die Länge von v irrelevant ist, folgt unmittelbar aus der Definition der Householder-Reflexion.

Für die beiden anderen Behauptungen beobachten wir, dass orthogonale Matrizen nach Proposition 1.13.4 die Euklidische Länge erhalten, d.h.

$$\|a\|_2^2 = \|\boldsymbol{Q}a\|_2^2 = \|\alpha e_1\|_2^2 = |\alpha|^2.$$

Somit ergibt sich für α:

$$\alpha = \pm\|a\|_2 = \sqrt{\boldsymbol{A}^\top a}.$$

Aus (3.6.15) folgt zudem

$$\alpha e_1 = \boldsymbol{Q}\boldsymbol{A} = \left(\mathrm{Id}_k - \frac{2}{v^\top v} v v^\top\right) \boldsymbol{A} = a - \frac{2}{v^\top v}(v^\top a)\, v. \tag{3.6.16}$$

Wir betrachten nun die Fälle $v^\top a \neq 0$ und $v^\top \boldsymbol{A} = 0$ separat.

Im Fall $v^\top a \neq 0$ erhalten wir $v \in \operatorname{span}(\alpha e_1 - a)$, d.h. es muss gelten $v = \lambda(a - \alpha e_1)$ wie behauptet. Falls a parallel zu e_1 ist, impliziert dies insbesondere $v = \lambda e_1$ für ein geeignetes $\lambda \in \mathbb{R} \setminus \{0\}$.

Im Fall $v^\top \boldsymbol{A} = 0$ schliessen wir aus (3.6.16), dass a parallel zu e_1 ist. Wie oben behauptet, bildet in diesem Fall die Householder-Reflexion zu jedem v mit $v^\top \boldsymbol{A} = 0$ den Vektor a auf ein Vielfaches von e_1 ab. □

Lemma 3.6.16 zeigt, dass die Householder-Reflexion, die einen Vektor a auf ein Vielfaches des Einheitsvektors e_1 abbildet, nicht eindeutig ist. Die verbliebenen Freiheiten bei der Wahl des Householder-Vektors v nutzt man aus, um numerische Stabilität bei der Bestimmung von v zu gewinnen und um bei der Speicherung der Vektoren v Platz zu sparen:

- Bei der Bestimmung von v mithilfe von Lemma 3.6.16 wählt man in der Praxis $\alpha = -\operatorname{sign}(a_1)\sqrt{\boldsymbol{A}^\top a}$, um Auslöschung bei der Auswertung von $v = (a_1 - \alpha, a_2, \ldots, a_k)^\top$ zu vermeiden [5].

- Die Länge von $v \in \mathbb{R}^k$ ist nicht wesentlich, d.h. die Householder-Reflexionen, die zu v und zu λv, $\lambda \neq 0$, gehören, stimmen überein. Die Wahl $\alpha = -\operatorname{sign}(a_1)\|a\|_2$ impliziert insbesondere, dass $v_1 = a_1 + \operatorname{sign}(a_1)\|a\|_2 \neq 0$. Man kann also v so normieren, dass $v_1 = 1$.

Aus diesen beiden Überlegungen ergibt sich schliesslich, dass man den Householder-Vektor $v \in \mathbb{R}^k$, dessen zugehörige Householder-Reflexion a auf αe_1 abbildet, so wählt:

$$v := \frac{1}{\beta}(a - \alpha e_1), \qquad \alpha = -\operatorname{sign}(a_1)\sqrt{\boldsymbol{A}^\top a}, \qquad \beta = a_1 + \operatorname{sign}(a_1)\sqrt{\boldsymbol{A}^\top a}. \tag{3.6.17}$$

Für die $\boldsymbol{QR}$-Zerlegung wählen wir die Householder-Vektoren wie in (3.6.17); auf diese Weise ist die erste Komponente gleich 1. Der Grund hierfür ist, dass auf diese Weise eine effiziente Abspeicherung der gesamten Information über die $\boldsymbol{QR}$-Zerlegung einer Matrix $\boldsymbol{A}$ direkt auf dem Speicher der Matrix $\boldsymbol{A}$ möglich ist: Im Speicherbereich A_{ij} mit $i \leq j$ kann der Faktor $\boldsymbol{R}$ abgelegt werden, und der Speicherbereich von A_{ij}, $i > j$ fasst die wesentliche Information der Householder-Vektoren v_k (d.h. die Einträge $(v_k)_i$, $i = 2, \ldots, n-k+1$ von $v_k \in \mathbb{R}^{n-k+1}$).

Der Algorithmus zur Bestimmung einer $\boldsymbol{QR}$-Zerlegung einer $n \times n$-Matrix ist damit festgelegt. Wir formalisieren das Vorgehen in Algorithmus 3.6.17.

Algorithmus 3.6.17 ($\boldsymbol{QR}$-Zerlegung mit Householder-Reflexionen)
input : Matrix $A \in \mathbb{R}^{n\times n}$, $\boldsymbol{A}$ invertierbar
output: überschreibt $\boldsymbol{A}$ mit der wesentlichen Information für $\boldsymbol{QR}$-Zerlegung:
$\boldsymbol{Q}\boldsymbol{A} = \boldsymbol{R}$ mit $\boldsymbol{Q} = \boldsymbol{Q}_{n-1} \cdots \boldsymbol{Q}_1$ mit $\boldsymbol{Q}_j$ von der Form (3.6.14)

[5] Wir definieren $\operatorname{sign}(x) = 1$ für $x \geq 0$ und $\operatorname{sign}(x) = -1$ für $x < 0$

$R_{ij} = A_{ij}$ für $i \leq j$
Vektoren $v_j \in \mathbb{R}^{n-j+1}$ aus (3.6.14): $(v_j)_{i-j+1} = A_{ij}$ für $i > j$ und implizit $(v_j)_1 = 1$

```
for k from 1 to n − 1 do {
    v(k : n) = house(A(k : n, k))                    % bestimme Householder-Vektor v_k
    A(k : n, k : n) = house_mult(A(k : n, k : n), v(k : n))    % berechne α^(k), b^(k), A^(k)
    A(k + 1 : n, k) = v(k + 1 : n)   %speichere nur wesentliche Information von v_k—(v_k)_1 = 1
}
return (A)
```

function: $v =$ **house** (a)
input: Vektor $a \neq 0$
output: Householder-Vektor v normiert derart, dass erste Komponente 1 ist
$n := \text{length}(a)$ % $a \in \mathbb{R}^n$
$\alpha := -\|a\|_2$; `if` $\alpha = 0$ `exit(‘‘`$A = 0$ `nicht zulässig!’’);`
$v := a$
$\beta := a(1) - \alpha \,\text{sign}\,(a(1))$
$v(2 : n) := \frac{1}{\beta} v(2 : n)$
$v(1) = 1$
`return(v)`

function: $\boldsymbol{A} =$ **house_mult** $(\boldsymbol{A}, v)$
input: Matrix $\boldsymbol{A}$, Householder-Vektor v
output: überschreibt Matrix $\boldsymbol{A}$ mit $(\text{Id} - \frac{2}{v^\top v} v v^\top) A$
$\beta := -2/v^\top v$
$w := v^\top A$ % w ist ein Zeilenvektor
$A := A + \beta v w$ % Rang-1 update der Matrix $\boldsymbol{A}$
`return(A)`

Kosten für die Berechnung einer $\boldsymbol{QR}$-Zerlegung: $\frac{2}{3}n^3$ Multiplikationen. Die $\boldsymbol{QR}$-Zerlegung ist damit ungefähr doppelt so teuer wie eine $\boldsymbol{LR}$-Zerlegung. Sie wird aber trotzdem wegen ihrer numerischen Stabilitätseigenschaften oft angewandt, insbesondere bei *Ausgleichsproblemen*.

Mit den Householder-Reflexionen haben wir explizit $\boldsymbol{QR}$-Zerlegungen von Matrizen konstruiert. Wie bei der $\boldsymbol{LR}$-Zerlegung kann man die Frage stellen, inwieweit $\boldsymbol{QR}$-Zerlegungen eindeutig sind. Da wir bereits in jedem Schritt zwischen zwei verschiedenen Householder-Reflexionen auswählen konnten (der Parameter α war nur bis auf das Vorzeichen festgelegt), bedarf es offensichtlich einer zusätzlichen Normierungsbedingung. Verlangt man bei invertierbaren Matrizen, dass die Diagonaleinträge der Rechtsdreiecksmatrix $\boldsymbol{R}$ positiv sein sollen, so ist die Zerlegung eindeutig:

Theorem 3.6.18 (Eindeutigkeit der $\boldsymbol{QR}$-Zerlegung) *Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ regulär. Dann existiert ein eindeutiges $\boldsymbol{Q} \in O_n$ und eine eindeutige Rechtsdreiecksmatrix $\boldsymbol{R}$ mit $R_{ii} > 0$, $i = 1, \ldots, n$, so dass $\boldsymbol{QR} = \boldsymbol{A}$.*

Beweis: Aus Algorithmus 3.6.17 folgt die Existenz einer $\boldsymbol{QR}$-Zerlegung $\boldsymbol{Q}'\boldsymbol{R}' = \boldsymbol{A}$. Sei $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ Diagonalmatrix mit $D_{ii} = \text{sign} R_{ii}$. Dann ist $\boldsymbol{R} := \boldsymbol{D}\boldsymbol{R}'$ eine Rechtsdreiecksmatrix mit $R_{ii} > 0$. $\boldsymbol{D}$ ist orthogonal ($\boldsymbol{D}^\top = \boldsymbol{D}$ und $\boldsymbol{D}^2 = \text{Id}_n$). Damit ist $\boldsymbol{A} = \boldsymbol{Q}'\boldsymbol{R}' = \boldsymbol{Q}'\boldsymbol{D}^{-1}\boldsymbol{D}\boldsymbol{R}' =$

$(\boldsymbol{Q}'\boldsymbol{D}^{-1})\boldsymbol{R}$. Da $\boldsymbol{D}$ orthogonal ist, ist $\boldsymbol{Q} := \boldsymbol{Q}'\boldsymbol{D}^{-1}$ nach Satz 3.6.12 ebenfalls orthogonal. Damit ist die Existenz einer $\boldsymbol{QR}$-Zerlegung der gewünschten Art gezeigt. Eindeutigkeit folgt aus Korollar 3.2.4: Sei $\boldsymbol{QR} = \boldsymbol{A}$ und $R_{ii} > 0$, $i = 1, \ldots, n$.

Die Matrix $\boldsymbol{A}^\top\boldsymbol{A}$ ist SPD. Es gilt

$$\boldsymbol{A}^\top\boldsymbol{A} = (\boldsymbol{QR})^\top\boldsymbol{QR} = \boldsymbol{R}^\top\boldsymbol{Q}^\top\boldsymbol{QR} = \boldsymbol{R}^\top\boldsymbol{R}.$$

Somit ist $\boldsymbol{R}^\top$ der Cholesky-Faktor von $\boldsymbol{A}^\top\boldsymbol{A}$. Nach Korollar 3.2.4 ist $\boldsymbol{R}$ eindeutig; damit ist auch $\boldsymbol{Q} = \boldsymbol{AR}^{-1}$ eindeutig. □

Bemerkung 3.6.19 Anstelle der Householder-Reflexionen werden auch *Givens-Rotationen* zur Orthogonalisierung einer Matrix verwendet.

Zum Abschluss berechnen wir eine $\boldsymbol{QR}$-Zerlegung mithilfe von Householder-Reflexionen für ein konkretes Beispiel:

Beispiel 3.6.20 Sei

$$\boldsymbol{A} = \begin{pmatrix} -4 & -2-2\sqrt{6} & -6-3\sqrt{2}-\sqrt{6} \\ 0 & -2\sqrt{3} & 9-\sqrt{3} \\ -4\sqrt{2} & -2\sqrt{2}+2\sqrt{3} & 3-6\sqrt{2}+\sqrt{3} \end{pmatrix}.$$

Gesucht ist eine $\boldsymbol{QR}$-Zerlegung von $\boldsymbol{A}$. Unsere Handrechnung stimmt im wesentlichen mit Algorithmus 3.6.17 überein. Im Unterschied zu Algorithmus 3.6.17 normieren wir nicht die Householder-Vektoren v_k zu $(v_k)_1 = 1$.

Die erste Spalte von $\boldsymbol{A}$ ist

$$\boldsymbol{A} = \begin{pmatrix} -4 \\ 0 \\ -4\sqrt{2} \end{pmatrix}, \qquad \|a\|_2 = \sqrt{16 + 16 \cdot 2} = \sqrt{48} = 4\sqrt{3}.$$

Also ist $\alpha = -\mathrm{sign}(a_1)\|a\|_2 = -\mathrm{sign}(-4)4\sqrt{3} = 4\sqrt{3}$ und damit v

$$v = a - \alpha e_1 = \begin{pmatrix} -4 \\ 0 \\ -4\sqrt{2} \end{pmatrix} - 4\sqrt{3}\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -4-4\sqrt{3} \\ 0 \\ -4\sqrt{2} \end{pmatrix}, \qquad \|v\|_2^2 = 16(6+2\sqrt{3}).$$

Die erste Householder-Reflexion ist somit

$$\begin{aligned}
\boldsymbol{Q}_1 &= \mathrm{Id}_3 - \frac{2}{\|v\|_2^2}vv^\top = \mathrm{Id}_3 - \frac{2}{16(6+2\sqrt{3})}\begin{pmatrix} -4-4\sqrt{3} \\ 0 \\ -4\sqrt{2} \end{pmatrix} \cdot \begin{pmatrix} -4-4\sqrt{3}, & 0, & -4\sqrt{2} \end{pmatrix} \\
&= \mathrm{Id}_3 - \frac{1}{3+\sqrt{3}}\begin{pmatrix} -1-\sqrt{3} \\ 0 \\ -\sqrt{2} \end{pmatrix} \cdot \begin{pmatrix} -1-\sqrt{3}, & 0, & -\sqrt{2} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \frac{1}{\sqrt{3}(1+\sqrt{3})}\begin{pmatrix} 4+2\sqrt{3} & 0 & \sqrt{2}(1+\sqrt{3}) \\ 0 & 0 & 0 \\ \sqrt{2}(1+\sqrt{3}) & 0 & 2 \end{pmatrix} \\
&= \frac{1}{\sqrt{3}}\begin{pmatrix} -1 & 0 & -\sqrt{2} \\ 0 & \sqrt{3} & 0 \\ -\sqrt{2} & 0 & 1 \end{pmatrix}.
\end{aligned}$$

(Man sieht, dass–wie erwartet–Q_1 eine orthogonale Matrix ist). Wir erhalten damit für $\boldsymbol{Q}_1\boldsymbol{A}$:

$$\begin{aligned}
\boldsymbol{Q}_1\boldsymbol{A} &= \frac{1}{\sqrt{3}}\begin{pmatrix} -1 & 0 & -\sqrt{2} \\ 0 & \sqrt{3} & 0 \\ -\sqrt{2} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -4 & -2-2\sqrt{6} & -6-3\sqrt{2}-\sqrt{6} \\ 0 & -2\sqrt{3} & 9-\sqrt{3} \\ -4\sqrt{2} & -2\sqrt{2}+2\sqrt{3} & 3-6\sqrt{2}+\sqrt{3} \end{pmatrix} \\
&= \frac{1}{\sqrt{3}}\begin{pmatrix} 12 & 6 & 18 \\ 0 & -6 & -3+9\sqrt{3} \\ 0 & 6\sqrt{3} & 9+3\sqrt{3} \end{pmatrix} = \sqrt{3}\begin{pmatrix} 4 & 2 & 6 \\ 0 & -2 & -1+3\sqrt{3} \\ 0 & 2\sqrt{3} & 3+\sqrt{3} \end{pmatrix}.
\end{aligned}$$

Die erste Spalte von $\boldsymbol{Q}_1\boldsymbol{A}$ stimmt mit αe_1 überein—so wurde die Householder-Reflexion $\boldsymbol{Q}_1$ schliesslich konstruiert!

Die erste Zeile von $\boldsymbol{Q}_1\boldsymbol{A}$ ist bereits die erste Zeile der gesuchten Rechtsdreiecksmatrix $\boldsymbol{R}$. Wir wiederholen nun obiges Vorgehen für die Untermatrix

$$\boldsymbol{A}^{(1)} = \sqrt{3}\begin{pmatrix} -2 & -1+3\sqrt{3} \\ 2\sqrt{3} & 3+\sqrt{3} \end{pmatrix}.$$

Die erste Spalten von $\boldsymbol{A}^{(1)}$ ist

$$\boldsymbol{A} = \begin{pmatrix} -2\sqrt{3} \\ 6 \end{pmatrix}, \qquad \|a\|_2 = \sqrt{12+36} = 4\sqrt{3}.$$

Also ist nun $\alpha = -\mathrm{sign}(a_1)\|a\|_2 = -\mathrm{sign}(-2\sqrt{3}) \cdot 4\sqrt{3} = 4\sqrt{3}$ und für $v = a - \alpha e_1$ erhalten wir

$$v = \begin{pmatrix} -2\sqrt{3} \\ 6 \end{pmatrix} - 4\sqrt{3}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -6\sqrt{3} \\ 6 \end{pmatrix}, \qquad \|v\|_2^2 = 144.$$

Damit ist $\boldsymbol{Q}^{(1)}$

$$\begin{aligned}
\boldsymbol{Q}^{(1)} &= \mathrm{Id}_2 - \frac{2}{\|v\|_2^2} v \cdot v^\top = \mathrm{Id}_2 - \frac{2}{144} \cdot \begin{pmatrix} -6\sqrt{3} \\ 6 \end{pmatrix} \cdot \begin{pmatrix} -6\sqrt{3}, & 6 \end{pmatrix} \\
&= \mathrm{Id}_2 - \frac{1}{2} \cdot \begin{pmatrix} -\sqrt{3} \\ 1 \end{pmatrix} \cdot \begin{pmatrix} -\sqrt{3}, & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 3 & -\sqrt{3} \\ -\sqrt{3} & 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} -1 & \sqrt{3} \\ \sqrt{3} & 1 \end{pmatrix}.
\end{aligned}$$

Die fehlenden Einträge der gesuchten Rechtsdreiecksmatrix $\boldsymbol{R}$ können somit bestimmt werden:

$$\boldsymbol{Q}^{(1)} \cdot \boldsymbol{A}^{(1)} = \frac{1}{2}\begin{pmatrix} -1 & \sqrt{3} \\ \sqrt{3} & 1 \end{pmatrix} \cdot \sqrt{3}\begin{pmatrix} -2 & -1+3\sqrt{3} \\ 2\sqrt{3} & 3+\sqrt{3} \end{pmatrix} = \frac{\sqrt{3}}{2}\begin{pmatrix} 8 & 4 \\ 0 & 12 \end{pmatrix}.$$

Die gesuchte Rechtsdreiecksmatrix $\boldsymbol{R}$ ist damit

$$R = \sqrt{3}\begin{pmatrix} 4 & 2 & 6 \\ 0 & 4 & 2 \\ 0 & 0 & 6 \end{pmatrix}.$$

Die orthogonale Matrix $\boldsymbol{Q}^{-1}$ mit $\boldsymbol{Q}^{-1}A = R$ ist somit

$$\begin{aligned}
\boldsymbol{Q}^{-1} &= \left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & \multicolumn{2}{c}{\multirow{2}{*}{$\boldsymbol{Q}^{(1)}$}} \\ 0 & & \end{array}\right) \cdot \boldsymbol{Q}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{1}{2} & \frac{1}{2}\sqrt{3} \\ 0 & \frac{1}{2}\sqrt{3} & \frac{1}{2} \end{pmatrix} \cdot \frac{1}{\sqrt{3}}\begin{pmatrix} -1 & 0 & -\sqrt{2} \\ 0 & \sqrt{3} & 0 \\ -\sqrt{2} & 0 & 1 \end{pmatrix} \\
&= \frac{1}{2\sqrt{3}}\begin{pmatrix} -2 & 0 & -2\sqrt{2} \\ -\sqrt{6} & -\sqrt{3} & \sqrt{3} \\ -\sqrt{2} & 3 & 1 \end{pmatrix}.
\end{aligned}$$

Zur Probe berechnen wir noch $\boldsymbol{QR}$ ($\boldsymbol{Q} = (\boldsymbol{Q}^{-1})^{-1} = (\boldsymbol{Q}^{-1})^\top$, weil Q orthogonal ist!)

$$\begin{aligned}
\boldsymbol{QR} &= \frac{1}{2\sqrt{3}} \begin{pmatrix} -2 & -\sqrt{6} & -\sqrt{2} \\ 0 & -\sqrt{3} & 3 \\ -2\sqrt{2} & \sqrt{3} & 1 \end{pmatrix} \cdot \sqrt{3} \begin{pmatrix} 4 & 2 & 6 \\ 0 & 4 & 2 \\ 0 & 0 & 6 \end{pmatrix} \\
&= \begin{pmatrix} -4 & -2-2\sqrt{6} & -6-3\sqrt{2}-\sqrt{6} \\ 0 & -2\sqrt{3} & 9-\sqrt{3} \\ -4\sqrt{2} & -2\sqrt{2}+2\sqrt{3} & 3-6\sqrt{2}+\sqrt{3} \end{pmatrix} = \boldsymbol{A}
\end{aligned}$$

■

Householder-Reflexionen $\boldsymbol{Q}$ sind Rang-1 Störungen der Identitätsmatrix; die Inverse $\boldsymbol{Q}^{-1}$ konnte einfach bestimmt werden (Satz 3.6.14), denn sie ist wegen $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^\top$ selbst wieder eine Rang-1 Störung von $\boldsymbol{I}d_n = \boldsymbol{I}d_n^{-1}$. Es gilt der folgende, allgemeinere Zusammenhang für Rang-1 Störungen invertierbarer Matrizen, die sog. *Sherman-Morrison-Woodbury Formel*:

Theorem 3.6.21 *Sei $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ regulär, u, $v \in \mathbb{R}^n$. Dann gilt:*

- *Falls $v^\top \boldsymbol{A}^{-1} u \neq -1$, dann ist*

$$(\boldsymbol{A} + uv^\top)^{-1} = \boldsymbol{A}^{-1} - \frac{1}{1 + v^\top \boldsymbol{A}^{-1} u} \boldsymbol{A}^{-1} uv^\top \boldsymbol{A}^{-1}.$$

- *Falls $v^\top \boldsymbol{A}^{-1} u = -1$, dann ist $\boldsymbol{A} + uv^\top$ nicht regulär.*

Beweis: Übung (vgl. Aufgabe 16). □

3.6.5 Anwendung auf Ausgleichsprobleme

Die $\boldsymbol{QR}$ Zerlegung und Satz 3.6.10 legen somit eine weitere Methode nahe, wie das Ausgleichsproblem (3.6.7) gelöst werden kann: man konstruiert eine orthogonale Matrix $\boldsymbol{Q} \in \boldsymbol{O_M}$, so dass $\boldsymbol{QA}$ die Rechtsdreiecksgestalt (3.6.12) hat. Diese orthogonale Matrix Q kann genauso erzeugt werden, wie bei der $\boldsymbol{QR}$-Zerlegung in Abschnitt 3.6.4: es wird zuerst eine Householder-Reflexion $\boldsymbol{Q}_1$ konstruiert, die die erste Spalte von $\boldsymbol{A}$ auf ein Vielfaches von e_1 abgebildet. Anschliessend wird eine orthogonale Matrix konstruiert, um die zweite Spalte von $\boldsymbol{Q}_1\boldsymbol{A}$ auf einen Vektor abzubilden, bei dem alle Einträge mit Ausnahme der obersten beiden Null sind u.s.w. Das Vorgehen ist also völlig analog zu Algorithmus 3.6.17—der einzige Unterschied besteht darin, dass im vorliegenden Fall einer Matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ mit $m > n$ nur n Householder-Reflexionen berechnet werden müssen und nicht $m-1$ wie im Falle einer $m \times m$ Matrix (Übung: warum reichen im Fall $m > n$ hier $n-1$ Householder-Reflexionen nicht aus?)

Algorithmus 3.6.22 ($\boldsymbol{QR}$-Zerlegung mit Householder-Reflexionen)
input : Matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $m > n$, Spalten von $\boldsymbol{A}$ linear unabhängig
output: orthogonale Matrix $\boldsymbol{Q}$ und Rechtsdreiecksmatrix $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ mit $\boldsymbol{QR} = \boldsymbol{A}$
$\boldsymbol{A}^{(1)} := \boldsymbol{A}$
$\boldsymbol{R} := 0 \in \mathbb{R}^{m\times n}$
`for` k `from` 1 `to` n `do` {

1. $a(1:m-k+1) := \boldsymbol{A}^{(k)}(1:m-k+1,1)$
   %$a :=$ Vektor der Länge $m-k+1$, der die 1. Spalte von $\boldsymbol{A}^{(k)}$ enthält

2. $\alpha := -\operatorname{sign}(a_1)\sqrt{\boldsymbol{A}^\top a}$

3. $v_k(1) := a(1) - \alpha$
   $v_k(2:m-k+1) := a(2:m-k+1)$
   % $v_k = a - \alpha e_1, \quad e_1 \in \mathbb{R}^{m-k+1}$ ist der 1. Einheitsvektor im $\mathbb{R}^{m-k+1}$

4. $\boldsymbol{Q}' := \mathrm{Id}_{m-k+1} - \dfrac{2}{v_k^\top v_k} v_k v_k^\top$

5. `berechne` $\boldsymbol{B} := \boldsymbol{Q}'\boldsymbol{A}^{(k)}$.

6. $\boldsymbol{R}(k,k:n) := \boldsymbol{B}(1,1:n-k+1)$ `%` k`-te Zeile von` $\boldsymbol{R}$

7. $\boldsymbol{A}^{(k+1)} := \boldsymbol{B}(2:m-k+1,2:n-k+1)$
   %$\boldsymbol{A}^{(k+1)}$ ist die Untermatrix von $\boldsymbol{B}$, die durch Streichen
   % der 1. Zeile und 1. Spalte entsteht

8. `setze`

$$\boldsymbol{Q}_k := \left(\begin{array}{c|c} \mathrm{Id}_{k-1} & 0 \\ \hline 0 & \boldsymbol{Q}' \end{array}\right) \qquad \% \text{ Konvention: } \quad \boldsymbol{Q}_1 = \boldsymbol{Q}'$$

}
$\boldsymbol{R}(n,n) := \boldsymbol{A}^{(m+1)}(1,1)$
`setze` $\boldsymbol{Q} := \boldsymbol{Q}_n\boldsymbol{Q}_{n-1}\boldsymbol{Q}_{n-2}\cdots\boldsymbol{Q}_1$
`return` $(\boldsymbol{Q}^\top, \boldsymbol{R})$

Beispiel 3.6.23 Wir führen das Vorgehen wieder mit den Zahlen aus Beispiel 3.6.1 vor. Aus Platzgründen geben wir nur 4 Ziffern an. Wie in Beispiel 3.6.20 unterscheidet sich unser Vorgehen hier von Algorithmus 3.6.17 dadurch, dass die Householder-Vektoren v_k nicht auf $(v_k)_1 = 1$ normiert werden.

$$\boldsymbol{A} = \begin{pmatrix} 0.1 & -0.005 \\ 0.4 & -0.08 \\ 0.5 & -0.125 \\ 0.9 & -0.405 \\ 1.0 & -0.5 \\ 1.2 & -0.72 \\ 2.0 & -2.0 \end{pmatrix}$$

Die erste Spalte von $\boldsymbol{A}$ ist

$$\boldsymbol{A} = \begin{pmatrix} 0.1 \\ 0.4 \\ 0.5 \\ 0.9 \\ 1.0 \\ 1.2 \\ 2.0 \end{pmatrix}, \qquad \|a\|_2 = 2.7695.$$

Damit ist $\alpha = -\text{sign}(0.1) \cdot 2.7695 = -2.7695$. Für $v = a - \alpha e_1$ ergibt sich

$$v = \begin{pmatrix} 0.1 \\ 0.4 \\ 0.5 \\ 0.9 \\ 1.0 \\ 1.2 \\ 2.0 \end{pmatrix} + 2.7695 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 2.8695 \\ 0.4000 \\ 0.5000 \\ 0.9000 \\ 1.0000 \\ 1.2000 \\ 2.0000 \end{pmatrix}$$

Die erste Householder-Reflexion ist somit

$$\boldsymbol{Q}_1 = \boldsymbol{I}d_7 - \frac{2}{\|v\|_2^2} vv^\top = \begin{pmatrix} -0.0361 & -0.1444 & -0.1805 & -0.3250 & -0.3611 & -0.4333 & -0.7222 \\ -0.1444 & 0.9799 & -0.0252 & -0.0453 & -0.0503 & -0.0604 & -0.1007 \\ -0.1805 & -0.0252 & 0.9685 & -0.0566 & -0.0629 & -0.0755 & -0.1258 \\ -0.3250 & -0.0453 & -0.0566 & 0.8981 & -0.1133 & -0.1359 & -0.2265 \\ -0.3611 & -0.0503 & -0.0629 & -0.1133 & 0.8742 & -0.1510 & -0.2517 \\ -0.4333 & -0.0604 & -0.0755 & -0.1359 & -0.1510 & 0.8188 & -0.3020 \\ -0.7222 & -0.1007 & -0.1258 & -0.2265 & -0.2517 & -0.3020 & 0.4967 \end{pmatrix}$$

Man berechnet $\boldsymbol{Q}_1\boldsymbol{A}$:

$$\boldsymbol{Q}_1\boldsymbol{A} = \begin{pmatrix} -2.7695 & 2.1027 \\ 0 & 0.2138 \\ 0 & 0.2423 \\ 0 & 0.2561 \\ 0 & 0.2345 \\ 0 & 0.1614 \\ 0 & -0.5309 \end{pmatrix}$$

Die erste Zeile von $\boldsymbol{R}$ ist somit $(-2.7695, 2.1027)$. Wir wiederholen obige Schritte für die "Untermatrix "

$$\boldsymbol{A}^{(1)} = \begin{pmatrix} 0.2138 \\ 0.2423 \\ 0.2561 \\ 0.2345 \\ 0.1614 \\ -0.5309 \end{pmatrix}$$

Wir setzen

$$\boldsymbol{A} = \begin{pmatrix} 0.2138 \\ 0.2423 \\ 0.2561 \\ 0.2345 \\ 0.1614 \\ -0.5309 \end{pmatrix}, \qquad \|a\|_2 = 0.7300; \qquad \alpha = -\text{sign}(0.2138) \cdot 0.73 = -0.73$$

und erhalten damit für $v = a - \alpha e_1$:

$$v = \begin{pmatrix} 0.2138 \\ 0.2423 \\ 0.2561 \\ 0.2345 \\ 0.1614 \\ -0.5309 \end{pmatrix} + 0.73 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.9438 \\ 0.2423 \\ 0.2561 \\ 0.2345 \\ 0.1614 \\ -0.5309 \end{pmatrix}$$

Somit ist

$$\boldsymbol{Q}^{(1)} = \boldsymbol{I}d_6 - \frac{2}{\|v\|_2^2} vv^\top = \begin{pmatrix} -0.2929 & -0.3319 & -0.3508 & -0.3213 & -0.2212 & 0.7273 \\ -0.3319 & 0.9148 & -0.0900 & -0.0825 & -0.0568 & 0.1867 \\ -0.3508 & -0.0900 & 0.9048 & -0.0872 & -0.0600 & 0.1973 \\ -0.3213 & -0.0825 & -0.0872 & 0.9202 & -0.0550 & 0.1807 \\ -0.2212 & -0.0568 & -0.0600 & -0.0550 & 0.9622 & 0.1244 \\ 0.7273 & 0.1867 & 0.1973 & 0.1807 & 0.1244 & 0.5909 \end{pmatrix}$$

Also

$$\boldsymbol{Q}^{(1)}\boldsymbol{A}^{(1)} = \begin{pmatrix} -0.73 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Die gesuchte Rechtsdreiecksmatrix $R \in \mathbb{R}^{2\times 2}$ ist damit

$$\boldsymbol{R} = \begin{pmatrix} -2.7695 & 2.1027 \\ 0 & -0.73 \end{pmatrix}.$$

Wir erhalten damit für die orthogonale Matrix $\boldsymbol{Q}_2$:

$$\boldsymbol{Q}_2 = \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \boldsymbol{Q}^{(1)} \end{array}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.2929 & -0.3319 & -0.3508 & -0.3213 & -0.2212 & 0.7273 \\ 0 & -0.3319 & 0.9148 & -0.0900 & -0.0825 & -0.0568 & 0.1867 \\ 0 & -0.3508 & -0.0900 & 0.9048 & -0.0872 & -0.0600 & 0.1973 \\ 0 & -0.3213 & -0.0825 & -0.0872 & 0.9202 & -0.0550 & 0.1807 \\ 0 & -0.2212 & -0.0568 & -0.0600 & -0.0550 & 0.9622 & 0.1244 \\ 0 & 0.7273 & 0.1867 & 0.1973 & 0.1807 & 0.1244 & 0.5909 \end{pmatrix}.$$

Setzt man

$$\boldsymbol{Q} \quad = \quad \boldsymbol{Q}_2\boldsymbol{Q}_1 \begin{pmatrix} -0.0361 & -0.1444 & -0.1805 & -0.3250 & -0.3611 & -0.4333 & -0.7222 \\ -0.0972 & -0.3064 & -0.3488 & -0.3813 & -0.3551 & -0.2618 & 0.6595 \\ -0.1684 & -0.3553 & 0.8855 & -0.1429 & -0.1412 & -0.1272 & 0.0693 \\ -0.3121 & -0.3943 & -0.1444 & 0.8069 & -0.1959 & -0.1905 & -0.0202 \\ -0.3493 & -0.3700 & -0.1433 & -0.1967 & 0.7984 & -0.2010 & -0.0628 \\ -0.4252 & -0.2804 & -0.1309 & -0.1934 & -0.2031 & 0.7844 & -0.1720 \\ -0.7488 & 0.6229 & 0.0562 & -0.0375 & -0.0802 & -0.1887 & 0.0690 \end{pmatrix}$$

so gilt nach Konstruktion

$$\boldsymbol{QA} = \begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} -2.7695 & 2.1027 \\ 0 & -0.73 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Wir berechnen

$$\boldsymbol{Q}y = \boldsymbol{Q}\begin{pmatrix} 0.96 \\ 3.26 \\ 3.82 \\ 5.11 \\ 5.2 \\ 5.05 \\ 0.58 \end{pmatrix} = \begin{pmatrix} -7.3404 \\ -7.1590 \\ -0.0037 \\ -0.0063 \\ 0.0058 \\ -0.0055 \\ 0.0046 \end{pmatrix}$$

Die gewünschte Partitionierung von Qy ist damit

$$\boldsymbol{Q}y = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \qquad b_1 = \begin{pmatrix} -7.3404 \\ -7.1590 \end{pmatrix}, \qquad b_2 = \begin{pmatrix} -0.0037 \\ -0.0063 \\ 0.0058 \\ -0.0055 \\ 0.0046 \end{pmatrix}$$

Somit erhalten wir als das zu lösende Gleichungssystem $\boldsymbol{R}x\boldsymbol{R}x = b_1$:

$$\begin{pmatrix} -2.7695 & 2.1027 \\ 0 & -0.73 \end{pmatrix} x = \begin{pmatrix} -7.3404 \\ -7.1590 \end{pmatrix}.$$

Aufgelöst nach $x = (v_y, g)^\top$ ergibt sich:

$$x = \begin{pmatrix} 10.083 \\ 9.807 \end{pmatrix}.$$

■

Bezüglich der Konditionszahl der Rechtsdreiecksmatrix $\boldsymbol{R}$ aus der $\boldsymbol{QR}$ Zerlegung von $\boldsymbol{A}$ und der Matrix $\boldsymbol{A}^\top\boldsymbol{A}$ aus den Normalengleichungen gilt folgendes Ergebnis.

Theorem 3.6.24 *Unter den Annahmen von Satz 3.6.10 gilt $\boldsymbol{A}^\top A = R^\top R$. Weiterhin seien $\sigma_1 \geq ... \geq \sigma_n > 0$ die Singulärwerte von $A \in \mathbb{R}^{m\times n}$. Dann gilt*

$$\kappa_2(\boldsymbol{R}) = \frac{\sigma_1}{\sigma_n}, \qquad \textit{und} \quad \kappa_2(\boldsymbol{A}^\top\boldsymbol{A}) = (\kappa_2(\boldsymbol{R}))^2 = \frac{\sigma_1^2}{\sigma_n^2}.$$

Beweis: Übung. Hinweis: Für SPD-Matrizen B gilt nach dem Rayleigh-Ritz Prinzip:

$$\lambda_{min}(\boldsymbol{B}) = \min_{0\neq x\in\mathbb{R}^m} \frac{x^\top \boldsymbol{B} x}{x^\top x}, \qquad \lambda_{max}(\boldsymbol{B}) = \max_{0\neq x\in\mathbb{R}^m} \frac{x^\top \boldsymbol{B} x}{x^\top x}.$$

□

Bemerkung 3.6.25 Wir hatten in Abschnitt 3.6.4 gesehen, dass die $\boldsymbol{QR}$-Zerlegung einer invertierbaren Matrix eindeutig ist, falls $\boldsymbol{A}$ vollen Spaltenrang hat und das Vorzeichen der Diagonaleinträge von $\boldsymbol{R}$ festliegt. Auch bei Zerlegungen wie in Satz 3.6.10 ist die Rechtsdreiecksmatrix $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ eindeutig, wenn das Vorzeichen der Diagonaleinträge festgelegt ist. Wird verlangt, dass $R_{ii} > 0$, $i = 1, \ldots, m$, dann ist $R^\top$ der Cholesky-Faktor von $\boldsymbol{A}^\top\boldsymbol{A}$. ■

3.6.6 Rechteckmatrizen mit Rangdefekt. Singulärwertzerlegung.

Bisher betrachteten wir Systeme

$$\boldsymbol{A}\underline{x} = \underline{b}$$

mit $\underline{x} \in \mathbb{R}^n$, $\underline{b} \in \mathbb{R}^m$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, und $m \geq n$ $\operatorname{rank}(\boldsymbol{A}) = n$, etwa aus dem Abgleich eines physikalischen Gesetzes

$$f(t) = \sum_{j=1}^{n} x_j\, \Phi_j(t) \tag{3.6.18}$$

mit $m > n$ Datenpaaren (t_i, f_i), $i = 1 : m$.

Insbesondere ist also $f(t) \in V := \operatorname{span}\{\Phi_j(t) : j = 1 : n\}$ und

$$A = (a_{ij})_{i=1:m, j=1:n} \quad \text{mit} \quad a_{ij} = \Phi_j(t_i). \tag{3.6.19}$$

Seien nun $n - r > 0$ der Basisfunktionen $\Phi_j(t)$ in (3.6.18) linear abhängig, d.h. die Dimension des Vektorraums $V = \operatorname{span}\{\Phi_1(t), ..., \Phi_n(t)\}$ ist $r < n$.

Seien weiter $\sigma_1 \geq ... \geq \sigma_n \geq 0$ die Singulärwerte der Matrix $\boldsymbol{A}$ in (3.6.19). Dann ist $\sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_n = 0$ und, mit einer Lösung $\underline{x}^* \in \mathbb{R}^n$ von

$$\Phi(x^*) := \|\boldsymbol{A}\underline{x}^* - \underline{b}\|_2^2 \leq \|\boldsymbol{A}\underline{y} - \underline{b}\|_2^2 \quad \forall \underline{y} \in \mathbb{R}^n \tag{3.6.20}$$

ist auch $\underline{x}^* + \underline{z}$ für jedes $\underline{z} \neq 0$, $\underline{z} \in \ker(\boldsymbol{A})$ Lösung von (3.6.20). Eine Zusatzbedingung ist also nötig, um (3.6.20) eindeutig lösbar zu machen. Wir fordern, dass $\underline{x}^*$ unter allen Lösungen von (3.6.20) die kleinste $\| \circ \|_2$ Norm hat:

finde $x^* \in \mathbb{R}^n$ mit $\|x^*\|_2 \stackrel{!}{=} \min$ so, dass

$$\|\boldsymbol{A}\underline{x}^* - \underline{b}\|_2 \leq \min_{\underline{y} \in \mathbb{R}^n} \|\boldsymbol{A}\underline{y} - \underline{b}\|_2 \tag{3.6.21}$$

ist.

Theorem 3.6.26 *Sei $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\underline{b} \in \mathbb{R}^m$ gegeben mit $r = \operatorname{rang}(\boldsymbol{A}) < n < m$. Dann ist $\underline{x}^* \in \mathbb{R}^n$ gegeben durch $\underline{x}^* = \boldsymbol{A}^+\underline{b}$, die eindeutige Lösung von (3.6.21).*

Beweis: Sei $\boldsymbol{A} = \boldsymbol{V}\,\boldsymbol{\Sigma}\,\boldsymbol{U}^\top$ die SVD von $\boldsymbol{A}$. Dann ist (3.6.21) äquivalent zur Minimierung von

$$\Phi(\underline{y}) = \|\boldsymbol{V}\,\boldsymbol{\Sigma}\,\boldsymbol{U}^\top\underline{y} - \underline{b}\|_2^2 = \|\boldsymbol{V}(\boldsymbol{\Sigma}\,\boldsymbol{U}^\top\underline{y} - \boldsymbol{V}^\top b)\|_2^2 = \|\boldsymbol{\Sigma}\underline{w} - \boldsymbol{V}^\top\underline{b}\|_2^2$$

über alle $\underline{w} = \boldsymbol{U}^\top\underline{y} \in \mathbb{R}^n$. Hieraus folgt, dass

$$\Phi(\underline{y}) = \Phi(\boldsymbol{U}\underline{w}) = \sum_{i=1}^{r} (\sigma_i w_i - (\boldsymbol{V}^\top\underline{b})_i)^2 + \sum_{i=r+1}^{n} (\boldsymbol{V}^\top\underline{b})_i^2 \,.$$

Also ist $\Phi(\underline{y})$ minimal für $w_i = \sigma_i^{-1}(\boldsymbol{V}^\top b)_i$, $i = 1, \ldots, r$. Da $\boldsymbol{U} \in O_n$, ist $\|\underline{y}\|_2 = \|\boldsymbol{U}\underline{w}\|_2 = \|\underline{w}\|_2$. Also ist $\|\underline{y}\|_2$ minimal für $w_{r+1} = \cdots = w_n = 0$. Somit ist $\underline{w}^* = \boldsymbol{\Sigma}^+\boldsymbol{V}^\top\underline{b}$ Lösung, woraus folgt $\underline{x}^* = \boldsymbol{U}\boldsymbol{\Sigma}^+\boldsymbol{V}^\top\underline{b}$. □

3.6.7 ε-Rang einer Matrix

Sind die Basisfunktionen Φ_j im linearen physikalischen Gesetz (3.6.18) 'fast' linear abhängig oder die Messpunkte t_i so, dass sich die Φ_j 'fast' nicht auf dem Bereich der t_i unterscheiden, haben die Spalten von $\boldsymbol{A}$ noch mathematisch vollen Rang, aber das Ausgleichsproblem ist schlecht konditioniert: kleine Fehler in den Daten können grosse Änderungen im Lösungsvektor des Ausgleichsproblems nach sich ziehen.

In diesem Fall sind in der SVD

$$\boldsymbol{A} = \boldsymbol{V}\,\boldsymbol{\Sigma}\,\boldsymbol{U}^\top \tag{3.6.22}$$

von $\boldsymbol{A}$ einige der Singulärwerte $\sigma_{r+1} \geq ... \geq \sigma_n \geq 0$ entweder gleich 0 oder aber sehr klein und daher ist $\kappa_2(\boldsymbol{A}^\top\boldsymbol{A}) = (\sigma_1/\sigma_n)^2$ gross: kleine Messfehler in den Daten $(f_i : i = 1 : m)$ werden verstärkt im Lösungsvektor der Normalgleichungen auftreten.

Sei deshalb jetzt $\varepsilon > 0$ eine Fehlerschranke der f_i (z.B. ist ε ein Messfehler in den Daten; aufgrund von Rundungsfehlern in Gleitkommaarithmetik $\mathbb{F}$ gilt übrigens immer $\varepsilon \geq \varepsilon_M(\mathbb{F})$ mit dem Maschinenepsilon aus (2.2.2)).

Dann definieren wir den **ε-Rang des Systems $\{\Phi_i\}$ auf den Datenpunkten t_i** wie folgt:

$$r(\varepsilon) = \max\{k : \sigma_k \geq \varepsilon\} \tag{3.6.23}$$

Offensichtlich ist der ε-Rang des Systems $\{\Phi_i\}$ eine Verallgemeinerung des mathematischen Rangs – der 0-Rang des Systems entspricht der Dimension des VR $\operatorname{span}\{\Phi_1(t), ..., \Phi_n(t)\}$.

Beachte, dass $r(\varepsilon)$ abhängt von der Wahl der t_i sowie der Wahl der Φ_j, nicht aber von den Werten f_i:

$$r = r(\varepsilon, \{t_1, ..., t_m\}, \{\Phi_1, ..., \Phi_n\}).$$

und es gilt natürlich immer

$$r(\varepsilon) \leq r(0) = \operatorname{rank}(\boldsymbol{A}) \leq n < m\,.$$

Kann im Falle eines kleinen ε-Rangs von $\boldsymbol{A}$ die SVD von $\boldsymbol{A}$ ausser zur Verbesserung der numerischen Stabilität noch weitergehend benutzt werden? In der Tat kann man mittels der Singulärvektoren $\underline{u}_1, ..., \underline{u}_{r(\varepsilon)}$, also den ersten $r(\varepsilon)$ Spalten der Matrix $\boldsymbol{U} = [\underline{u}_1, ..., \underline{u}_n]$ in der SVD (3.6.22) von $\boldsymbol{A}$, aus dem Vektor

$$\underline{\Phi}(t) := (\Phi_1(t), ..., \Phi_n(t))^\top$$

der Basisfunktionen $\Phi_j(t)$ die $r(\varepsilon)$ 'dominanten' Basisfunktionen für den VR $\operatorname{span}\{\Phi_1, ..., \Phi_n\}$ herausziehen: es sind dies die Funktionen

$$\phi_j(t) := \underline{u}_j^\top \underline{\Phi}(t), \qquad j = 1, ..., r(\varepsilon)$$

wobei $\underline{u}_j$ die j-te Spalte der Matrix $\boldsymbol{U}$ in der SVD von $\boldsymbol{A}$ ist.