Skript zur Vorlesung

Angewandte Mathematik: Numerik

Wintersemester 2022/23

universitätbonn

Institut für Informatik II
Computer Graphik

Professor:
Prof. Dr. Reinhard Klein

Assistenten:
Dipl.-Inform. Alexander Schier

Bonn, 26. Januar 2023

Inhaltsverzeichnis

Vorwort

Dieses Skript basiert auf und folgt in weiten Teilen den folgenden Lehrbüchern:

- Numerical Linear Algebra
  Lloyd N. Trefethen, David Bau, III
  ISBN: 978-0-898713-61-9 [3]

- Analysis 2
  2. erweiterte Auflage
  Konrad Königsberger
  ISBN: 978-3540203896 [2]

Zum Lernen und zur weiteren Vertiefung empfehlen wir, diese Bücher zusätzlich zum Skript zu verwenden.

Hinweis:
Fehler und Anmerkungen für Ergänzungen zum Skript können an *schier@uni-bonn.de* geschickt werden.

Teil I

Numerische Lineare Algebra

Kapitel 1

Grundlagen

1.1 Vektoren und Matrizen

Seien x ein $n-$dimensionaler Spaltenvektor und A eine $(m \times n)$-Matrix (m Zeilen, n Spalten) jeweils über $\mathbb{C}$. $b = Ax$ ist der m-dimensionale Spaltenvektor definiert durch

$$b_i = \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, \ldots, m \tag{1.1}$$

wobei b_i der $i-te$ Eintrag von b und (a_{ij}) der Eintrag an ij-ter Stelle (i-te Zeile, j-te Spalte) in A ist.

Interpretation:

Sei a_j die j-te Spalte von A (d.h. a_j ist ein Vektor mit m Einträgen). Dann können wir das Vektor-Matrixprodukt in Gleichung (1.1) schreiben als

$$b = Ax = \sum_{j=1}^{n} x_j a_j, \tag{1.2}$$

d.h. b ist eine *Linearkombination der Spaltenvektoren* von A. Schematisch:

$$\begin{pmatrix} \\ b \\ \\ \end{pmatrix} = \left(\begin{array}{c|c|c|c} & & & \\ a_1 & a_2 & \ldots & a_n \\ & & & \end{array}\right) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \underbrace{x_1 \begin{pmatrix} \\ a_1 \\ \\ \end{pmatrix} + x_2 \begin{pmatrix} \\ a_2 \\ \\ \end{pmatrix} + \cdots + x_n \begin{pmatrix} \\ a_n \\ \\ \end{pmatrix}}_{\text{Linearkombination}}$$

Beispiel (Matrix-Vektor-Multiplikation):

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} = 0.5 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.5 \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Bemerkung 1.1: Bilder der kanonischen Basisvektoren

Die Spaltenvektoren $a_i, i = 1, ..., n$, von A geben an, wohin die kanonischen Einheitsvektoren e_i, d.h. die Vektoren mit 1 an der i-ten Stelle und 0 sonst, abgebildet werden.

Bemerkung: Vergleiche die Faustregel zur Matrix-Multiplikation:
Zeile $\times$ *Spalte*. Seien A eine $(l \times m)$-Matrix und C eine $(m \times n)$-Matrix definiert durch

$$b_{ij} = \sum_{k=1,...,m} a_{ik} c_{kj}, \quad i = 1, \dots, l, j = 1, \dots, n. \tag{1.3}$$

Schematisch:

$$\left(\begin{array}{c|c|c|c} b_1 & b_2 & \dots & b_n \end{array} \right) = \left(\begin{array}{c|c|c|c} a_1 & a_2 & \dots & a_m \end{array} \right) \left(\begin{array}{c|c|c|c} c_1 & c_2 & \dots & c_n \end{array} \right)$$

d.h. $b_j = Ac_j = \sum_{k=1,...,m} c_{kj} a_k$, also ist b_j eine *Linearkombination der Spalten* a_k *mit den Koeffizienten* c_{kj}.

Das Produkt eines Spaltenvektors u ($l \times 1$-Matrix) mit einem Zeilenvektor v ($1 \times n$-Matrix) heißt *Äußeres Produkt*:

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_l \end{pmatrix} \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix} = \left(\begin{array}{c|c|c|c} v_1 u & v_2 u & \dots & v_n u \end{array} \right) = \left(\begin{array}{c} u_1 v \\ \hline u_2 v \\ \hline \vdots \\ \hline u_l v \end{array} \right)$$

$$= \begin{pmatrix} u_1 v_1 & \cdots & u_1 v_n \\ \vdots & & \vdots \\ u_l v_1 & \cdots & u_l v_n \end{pmatrix}$$

Alle Spalten sind Vielfache des Vektors u und alle Zeilen sind Vielfache von v.

1.2 Vektorräume

Definition 1.1: Bild einer Matrix

Das Bild einer Matrix $A \in \mathbb{C}^{m \times n}$, geschrieben als $Bild(A)$, ist die Menge der Vektoren $v \in \mathbb{C}^m$, für die es ein $x \in \mathbb{C}^n$ gibt, so dass gilt $v = Ax$, d.h.

$$Bild(A) = \{Ax \mid x \in \mathbb{C}^n\}$$

Definition 1.2: Nullraum (Kern)

Der Nullraum (auch als Kern bezeichnet) einer Matrix $A \in \mathbb{C}^{m \times n}$, geschrieben als $Null(A)$, ist die Menge der Vektoren $x \in \mathbb{C}^n$, für die gilt $Ax = 0$, wobei 0 der 0-Vektor im $\mathbb{C}^m$ ist, d.h.

$$Null(A) = \{x \in \mathbb{C}^n \mid Ax = 0\}$$

Theorem 1.1: Bildraum

$Bild(A)$ ist der lineare Vektorraum, der von den Spalten von A aufgespannt wird.

Beweis: **Bemerkung:** Jeder Vektor $x \in Kern(A)$ beschreibt eine Darstellung der 0 als Linearkombination der Spalten von A: $0 = x_1a_1 + x_2a_2 + ... + x_na_n$.
Sei $y = Ax$. Dann ist $y = \sum_{1 \leqslant j \leqslant n} x_j a_j$ Linearkombination von Spaltenvektoren, d.h. $y \in L\{a_1, ..., a_n\}$. Sei umgekehrt y in der linearen Hülle der Spalten von A, so gilt $y = \sum_{1 \leqslant j \leqslant n} x_j a_j$ für Koeffizienten $x_j, j = 1, ..., n$. Wählt man $x = (x_1, ..., x_n)^T$, so ist $y = Ax$. □

Bemerkung: Zeilenrang und Spaltenrang sind gleich (folgt aus der SVD, Kapitel 3).

Definition 1.3: Rang einer Matrix

Der Spaltenrang einer Matrix ist die Dimension des Spaltenraums. Der Zeilenrang einer Matrix ist die Dimension des Zeilenraums. Eine $(m \times n)$-Matrix ist von *vollem Rang*, falls sie maximal möglichen Rang besitzt, d.h. eine $(m \times n)$-Matrix mit $m \geqslant n$ mit vollem Rang hat n linear unabhängige Spalten.

Bemerkung: Nur quadratische Matrizen können invertiert werden. Eine Verallgemeinerung der Inversen einer Matrix ist die *Pseudoinverse*
(vgl. Definition 3.5).

Definition 1.4: invertierbar

Eine invertierbare (nicht singuläre) Matrix A ist eine Matrix von vollem Rang, d.h. die Spalten einer invertierbaren $(m \times m)$-Matrix bilden eine Basis des $\mathbb{C}^m$.

Wir können also jeden Vektor in $\mathbb{C}^m$ eindeutig als Linearkombination der Spaltenvektoren von A schreiben. Insbesondere gibt es für jeden kanonischen Einheitsvektor e_j einen eindeutigen Vektor z_j, sodass gilt:

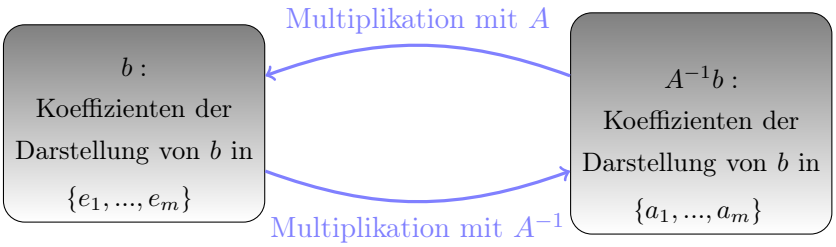
$$e_j = Az_j = \sum_{i=1}^{m} z_{ij} a_i, \qquad j = 1, ..., m, \; A \in \mathbb{C}^{m \times m} \tag{1.4}$$

Sei Z die Matrix mit den Spaltenvektoren z_j, dann gilt:

$$\left(\begin{array}{c|c|c|c} e_1 & e_2 & \dots & e_n \end{array} \right) = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} = I = AZ$$

I ist hierbei die Einheitsmatrix und Z ist die *Inverse* von A. Sie ist eindeutig bestimmt und wird auch als A^{-1} geschrieben. Es gilt $I = AA^{-1} = A^{-1}A$.

Produkt von Inverser Matrix und Vektor:
Sei $x = A^{-1}b$. Dann gilt $Ax = AA^{-1}b = Ib = b$. x ist also der eindeutige Vektor, der die Gleichung $Ax = b$ erfüllt. D.h. die x_i sind die Koeffizienten der Darstellung von b als Linearkombination der Spaltenvektoren von A. Die Multiplikation mit A^{-1} ist ein Basiswechsel:



Definition 1.5: Adjungierte Matrix

Die hermitesch adjungierte Matrix A^* (oder nur Adjungierte) einer Matrix $A \in \mathbb{C}^{m\times n}$ ist die $(n \times m)$-Matrix, deren (i, j)-ter Eintrag das komplex Konjugierte des (j, i)-ten Eintrags ist, d.h. $A^* = \overline{A}^T$.

Es gilt:

- Falls $A = A^*$ gilt, so heißt A *hermitesch*.
- Für reelle A ist $A^* = A^T$.
- Falls $A = A^T$ gilt, so heißt A *symmetrisch*.

Definition 1.6: Skalarprodukt

Für zwei Spaltenvektoren $x, y \in \mathbb{C}^m$ definieren wir das Skalarprodukt als

$$x^*y = \begin{pmatrix} \overline{x}_1 & \overline{x}_2 & \cdots & \overline{x}_m \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \sum_{i=1}^{m} \overline{x}_i y_i$$

Für $x, y \in \mathbb{R}^m$ entfällt die komplexe Konjugation und man erhält $x^*y = \sum_{i=1}^{m} x_i y_i$.

Bemerkung 1.2: Alternative Schreibweisen für das Skalarprodukt

Statt x^*y schreiben wir auch $\langle x, y\rangle$ oder $\langle x|y\rangle$.

Definition 1.7: Orthogonale Vektoren

Ein Paar von Vektoren x, y heißt orthogonal falls $x^*y = 0$. D.h. falls $x, y \in \mathbb{R}^m$, so bedeutet das, dass $x \perp y$ ist.
Eine Menge von Null verschiedener Vektoren S ist *orthogonal*, falls ihre Elemente paarweise orthogonal sind.
Eine Menge von Vektoren S heißt *orthonormal*, falls sie orthogonal ist und $\forall x \in S$

gilt $\|x\| = 1$.

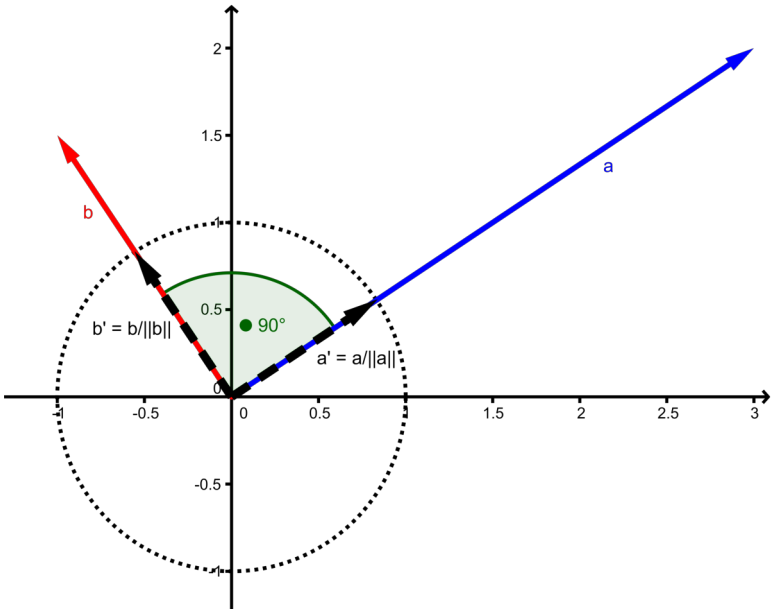
Beispiel (Orthogonale Vektoren):

Die Vektoren $a = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$ und $b = \begin{pmatrix} -1 \\ 1.5 \end{pmatrix}$ stehen orthogonal zueinander, denn:

$$\begin{aligned} a^* b &= a_1 * b_1 + a_2 * b_2 \\ &= 3 * (-1) + 2 * 1.5 = 0 \end{aligned}$$

$a' := \frac{a}{\|a\|}$ und $b' := \frac{b}{\|b\|}$ sind *orthonormale* Vektoren:

$$\begin{aligned} \|a\| &= \sqrt{a_1^2 + a_2^2} = \sqrt{3^2 + 2^2} = \sqrt{13} \\ \|a'\| &= \left\| \frac{a}{\|a\|} \right\| = \left\| \frac{1}{\sqrt{13}} \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right\| = \sqrt{\left(\frac{3}{\sqrt{13}}\right)^2 + \left(\frac{2}{\sqrt{13}}\right)^2} = \sqrt{\frac{\sqrt{13}}{\sqrt{13}}} = 1 \\ \|b'\| &\quad \text{analog} \end{aligned}$$



`Abbildung` 1.1: Orthogonale und orthonormale Vektoren.

Theorem 1.2: Vektoren in einer orthogonalen Menge

Die Vektoren in einer orthogonalen Menge $S := \{v_1, \ldots, v_n\} \subset \mathbb{C}^m \backslash \{0\}$ sind linear unabhängig.

Beweis: (durch Widerspruch)

Angenommen, die Vektoren einer orthogonalen Menge sind nicht linear unabhängig.

Dann kann der Vektor $v_k \in S$ als Linearkombination der übrigen Vektoren $v \in S\backslash\{v_k\}$ geschrieben werden, d.h. $v_k = \sum_{i=1, i\neq k}^{n} \lambda_i v_i$ $(\lambda_1, \ldots, \lambda_n \in \mathbb{K})$. Da $v_k \neq 0$ folgt $v_k^* v_k = \|v_k\|^2 > 0$:

$$0 < v_k^* v_k = \sum_{i=1, i\neq k}^{n} \lambda_i v_k^* v_i = 0.$$

Wegen $v_k^* v_i = 0$ folgt der Widerspruch. □

Das Skalarprodukt kann verwendet werden, um einen beliebigen Vektor in orthogonale Komponenten zu zerlegen.
Sei $\{q_1, ..., q_n\}$ eine orthonormale Menge von Vektoren und v ein beliebiger Vektor. Dann ist der Vektor

$$r = v - (q_1^* v)q_1 - (q_2^* v)q_2 - ... - (q_n^* v)q_n$$

orthogonal zu $\{q_1, ..., q_n\}$:

$$q_i^* r = q_i^* v - \sum_{j=1}^{n} (q_j^* v) \underbrace{(q_i^* q_j)}_{=0, \text{wenn } i\neq j} = q_i^* v - (q_i^* v) \underbrace{(q_i^* q_i)}_{=1} = 0$$

Bemerkung: Diese Tatsache ist aus der linearen Algebra bekannt als **Basisergänzungssatz** (BES).
Zur Erinnerung:
In einem endlich erzeugten Vektorraum V seien linear unabhängige Vektoren $w_1, ..., w_n$ gegeben. Dann kann man $w_{n+1}, ..., w_r$ finden, sodass $B = (w_1, ..., w_n, w_{n+1}, ..., w_r)$ eine Basis von V ist. [**?**]
Es ist also möglich, v als Linearkombination von $n+1$ orthogonalen Vektoren $\{q_1, \ldots, q_n, r\}$ darzustellen:

$$v = r + \sum_{j=1}^{n} (q_j^* v) q_j.$$

Falls $\{q_1, \ldots, q_n\}$ eine Basis des $\mathbb{C}^n$ darstellen, so kann es keine Vektoren $r \neq 0$ geben, die orthogonal zu allen q_i sind und somit ist $r = 0$.
Jeder Vektor v kann daher in diesem Fall darstellt werden als:

$$v = \sum_{j=1}^{n} (q_j^* v) q_j$$

Definition 1.8: Unitäre Matrizen

Eine quadratische Matrix Q ist unitär (im reellen Fall orthogonal), falls $Q^{-1} = Q^*$ ist, und daher gilt: $Q^*Q = I$.

$$\left(\begin{array}{c} q_1^* \\ \hline q_2^* \\ \hline \vdots \\ \hline q_m^* \end{array}\right) \left(\begin{array}{c|c|c|c} & & & \\ q_1 & q_2 & \cdots & q_m \\ & & & \end{array}\right) = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}$$

Bemerkung: δ_i, j ist das Kronecker-Delta:

$\delta_{i,j} = \begin{cases} 1, & \text{falls } i = j \\ 0 & sonst. \end{cases}$ Es gilt also $q_i^* q_j = \delta_{ij}$. Die Spalten einer unitären Matrix bilden somit eine orthonormale Basis des $\mathbb{C}^m$.

Satz 1.3: Invarianz des Skalarprodukts

Die Multiplikation mit einer unitären Matrix erhält das Skalarprodukt.

Beweis:

$$(Qx)^*(Qy) = x^*Q^*Qy = x^*Iy = x^*y$$

□

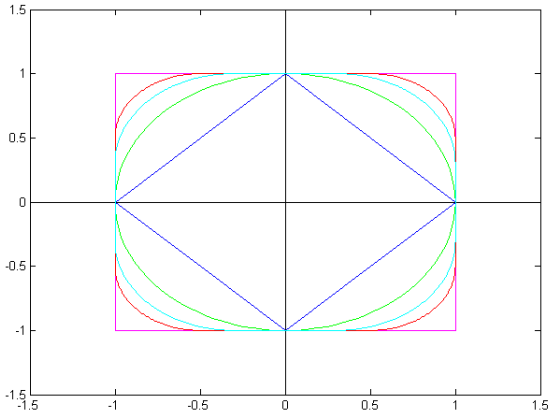
Unitäre Matrizen erhalten daher Winkel und Längen. Im reellen Fall entspricht die Multiplikation mit einer unitären Matrix einer Rotation ($\det Q = 1$) oder einer Spiegelung ($\det Q = -1$). In diesem Fall spricht man auch von *orthogonalen Matrizen*.

1.3 Normen

Definition 1.9: Norm

Eine Norm ist eine Abbildung $\|\cdot\| : \mathbb{C}^n \to \mathbb{R}$ mit folgenden Eigenschaften:

1. $\|x\| \geqslant 0$, $\|x\| = 0$ gdw. $x = 0$
2. $\|\alpha x\| = |\alpha| \cdot \|x\|$
3. $\|x + y\| \leqslant \|x\| + \|y\|$



`Abbildung` 1.2: Beispiel für p-Normen: von innen (1-Norm) nach außen (∞-Norm).

Definition 1.10: p-Norm (Vektoren)

Für eine relle Zahl $p \geqslant 1$ definiert man für einen Vektor $x \in \mathbb{C}^n$ die *p-Norm* durch

$$\|x\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}.$$

Beispiel (Euklidsche Norm):
Die p-Norm für $p = 2$ wird *euklidische Norm* genannt. Betrachte den Vektor $x = (1, 2, 3)$:

$$\|x\|_2 = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}$$

Bemerkung: $\|A\|_{(m,n)} = \sup \frac{\|Ax\|_m}{\|x\|_{(m)}} = \sup_{x \in \mathbb{C}^m, \|x\|_n = 1} \|Ax\|_{(m)}$. Eine $(m \times n)$-Matrix kann als Vektor im $\mathbb{C}^{m \cdot n}$ aufgefasst werden. Damit kann jede Vektornorm auf dem $\mathbb{C}^{m \cdot n}$ auch

als Matrixnorm verwendet werden. Es gibt jedoch andere Matrixnormen, die in vielen Fällen nützlicher sind, die *induzierten Matrixnormen*:

Definition 1.11: Induzierte Matrixnorm

Seien $\|\cdot\|_n$ und $\|\cdot\|_m$ Vektornormen des Definitions- bzw. Bildbereichs von $A \in \mathbb{C}^{m\times n}$. Dann ist $\|A\|_{(m,n)}$ definiert als die kleinste Zahl C, für die $\forall x \in \mathbb{C}^n$ gilt

$$\|Ax\|_m \leqslant C \cdot \|x\|_n,$$

d.h.

$$\|A\|_{(m,n)} = \sup_{x\in\mathbb{C}^n\setminus\{0\}} \frac{\|Ax\|_m}{\|x\|_n}.$$

Satz 1.4: Eigenschaften induzierter Matrixnormen

Induzierte Matrixnormen sind Normen und es gilt für alle $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{n\times l}$ und $x \in \mathbb{C}^n$:

- $\|A \cdot x\|_m \leqslant \|A\|_{(m,n)} \cdot \|x\|_n$
- $\|A \cdot B\|_{(m,l)} \leqslant \|A\|_{(m,n)} \cdot \|B\|_{(n,l)}$

Beweis: Die Norm ist Quotient nicht-negativer Zahlen und somit nicht negativ. Es ist $\|0\|_{(m,n)} = 0$ und aus $A \neq 0$ folgt, dass es ein $y \in \mathbb{C}^n$ mit $A \cdot y \neq 0$ gibt, so dass $\|A\|_{(m,n)} > 0$ ist. Weiter gilt für $\alpha \in \mathbb{C}$ und $C \in \mathbb{C}^{m\times n}$:

$$\begin{aligned}
\|\alpha \cdot A\|_{(m,n)} &= \sup_{y\in\mathbb{C}^n\setminus\{0\}} \frac{\|\alpha \cdot A \cdot y\|_m}{\|y\|_n} = \alpha \cdot \sup_{y\in\mathbb{C}^n\setminus\{0\}} \frac{\|A \cdot y\|_m}{\|y\|_n} = \alpha \cdot \|A\|_{(m,n)} \\
\|A + C\|_{(m,n)} &= \sup_{y\in\mathbb{C}^n\setminus\{0\}} \frac{\|A \cdot y + C \cdot y\|_m}{\|y\|_n} \\
&\leqslant \sup_{y\in\mathbb{C}^n\setminus\{0\}} \frac{\|A \cdot y\|_m}{\|y\|_n} + \sup_{y\in\mathbb{C}^n\setminus\{0\}} \frac{\|C \cdot y\|_m}{\|y\|_n} = \|A\|_{(m,n)} + \|C\|_{(m,n)}
\end{aligned}$$

Die Ungleichungen ergeben sich aus folgenden Abschätzungen:

$$\|A \cdot x\|_m = \frac{\|A \cdot x\|_m}{\|x\|_n} \cdot \|x\|_n \leqslant \sup_{y\in\mathbb{C}^n\setminus\{0\}} \frac{\|A \cdot y\|_m}{\|y\|_n} \cdot \|x\|_n = \|A\|_{(m,n)} \cdot \|x\|_n$$

$$\|A \cdot B\|_{(m,l)} = \sup_{y\in\mathbb{C}^l\setminus\{0\}} \frac{\|A \cdot B \cdot y\|_m}{\|y\|_l} \leqslant \|A\|_{(m,n)} \cdot \sup_{y\in\mathbb{C}^l\setminus\{0\}} \frac{\|B \cdot y\|_n}{\|y\|_l} = \|A\|_{(m,n)} \cdot \|B\|_{(n,l)}$$

□

Lemma 1.5: 2-Norm einer Diagonalmatrix

Sei D eine Diagonalmatrix mit $D = \begin{pmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{pmatrix}$.

Dann gilt:

$$\|D\|_2 = \sup_{\|x\|=1} \|Dx\|_2 = \max_{1 \leqslant i \leqslant m} |d_i|.$$

Bemerkung: Jede Matrix bildet die Einheitskugel der 2-Norm auf eine Hyperellipse ab (Beweis mittels SVD, Kapitel 3). ***Beweis:*** Die Matrix D skaliert die Einheitskugel entlang der Koordinatenachsen. Betrachte dazu die Multiplikation von D mit den kanonischen Einheitsvektoren e_i. Jeder Vektor e_i wird um den Faktor d_i gestreckt. Dabei entspricht die maximale Streckung dem betragsmäßig größtem Diagonaleintrag von D. □

Definition 1.12: 1-Norm einer Matrix (maximale Spaltensumme)

Sei A eine $(m \times n)$-Matrix. Dann entspricht $\|A\|_1$ der maximalen Spaltensumme:

$$\|A\|_1 = \max_{1 \leqslant j \leqslant n} \|a_j\|_1.$$

Theorem 1.6: 1-Norm einer Matrix

Die 1-Norm $\|A\|_1$ einer Matrix A mit $A \in \mathbb{C}^{m \times n}$ ist eine induzierte Matrixnorm.

Beweis: Um dies zu zeigen, betrachten wir die Einheitskugel $\{x \in \mathbb{C}^m | \sum_{j=1}^n |x_j| \leqslant 1\}$ unter der $\|\cdot\|_1$ Norm. Für jeden Vektor Ax im Bild dieser Menge gilt

$$\|Ax\|_1 = \|\sum_{j=1}^n x_j a_j\|_1 \overset{\text{Def. 1.9}}{\leqslant} \sum_{j=1}^n |x_j| \|a_j\|_1 \leqslant \max_{1 \leqslant j \leqslant n} \|a_j\|_1.$$

Es gilt also

$$\|Ax\|_1 \leqslant \max_{1 \leqslant j \leqslant n} \|a_j\|_1.$$

Wählt man $x = e_j$, wobei j so gewählt wird, dass $\|a_j\|_1$ maximal ist, d.h. $j = \arg\max \|a_j\|_1$, so gilt $\|A\|_1 = \max_{1 \leqslant j \leqslant n} \|a_j\|_1$. □

Definition 1.13: Unendlich-Norm einer Matrix (max. Zeilensumme)

Sei A eine $(m \times n)$-Matrix. Dann entspricht $\|A\|_\infty$ der maximalen Zeilensumme:

$$\|A\|_\infty = \max_{1 \leqslant i \leqslant m} \|a_i^*\|_1$$

Theorem 1.7: Unendlich-Norm einer Matrix

Die ∞-Norm $\|A\|_\infty$ einer Matrix A mit $A \in \mathbb{C}^{m \times n}$ ist eine induzierte Matrixnorm.

Die Berechnung der Matrix-p-Normen mit $p \neq 1, \infty$ ist schwieriger. Hier kann man die Hölderungleichung verwenden:

Lemma 1.8: Hölder-Ungleichung

Seien $p, q \in \mathbb{R}$ mit $1 < p, q < \infty$ so gewählt, dass $\frac{1}{p} + \frac{1}{q} = 1$. Dann gilt für alle Vektoren $x, y \in \mathbb{C}^n$:

$$|\sum_{i=1}^{n} \overline{x_i} y_i| = |x^* y| \leqslant \|x\|_p \|y\|_q.$$

Bemerkung: Youngsche-Ungleichung ([?])
Seien $p, q \in \mathbb{R}$ mit $1 < p, q < \infty$ so gewählt, dass $\frac{1}{p} + \frac{1}{q} = 1$. Dann gilt für alle $x, y \in \mathbb{R}_+$:

$$x^{1/p} \cdot y^{1/q} \leqslant \frac{x}{p} \cdot \frac{y}{q}.$$

Beweis: Der klassische Beweis greift auf die Youngsche-Ungleichung zurück. □

Spezialfall der Hölder-Ungleichung für $p = q = 2$:

Lemma 1.9: Cauchy-Schwarz-Ungleichung

Es gilt für alle Vektoren $x, y \in \mathbb{C}^n$:

$$|x^* y| \leqslant \|x\|_2 \|y\|_2.$$

Beispiel (1-Norm):

$A : \mathbb{R}^2 \to \mathbb{R}^2, \quad A = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix}$

$$\begin{aligned} \|A\|_1 &= \sup_{\|x\|=1} \|Ax\|_1 \\ &= 2 + 2 \\ &= 4. \end{aligned}$$

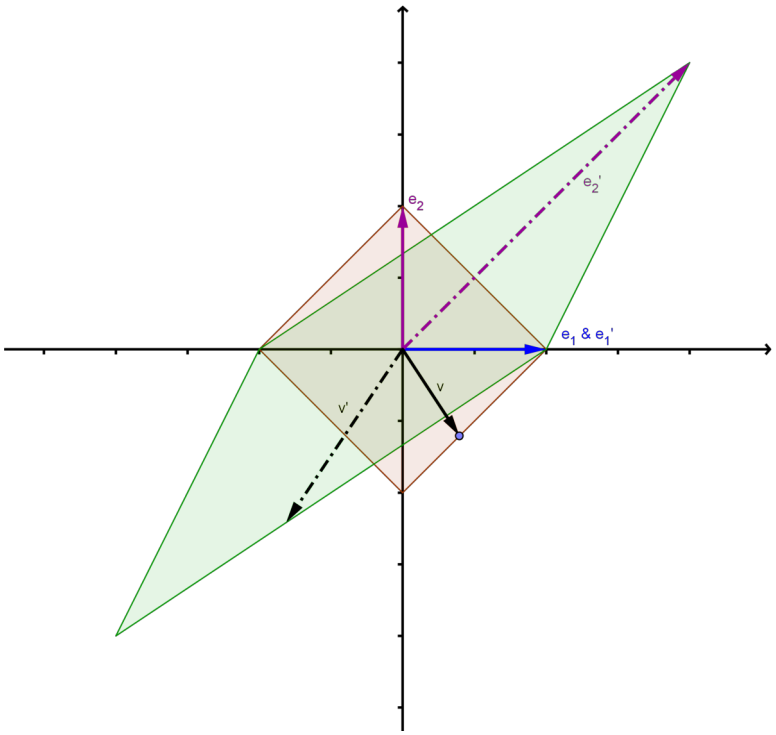


Abbildung 1.3: Anwendung der 1-Norm.

Enthalte A nur eine Zeile, d.h. $A = a^*$, wobei a ein Spaltenvektor ist:

$$\|Ax\|_2 = |a^* x| \leqslant \|a\|_2 \|x\|_2.$$

Für $x = a$ gilt:

$$\|Aa\|_2 = \|a\|_2^2 \Rightarrow \|A\|_2 = \sup_{x \neq 0} \left\{ \frac{\|Ax\|_2}{\|x\|_2} \right\} = \|a\|_2. \tag{1.5}$$

Begrenzung von $\|AB\|$ in induzierter Matrixnorm:

$$\|ABx\|_l \leqslant \|A\|_{(l,m)} \|Bx\|_m \leqslant \|A\|_{(l,m)} \|B\|_{(m,n)} \|x\|_n$$

$$\Rightarrow \|AB\|_{(l,n)} \leqslant \|A\|_{(l,m)} \|B\|_{(m,n)} \quad \text{(wird im Allgemeinen nicht angenommen).}$$

Definition 1.14: Allgemeine Matrixnormen

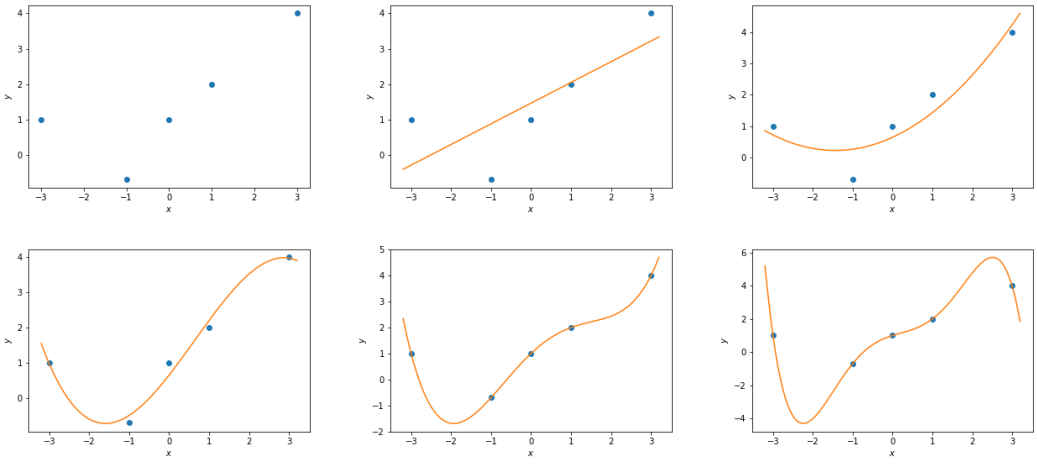
Eine Matrixnorm kann analog zu einer Vektornorm definiert werden. Sie muss die folgenden Bedingungen erfüllen:

1. $\|A\| \geqslant 0$ und $\|A\| = 0 \Leftrightarrow A = 0$
2. $\|\alpha A\| = |\alpha| \|A\|$
3. $\|A + B\| \leqslant \|A\| + \|B\|$

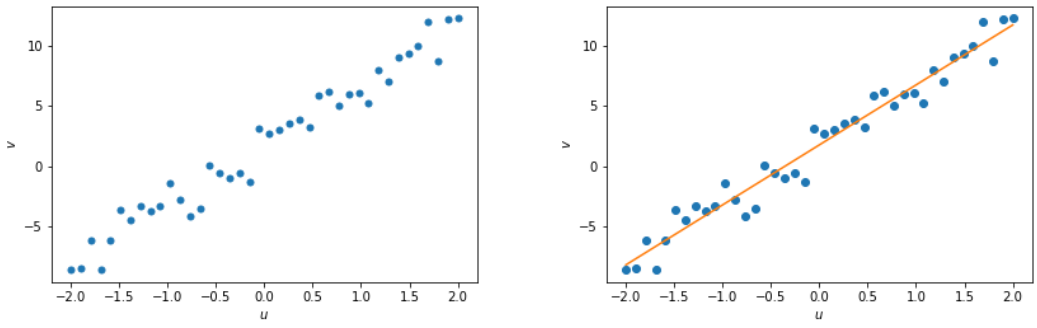
Kapitel 2

Lineare Ausgleichsrechnung

2.1 Lineare Ausgleichsrechnung



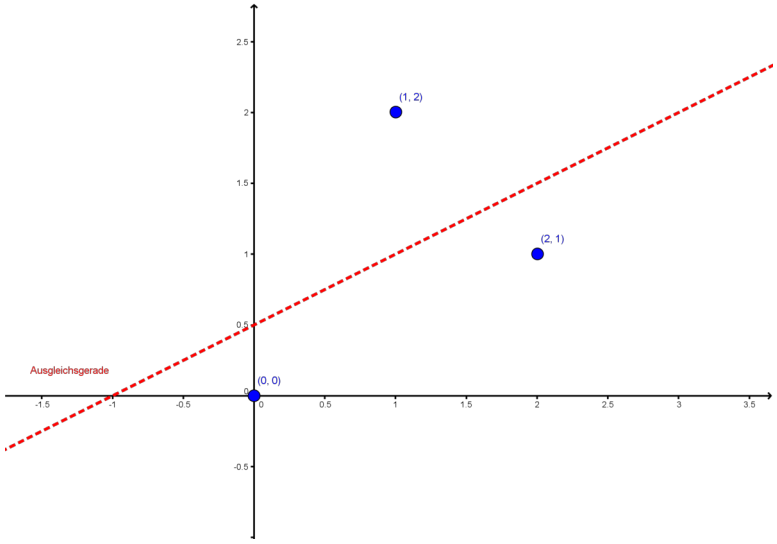
Die Daten im Bild oben links werden von links oben nach rechts unten mit Polynomen von wachsendem Grad approximiert. Ab Polynomgrad 4 erreichen wir Interpolation.



Daten ohne und mit Ausgleichsgerade

Gegeben: $(u_i, v_i) \in \mathbb{R}^2, \quad i = 1, .., n$, z.B. gemessene Punkte.

Gesucht: Gerade, die die Punkte (u_i, v_i) „am besten" approximiert.



Idee: Wähle Gerade $g(u) := x_0 + x_1 u$ die den quadratischen Approximationsfehler minimiert, d.h.

$\sum_{i=1}^{n} |v_i - g(u_i)|^2 = \sum_{i=1}^{n} (v_i - x_0 - x_1 u_i)^2 \longrightarrow \min \quad \star$

oder $\underbrace{\|b - Ax\|_2 \longrightarrow \min}_{\text{lineares Ausgleichsproblem}}$ mit $A = \begin{pmatrix} 1 & \cdots & 1 \\ u_1 & \cdots & u_n \end{pmatrix}^T$,

$b = (v_1, \ldots, v_n)^T$, $x = (x_0, x_1)^T$. So können z.B. Messfehler in b ausgeglichen werden (siehe obige Abbildung).

Lösungen des Problems heißen **Kleinste-Quadrate-Lösungen** (auf Englisch **least squares solutions**).

Anmerkung: Die Äquivalenz der beiden Formulierungen in Gleichung ($\star$) lässt sich wie folgt nachvollziehen:

$$\sum_{i=1}^{n} (v_i - (x_0 + x_1 u_i))^2 = \left\| \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} - \begin{pmatrix} 1 & u_1 \\ \vdots & \vdots \\ 1 & u_n \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \right\|_2^2 = \|b - Ax\|_2^2$$

Da weiterhin $\|\cdot\|^2 \to \|\cdot\|$ eine monotone Umformung ist, gilt, dass die quadratische Minimierung $\|b - Ax\|_2^2 \to \min$ exakt dieselben Lösungen wie $\|b - Ax\|_2 \to \min$ besitzt.

Definition 2.1: Lineares Ausgleichsproblem

Seien $A \in \mathbb{C}^{m\times n}$, $m \geqslant n$, $b \in \mathbb{C}^m$. Dann heißt das Minimierungsproblem:
Gesucht sei $x \in \mathbb{C}^n$ mit

$$x = \arg\min_{y\in\mathbb{R}^n} \|b - Ay\|_2 \tag{2.1}$$

lineares Ausgleichsproblem (LAGP).

Das Ausgleichproblem heißt **linear**, da die Parameter x nur linear in den Defekt $Ax - b$ eingehen.
Wird statt der Ausgleichsgerade ein Ausgleichpolynom vom Grad k

$$g(u) = \sum_{i=0}^{k} a_i u^i$$

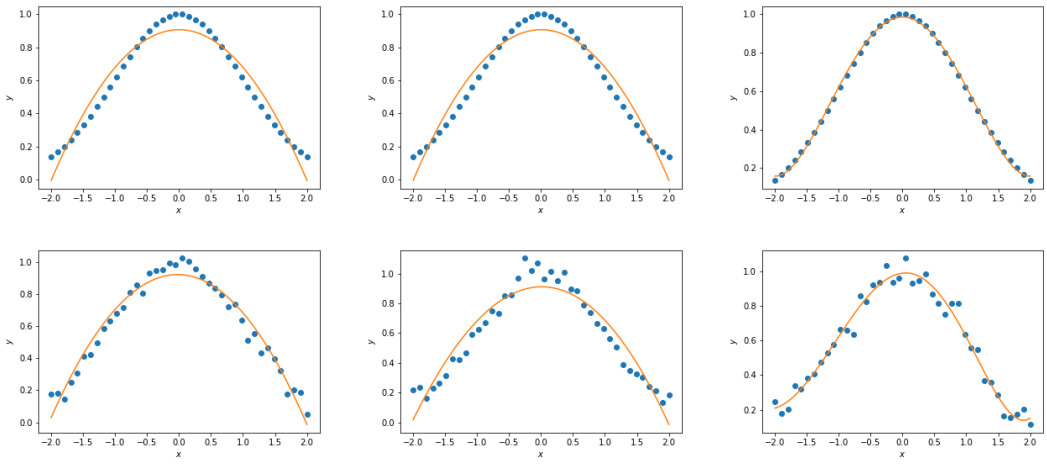
zu den Daten $(u_i, v_i) \in \mathbb{R}^2, i = 1, \cdots, n$ gewünscht, so ergibt sich folgendes lineares Ausgleichsproblem:

$$x = \arg\min_{y\in\mathbb{R}^n} \|b - Ay\|_2$$

mit

$$A = \begin{pmatrix} 1 & u_1 & u_1^2 & \cdots & u_1^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u_n & u_n^2 & \cdots & u_n^k \end{pmatrix}, x = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix}, b = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

Beispiel (Gaußsche Glockenkurve):



Approximation der Gaußschen Glockenkurve durch Polynome unterschiedlichen Grades: Links Grad 2, Mitte Grad 3 und Rechts Grad 4. Die Daten in der zweiten Zeile wurde zufällig verauscht.

Bemerkung: $A^*Ax = A^*b$ ist ein quadratisches lineares Gleichungssystem.

Satz 2.1: Lösung des linearen Ausgleichsproblems

Sei $A \in \mathbb{C}^{m\times n}, b \in \mathbb{C}^m$. Jede Lösung x des linearen Ausgleichsproblems

$$x = \arg\min_{y\in\mathbb{R}^n} \|b - Ay\|_2 \tag{2.2}$$

ist eine Lösung der „Gaußschen Normalengleichung"

$$A^*Ax = A^*b. \tag{2.3}$$

Umgekehrt ist jede Lösung x der „Gaußschen Normalengleichung" auch eine Lösung des linearen Ausgleichsproblems.

Lemma 2.2: Trivialer Nullraum

Die Matrix A^*A ist hermitesch und positiv semidefinit, d.h. $x^*A^*Ax \geqslant 0$.
A^*A ist genau dann positiv definit, wenn der Nullraum $Kern(A)$ trivial ist, d.h. $Kern(A) = 0$.
Ferner ist $Kern(A^*A) = Kern(A)$ und $Bild(A^*A) = Bild(A^*) = Kern(A)^\perp$.

Beweis: (Lemma 2.2)

- $(A^*A)^* = A^*(A^*)^* = A^*A$
- $x^*A^*Ax = \|Ax\|_2^2 \geqslant 0 \; \forall x \in \mathbb{C}^n$, d.h. A^*A ist positiv semidefinit.
- Sei nun $x \in Kern(A^*A)$.
  $A^*Ax = 0 \Rightarrow x^*A^*Ax = 0 \Rightarrow \|Ax\|_2^2 = 0 \Rightarrow Ax = 0$.
  D.h. $Kern(A^*A) \subseteq Kern(A)$ und wegen $Ax = 0 \Rightarrow A^*Ax = 0$ $Kern(A^*A) = Kern(A)$.
- Wenn $Kern(A) = \{0\}$ ist $Kern(A^*A) = 0$ und somit für alle $x \in \mathbb{C}^n\backslash\{0\}$ $x^*A^*Ax > 0$, d.h. A^*A ist positiv definit.
- Weiter gilt $dimBild(A^*A) = n - dimKern(A^*A) = n - dimKern(A) = Rang(A) = dimBild(A^*)$. Damit ist $Bild(A^*A) = Bild(A^*)$.
- **Bemerkung:** Zur Erinnerung:
  Dimensionssatz

Sei $f : V \longrightarrow W$ eine lineare Abbildung zwischen zwei Vektorräumen V und W.
Dann gilt:
$dimV = dimKern(f)+dimBild(f)$. Zu $Bild(A^*) = Kern(A)^\perp$: Sei $z \in Bild(A^*), x \in Kern(A)$ beliebig. Dann existiert ein $y \in \mathbb{C}^m$ mit $z = A^*y$ und es gilt

$$x^*z = x^*A^*y = (Ax)^*y = 0 \cdot y = 0,$$

d.h. $Kern(A)$ und $Bild(A^*)$ sind orthogonal und wegen
$dimKern(A) + dimBild(A^*) = n$ folgt $Bild(A^*) = Kern(A)^\perp$.

□

Beweis: (Satz 2.1) Setze für $x \in \mathbb{C}^n$

$$\phi(x) = \frac{1}{2}\|b - Ax\|_2^2 = \frac{1}{2}(b - Ax)^*(b - Ax).$$

Dann gilt für jedes $\hat{x}$ mit $A^*A\hat{x} = A^*b$

$$\phi(x) - \phi(\hat{x}) = \frac{1}{2}(x - \hat{x})^*A^*A(x - \hat{x}) \qquad \text{(quadratische Ergänzung).}$$

Da A^*A positiv semi-definit (2.2), ist folgt

$$\phi(x) - \phi(\hat{x}) \geqslant 0 \quad \Leftrightarrow \quad \phi(x) \geqslant \phi(\hat{x}) \quad \forall \hat{x} \text{ mit } A^*A\hat{x} = A^*b,$$

d.h. das Minimum von ϕ wird an den $\hat{x}$ mit $A^*A\hat{x} = A^*b$ angenommen. Solche $\hat{x}$ existieren, da A^*b zu $Bild(A^*) = Bild(A^*A)$ gehört.
Es bleibt zu zeigen, dass Gleichheit genau dann gilt, wenn eine Lösung x des Ausgleichsproblems auch die Normalengleichung erfüllt.
Um zu zeigen, dass Gleichheit genau dann gilt, wenn eine Lösung x des Ausgleichsproblems auch die Normalengleichung erfüllt, betrachten wir ein x, welches $\phi(x)$ minimiert.
Mit einem wie oben gewählten $\hat{x}$ gilt:

$$\begin{aligned}
& \phi(x) - \phi(\hat{x}) && = 0 \\
\Leftrightarrow \quad & \tfrac{1}{2}(x - \hat{x})^*A^*A(x - \hat{x}) && = 0 \\
\Leftrightarrow \quad & \tfrac{1}{2}\|A(x - \hat{x})\|_2^2 && = 0 \\
\Leftrightarrow \quad & (x - \hat{x}) \in Kern(A).
\end{aligned}$$

Unter Ausnutzung von $(x - \hat{x}) \in Kern(A)$ setzen wir in die Normalengleichung $\hat{x}$ ein:

$$\begin{aligned}
A^*b &= A^*A\hat{x} \\
&= A^*(A\hat{x} + \underbrace{A(x - \hat{x})}_{0}) \\
&= A^*A(\hat{x} + (x - \hat{x})) \\
&= A^*Ax.
\end{aligned}$$

Somit erfüllt jede Lösung x des linearen Ausgleichsproblems die Normalengleichung. Falls A^*A singulär ist, so hat die Gaußsche Normalengleichung mehrere Lösungen und ϕ wird in allen Lösungen minimal. □

Bemerkung 2.1:

Das *Residuum* (Rest) $r = b - A\hat{x}$ gehört zu $Kern(A^*)$:

$$A^*r = A^*b - A^*A\hat{x} = 0.$$

Da nach dem Lemma 2.2 $Bild(A^*) = Kern(A)^\perp$ gilt, ist auch $Bild(A) = Kern(A^*)^\perp$ beziehungsweise $Kern(A^*) = Bild(A)^\perp$, d.h. r steht orthogonal auf Bild(A).

Bemerkung 2.2: Minimierung mit Methoden der Differentialrechnung

Im reellen Fall kann das Minimum in Gleichung 2.1 alternativ mit Methoden der Differentialrechnung bestimmt werden.

Sei dazu $d \in \mathbb{R}^n$, so berechnen wir die Richtungsableitung nach d an der Stelle x gemäß

$$\frac{\partial}{\partial d}\Phi(x) = \lim_{t\to 0}\frac{\Phi(x+td)-\Phi(x)}{t}.$$

Wegen

$$\begin{aligned}\Phi(x+td)-\Phi(x) &= \frac{1}{2}(b-Ax-tAd)^*(b-Ax-tAd) - \frac{1}{2}(b-Ax)^*(b-Ax)\\ &= t(Ad)^*(Ax-b) + \frac{1}{2}t^2(Ad)^*Ad.\end{aligned}$$

folgt

$$\frac{\partial}{\partial d}\Phi(x) = (Ad)^*(Ax-b)$$

Nullsetzen liefert eine notwendige Bedingung für ein Minimum.

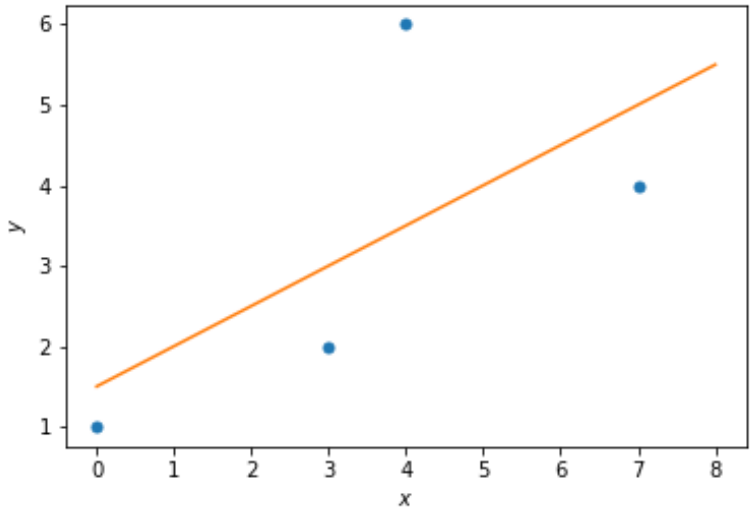
Beispiel (Normalengleichung):

Wir versuchen die zugehörige Normalengleichung zu folgendem Problem zu finden:

$$A^*Ax = A^*b$$

mit $A = \begin{pmatrix} 1 & 0 \\ 1 & 3 \\ 1 & 4 \\ 1 & 7 \end{pmatrix}$ und $b = \begin{pmatrix} 1 \\ 2 \\ 6 \\ 4 \end{pmatrix}$. Dazu berechnen wir:

$$
\begin{aligned}
A^*A &= \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 3 & 4 & 7 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 3 \\ 1 & 4 \\ 1 & 7 \end{pmatrix} = \begin{pmatrix} 4 & 14 \\ 14 & 74 \end{pmatrix} \\
A^*b &= \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 3 & 4 & 7 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 6 \\ 4 \end{pmatrix} = \begin{pmatrix} 13 \\ 58 \end{pmatrix} \\
(A^*A)^{-1} &= \frac{1}{4 \cdot 74 - 14 \cdot 14} \cdot \begin{pmatrix} 74 & -14 \\ -14 & 4 \end{pmatrix} \\
\Rightarrow \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} &= \begin{pmatrix} 3/2 \\ 1/2 \end{pmatrix}.
\end{aligned}
$$



`Abbildung` 2.4: Daten als Punkte und Ausgleichsgerade

Bemerkung 2.3: Tikhonov Regularisierung

Phillips and Tikhonov minimieren nicht nur die kleinsten Fehlerquadrate, sondern schlagen vor, gleichzeitig die Norm des Lösungsvektors klein zu machen und betrachten für $\lambda > 0, A \in \mathbb{C}^{m\times n}, \mathbb{I} = \mathrm{Id}_n \in \mathbb{R}^{n\times n}$.

$$\hat{x} = \underset{x\in\mathbb{R}^n}{\arg\min} \left(\|Ax - b\|_2^2 + \lambda\|x\|_2^2\right) \tag{2.4}$$

Diese Problem kann auch als gewöhnliches Kleinste Quadrate Problem geschrieben werden:

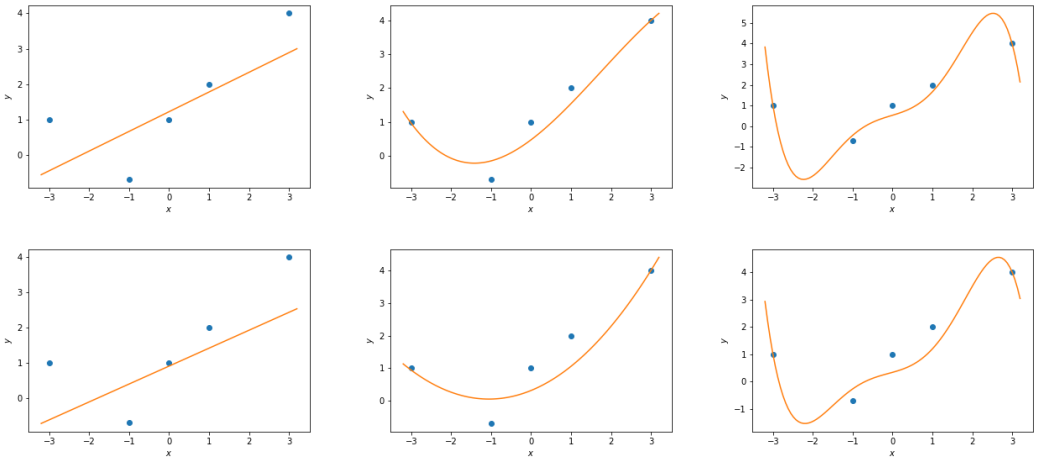
$$\hat{x} = \underset{x\in\mathbb{R}^n}{\arg\min} \left(\left\| \begin{pmatrix} A \\ \sqrt{\lambda}\mathbb{I} \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2^2 \right)$$

Bemerkung 2.4: Tikhonov Regularisierung

Für die zugehörige Normalengleichung ergibt sich

$$\begin{aligned} \begin{pmatrix} A^* & \sqrt{\lambda}Id \end{pmatrix} \begin{pmatrix} A \\ \sqrt{\lambda}Id \end{pmatrix} x &= \begin{pmatrix} A^* & \sqrt{\lambda}Id \end{pmatrix} \begin{pmatrix} b \\ 0 \end{pmatrix} \\ \Leftrightarrow (A^*A + \lambda Id)\, x &= A^*b \end{aligned}$$

Beispiel (Thikhonov Regularisierung):



In den Bildern der Spalten Approximation der Daten jeweils mit Polynomgrad 1,3 und 5. In der oberen Reihe mit Regularisierung $\lambda = 1$. Untere Zeile mit $\lambda = 5$.

Kapitel 3

Singulärwertzerlegung (SVD)

3.1 Einführung in die SVD

Definition 3.1: Selbstadjungierter Endomorphismus

Sie $V, < \cdot, \cdot >$ ein euklidischer Vektorraum. Ein Endomorphismus $f : V \to V$ heißt selbstadjungiert (bzgl. des gegebenen Skalarprodukt $\langle \cdot, \cdot \rangle$), falls für beliebige $v, w \in V$ gilt:

$$\langle f(v), w \rangle = \langle v, f(w) \rangle$$

Bemerkung 3.1: Matrixdarstellung selbstadjungierter Endomorphismen

Sei $B = (b_1, \cdots, b_n)$ eine Orthonormalbasis von V und $A = DM_B(f)$, so gilt

$$f \text{ selbstadjungiert } \Leftrightarrow A^t = A$$

Satz 3.1: Spektralsatz

Sei $f : V \to V$ ein selbstadjungierter Endomorphismus eines $n-$ dimensionalen euklidischen Vektorraums V. Dann besitzt V eine Orthonormalbasis, die aus Eigenvektoren von f besteht.

Korollar 3.2: Diagonalisierung hermitscher Matrizen

Sei $A \in \mathbb{C}^{n\times n}$ hermitsch, d.h. $A^* = A$. Dann gibt es eine unitäre Matrix T, d.h. $T^* \cdot T = Id_n$, so dass

$$T^*AT = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix}$$

Definition 3.2: Unitäres Skalarprodukt und unitärer Vektorraum

Ist V ein komplexer Vektorraum, so heißt eine Abb. von $V \times V \to \mathbb{C}$ mit

$$(v, w) \mapsto v \cdot w = \langle v, w \rangle$$

ein unitäres Skalarprodukt, wenn für alle $v, v', w \in V$ und $\lambda \in \mathbb{C}$ die folgenden Eigenschaften erfüllt sind:

1. $(u + v') \cdot w = v \cdot w + v' \cdot w$ und $(\lambda v) \cdot w = \lambda(v \cdot w)$ (Linearität im ersten Argument),
2. $v \cdot w = \overline{v \cdot w}$ (hermitesch)
3. $v \cdot v \geqslant 0$ und $v \cdot v = 0 \Leftrightarrow v = 0$ (positive Definitheit).

Ist $\langle \cdot, \cdot \rangle$ ein unitäres Skalarprodukt in V, so nennt man V einen unitären Vektorraum.

Sei p der Rang von A (= Rang von A^*A) und seien $\lambda_1, ..., \lambda_n$ die absteigend sortierten Eigenwerte von A^*A, d.h.

$$\lambda_1 \geqslant \lambda_2 \geqslant ... \geqslant \lambda_p > \lambda_{p+1} = ... = \lambda_n = 0,$$

sowie $v_1, ..., v_n \in \mathbb{C}^n$ eine Basis orthonormaler Eigenvektoren ($A^*Av_i = \lambda_i v_i \; \forall i = 1, .., n$, $v_i^* v_j = \delta_{ij}$, $i, j = 1, ..., n$). Eine solche *Spektralzerlegung* existiert, da A^*A hermitesch und positiv semidefinit ist.
Nun definieren wir Vektoren $u_i \in \mathbb{C}^m$ durch

$$u_i = \frac{1}{\sqrt{\lambda_i}} Av_i, \qquad i = 1, ..., p.$$

Für $1 \leqslant i, j \leqslant p$ gilt dann

$$u_i^* u_j = \frac{1}{\sqrt{\lambda_i}} \frac{1}{\sqrt{\lambda_j}} (Av_i)^* (Av_j) = \frac{1}{\sqrt{\lambda_i \lambda_j}} v_i^* (A^* A v_j) = \frac{\lambda_j}{\sqrt{\lambda_i \lambda_j}} v_i^* v_j = \delta_{ij}.$$

Die Vektoren $u_i, i = 1, ..., p$, bilden also eine Orthonormalbasis des gesamten Bildraumes $Bild(A)$, denn

$$dimBild(A) = n - dimKern(A) = n - dimKern(A^* A) = n - (n - p) = p.$$

Diese Basis kann durch weitere $m - p$ Vektoren $u_{p+1}, ..., u_m$ zu einer orthonormalen Basis des $\mathbb{C}^m$ erweitert werden (vgl. Bemerkung im Kapitel 1: Basisergänzungssatz).
Des Weiteren gilt

$$A^* u_i = \begin{cases} \frac{1}{\lambda_i} A^* A v_i = \frac{\lambda_i}{\sqrt{\lambda_i}} v_i = \sqrt{\lambda_i} v_i & \forall i = 1, .., p, \\ 0 & \forall i \geqslant p + 1, ..., m. \end{cases}$$

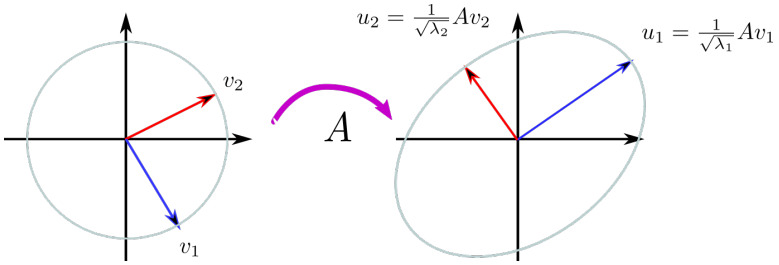


Abbildung 3.5: Grapische Darstellung der SVD

Definition 3.3: Singulärwertzerlegung

Sei $A \in \mathbb{C}^{m \times n}$ eine Matrix mit Rang p. Ein System

$$\{\sigma_i, u_j, v_k \ : \ i = 1, ..., p, j = 1, ..., m, k = 1, ..., n\} \tag{3.1}$$

mit $\sigma_1 \geqslant \sigma_2 \geqslant ... \geqslant \sigma_p > 0$ und Orthonormalbasen $\{u_j\}_{j=1}^m, \{v_k\}_{k=1}^n$ des $\mathbb{C}^m$ bzw. $\mathbb{C}^n$, wobei

$$\begin{aligned} Av_i &= \sigma_i u_i, \quad A^* u_i = \sigma_i v_i, \quad \forall i = 1, ..., p, \\ Av_k &= 0, \qquad A^* u_j = 0, \qquad j, k > p, \end{aligned}$$

heißt *Singulärwertzerlegung* von A. Die σ_i heißen Singulärwerte von A. Ihre Quadrate σ_i^2 sind die von 0 verschiedenen Eigenwerte von $A^* A$.

Satz 3.3: Existenz der Singulärwertzerlegung

Jede Matrix $A \in \mathbb{C}^{m\times n}$ mit Rang p besitzt eine Singulärwertzerlegung.

Bemerkung: **SVD** steht für *engl.* **S**ingular **V**alue **D**ecomposition

3.2 Die Singulärwertzerlegung in Matrixnotation

Seien $U \in \mathbb{C}^{m\times m}$ und $V \in \mathbb{C}^{n\times n}$ mit

$$U = \left(\begin{array}{c|c|c|c} & & & \\ u_1 & u_2 & \dots & u_m \\ & & & \end{array}\right), \quad V = \left(\begin{array}{c|c|c|c} & & & \\ v_1 & v_2 & \dots & v_n \\ & & & \end{array}\right)$$

$$\text{und } \Sigma = \begin{pmatrix} \sigma_1 & & 0 & 0 \\ & \ddots & & 0 \\ 0 & & \sigma_p & \\ 0 & 0 & & 0 \end{pmatrix}.$$

Dann gilt:

$$AV = U\Sigma \Leftrightarrow A = U\Sigma V^* \Leftrightarrow A^* = V\Sigma^* U^* \Leftrightarrow A = \Sigma_{i=1}^{p} \sigma_i u_i v_i^*. \tag{3.2}$$

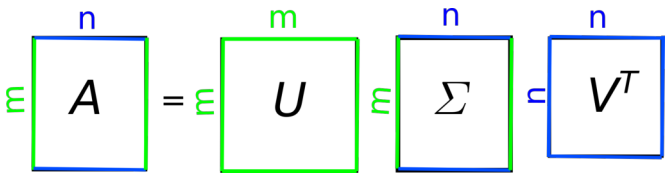


Abbildung 3.6: Schematische Darstellung der SVD

$m > n$: $m = n$: $m < n$:

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & \vdots & \ddots & \vdots \\ 0 & 0 & \sigma_n & 0 & \dots & 0 \end{pmatrix}$$

Zur geometrischen Interpretation beschränken wir uns auf reelle Matrizen.

Bemerkung: *Linkseigenvektoren* werden definiert als

$$x^T \cdot A = \lambda \cdot x^T.$$

Wegen

$$(x^T A)^T = A^T x$$

sind die Linkseigenvektoren die (Rechts-) Eigenvektoren der Matrix A^T. [onlytext-width,c]

Sei S die Einheitssphäre im $\mathbb{R}^n, A \in \mathbb{R}^{m \times n}, m \geqslant n$. A habe vollen Rang. Seien dann $\sigma_1, ..., \sigma_n$ die Längen der Halbachsen von AS (siehe Abbildung) und $\sigma_1 \geqslant \sigma_2 \geqslant ... \geqslant \sigma_n > 0$ die Singulärwerte von A.

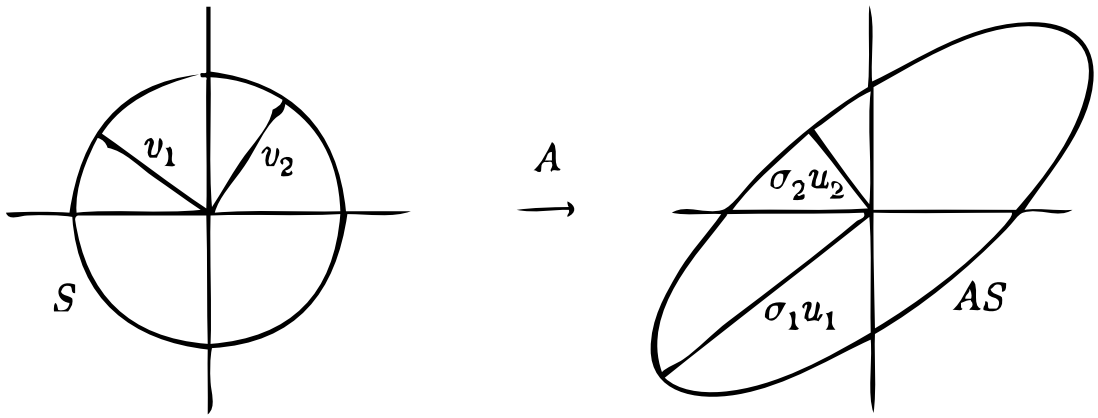


Abbildung 3.7: Ellipse mit gestreckten bzw. gestauchten Hauptachsen

Die normierten Einheitsvektoren entlang der Hauptachsen $\{u_1, ..., u_n\}$ sind die *Linkseigenvektoren*. Die *Rechtseigenvektoren* von A sind die Einheitsvektoren $\{v_1, ..., v_n\} \in S$, die durch $Av_j = \sigma_j u_j$ $(\forall j \in \{1, ..., n\})$ auf die Hauptachse abgebildet werden.

In Matrixschreibweise:

$$\left(\quad A \quad \right) \cdot \left(\begin{array}{c|c|c|c} & & & \\ v_1 & v_2 & \dots & v_n \\ & & & \end{array} \right) = \left(\begin{array}{c|c|c|c} & & & \\ u_1 & u_2 & \dots & u_n \\ & & & \end{array} \right) \cdot \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{pmatrix}$$

Korollar 3.4: Bild der Einheitkugel

Das Bild der Einheitskugel bezüglich der 2-Norm $\|\cdot\|_2$ unter einer beliebigen $(m\times n)$-Matrix ist ein verallgemeinertes Ellipsoid.

Wenn die Vektoren $v = \sum_{i=1}^{n} a_i v_i$ die Einheitssphäre des $\mathbb{R}^n$ durchlaufen, d.h. $\sum_{i=1}^{n} a_i^2 = 1$, dann durchlaufen ihre Bilder $Av = \sum_{i=1}^{n} \sigma_i a_i u_i$ ein verallgemeinertes Ellipsoid, denn es gilt

$$\sum_{i=1}^{n} \frac{1}{\sigma_i^2}\sigma_i^2 a_i^2 = \sum_{i=1}^{n} a_i^2 = 1.$$

Dies ist die Gleichung eines verallgemeinerten Ellipsoids mit Scheitelpunkten $\pm\sigma_i e_i$, wobei e_i der i-te Basisvektor ist.

Beispiel (Rezept zur Berechung einer SVD):
Die Singulärwertzerlegung $A = U\Sigma V^T$ einer Matrix $A \in \mathbb{R}^{m\times n}$ kann man folgendermaßen berechnen:

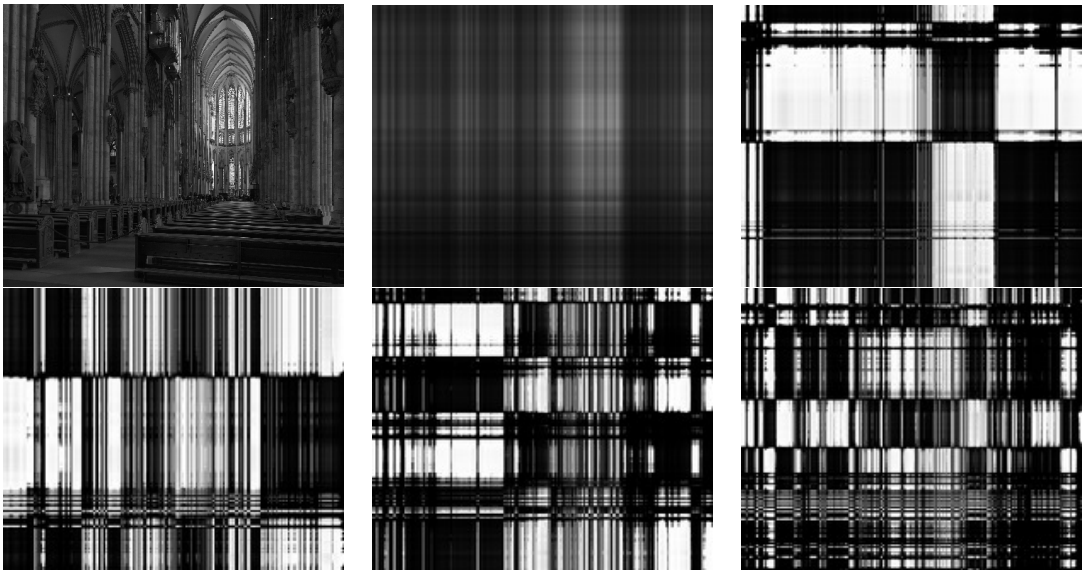
1. Berechne $B = A^T A$.
2. Berechne die Eigenwerte von B: $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_k > \lambda_{k+1} = \cdots \lambda_n = 0$.
3. Die normierten Eigenvektoren $\{v_1, ..., v_n\}$ sind eine Orthonormal-Basis des $\mathbb{R}^n$, wobei v_i ein Eigenvektor zum Eigenwert λ_i ist. Dann ist $V := [v_1, ..., v_n]$. Es gilt: $V^T = V^{-1}$.
4. Bilde die Diagonalmatrix Σ aus den Singulärwerten von A: $\sigma_i = \sqrt{\lambda_i}$.
5. Finde die Matrix U durch Berechnen der Vektoren $u_i = \frac{1}{\sqrt{\lambda_i}} A v_i$ für $i \leqslant k$. Ergänze diese Vektoren zu einer ON-Basis des $\mathbb{R}^m$ (vgl. Basisergänzungsatz in Kapitel 1).

Beispiel (Beispiel Bildkompression):
Gegeben sei ein Grauwertbild, dessen Speicherbedarf reduziert werden soll.
Lösung:

1. Fasse das Grauwertbild mit $n \times m$ Pixeln als Matrix $A \in \mathbb{R}^{n\times m}$ auf.
2. Berechne die SVD von $A = U\Sigma V^*$.
3. Berechne eine **Rank-k Approximation** von A:

$$\hat{A}(k) := \sum_{i=1}^{k} \sigma_i u_i v_i^t$$

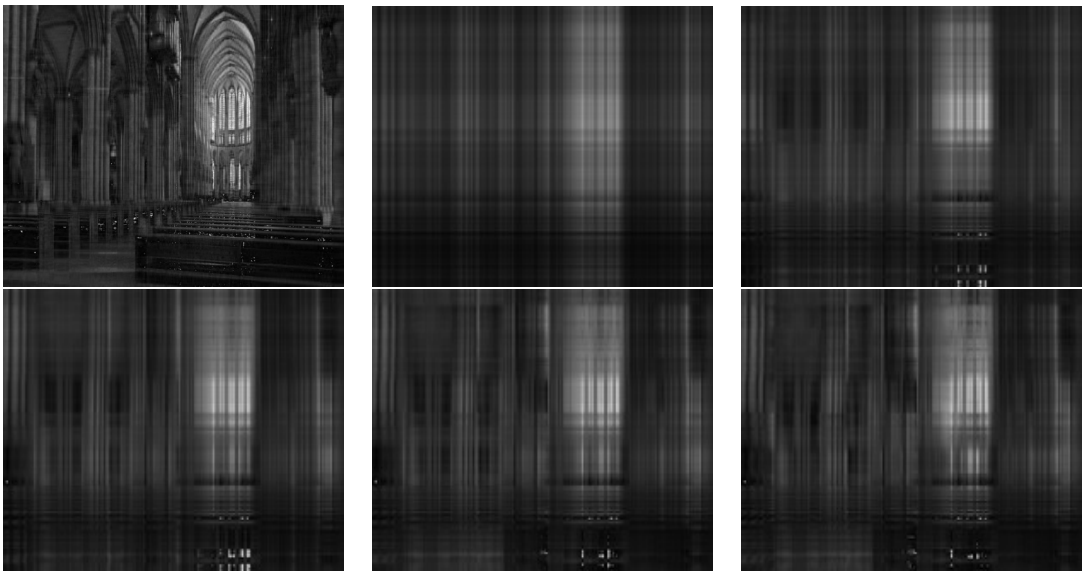
4. Speichere Matrix als Bild.



Beispiel (Beispiel Bildkompression):

Die ersten fünf Hauptkomponenten des links oben gezeigten Bilds aus dem Kölner Dom.

Beispiel (Beispiel Bildkompression):



Oben links eine Rang 50 Approximation des Bilds aus dem Kölner Dom. Folgend Rang-1 bis Rang-5 Approximationen.

Theorem (Spektralnorm) Sei $A \in \mathbb{C}^{m\times n}$ und σ_1 der größte Singulärwert von A. Dann gilt

$$\|A\|_2 = \sup \frac{\|Ax\|_2}{\|x\|_{(2)}} = \sigma_1.$$

Theorem (Eckart-Young Theorem) Sei $A \in \mathbb{C}^{m\times n}$ ein Matrix vom Rang r und sei $B \in \mathbb{C}^{m\times n}$ eine Matrix vom Rang k. Für jedes $k \leqslant r$ und jede Rang-k Approximation $\hat{A}(k) := \sum_{i=1}^{k} \sigma_i u_i v_i^t$ von A gilt

$$\begin{aligned} \hat{A}(k) &= \underset{\{B\in\mathbb{C}^{m\times n} | rang(B)=k\}}{\arg\min} \|A - B\|_2, \\ \|A - \hat{A}(k)\|_2 &= \sigma_{k+1}. \end{aligned}$$

Für die Beweise verweisen wir auf die Übungen bzw. die Literatur.

3.3 Unterschiede zwischen SVD und Eigenwertzerlegung

Bemerkung: **EVD** steht für *engl.* **E**igen**V**alue **D**ecomposition

Definition 3.4: Eigenwertzerlegung

Eine Zerlegung der Form

$$A = V\Lambda V^{-1} \tag{3.3}$$

heißt *Eigenwertzerlegung* (oder *Spektralzerlegung*) von A.

Satz 3.5: Diagonalisierbarkeit

Eine Matrix $A \in \mathbb{C}^{n\times n}$ ist genau dann diagonalisierbar, wenn sie n linear unabhängige Eigenvektoren besitzt.

Bemerkung 3.2: Eigenwertzerlegung in Matrixschreibweise

Besitzt $A \in \mathbb{C}^{n\times n}$ n linear unabhängige Eigenvektoren v_i, so gilt für alle $i = 1, \cdots, n$

$$Av_i = \lambda_i v_i$$

.

In Gleichung (3.3) erhalten wir damit

$$A = \left(v_1 \;\middle|\; v_2 \;\middle|\; \ldots \;\middle|\; v_n \right) \underbrace{\begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}}_{\Lambda=\mathrm{diag}(\lambda_1,\ldots,\lambda_n)} \left(v_1 \;\middle|\; v_2 \;\middle|\; \ldots \;\middle|\; v_n \right)^{-1}$$

Zusammenfassung:

Singulärwertzerlegung und Eigenwertzerlegung im Vergleich

SVD	vs.	**EVD**
$A = U\Sigma V^*$		$A = V\Lambda V^{-1}$

- SVD verwendet zwei Basen (U, V)
  EVD verwendet eine Basis (V)
- SVD verwendet orthonormale Basen
  Basen der EVD sind nicht zwingend orthonormal
- SVD existiert für jede Matrix A
  EVD existiert nicht für jede Matrix A
- SVD ist interessant zum Studium der Matrix A selbst
  EVD ist interessant zum Studium von Matrixpotenzen A^k, e^{tA}:
  $Av_i = \lambda_i v_i \quad \Leftrightarrow \quad A^k v_i = \lambda_i^k v_i$

3.4 Singulärwertzerlegung und Lineare Ausgleichsrechnung

Bemerkung: Für invertierbare Matrizen $A \in \mathbb{C}^{n\times n}$ ist $A^+ = A^{-1}$. Daher ist die Pseudoinverse eine Verallgemeinerung der klassischen Inversen für singuläre oder nicht quadratische Matrizen.

Definition 3.5: Pseudoinverse

Sei $A = U\Sigma V^*$ die Singulärwertzerlegung von $A \in \mathbb{C}^{m\times n}$. Dann heißt

$$A^+ = V\Sigma^+ U^* \in \mathbb{C}^{n\times m}$$

mit

$$\Sigma^+ = \begin{pmatrix} \sigma_1^{-1} & & & 0 \\ & \ddots & & 0 \\ & & \sigma_p^{-1} & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix} \in \mathbb{C}^{n\times m}$$

Pseudoinverse oder *Moore-Penrose-Inverse* von A. Man kann nachrechnen:

$$A^+ = \sum_{i=1}^{p} \sigma_i^{-1} v_i u_i^*,$$

$$Kern(A^+) = Kern(A^*) = Bild(A)^\perp, \qquad Bild(A^+) = Bild(A^*) = Kern(A)^\perp.$$

Satz 3.6: Lösung des Ausgleichsproblems

Der Vektor A^+b ist die eindeutig bestimmte Lösung des linearen Ausgleichsproblems

$$\|b - Ax\|_2 \longrightarrow \min$$

(vgl. Gleichung (★) am Anfang von Kapitel 2) mit minimaler euklidischer Norm.

Wir beweisen die Aussage in mehreren Schritten:

1. $\forall b \in \mathbb{C}^m : \; AA^+b - b \in Kern(A^+)$
   $A^+AA^+ = V\Sigma^+U^*U\Sigma V^*V\Sigma^+U^* = V\Sigma^+U^* = A^+$
   Nun gilt: $A^+(AA^+b - b) = A^+AA^+b - A^+b = A^+b - A^+b = 0$.

2. $Kern(A^+) = Bild(A)^\perp = Kern(A^*)$
   Also erfüllt A^+b die Normalengleichung $A^*A(A^+b) = A^*b$:
   $A^*A(A^+b) - A^*b = A^*(AA^+b - b) = 0$, da $AA^+b - b \in Kern(A^+) = Kern(A^*)$.

3. Ist z eine weitere Lösung der Normalengleichung, so gilt:
   $A^*Az = A^*A(A^+b) = A^*b$ und somit $w = A^+b - z \in Kern(A^*A) = Kern(A)$
   Andererseits ist $A^+b \in Bild(A^+) = Kern(A)^\perp$. Daher ist $z = \underbrace{A^+b}_{Kern(A)^\perp} - \underbrace{w}_{Kern(A)}$
   eine orthogonale Zerlegung und nach Pythagoras gilt

   $$\|z\|_2^2 = \|A^+b\|_2^2 + \|w\|_2^2 \geqslant \|A^+b\|_2^2$$

   mit Gleichheit genau dann, wenn $\|w\|_2^2 = 0$, d.h. für $z = A^+b$.

□

Korollar 3.7: Trivialer Nullraum

Ist $Kern(A) = \{0\}$, so gilt $A^+ = (A^*A)^{-1}A^*$.

Beweis: Die Lösung des linearen Ausgleichsproblems $\|b - Ax\| \to \min$ ist laut Normalengleichung gegeben durch $(A^*A)^{-1}A^*b$, aber auch durch A^+b. Da dies für beliebiges b gilt folgt die Behauptung. □

Anschaulich ist die Pseudoinverse A^+ also eine Matrix, die die Lösungen des linearen Ausgleichsproblems zur Matrix A für beliebiges b beschreibt. Entsprechend stellt AA^+ die orthogonale Projektion auf das Bild von A dar.

Kapitel 4

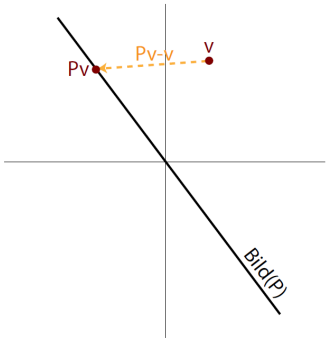
QR-Zerlegung

4.1 Projektoren und Projektionsmatrizen

Definition 4.1: Projektionsmatrix

Eine quadratische Matrix $P \in \mathbb{C}^{n\times n}$ heißt *Projektionsmatrix*, falls

$$P^2 = P.$$



`Abbildung` 4.8: Beispiel einer Projektion.

Bemerkung 4.1: Bild und Kern

Sei $v \in \mathbb{C}^n, P \in \mathbb{C}^{n\times n}$.

1. Falls $\nu \in Bild(P)$, so gilt $P\nu = \nu$.
2. Sei $\nu \in \mathbb{C}^n$. Dann liegt $P\nu - \nu \in Kern(P)$.
   Denn:
   $P(P\nu - \nu) = P^2\nu - P\nu = P\nu - P\nu = 0$.

Bemerkung: Eine (quadratische) Matrix A heißt *idempotent*, wenn gilt: $A^2 = A$.

Satz 4.1: Komplementäre Projektionsmatrix

Sei $P \in \mathbb{C}^{n\times n}$ eine Projektionsmatrix. Dann ist auch $I - P$ eine Projektionsmatrix. $I - P$ heißt *komplementäre Projektionsmatrix*.

Beweis: Da P eine Projektionsmatrix ist, gilt zunächst $P^2 = P$. Daher gilt für $I - P$:

$$(I-P)^2 = I - 2P + \underbrace{P^2}_{=P} = I - P.$$

□

Wir können unter anderem folgende Beobachtungen machen:

1. $I - P$ projiziert auf $Kern(P)$, d.h. $Bild(I - P) = Kern(P)$.
2. Es gilt: $Kern(I - P) = Bild(P)$.
3. Es ist $Kern(I - P) \cap Kern(P) = \{0\}$ (und damit auch $Bild(I - P) \cap Bild(P) = \{0\}$).

(Beweis der Beobachtungen zu komplementären Matrizen:)
Beweis:

1. Z.z.: $Bild(I - P) = Kern(P)$:
   (i) Es gilt zum einen die Mengeninklusion $Bild(I - P) \supseteq Kern(P)$, denn:
   $\forall v \in Kern(P): \quad v = Iv - \underbrace{Pv}_{=0} = (I - P)v \quad \in Bild(I - P)$.
   (ii) Zum anderen gilt aber auch die Inklusion $Bild(I - P) \subseteq Kern(P)$, denn:
   $\forall v \in Bild(I - P): \quad (I - P)v = v - Pv \quad \in Kern(P)$.
2. Z.z.: $Kern(I - P) = Bild(P)$:
   Da $(I - P)$ eine Projektionsmatrix ist (Satz 4.1), folgt:
   $Kern(I - P) = Bild(I - (I - P)) = Bild(P)$.
3. Z.z. $Kern(I - P) \cap Kern(P) = \{0\}$:
   Es gilt $\forall v \in Kern(I - P) \cap Kern(P)$: $0 = (I - P)v = v - \underbrace{Pv}_{=0} = v$.

□

Bemerkung: Innere direkte Summe:
Sei $(U_i)_{i \in I}$ eine Familie von Untervektorräumen eines Vektrorraums V. $V = \sum_{i \in I} U_i$ heißt *innere direkte Summe*, wenn $\forall j \in I$ gilt:

$$U_j \cap \sum_{i \in I \setminus \{j\}} U_i = \{0\}.$$

Im Spezialfall $U_1 \oplus U_2 = V$ ennt man U_1 und U_2 *zueinander komplementär*:

$$\begin{aligned} & U_1 \oplus U_2 = V \\ \Leftrightarrow \quad & U_1 + U_2 = V \\ \wedge \quad & U_1 \cap U_2 = \{0\}. \end{aligned}$$

Wir haben also gezeigt, dass eine Projektion den $\mathbb{C}^n$ in zwei Teilräume teilt.
Sind umgekehrt S_1, S_2 zwei Teilräume des $\mathbb{C}^n$ mit $S_1 \cap S_2 = \{0\}$ und $S_1 \oplus S_2 = \mathbb{C}^n$.
Dann gibt es eine Projektion P, sodass

$$Bild(P) = S_1 \qquad \text{und} \qquad Kern(P) = S_2.$$

P heißt dann Projektion auf S_1 entlang S_2.

4.1.1 Orthogonale Projektionen

Definition 4.2: Orthogonale Projektion

Eine *orthogonale Projektion* ist eine Projektion, die entlang eines Unterraums S_2 auf S_1 projiziert, wobei S_1 und S_2 orthogonal sind.

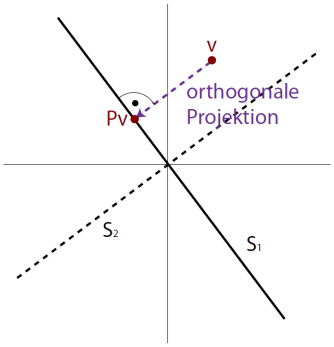


Abbildung 4.9: Beispiel einer orthogonalen Projektion.

Theorem 4.2: Orthogonale Projektion

Eine Projektion ist genau dann orthogonal, wenn gilt: $P = P^*$.

Bemerkung: Achtung:
Ein Projektor wird zwar durch eine Matrix dargestellt. Ein *orthogonaler Projektor* ist aber nicht zwangsläufig eine *orthogonale Matrix*.

Beweis: Sei $P = P^*$, dann gilt für beliebige $x, y \in \mathbb{C}^n$:

$$\underbrace{(Px)^*}_{\in Bild(P)=S_1} \overbrace{(I-P)y}^{\in Bild(I-P)=S_2} = x^* P^* (I - P^2) y = 0.$$

D.h. Px und $(I-P)y$ sind orthogonal zueinander.
Sei umgekehrt P eine orthogonale Projektion auf S_1 entlang S_2, wobei S_1 und S_2 orthogonal sind und $dim S_1 = k$.
Dann wählen wir eine Orthonormalbasis $\{q_1, ..., q_n\}$ des $\mathbb{C}^n$, sodass $\{q_1, ..., q_k\}$ eine Basis von S_1 und $\{q_{k+1}, ..., q_n\}$ eine Basis von S_2 ist (vgl. BES 11).
Dann gilt

$$\forall j \leqslant k: \quad Pq_j = q_j \quad \text{und}$$
$$\forall j > k: \quad Pq_j = 0.$$

Sei nun Q eine unitäre Matrix mit den Spalten q_j. Dann gilt

$$Q^*PQ = \begin{pmatrix} \mathbf{1} & 0 & \dots & & & \dots & 0 \\ 0 & \mathbf{1} & & & & & \vdots \\ \vdots & & \ddots & & & & \\ & & & \mathbf{1} & & & \\ & & & & 0 & & \\ \vdots & & & & & \ddots & \vdots \\ 0 & \dots & & & & \dots & 0 \end{pmatrix} =: \Sigma.$$

Bemerkung: Die Orthogonalprojektionsmatrix hat demnach k Eigenwerte $= 1$ und $n - k$ Eigenwerte $= 0$:
$\lambda_1 = ... = \lambda_k = 1$ und $\lambda_{k+1} = ... = \lambda_n = 0$. Die Matrix Q^*PQ ist diagonal, wobei die ersten k Einträge 1 sind.
Damit haben wir (wegen $Q^*Q = I$) mit $P = Q\Sigma Q^*$ eine SVD (Kap. 3) von P, für die gilt:

$$P^* = (Q\Sigma Q^*)^* = Q\Sigma Q^* = P.$$

□

Da ein orthogonaler Projektor – mit Ausnahme des trivialen Falls $P = I$ – mehrere Singulärwerte gleich 0 enthalten kann, liegt es nahe, mit einer reduzierten Form (vgl. 3) zu arbeiten. Im Folgenden verwenden wir folgende Schreibweise: Eine Matrix A habe k Singulärwerte ungleich 0. Dann ist

$$\hat{Q} := (q_1|...|q_k)$$

die reduzierte Form der Matrix Q.

4.1.2 Projektion mit orthonormaler Basis

Bemerkung: $\hat{Q}$ kann von einer SVD kommen, muss aber nicht. $\{q_1, ..., q_n\}$ ist ein beliebiges Orthonormalsystem (ONS).

Sei $\{q_1, ..., q_n\}$ eine Menge orthonormaler Vektoren des $\mathbb{C}^m$ und $\hat{Q}$ die zugehörige $(m \times n)$-Matrix. Dann gilt für alle $v \in \mathbb{C}^m$:

$$v = r + \sum_{i=1}^{n} (q_i q_i^*) v,$$

$$\text{mit} \quad r \in < q_1, ..., q_n >^{\perp}$$
$$\text{und} \quad \|q_i\| = 1 \quad \forall i = 1, .., n.$$

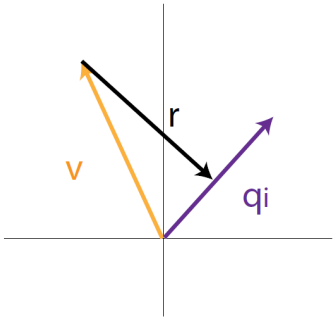


Abbildung 4.10: Projektion mit orthonormaler Basis.

Bemerkung: Jeder beliebige Vektor v kann in eine Linearkombination mit orthogonalen Komponenten zerlegt werden. Der Vektor v wird also in eine Summe aus einem Element aus dem Spaltenraum von $\hat{Q}$ und einem Element aus dem orthogonalen Komplement zerlegt. Die einzige Voraussetzung ist die Orthonormalität der Spalten von $\hat{Q}$.
Daher ist die Abbildung $v \longmapsto \sum_{i=1}^{n}(q_i q_i^*)v$ eine orthogonale Projektion auf $Bild(\hat{Q})$:

$$\begin{pmatrix} \\ w \\ \\ \end{pmatrix} = \begin{pmatrix} \\ Q \\ \\ \end{pmatrix} \cdot \begin{pmatrix} \\ Q^* \\ \\ \end{pmatrix} \begin{pmatrix} \\ v \\ \\ \end{pmatrix}.$$

Bemerkung:

Somit ist jedes Produkt $\hat{Q}\hat{Q}^*$ eine Projektion auf den Spaltenraum von $\hat{Q}$.
Auch das Komplement eines orthogonalen Projektors ist eine orthogonale Projektion, da $(I - \hat{Q}\hat{Q}^*)$ hermitesch ist.

Bemerkung 4.2: Spezialfall: Rang-1-Projektion

Für nur einen Vektor q mit $||q||_2 = 1$ ergibt sich

$$P_q = qq^* \qquad \text{und} \qquad P_{\perp q} = I - qq^*. \tag{4.1}$$

Dies ist eine Projektion in eine einzelne Richtung q.
Ist $\tilde{q}$ ein allgemeiner Vektor, so verallgemeinert sich (4.1) zu

$$P\tilde{q} = \frac{\tilde{q}\tilde{q}^*}{\tilde{q}^*\tilde{q}} \quad \text{und} \quad P_{\perp\tilde{q}} = I - \frac{\tilde{q}\tilde{q}^*}{\tilde{q}^*\tilde{q}}.$$

4.2 Die QR-Zerlegung

Seien zunächst $A \in \mathbb{C}^{m\times n}$ eine Matrix mit vollem Rang (1.2) und a_j, $j = 1, ..., n$, deren Spaltenvektoren. Wir betrachten die geschachtelten Unterräume von $\mathbb{C}^m$, die von den Spalten a_j aufgespannt werden:

$$\underbrace{< a_1 >}_{dim\leqslant 1} \subseteq \underbrace{< a_1, a_2 >}_{dim\leqslant 2} \subseteq \underbrace{< a_1, a_2, a_3 >}_{dim\leqslant 3} \subseteq ... \underbrace{< a_1, a_2, a_3, ..., a_n >}_{dim\leqslant n} \tag{4.2}$$

Dabei ist $< a_1, ..., a_j >$ der von den Vektoren $a_1, ..., a_j$ aufgespannte Unterraum. **Bemerkung:**
Achtung:
Die Notation $\langle \cdot, \cdot \rangle$ ist mehrfach belegt. Sie bezeichnet zum einen das Skalarprodukt (vgl. 1.2), zum anderen aber auch den *span*, d.h. das Erzeugendensystem eines Vektorraums. Während das Skalarprodukt nur zwei Komponenten enthalten kann, können beim *span* mehrere Vektoren aufgelistet werden. Das Skalarprodukt kann auch mit einer Trennlinie geschrieben werden ($\langle \cdot | \cdot \rangle$), der *span* hingegen nur mit Komma ($\langle \cdot, ..., \cdot \rangle$).
Konstruiere nun eine Folge *orthonormaler Vektoren* $q_1, q_2, ...$ etc., die diese verschachtelten Unterräume aufspannen, d.h.:

$$< q_1, q_2, ..., q_j > = < a_1, a_2, ..., a_j > \qquad \text{für } j = 1, ..., n. \tag{4.3}$$

Somit haben wir mit (4.2) und (4.3)

$$\left(\begin{array}{c|c|c|c} a_1 & a_2 & \dots & a_n \end{array}\right) = \underbrace{\left(\begin{array}{c|c|c|c} q_1 & q_2 & \dots & q_n \end{array}\right)}_{\hat{Q}\in\mathbb{C}^{m\times n}} \underbrace{\begin{pmatrix} r_{11} & r_{12} & \dots & \dots & r_{1n} \\ 0 & r_{22} & & & \vdots \\ \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \dots & r_{nn} \end{pmatrix}}_{\hat{R}\in\mathbb{C}^{n\times n}}, \tag{4.4}$$

wobei die Einträge auf der Diagonalen ungleich null sind, also $r_{kk} \neq 0 \ \forall k = 1, ..., n$.
Bemerkung: Schreibweise: r_{ij} ist der Eintrg der i-ten Zeile und j-ten Spalte von R. Alternativ schreibt man auch $r_{i,j}$ oder $(R)_{i,j}$.
Ausgeschrieben erhält man demnach für die einzelnen Spalten

$$\begin{aligned} a_1 &= r_{11}q_1 \\ a_2 &= r_{12}q_1 + r_{22}q_2 \\ a_3 &= r_{13}q_1 + r_{23}q_2 + r_{33}q_3 \\ &\vdots \\ a_n &= r_{1n}q_1 + r_{2n}q_2 + ... + r_{nn}q_n \end{aligned} \tag{4.5}$$

und als Matrix

$$A = \hat{Q}\hat{R},$$

wobei $\hat{Q}$ eine $(m \times n)$-Matrix mit orthonormalen Spalten und $\hat{R}$ eine rechte obere $(n \times n)$-Dreiecksmatrix ist.

Definition 4.3: Reduzierte QR-Faktorisierung

Die in Gleichung (4.4) dargestellte Repräsentation heißt *reduzierte QR-Faktorisierung*.

Bei der *vollständigen QR-Faktorisierung* von $A \in \mathbb{C}^{m \times n}$ werden $(m - n)$ orthonormale Spalten zu $\hat{Q}$ hinzugefügt, sodass $\hat{Q} \in \mathbb{C}^{m \times m}$ unitär wird. Gleichzeitig werden $(m - n)$ Zeilen mit Nullen an die rechte obere Dreiecksmatrix angehängt.
Schematisch sieht dies wie folgt aus:

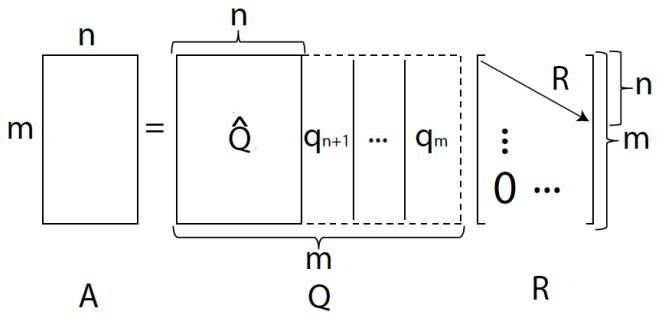


Abbildung 4.11: Projektion mit orthonormaler Basis

Theorem 4.3: QR-Faktorisierung

Jede Matrix $A \in \mathbb{C}^{m \times n}$, $m \geqslant n$, besitzt eine QR-Faktorisierung.

Genauer: Jede Matrix A besitzt eine *vollständige* QR-Faktorisierung. Diese ist aufgrund der erweiterten Spalten von Q nicht eindeutig. Eine *reduzierte* QR-Faktorisierung hingegen ist eindeutig, wenn $r_{ii} > 0$ gilt. (Für einen Beweis sei hier auf Kapitel 7 in [3] verwiesen.)

Eine mögliche Anwendung der QR-Zerlegung ist das Lösen von Gleichungssystemen:

Bemerkung 4.3:

Ein Gleichungssystem kann wie folgt gelöst werden

Gegeben: $A \in \mathbb{C}^{m\times m}$, A nicht singulär, und $b \in \mathbb{C}^m$

Gesucht: x mit $Ax = b$

Lösung: QR-Faktorisierung

(i) Berechne die QR-Faktorisierung $A = QR$:

$$Ax = QRx = b \Leftrightarrow Rx = Q^*b$$

(ii) $Rx = Q^*b$ wird einfach durch Rücksubstitution gelöst.

4.2.1 Einschub: Vorwärts- und Rückwärtssubstitution

Gegeben seien die Matrizen $L \in \mathbb{C}^{n\times n}$ und $U \in \mathbb{C}^{n\times n}$ sowie der Vektor $b \in \mathbb{C}^n$. Dabei sei L eine untere Dreiecksmatrix und U eine obere Dreiecksmatrix, wobei die Digonaleinträge $L_{i,i}$ und $U_{i,i}$, $i \in \{1, ..., n\}$ ungleich 0 sind.
Gesucht sind Methoden zum Lösen der Gleichungssysteme $Lx = b$ und $Ux = b$.
Man unterscheidet dabei zwischen *Vorwärts-* und *Rückwärtssubstitution*, jenachdem ob die Matrix *untere* oder *obere* Dreiecksgestalt hat.

Vorwärtssubstitution:

$$Lx = b$$

L ist eine *untere* Dreiecksmatrix

Algorithmus 4.1: Vorwärtssubstitution

for $i = 1 \to n$ **do**
  $x(i) = b(i)/L_{i,i}$
  $b(i+1:n) = b(i+1:n) - L_{(i+1):n,i} \cdot x(i)$
end for

Fall $n = 2$:

$$\begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Dann berechnet man:

$$\begin{aligned} x_1 &= b_1/L_{11} \\ x_2 &= (b_2 - L_{21}x_1)/L_{22} \end{aligned}$$

Allgemein:

$$x_i = \left(b_i - \sum_{j=1}^{i-1} L_{ij}x_j \right) / L_{ii}$$

Rückwärtssubstitution:

$$Ux = b$$

U ist eine *obere* Dreiecksmatrix

Fall $n = 2$:

$$\begin{pmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Dann berechnet man:

$$\begin{aligned} x_2 &= b_2/U_{22} \\ x_1 &= (b_1 - U_{12}x_2)/U_{11} \end{aligned}$$

Allgemein:

$$x_i = \left(b_i - \sum_{j=i+1}^{n} U_{ij}x_j \right) / U_{ii}$$

Algorithmus 4.2: Rückwärtssubstitution

for $i = n \to 1$ **do**
  $x(i) = b(i)/U_{i,i}$
  $b(1 : i-1) = b(1 : i-1) - U_{1:(i-1),i} \cdot x(i)$
end for

4.3 Das Gram-Schmidt-Verfahren

4.3.1 Das klassische Gram-Schmidt-Verfahren

Bemerkung: Problem: Die klassische Gram-Schmidt-Orthogonalisierung ist wegen fortgeführter Rundungsfehler numerisch instabil! (5) Wie konstruiert man eine orthogonale Matrix Q mit den Spalten $\{q_1, ..., q_n\}$?
Idee: *Gram-Schmidt-Orthogonalisierung.*
Seien $a_1, a_2, ..., a_n$ wie in Gleichung (4.2) als Spalten der Matrix $A \in \mathbb{C}^{m\times n}$ gegeben. Im j-ten Schritt soll $q_j \in < a_1, ..., a_j >$ berechnet werden, sodass

$$q_j \perp < q_1, ..., q_{j-1} > \qquad \text{und} \qquad \|q_j\| = 1.$$

Bemerkung: $(q_i^* a_j)$ ist äquivalent zu $\langle q_i | a_j \rangle$. Die erste Gleichung kann als Matrix-Vektorprodukt aufgefasst werden, wobei die erste Matrix aus dem Adjungierten zu q_i besteht, also einer (komplex konjugierten) $(1\times m)$-Matrix im Produkt mit einem $(m\times 1)$-Vektor. Lösung: Setze $q_j := \frac{v_j}{\|v_j\|}$, wobei

$$v_j = a_j - (q_1^* a_j)q_1 - (q_2^* a_j)q_2 - ... - (q_{j-1}^* a_j)q_{j-1}. \tag{4.6}$$

Stellen wir nun jeweils die i-te die Gleichung in (4.5) nach q_i um, so ergibt sich:

$$\begin{aligned} q_1 &= \tfrac{a_1}{r_{11}} \\ q_2 &= \tfrac{a_2 - r_{12}q_1}{r_{22}} \\ q_3 &= \tfrac{a_3 - r_{13}q_1 - r_{23}q_2}{r_{33}} \\ &\vdots \\ q_n &= \tfrac{a_n - \sum_{i=1}^{n-1} r_{in}q_i}{r_{nn}}. \end{aligned} \tag{4.7}$$

Die Koeffizienten in (4.6) lauten daher wie folgt:

$$r_{ij} := \begin{cases} q_i^* a_j, & (i \neq j) \\ \left\| a_j - \sum_{l=1}^{j-1} r_{lj} q_l \right\|_2, & (i = j) \end{cases} \tag{4.8}$$

Bemerkung: Da $\|\cdot\| \geqslant 0$, ist auch $r_{ii} \geqslant 0$.

Algorithmus 4.3: Klassisches Gram-Schmidt-Verfahren (CGS)

for $j = 1 \to n$ **do**
  $v_j = a_j$
  for $i = 1 \to j-1$ **do**
    $r_{ij} = q_i^* a_j$
    $v_j = v_j - r_{ij} q_i$
  end for
  $r_{jj} = \|v_j\|_2$
  $q_j = v_j / r_{jj}$
end for

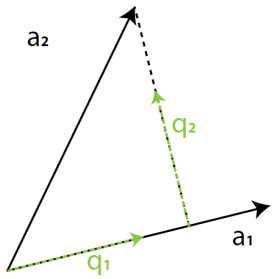


Abbildung 4.12: Beispiel für das klassische Gram-Schmidt-Verfahren

Beispiel (Der Fall $n = 2$):

Die obige Abbildung veranschaulicht das Verfahren für zwei Vektoren: a_2 ließe sich wie folgt errechnen:

$$\begin{aligned} q_1 &= \frac{a_1}{\|a_1\|} \\ q_2 &= a_2 - q_1 q_1^* a_2 \\ a_2 &= q_1 q_1^* a_2 + q_2 q_2^* a_2 \end{aligned}$$

Beispiel (CGS für zwei Vektoren im $\mathbb{R}^2$):

Betrachte die beiden Vektoren $a_1 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$ und $a_2 = \begin{pmatrix} 1.5 \\ 3 \end{pmatrix}$:

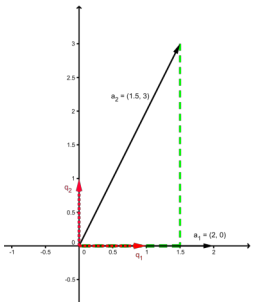


Abbildung 4.13: GS im $\mathbb{R}^2$

$$\begin{aligned} q_1 &= \frac{a_1}{\|a_1\|} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ q_2' &= a_2 - \underbrace{\langle a_2, q_1 \rangle}_{=1.5 \cdot 1 + 3 \cdot 0 = 1.5} q_1 \\ &= \begin{pmatrix} 1.5 \\ 3 \end{pmatrix} - 1.5 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \end{pmatrix} \\ q_2 &= \frac{q_2'}{\|q_2'\|} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{aligned}$$

Der Vektor a_2 lässt sich als Linearkombination von q_1 und q_2 darstellen (gestrichelte grüne Linie):

$$a_2 = q_1\langle q_1, a_2\rangle + q_2\langle q_2, a_2\rangle = 1.5\begin{pmatrix}1\\0\end{pmatrix} + 3\begin{pmatrix}0\\1\end{pmatrix}$$

Ist der $j-$te Spaltenvektor $a_j \notin \langle q_1, \ldots, q_{j-1}\rangle$, so berechnet sich q_j nach Gleichung 4.6 gemäß

$$q_j = \frac{a_j - (q_1^* a_j)q_1 - (q_2^* a_j)q_2 - ... - (q_{j-1}^* a_j)q_{j-1}}{\left\|a_j - (q_1^* a_j)q_1 - (q_2^* a_j)q_2 - ... - (q_{j-1}^* a_j)q_{j-1}\right\|_2}$$

Schreiben wir die Skalarprodukte $q_i^* a_j$ rechts der Vektoren q_i und Klammern wir a_j nach rechts aus so erhalten wir

$$q_j = \frac{\left(I - q_1 q_1^* - q_2 q_2^* - ... - q_{j-1} q_{j-1}^*\right) a_j}{\left\|\left(I - q_1 q_1^* - q_2 q_2^* - ... - q_{j-1} q_{j-1}^*\right) a_j\right\|_2}$$

Bezeichnen wir mit $\hat{Q}_{j-1}$ die $(m \times (j-1))$-Matrix ist, welche die $(j-1)$ ersten Spalten von $\hat{Q}$ enthält, d.h.

$$\hat{Q}_{j-1} = \left(\begin{array}{c|c|c|c} & & & \\ q_1 & q_2 & . \ . \ . & q_{j-1} \\ & & & \end{array}\right),$$

so erhalten wir

$$\begin{aligned} q_j &= \frac{\left(I - q_1 q_1^* - q_2 q_2^* - ... - q_{j-1} q_{j-1}^*\right) a_j}{\left\|\left(I - q_1 q_1^* - q_2 q_2^* - ... - q_{j-1} q_{j-1}^*\right) a_j\right\|_2} \\ &= \frac{\left(I - \hat{Q}_{j-1}\hat{Q}_{j-1}^*\right) a_j}{\left\|\left(I - \hat{Q}_{j-1}\hat{Q}_{j-1}^*\right) a_j\right\|_2}. \end{aligned}$$

Die Matrix $\hat{Q}_{j-1}\hat{Q}_{j-1}^*$ projiziert orthogonal auf $< q_1, ..., q_{j-1} >$ und somit ist

$$P_j = I - \hat{Q}_{j-1}\hat{Q}_{j-1}^* \tag{4.9}$$

die $(m \times m)$-Matrix vom Rang $m-(j-1)$, die $\mathbb{C}^m$ orthogonal auf den zu $< q_1, ..., q_{j-1} >$ orthogonalen Unterraum projiziert. Für $j = 1$ ist $P_1 = I$.
Mit diesen Projektionsmatrizen läßt sich das klassische Gram-Schmidtverfahren als Folge von Zuweisungen schreiben:

$$q_1 = \frac{P_1 a_1}{\|P_1 a_1\|_2}, \quad q_2 = \frac{P_2 a_2}{\|P_2 a_2\|_2}, ..., \quad q_n = \frac{P_n a_n}{\|P_n a_n\|_2}. \tag{4.10}$$

4.3.2 Das modifizierte Gram-Schmidt-Verfahren

Das klassische Gram-Schmidt-Verfahren ist numerisch instabil, d.h. maschinell errechnete Ergebnisse können von den tatsächlichen (exakten) Lösungen teilweise deutlich abweichen. Zur Stabilisierung wurde das Verfahren daher wie folgt modifiziert: anstatt wie

der klassische Algorithmus 4.3 eine einzige orthogonale Projektion vom Rang $m-(j-1)$ zu berechnen

$$q_j = \frac{P_j a_j}{\|P_j a_j\|_2}, \tag{4.11}$$

berechnet das modifizierte Verfahren dieselbe Projektion von P_j als Folge von $j-1$ Projektionen

$$P_{\perp q_i} = I - q_i q_i^*, i = 1, \cdots, j-1 \tag{4.12}$$

vom Rang $m-1$ auf die jeweils zu $q_1, ..., q_{j-1}$ orthogonalen Unterräume.
Ausmultiplizieren und Ausnutzen der Orthogonalität von q_i und $q_j, j \neq i$ liefert

$$\begin{aligned} P_{\perp q_{j-1}} ... P_{\perp q_2} P_{\perp q_1} &= (I - q_{j-1}q_{j-1}^*) \cdots (I - q_2 q_2^*)(I - q_1 q_1^*) \\ &= I - q_{j-1}q_{j-1}^* - \cdots - q_2 q_2^* - q_1 q_1^* \\ &= P_j. \end{aligned}$$

Das heißt eine zu Gleichung (4.11) äquivalente Aussage ist

$$q_j = \frac{P_{\perp q_{j-1}} ... P_{\perp q_2} P_{\perp q_1} a_j}{\|P_{\perp q_{j-1}} ... P_{\perp q_2} P_{\perp q_1} a_j\|}. \tag{4.13}$$

Der modifizierte Algorithmus berechnet also v_j auf die folgende Weise:

$$\begin{aligned} v_j^{(1)} &= a_j \\ v_j^{(2)} &= P_{\perp q_1} v_j^{(1)} = (I - q_1 q_1^*) v_j^{(1)} = v_j^{(1)} - q_1 q_1^* v_j^{(1)} \\ v_j^{(3)} &= P_{\perp q_2} v_j^{(2)} = v_j^{(2)} - q_2 q_2^* v_j^{(2)} \\ &\vdots \\ v_j = v_j^{(j)} &= P_{\perp q_{j-1}} v_{j-1}^{(j-1)} = v_j^{(j-1)} - q_{j-1} q_{j-1}^* v_j^{(j-1)} \end{aligned} \tag{4.14}$$

Dabei ist $v_j^{(i)}$ der Vektor v_j in der i-ten Iteration des Algorithmus, d.h. nach der $(i-1)$-ten Projektion zur Berechnung von q_j.

Algorithmus 4.4: Modifiziertes Gram-Schmidt-Verfahren (MGS)

for $i = 1 \to n$ **do**
  $v_i = a_i$
end for
for $i = 1 \to n$ **do**
  $r_{ii} = \|v_i\|_2$
  $q_i = v_i / r_{ii}$
  for $j = i+1 \to n$ **do**
    $r_{ij} = q_i^* v_j$
    $v_j = v_j - r_{ij} q_i$
  end for

end for

Tabelle 4.1: Vergleich von klassischem und modifiziertem GS-Verfahren

Klassisches GS-Verfahren	**Modifiziertes GS-Verfahren**
• numerisch instabil	• numerisch stabiler
• 1 Projektion vom Rang $m-(j-1)$	• $j-1$ Projektionen vom Rang $m-1$
• $v_j = P_j a_j$	• $v_j = P_{\perp q_{j-1}} \dots P_{\perp q_2} P_{\perp q_1} a_j$

Satz 4.4: Komplexität der Gram-Schmidt-Algorithmen

Die Algorithmen (4.3) (klassisches Gram-Schmidt-) und (4.4) (modifiziertes Gram-Schmidt-Verfahren) benötigen $O(mn^2)$ Operationen, wobei eine Addition, eine Multiplikation, eine Wurzeloperation, eine Division und eine Subtraktion jeweils eine Operation darstellen.

Beweis: Wir betrachten hier denn Fall des modifizierten Gram-Schmidt-Verfahrens (für CGS analog):

for $i = 1 \to n$ **do**
  for $j = i + 1 \to n$ **do** — m Multiplikationen + $(m-1)$ Additionen
    $r_{ij} = q_i^* v_j$ — m Multiplikationen + m Subtraktionen
    $v_j = v_j - r_{ij} q_i$
  end for — $\Rightarrow 4m - 1$ Operationen
end for

Der Gesamtaufwand beträgt somit also:

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(4m-1) &= (4m-1)\sum_{i=1}^{n}\sum_{j=i+1}^{n}1 \\ &= (4m-1)\sum_{i=1}^{n}(n-i) = (4m-1)\sum_{i=0}^{n-1} i \\ &= (4m-1)\frac{n(n-1)}{2} \approx 2mn^2\end{aligned}$$

□

Geometrische Anschauung der Komplexität:

In der äußeren Schleife von Algorithmus 4.3 wird die gesamte Matrix bearbeitet, indem ein Vielfaches der ersten Spalte von der anderen Spalte abgezogen wird. Im zweiten Schritt, muss nur noch eine Untermatrix bearbeitet werden, da ein Vielfaches der zweiten Spalte von der Spalte $3, ..., n$ abgezogen wird usw.
Dies bedeutet, dass für die Gram-Schmidt-Orthogonalisierung $\approx n^2 m$ Operationen benötigt werden:
$\approx n^2 m$ Operationen $\widehat{=}$ Volumen der Figur.

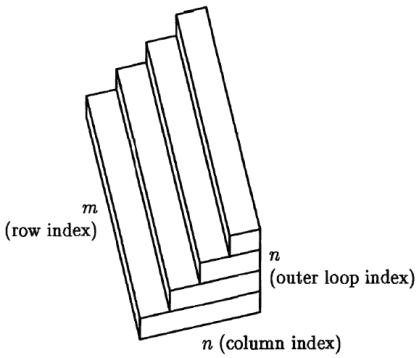


Abbildung 4.14: Geometrische Anschauung

4.3.3 Gram-Schmidt als Dreiecksorthonormalisierung

Jeder äußere Schritt des modifizierten Gram-Schmidt-Algorithmus kann als Multiplikation mit einer quadratischen oberen Dreiecksmatrix von rechts interpretiert werden.

Schritt 1:

$$\left(\begin{array}{c|c|c|c} v_1^{(1)} & v_2^{(1)} & \cdots & v_n^{(1)} \end{array}\right) \underbrace{\begin{pmatrix} \frac{1}{r_{11}} & -\frac{r_{12}}{r_{11}} & -\frac{r_{13}}{r_{11}} & \cdots \\ & 1 & & \\ & & 1 & \\ & & & \ddots \end{pmatrix}}_{R_1} = \left(\begin{array}{c|c|c|c|c} q_1 & v_2^{(2)} & v_3^{(2)} & \cdots & v_n^{(2)} \end{array}\right)$$

Schritt 2:

$$R_2 = \begin{pmatrix} 1 & & & & \\ & \frac{1}{r_{22}} & -\frac{r_{23}}{r_{22}} & -\frac{r_{24}}{r_{22}} & \cdots \\ & & 1 & & \\ & & & 1 & \\ & & & & \ddots \end{pmatrix}$$

Schritt 3:

$$R_3 = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \frac{1}{r_{33}} & -\frac{r_{34}}{r_{33}} & -\frac{r_{35}}{r_{33}} & \cdots \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & \ddots \end{pmatrix}$$

Am Ende der Iteration wurden folgende Multiplikationen ausgeführt:

$$A \underbrace{R_1 R_2 \cdots R_n}_{\hat{R}^{-1}} = \hat{Q}$$

Dabei ist $\hat{R}^{-1}$ eine reelle rechte obere Dreiecksmatrix.

4.4 Householder-Triangularisierung

Wir betrachten nun eine Alternative, die QR-Zerlegung einer Matrix $A \in \mathbb{C}^{m\times n}$ zu berechnen:
Anstatt A von rechts mit einer oberen Dreiecksmatrix zu multiplizieren, wird A von links mit unitären Matrizen Q_k multilpiziert, sodass

$$\underbrace{Q_n \cdots Q_2 Q_1}_{Q^*} A = R \tag{4.15}$$

eine obere Dreiecksmatrix wird. Da $Q = Q_1^* Q_2^* \cdots Q_n^*$ auch unitär ist, erhält man so ebenfalls eine vollständige QR-Zerlegung.
Dieses Verfahren wird *Householder-Triangularisierung* genannt.

Insgesamt haben wir damit zur Berechnung einer QR-Faktorisierung nun zwei Methoden kennen gelernt:

(i) **Gram-Schmidt:** Dreiecksorthonormalisierung

(ii) **Householder:** orthogonale Triangularisierung

4.4.1 Idee der Householder-Transformation

Bemerkung: Dieses Verfahren wurde begründet durch und benannt nach Alston Scott Householder (1958). Schematisch lässt sich die Householder-Transformation wie folgt darstellen:

$$\underbrace{\begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} \end{pmatrix}}_{A} \xrightarrow{Q_1} \underbrace{\begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} \end{pmatrix}}_{Q_1 A} \xrightarrow{Q_2} \underbrace{\begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} \\ 0 & 0 & \mathbf{x} \\ 0 & 0 & \mathbf{x} \end{pmatrix}}_{Q_2 Q_1 A} \xrightarrow{Q_3} \underbrace{\begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{Q_3 Q_2 Q_1 A}$$

Konkret bedeutet dies: Q_k verändert die Zeilen $k, ..., m$. Dabei bildet Q_k eine Linearkombination dieser Zeilen, wobei die Nullen in den Spalten $1, ..., k-1$ erhalten bleiben. Nach n Schritten ist die Matrix $Q_n ... Q_1 A$ eine obere Dreiecksmatrix.

4.4.2 Householder-Spiegelungen

Definition 4.4: Householder-Transformation

Die Matrix

$$H := I - \frac{2}{v^*v} vv^* \qquad \in \mathbb{C}^{m\times m} \qquad \text{mit } v \in \mathbb{C}^m \backslash \{0\}$$

heißt *Householder-Transformation.*

Lemma 4.5: Householder-Transformation

Die Householder-Transformation ist eine hermitesch unitäre Matrix mit

$$Hv = -v, \qquad Hw = w \qquad \forall w \in \{v\}^\perp.$$

Beweis:

- H ist nach Definition hermitesch. ✓
- H ist unitär:

$$\begin{aligned} H^*H = H^2 &= I - \frac{4}{v^*v} vv^* + \frac{4}{(v^*v)^2} v(v^*v)v^* \\ &= I - \frac{4}{v^*v} vv^* + \frac{4}{v^*v} vv^* \\ &= I. \end{aligned}$$

- Sei $w \perp v$. Dann gilt:

$$\begin{aligned} Hw &= Iw - \frac{2}{v^*v} \underbrace{vv^*w}_{=0} = w, \\ Hv &= Iv - \frac{2}{v^*v} vv^*v = v - 2v = -v. \end{aligned}$$

□

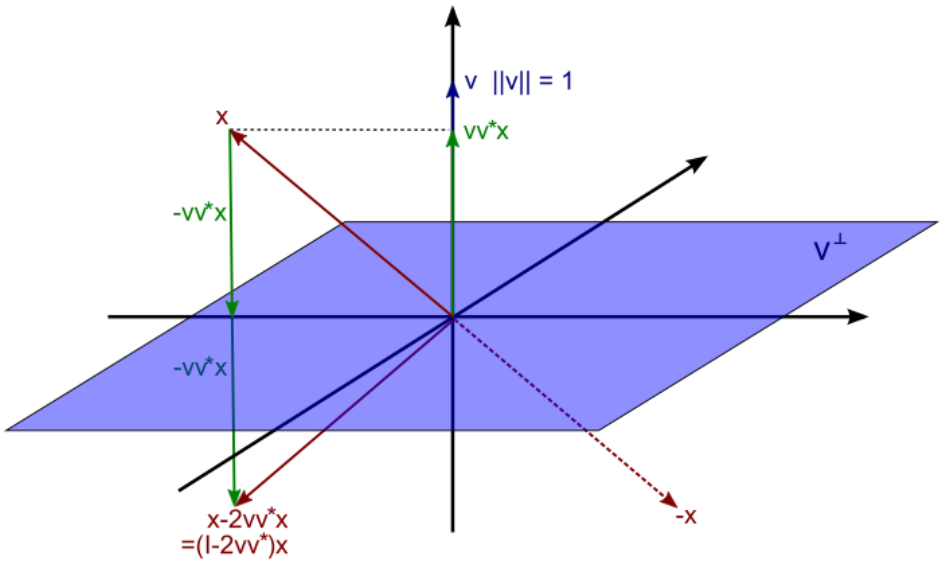


Abbildung 4.15: Geometrische Anschauung der Householder-Transformation

Bemerkung 4.4:

Die Householder-Transformation ist unter der $\|\cdot\|_2$ invariant:

$$\|Hx\|_2^2 = (Hx)^* Hx = x^* H^* Hx = x^* x = \|x\|_2^2$$

Vertiefung des Householder-Verfahrens:

Mit Hilfe der Householder-Transformation konstruieren wir die unitären Matrizen Q_k wie folgt:

$$Q_k = \begin{pmatrix} I & 0 \\ 0 & H \end{pmatrix},$$

wobei I eine $(k-1) \times (k-1)$ Einheitsmatrix und H eine $(m-k+1) \times (m-k+1)$ unitäre Matrix ist, welche die Nullen in der k-ten Spalte einführt. H wird als Householder-Matrix (Definition 4.4) gewählt.

Sei dazu $x \in \mathbb{C}^{m-k+1}$ der Vektor mit den Einträgen $k, ..., m$ aus der k-ten Spalte der

Matrix $Q_{k-1}Q_{k-2}\cdots Q_1 A$. Dann soll H zu der folgenden Abbildung führen:

$$x = \begin{pmatrix} x_k^{(k)} \\ x_{k+1}^{(k)} \\ \vdots \\ x_m^{(k)} \end{pmatrix} \quad \xrightarrow{H} \quad Hx = \begin{pmatrix} \pm\|x\|_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \pm\|x\|_2 e_1.$$

Daher ist:

$$\begin{aligned} Hx &= x - \frac{2}{v^*v}v(v^*x) = \xi e_1, \\ \text{mit } |\xi| &= \|x\|_2, \end{aligned}$$

d.h. also, dass v ein Vielfaches von $x - \xi e_1$ sein muss.
x wird also durch diese Transformation wie gewünscht gespiegelt (siehe Abbildung rechts).

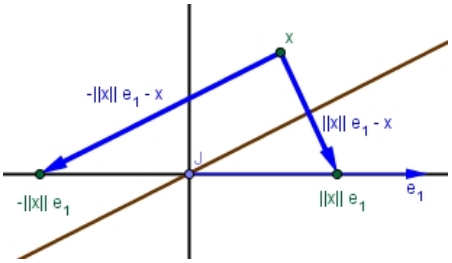


Abbildung 4.16: Geometrische Anschauung

Damit bei dieser Subtraktion keine Auslöschung auftritt, wählen wir

$$\xi = \begin{cases} -\frac{x_1}{|x_1|}\|x\|_2 & \text{für } x_1 \neq 0 \\ -\|x\|_2 & \text{für } x_1 = 0 \end{cases}$$

Bemerkung: $-\frac{x_1}{|x_1|}\|x\| = -\text{sign}(x_1)$. ξe_1 sollte aus numerischen Gründen nicht zu nahe bei x liegen. Um die Stabilität des Algorithmus zu erhöhen, kehren wir das Vorzeichen entsprechend um!
Nun wählen wir v so, dass x auf ξe_1 abgebildet wird, d.h.

$$v = \lambda(x - \xi e_1).$$

Setzen wir nun Werte für ξ ein und wählen $\lambda = \frac{1}{\|x\|_2}$, $\|x\|_2 \neq 0$, so ergibt sich:

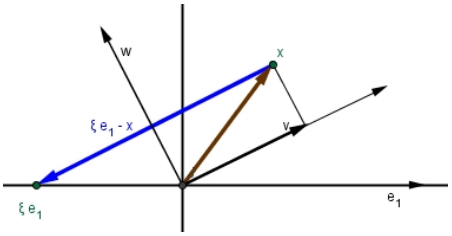


Abbildung 4.17: Geometrische Anschauung

$$\begin{aligned} v &= \lambda(x - \xi e_1) \\ &= \frac{1}{\|x\|_2}\left(x + \frac{x_1\|x\|_2}{|x_1|}e_1\right) \\ &= \frac{1}{|x_1|\|x\|_2}\left(|x_1|x + x_1\|x\|_2 e_1\right) \qquad \text{für } x_1 \neq 0 \end{aligned}$$

In beiden Fällen gilt:

$$\begin{aligned} v^*x &= \|x\|_2 + |x_1| \\ v^*v &= 2 + 2\frac{|x_1|}{\|x\|_2}. \end{aligned}$$

Analog ergibt sich

$$\begin{aligned} v &= \lambda(x - \xi e_1) \\ &= \frac{1}{\|x\|_2}(x + \|x\|_2 e_1) \\ &= \frac{x}{\|x\|_2} + e_1 \qquad \text{für } x_1 = 0 \end{aligned}$$

Mit diesem v ergibt sich in der Tat für $x_1 \neq 0$

$$\begin{aligned} Hx = x - \frac{2}{v^*v} v(v^*x) &= x - 2\frac{\|x\|_2 + |x_1|}{2 + 2\frac{|x_1|}{\|x\|_2}} \frac{|x_1| x + x_1 \|x\|_2 e_1}{|x_1|\|x\|_2} \\ &= x - \|x\|_2 \frac{\|x\|_2 + |x_1|}{\|x\|_2 + |x_1|} \frac{|x|_1 x + x_1 \|x\|_2 e_1}{|x|_1 \|x\|_2} \\ &= x - \frac{|x|_1 x + x_1 \|x\|_2 e_1}{|x|_1 \|x\|_2} \\ &= x - x - \frac{x_1}{|x_1|}\|x\|_2 e_1 \\ &= -\frac{x_1}{|x_1|}\|x\|_2 e_1 \end{aligned}$$

und analog für $x_1 = 0$

$$Hx = x - \|x\|_2 v = -\|x\|_2 e_1.$$

⇝ Diese Überlegungen führen zu folgendem Algorithmus:

Wir benutzen eine Matlab-ähnliche Notation zur Kennzeichnung von Teilen einer Matrix: $A_{[i:j,k:l]}$ ist die $(j - i + 1) \times (l - k + 1)$ Untermatrix mit linker oberer Ecke a_{ij} und unterer rechter Ecke a_{kl}. Falls die Untermatrix ein Vektor ist, so schreiben wir $A_{[i,k:l]}$ oder $A_{[i:j,k]}$ und wir definieren $\text{sign}(x_1) = +1$, falls $x_1 \geqslant 0$ und $\text{sign}(x_1) = -1$ falls $x_1 < 0$.

Algorithmus 4.5: Householder QR-Zerlegung

for $i = 1 \to n$ **do**
- $x = A_{[i:m,i]}$ (i-te Spalte der $(m - i + 1) \times (n - i + 1)$ Teilmatrix)
- $\hat{v}_i = \text{sign}(x_1)\|x\|_2 e + x$ (mit $e = (1, 0, \ldots, 0)^T \in \mathbb{R}^{m-i+1}$)
- $v_i = \dfrac{\hat{v}_i}{\|\hat{v}_i\|_2}$
- $A_{[i:m,i:n]} = A_{[i:m,i:n]} - 2v_i v_i^* A_{[i:m,i:n]}$

end for

Bemerkung 4.5: Berechnung von Q ist optional

Der Algorithmus 4.5 berechnet die obere Dreiecksmatrix R der Faktorisierung $A = QR$. Q selbst wird nicht explizit berechnet. Der Grund dafür ist, dass die Berechnung von Q zusätzlichen Aufwand bedeuten würde, der meist nicht unbedingt notwendig ist.

Beispiel (Anwendung der QR-Zerlegung):
Lösung von $Ax = b$ durch QR-Zerlegung. Dabei sei A quadratisch und habe vollen Rang.

$$Ax = QRx = b$$

Strategie: Berechne $y = Q^*b$ und löse dann $Rx = y$. Wende dann sukzessive die Operation wie bei der Berechnung von R an.
Im vorangehenden Beispiel benötigt man die explizite Berechnung von Q, die im Algorithmus 4.5 jedoch nicht gegeben ist.
Q kann mittels folgender Algorithmen berechnet werden:

Algorithmus 4.6: Berechnung von Q^*b

for $k = 1 \to n$ **do**
  $b_{k:m} = b_{k:m} - 2v_k v_k^* b_{k:m}$
end for

Algorithmus 4.7: Berechnung eines Produktes Qy

for $k = n \to 1$ **do**
  $y_{k:m} = y_{k:m} - 2v_k v_k^* y_{k:m}$
end for

Somit kann man Q auf zwei Arten berechenen: Berechnung von QI

1. mittels $Q^*e_1, Q^*e_2, ..., Q^*e_m$ und anschließendem Konjugieren *oder*
2. mittels $Qe_1, Qe_2, ..., Qe_m$.

Satz 4.6: Komplexität der Householder-Faktorisierung

Die Householder-Faktorisierung benötigt $mn^2 - \frac{1}{3}n^3 + O(mn)$ Operationen.

Die Komplexität liegt also wie beim Gram-Schmidt-Verfahren in $O(mn^2)$ und kann daher anschaulisch mit dem Volumen einer Pyramide verglichen werden (vgl. oben).

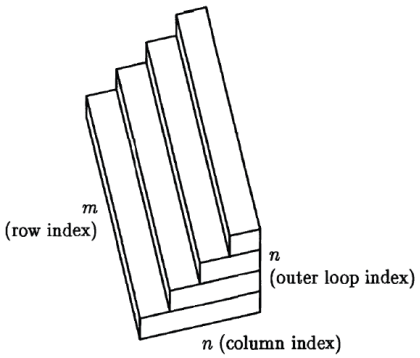


Abbildung 4.18: Geometrische Anschauung

QR-Zerlegung und Householder-Transformation:

Bemerkung 4.6: Anwenden der QR-Zerlegung

1. Die QR-Zerlegung ist anwendbar auf das Ausgleichsproblem

$$\|b - Ax\|_2 = \|Q^*(b - Ax)\|_2 = \|Q^*b - Q^*QRx\|_2 = \|c - Rx\|_2$$

mit $c = Q^*b$. Zerlegen wir R und c in

$$R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}, c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \; R_1 \in \mathbb{C}^{n\times n}, c_1 \in \mathbb{C}^n,$$

dann ist nach Pythagoras

$$\|b - Ax\|_2^2 = \|c - Rx\|_2^2 = \|c_1 - R_1x\|_2^2 + \|c_2\|_2^2 \geqslant \|c_2\|_2^2,$$

mit Gleichheit genau dann, wenn $c_1 = R_1x \Leftrightarrow x = R_1^{-1}c$.

2. Die QR-Zerlegung (mit der Householder-Transformation) gehört zu den numerisch stabilsten Algorithmen der Numerischen Linearen Algebra. Der Grund liegt darin, dass unitäre Transformationen keinerlei Fehlerverstärkung hervorrufen.

Kapitel 5

Kondition und Stabilität

5.1 Kondition

Betrachte zunächst die Auswertung einer reellwertigen Funktion:

$$f : \mathbb{R} \longrightarrow \mathbb{R}, \quad x \mapsto f(x);$$
$$\tilde{f} \text{ Algorithmus zur Berechnung von } f.$$

Dann heißt

$$\|\tilde{f}(x) - f(x)\| \quad \textit{absoluter Fehler} \quad \text{und} \tag{5.1}$$
$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \quad \textit{relativer Fehler.} \tag{5.2}$$

Aufgrund von *Daten-* oder *Rundungsfehlern* wird die Funktion f i.A. nicht an der Stelle x, sondern an der Stelle $\tilde{x} = x + \Delta x$ ausgewertet.
Wie wirkt sich der Fehler aber auf das Ergebnis aus?

Bezeichnen wir mit

$$\Delta y = f(x + \Delta x) - f(x) \tag{5.3}$$

den *fortgepflanzten absoluten Fehler*, so gilt nach dem Mittelwertsatz für $f \in \mathcal{C}^1$ **Bemerkung:**
Zur Erinnerung:
MWS: $f : [a, b] \mapsto \mathbb{R}$, $a < b$, stetig auf $[a, b]$ und diffbar in (a, b)
$\Rightarrow \exists \xi \in (a, b)$:
$f'(\xi) = \frac{f(b)-f(a)}{b-a}$

$$\Delta y = f(x + \Delta x) - f(x) = f'(\xi)\Delta x, \qquad \text{wobei} \quad \xi \in [x, x + \Delta x].$$

Bemerkung: Eine Funktion $f : \mathbb{R} \to \mathbb{R}$ heißt *Lipschitz-stetig*, falls eine Konstante L existiert, sodass gilt: $|f(x_1) - f(x_2)| \leqslant L \cdot |x_1 - x_2|$. Ist die Ableitung *Lipschitz-stetig*, dann gilt sogar

$$\Delta y = f'(x)\Delta x + O(|\Delta x|^2).$$

Dabei heißt $a_\varepsilon \in O(b_\varepsilon)$, falls $\exists C > 0$ mit $|a_\varepsilon| \leqslant C b_\varepsilon \quad \forall \varepsilon$ aus einer vereinbarten Grundmenge $E \subset \mathbb{R}^+$. Das heißt aber $\frac{a_\varepsilon}{b_\varepsilon} \to \varepsilon_0$. Falls $a_\varepsilon = O(b_\varepsilon)$ und $b_\varepsilon = O(a_\varepsilon)$, so schreiben wir auch $a_\varepsilon \sim b_\varepsilon$.

Vernachlässigen wir den quadratischen Term, so ist

$$K_{abs} = |f'(x)| \tag{5.4}$$

ein Maß für die *Fehlerverstärkung* des *absoluten Eingabefehlers*.

Üblicherweise ist der *relative Fehler* von größerer Bedeutung:

$$\frac{\Delta y}{y} \approx \frac{f'(x)}{f(x)} \Delta x = f'(x) \frac{x}{f(x)} \frac{\Delta x}{x}.$$

Definition 5.1: Absolute und relative Konditionszahl

Die Zahl

$$K_{abs} = |f'(x)| \tag{5.5}$$

heißt *absolute Konditionszahl* des Problems $x \mapsto f(x)$. Für $x \cdot f(x) \neq 0$ ist

$$K_{rel} = \left| \frac{f'(x) \cdot x}{f(x)} \right| \tag{5.6}$$

die *relative Konditionszahl* des Problems.

Bemerkung 5.1: Gut und schlecht konditioniert

Die Konditionszahlen beschreiben die Verstärkung des absoluten bzw. relativen Fehlers. Ein Problem heißt *schlecht konditioniert*, falls eine der beiden Konditionszahlen deutlich größer als 1 ist. Ansonsten heißt es *gut konditioniert*.

Beispiel ($f(x) = x + a$):

$$f(x) = x + a, \quad f'(x) = 1 \qquad \Rightarrow K_{abs} = |f'(x)| = 1$$

Für die relative Konditionszahl ergibt sich

$$K_{rel} = \left| \frac{f'(x) \cdot x}{f(x)} \right| = \left| \frac{x}{x + a} \right|.$$

K_{rel} wird groß, falls $|x+a| \ll |x|$, also wenn $x \approx -a$. Diesen schlecht konditionierten Fall bezeichnet man als Auslöschung: Für $a=-1$, $x=1,000001$ und $\Delta x = 0,001$ ist

$$f(x) = x + a = 0,000001 \qquad \text{und} \qquad f(x+\Delta x) + a = 0,001001.$$

Der absolute Fehler ist also $0,001$, d.h. gleich dem Eingangsfehler. Der relative Fehler ist

$$\frac{\Delta y}{y} = \frac{0,001}{0,000001} \approx 10^6 \frac{0,001}{1,000001},$$

wobei $\frac{0,001}{1,000001}$ um den Faktor 10^6 verstärkt wird.

Beispiel ($f(x) = ax$):
Betrachte die Funktion $f(x) = ax$. Die absolute und relative Konditionszahl lautet

$$\begin{aligned} K_{abs} &= |f'(x)| = |a|, \\ K_{rel} &= \left| \frac{f'(x) \cdot x}{ax} \right| = 1. \end{aligned}$$

In diesem Fall ist die absolute Konditionszahl schlecht, falls $|a| \gg 1$. Der relative Fehler bleibt fest.

Beispiel ($f(x) = x^2 - 2x + 1 = a_2x^2 + a_1x + a_0$):
Nullstellenbestimmung: $f(x) = x^2 - 2x + 1 = a_2x^2 + a_1x + a_0$. Die Nullstellen sind abhängig von den Koeffizienten a_i.
Betrachte die Abhängigkeit von a_0: $f(x) = x^2 - 2x + 0,9999 = (x - 0,99)(x - 1,01)$. D.h. die Wurzeln des Polynoms ändern sich in der Größenordnung der Wurzel der Koeffizienten.

5.1.1 Kondition einer Matrix

Betrachte das Problem $x \mapsto F(x)$, wenn x und $F(x) \in \mathbb{C}^n$. Wir beschränken uns auf den Spezialfall, in dem ein LGS der Form $Az = b$ zu lösen ist, wobei $A \in \mathbb{C}^{n\times n}$ invertierbar ist. In diesem Fall ist $F(b) = A^{-1}b$. Bei Eingangsfehler Δb ergibt sich

$$z = A^{-1}b, \qquad z + \Delta z = A^{-1}(b + \Delta b) = A^{-1}b + A^{-1}\Delta b,$$

d.h. die berechnete Lösung $z + \Delta z$ enthält den fortgepflanzten Fehler $\Delta z = A^{-1}\Delta b$.
Sind $\|\cdot\|_M$ und $\|\cdot\|$ ein verträgliches Matrix-/Vektornormpaar, d.h. $\|Ax\| \leqslant \|x\|\|A\|_M$ $\forall A \in \mathbb{C}^{m\times n}$ and $\forall x \in \mathbb{C}^n$, dann folgt:

$$\frac{\|\Delta z\|}{\|z\|} = \frac{\|A^{-1}\Delta b\|}{\|z\|} \leqslant \|A^{-1}\|_M \frac{\|\Delta b\|}{\|b\|} \frac{\|Az\|}{\|z\|} \leqslant \|A^{-1}\|_M \|A\|_M \frac{\|\Delta b\|}{\|b\|} \tag{5.7}$$

Das heißt, ein relativer Eingangsfehler in der Größenordnung $\frac{\|\Delta b\|}{\|b\|}$ führt zu einem relativen Fehler der Größenordnung $\frac{\|\Delta z\|}{\|z\|} \leqslant \|A^{-1}\|_M \|A\|_M \frac{\|\Delta b\|}{\|b\|}$ in der Lösung.

Definition 5.2: Kondition einer Matrix

Der Faktor

$$K_M(A) := \|A^{-1}\|_M \cdot \|A\|_M \tag{5.8}$$

wird als *Kondition der Matrix* A bezüglich der Norm $\|\cdot\|_M$ bezeichnet.

Bemerkung 5.2:

Falls $\|\cdot\|_M$ durch eine Vektornorm induziert wird, kann man Beispiele für b und Ab konstruieren, für die in (5.7) Gleichheit herrscht.

Satz 5.1: Kondition der Berechnung $b = Ax$

Das Problem der Berechnung $b = Ax$, gegeben x und eine nicht-singuläre Matrix $A \in \mathbb{C}^{m\times m}$, hat ebenfalls die Kondition

$$K_M(A) := \|A\|_M \|A^{-1}\|_M \tag{5.9}$$

bezüglich Störungen von x.

Beweis:

$$\begin{aligned}
\frac{\|\Delta b\|}{\|b\|} &= \frac{\|A\Delta x\|}{\|b\|} \\
&\leqslant \frac{\|A\|_M\|\Delta x\|}{\|b\|} = \|A\|_M \frac{\|\Delta x\|}{\|x\|}\frac{\|A^{-1}b\|}{\|b\|} \\
&\leqslant \|A\|_M\|A^{-1}\|_M \frac{\|\Delta x\|}{\|x\|}
\end{aligned}$$

□

Bemerkung: Die Pseudoinverse einer Matrix ist eine Verallgemeinerung der inversen Matrix für nicht-quadratische oder singuläre Matrizen $A \in \mathbb{C}^{m\times n}$.
Es gilt: Hat A vollen Zeilenrang, berechnet sich die Pseudoinverse wie folgt:

$$A^+ = A^*(AA^*)^{-1}.$$

Hat A vollen Spaltenrang, so gilt:

$$A^+ = (A^*A)^{-1}A^*.$$

Bemerkung 5.3: Kondition und Singulärwerte in nicht singulären Fall

Falls $\|\cdot\|_M = \|\cdot\|_2$ und A nicht singulär ist, gilt wegen $\|A\|_2 = \|U\Sigma V^*\|_2 = \|\Sigma\|_2$ insbesondere $\|A\|_2 = \sigma_1$ und $\|A^{-1}\|_2 = \frac{1}{\sigma_n}$ und daher $K(A) = \frac{\sigma_1}{\sigma_n}$. $\epsilon = \frac{1}{K(A)}\sqrt{K(A)^2 - 1}$ kann als Exzentrizität der Ellipse, die das Bild der Einheitskugel unter A ist, interpretiert werden.

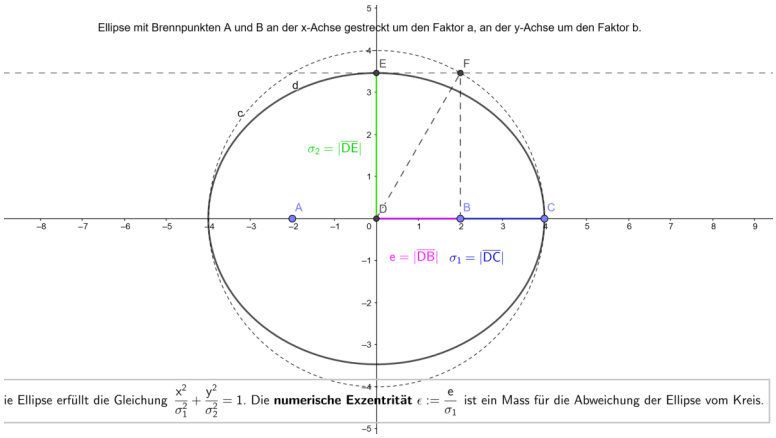


Abbildung 5.19: Exzentrizität einer Ellipse.

Die Exzentritt ist ein Maß für die Abweichung der Ellipse vom Kreis. Es gilt $\left(\varepsilon = \sqrt{1 - \left(\frac{b}{a}\right)^2}\right)$

Bemerkung 5.4: Kondition und Singulärwerte im singulären Fall

Für eine Matrix $A \in \mathbb{C}^{m\times n}$, $m \geqslant n$, mit vollem Rang wird die Kondition mit Hilfe der *Pseudoinversen* definiert:

$$K(A) = \|A\|\|A^+\|.$$

Für die $\|\cdot\|_2$ gilt in diesem Fall

$$K(A) = \frac{\sigma_1}{\sigma_n}.$$

5.1.2 Kondition eines Gleichungssystems

Wie ändert sich $b = A^{-1}x$, wenn A um ΔA variiert?
Sei dazu A nicht singulär. Betrachte dazu die Gleichung

$$\begin{aligned} & (A + \Delta A)(x + \Delta x) = b \\ \Leftrightarrow \quad & \underbrace{Ax}_{b} + \Delta A x + A\Delta x + \underbrace{\Delta A \Delta x}_{\approx 0} = b \\ & \text{d.h.} \qquad \Delta A x \approx -A\Delta x \\ \Leftrightarrow \quad & \|\Delta x\| \leqslant \|A^{-1}\|\|\Delta A\|\|x\| \\ \Leftrightarrow \quad & \tfrac{\|\Delta x\|}{\|x\|} \leqslant \|A^{-1}\|\|A\|\tfrac{\|\Delta A\|}{\|A\|}. \end{aligned}$$

(ΔA benötigt ein anderes Δx)

5.2 Floating Point Arithmetik

Reelle Zahlen können bekanntlicher Weise mit einem Computer nicht exakt dargestellt werden. Dies hat zur Folge, dass sich Rechenfehler wegen der Ungenauigkeit der Darstellung anhäufen können und zu extremen Abweichungen vom tatsächlichen Ergebnis führen.

Definition 5.3: Floating Point Zahlen

Die diskrete endliche Menge aller in einem Rechner darstellbaren Zahlen bezeichnen wir als *Floating Point Zahlen*:

$$F = \left\{ x \in \mathbb{R} \;\middle|\; \begin{array}{c} \exists \beta \geqslant 2 \text{ und } e \geqslant 1 \text{ mit } m \in \mathbb{N}, 1 \leqslant m \leqslant \beta^t : \\ x = 0 \text{ oder } x = \pm \left(\frac{m}{\beta^t}\right) \beta^e \end{array} \right\},$$

wobei t die *Präzision* ist und in IEEE die Werte 24 bzw. 53 annimmt. β heißt *Basis* und nimmt normalerweise den Wert 2 an.

Bemerkung 5.5:

1. Wir können m auf den Bereich $\beta^{t-1} \leqslant m \leqslant \beta^t - 1$ beschränken und damit die Wahl von m eindeutig machen.
2. Der Term $\left(\pm\frac{m}{\beta^t}\right)$ heißt *Mantisse*, e heißt *Exponent*.
3. Relative Genauigkeit bei *single precision*: $2^{-24} \approx 5,96 \times 10^{-8}$

Relative Genauigkeit bei *double precision*: $2^{-53} \approx 1,11 \times 10^{-16}$

Modellierung der Rechenarithmetik Zur Modellierung der Rechenarithmetik machen wir folgende beiden Modellannahmen:
Bemerkung: ε_M steht für $\varepsilon_{machine}$.

1. $x \bullet y := \square(x \circ y)$, wobei $x, y \in F$, $\circ$ die mathematische Grundoperation, $\square x$ die Rundung von $x \in \mathbb{R}$ zur nächstgelegenen Maschinenzahl und $\bullet$ die Realisierung dieser Grundoperation auf dem Rechner ist.

2. Die *Rechneroperation* soll den tatsächlichen Wert innerhalb einer *maximalen relativen Genauigkeit* bestimmen, d.h. $\forall x \in \mathbb{R}, \exists \varepsilon : |\varepsilon| \leqslant \varepsilon_M$ mit

$$\square\, x = x(1 + \varepsilon). \tag{5.10}$$

   Dabei ist $\varepsilon_M = \inf\{x > 0 : 1 \oplus x \neq 1\}$.

Bemerkung 5.6: Machine Epsilon

1. Der genaue Wert von ε_M ist rechnerabhängig. In der Regel ist $\varepsilon_M = 2^{-d}$ für ein $d > 0$.

2. Die Gleichung (5.10) wird falsch, wenn eine von Null verschiedene Zahl x auf Null gerundet wird. Diese Situation nennt man *Underflow*.
   Ähnlich spricht man von *Overflow*, wenn das Rechenergebnis größer als die darstellbaren Zahlen wird (also über den gegebenen Zahlenbereich hinausreicht).
   Sofern weder Overflow noch Underflow auftreten, ist der relative Rundungsfehler nach diesem Modell beschränkt durch

$$\frac{|\square x - x|}{|x|} \leqslant \varepsilon_M. \tag{5.11}$$

Unter beiden Modellannahmen 1. und 2. sind alle Elementaroperationen auf dem Rechner in der folgenden Weise realisiert:

$$\begin{aligned} &\forall x, y \in \mathbb{R}, \exists \varepsilon : |\varepsilon| \leqslant \varepsilon_M, \text{ s.d.} \\ &\qquad x \bullet y = (x \circ y)(1 + \varepsilon) \end{aligned} \tag{5.12}$$

d.h. jede Operation der Floating Point Arithmetik ist *genau* bis auf einen *relativen Fehler* in der Größenordnung der Maschinengenauigkeit.

5.3 Stabilität

Notation: Sei $\mathbf{f} : \mathbb{R} \to F$ eine Funktion, die $f(x)$ unter Verwendung von Fließkommazahlen berechnet, d.h. $\mathbf{f}$ beschreibt einen Realisierungsalgorithmus.
Bemerkung: Zur Erinnerung:

$$K_{rel} = \left| \frac{f'(x) \cdot x}{f(x)} \right|$$

Genauigkeit: Betrachten wir die Implementierung von $\mathbf{f}$ zur Lösung eines Problems $x \mapsto f(x) = y$, $x \in D(f) \subset \mathbb{R}$. Seien dazu x und y von Null verschieden. $\mathbf{f}$ ist ein *guter* Realisierungsalgorithmus, wenn

$$\left| \frac{\mathbf{f}(x) - f(x)}{f(x)} \right| \leqslant c_V K_{rel} \varepsilon_M \tag{5.13}$$

mit einem mäßig großen $c_V > 0$, das von x unabhängig ist.

Definition 5.4: Vorwärtsstabilität

Diese Form der Stabilitätsanalyse heißt *Vorwärtsanalyse* und $\mathbf{f}$ heißt *vorwärts stabil*, wenn Gleichung (5.13) erfüllt ist, d.h.

$$\left| \frac{\mathbf{f}(x) - f(x)}{f(x)} \right| \leqslant c_V K_{rel} \varepsilon_M$$

Bemerkung 5.7: Problem der Vorwärtsanalyse

Die Vorwärtsanalyse ist häufig sehr schwierig, da die Abhängigkeit von der Konditionszahl subtil ist. Die Lösung des Problems ist die Rückwärtsanalyse:

$$\frac{\Delta y}{y} \approx \frac{f'(x)}{f(x)} \Delta x = \underbrace{f'(x) \frac{x}{f(x)}}_{K_{rel}} \frac{\Delta x}{x} \tag{5.14}$$

Definition 5.5: Rückwärtsanalyse

Bei der *Rückwärtsanalyse* interpretiert man die berechnete Näherung als exakte Lösung mit gestörten Eingangsdaten, d.h. $\mathbf{f}(x) = f(x + \Delta x)$ und untersucht $|\Delta x|$. Gibt es mehrere Urbilder $x + \Delta x$, so wählt man das mit kleinster Störung Δx. Gilt dann

$$\left|\frac{\Delta x}{x}\right| \leqslant c_R \varepsilon_M, \tag{5.15}$$

mit einem mäßig großen $c_R > 0$, das von x unabhängig ist, so heißt $\mathbf{f}$ *rückwärts stabil.*

Bemerkung 5.8: Rückwärtsstabilität

Für einen rückwärts stabilen Algorithmus gilt nach Gleichung (5.14) mit $\tilde{x} = x + \Delta x$:

$$\left|\frac{\mathbf{f}(x) - f(x)}{f(x)}\right| = \left|\frac{f(\tilde{x}) - f(x)}{f(x)}\right| \leqslant K_{rel} \left|\frac{\tilde{x} - x}{x}\right| \leqslant c_R K_{rel} \varepsilon_M.$$

Bis auf den Einfluss des Approximationsfehlers in (5.14) ist damit jeder rückwärts stabile Algorithmus auch vorwärts stabil. Die Umkehrung gilt i.A. nicht.

5.3.1 Beispiele für Stabilitäten

Rückwärtsstabilität der Subtraktion

Theorem 5.2: Rückwärtsstabilität der Subtraktion

Die Subtraktion ist rückwärts stabil.

Beweis: Zur Erinnerung:

$$\forall x \in \mathbb{R}, \exists \varepsilon, |\varepsilon| \leqslant \varepsilon_M, \text{ s.d.} \qquad \square(x) = x(1 + \varepsilon) \qquad \text{(siehe (5.10))}$$
$$\forall x, y \in \mathbb{R}, \exists \varepsilon, |\varepsilon| \leqslant \varepsilon_M, \text{ s.d.} \qquad x \bullet y = (x \circ y)(1 + \varepsilon) \qquad \text{(siehe (5.12))}$$

Beweis der Rückwärtsstabilität:

$$\begin{aligned}
(\square x \boxminus \square y) &= (x(1 + \varepsilon_1) - y(1 + \varepsilon_2))(1 + \varepsilon_3) \\
&= x(1 + \varepsilon_1 + \varepsilon_3 + \varepsilon_1 \varepsilon_3) - y(1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_2 \varepsilon_3) \\
&= x(1 + \varepsilon_4) - y(1 + \varepsilon_5)
\end{aligned}$$

wobei $|\varepsilon_4|, |\varepsilon_5| \leqslant 2\varepsilon_M + O(\varepsilon_M^2)$, d.h. $(\square x \boxminus \square y) = \tilde{x} - \tilde{y}$ mit $\frac{|\tilde{x}-x|}{|x|} = O(\varepsilon_M)$, $\frac{|\tilde{y}-y|}{|y|} = O(\varepsilon_M)$. Das heißt: $(\square x \boxminus \square y)$ ist rückwärts stabil. □

Rückwärtsstabilität der Rückwärtssubstitution

$$\begin{pmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ & r_{22} & \dots & r_{2m} \\ & & \ddots & \vdots \\ & & & r_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Theorem 5.3: Rückwärtsstabilität der Rückwärtssubstitution

Die Rückwärtssubstitution ist rückwärts stabil, d.h. für die berechnete Lösung $\tilde{x}$ gilt:

$$(R + \delta R)\tilde{x} = b \tag{5.16}$$

für eine obere Dreiecksmatrix $\delta R \in \mathbb{C}^{m\times m}$ mit $\frac{\|\delta R\|}{\|R\|} = O(\varepsilon_M)$.
Insbesondere gilt $\forall i, j$:

$$\frac{|\delta r_{ij}|}{|r_{ij}|} \leqslant m\varepsilon_M + O(\varepsilon_M^2). \tag{5.17}$$

Aus der komponentenweise Rückwärtsstabilität (5.17) folgt die normweise Stabilität (5.16).

Beweis: Wir beweisen die Aussage von Theorem 5.3 für $m = 1, 2, 3$.
$\underline{m = 1:}$

$$\tilde{x}_1 = b_1 \boxdiv r_{11} = \frac{b_1}{r_{11}}(1 + \varepsilon_1) \qquad \text{für } |\varepsilon_1| \leqslant \varepsilon_M.$$

Setze $\varepsilon_1' := \frac{-\varepsilon_1}{1+\varepsilon_1}$, dann folgt

$$\tilde{x}_1 = \frac{b_1}{r_{11}(1 + \varepsilon_1')} \qquad \text{für } |\varepsilon_1'| \leqslant \varepsilon_M + O(\varepsilon_M^2),$$

wobei die Abschätzung für $|\varepsilon_1'|$ aus der Taylor-Entwicklung folgt.
Damit gilt für $(R + \delta R)\tilde{x} = b$:

$$(r_{11} + \underbrace{r_{11}\varepsilon_1'}_{\delta r_{11}})\tilde{x}_1 = b_1 \qquad \text{für } \frac{|\delta r_{11}|}{|r_{11}|} \leqslant \varepsilon_M + O(\varepsilon_M^2).$$

$\underline{m = 2:}$

$$\tilde{x}_2 = b_2 \boxdiv r_{22} = \frac{b_2}{r_{22}(1 + \varepsilon_1)} \qquad \text{für } |\varepsilon_1| \leqslant \varepsilon_M + O(\varepsilon_M^2).$$

Im zweiten Schritt haben wir mehrere Floating Point Operationen zu berücksichtigen:
Bemerkung: Dabei ist $\varepsilon'_{3,4} = \frac{-\varepsilon_{3,4}}{1+\varepsilon_{3,4}}$ zu verstehen als Kurzschreibweise für

$$\begin{aligned} \varepsilon'_3 &= \frac{-\varepsilon_3}{1+\varepsilon_3} \text{ und} \\ \varepsilon'_4 &= \frac{-\varepsilon_4}{1+\varepsilon_4} \end{aligned}$$

.

$$\begin{aligned} \tilde{x}_1 &= (b_1 \boxminus (\tilde{x}_2 \boxtimes r_{12})) \boxdiv r_{11} \\ &= (b_1 \boxminus (\tilde{x}_2 \cdot r_{12}(1+\varepsilon_2))) \boxdiv r_{11}, && |\varepsilon_2| \leqslant \varepsilon_M \\ &= \frac{(b_1 - \tilde{x}_2 \cdot r_{12}(1+\varepsilon_2))(1+\varepsilon_3)}{r_{11}} \cdot (1+\varepsilon_4), && |\varepsilon_3|, |\varepsilon_4| \leqslant \varepsilon_M \\ \overset{\varepsilon'_{3,4}=\frac{-\varepsilon_{3,4}}{1+\varepsilon_{3,4}}}{=} & \frac{b_1 - \tilde{x}_2 r_{12}(1+\varepsilon_2)}{r_{11}(1+\varepsilon'_3)(1+\varepsilon'_4)} && |\varepsilon'_3|, |\varepsilon'_4| \leqslant \varepsilon_M + O(\varepsilon_M^2) \\ &= \frac{b_1 - \tilde{x}_2 r_{12}(1+\varepsilon_2)}{r_{11}(1+2\varepsilon_5)} && |\varepsilon_5| \leqslant \varepsilon_M + O(\varepsilon_M^2), \end{aligned}$$

Damit können wir die Rückwärtsanalyse nun als $(R+\delta R)\tilde{x} = b$ schreiben mit

$$\frac{|\delta R|}{|R|} = \begin{pmatrix} \frac{|\delta r_{11}|}{|r_{11}|} & \frac{|\delta r_{12}|}{|r_{12}|} \\ 0 & \frac{|\delta r_{22}|}{|r_{22}|} \end{pmatrix} = \begin{pmatrix} 2|\varepsilon_5| & |\varepsilon_2| \\ 0 & |\varepsilon_1| \end{pmatrix} \leqslant \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \varepsilon_M + O(\varepsilon_M^2),$$

wobei Betrag und Division in $|\delta R|/|R|$ komponentenweise zu verstehen sind.

$\underline{m = 3:}$

Die ersten zwei Schritte erfolgen wie oben. Dabei ist $\varepsilon_5 = \varepsilon_3$.

$$\begin{aligned} \tilde{x}_1 &= [(b_1 \boxminus (\tilde{x}_2 \boxtimes r_{12})) \boxminus (\tilde{x}_3 \boxtimes r_{13})] \boxdiv r_{11} \\ &= \frac{[(b_1 - \tilde{x}_2 r_{12}(1+\varepsilon_4))(1+\varepsilon_6) - \tilde{x}_3 r_{13}(1+\varepsilon_5)](1+\varepsilon_7)}{r_{11}(1+\varepsilon'_8)} \\ &= \frac{[(b_1 - \tilde{x}_2 r_{12}(1+\varepsilon_4))(1+\varepsilon_6) - \tilde{x}_3 r_{13}(1+\varepsilon_5)]}{r_{11}(1+\varepsilon'_8)(1+\varepsilon'_7)} \\ &= \frac{(b_1 - \tilde{x}_2 r_{12}(1+\varepsilon_4)) - \tilde{x}_3 r_{13}(1+\varepsilon_5)(1+\varepsilon'_6)}{r_{11}(1+\varepsilon'_6)(1+\varepsilon'_8)(1+\varepsilon'_7)}. \end{aligned}$$

Somit gilt für $(R+\delta R)\tilde{x} = b$:

$$\frac{|\delta R|}{|R|} = \begin{pmatrix} \frac{|\delta r_{11}|}{|r_{11}|} & \frac{|\delta r_{12}|}{|r_{12}|} & \frac{|\delta r_{13}|}{|r_{13}|} \\ 0 & \frac{|\delta r_{22}|}{|r_{22}|} & \frac{|\delta r_{23}|}{|r_{23}|} \\ 0 & 0 & \frac{|\delta r_{33}|}{|r_{33}|} \end{pmatrix} \leqslant \begin{pmatrix} 3 & 1 & 2 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix} \varepsilon_M + O(\varepsilon_M^2)$$

$\underline{m \geqslant 3:}$

Die Analyse für größere m funktioniert analog.

□

Rückwärtsstabilität des Produkts Q^*b (ohne Beweis)

Bei der Berechnung von Q^*b treten Fehler auf. Anstatt Q^*b genau zu berechnen, wird ein $\tilde{y}$ berechnet.

Theorem 5.4: Rückwärtsstabilität des Produkts Q^*b

Die Operation Q^*b ist rückwärts stabil, d.h. $\exists \delta Q, \|\delta Q\| = O(\varepsilon_M)$ mit

$$(Q + \delta Q)\tilde{y} = b. \tag{5.18}$$

Das heißt, das Resultat der Berechnung von Householder-Reflektoren in Floating Point Arithmetik entspricht einer Multiplikation mit einer Matrix

$$(Q + \delta Q)^{-1}.$$

Rückwärtsstabilität der Householder-Triangularisierung (ohne Beweis)

Theorem 5.5: Rückwärtsstabilität der Householder-Triangularisierung

Sei $A = QR$, $A \in \mathbb{C}^{m\times n}$, eine mittels Householder-Transformation berechnete Faktorisierung von A. Sei weiterhin $\tilde{R}$ die mit Floating-Point-Genauigkeit berechnete obere Dreiecksmatrix und $\tilde{Q} = \tilde{Q}_1\tilde{Q}_2...\tilde{Q}_n$, wobei $\tilde{Q}_k$ der exakt unitäre Reflektor des entsprechenden Vektors $\tilde{v}_k$ in Floating-Point-Arithmetik ist.
Dann gilt

$$\tilde{Q}\tilde{R} = A + \delta A, \qquad \frac{\|\delta A\|}{\|A\|} = O(\varepsilon_M) \tag{5.19}$$

für $\delta A \in \mathbb{C}^{m\times n}$.

Lösen von Gleichungssystemen $Ax = b$ mittels QR-Faktorisierung

Algorithmus 5.1: Lösung von $Ax = b$ mittels QR-Faktorisierung

$$\begin{aligned} QR &= A \\ y &= Q^* b \\ x &= R^{-1} y \end{aligned}$$

Theorem 5.6: Rückwärtsstabilität der Lösung von $Ax = b$ mittels QR-Faktorisierung

Der Algorithmus 5.1 ist rückwärts stabil, d.h.

$$(A + \delta A)\tilde{x} = b, \qquad \frac{\|\delta A\|}{\|A\|} = O(\varepsilon_M) \tag{5.20}$$

für ein $\delta A \in \mathbb{C}^{n\times n}$

Beweis: Die Gleichungen (5.16) und (5.18) implizieren:

$$\begin{aligned} b = (\tilde{Q} + \delta Q)(\tilde{R} + \delta R)\tilde{x} &= [\underbrace{\tilde{Q}\tilde{R}}_{=A+\delta A} + (\delta Q)\tilde{R} + \tilde{Q}\delta R + (\delta Q)(\delta R)]\tilde{x} \\ &= [A + \underbrace{\delta A + \delta Q\tilde{R} + \tilde{Q}\delta R + \delta Q\delta R}_{\Delta A}]\tilde{x} \\ &= [A + \Delta A]\tilde{x}. \end{aligned}$$

Z.z.: Jeder Term von ΔA ist relativ zu A klein.

Nach Theorem 5.3.1 gilt

$$\tilde{Q}\tilde{R} = A + \delta A, \qquad \tilde{Q} \quad \text{unitär}, \qquad \frac{\|\delta A\|}{\|A\|} = O(\varepsilon_M)$$

$$\frac{\|\tilde{R}\|}{\|A\|} \leqslant \|\tilde{Q}^*\| \frac{\|A + \delta A\|}{\|A\|} = O(1), \qquad \text{falls } \varepsilon_M \to 0.$$

Daraus erhalten wir

$$\frac{\|\delta Q\tilde{R}\|}{\|A\|} \leqslant \|\delta Q\| \frac{\|\tilde{R}\|}{\|A\|} = O(\varepsilon_M)$$

und

$$\frac{\|\tilde{Q}\delta R\|}{\|A\|} \leqslant \|\tilde{Q}\| \frac{\|\delta R\|}{\|\tilde{R}\|} \frac{\|\tilde{R}\|}{\|A\|} = O(\varepsilon_M).$$

Schließlich gilt noch

$$\frac{\|(\delta Q)(\delta R)\|}{\|A\|} \leqslant \|\delta Q\| \frac{\|\delta R\|}{\|A\|} = O(\varepsilon_M^2).$$

□

Theorem 5.7: Genauigkeit der Lösung

Die mit Algorithmus 5.1 berechnete Lösung erfüllt

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(K(A) \cdot \varepsilon_M).$$

Kapitel 6

LU-Zerlegung

6.1 LU-Faktorisierung (Gauß-Elimination)

Sei $A \in \mathbb{C}^{m\times m}$. Im Folgenden wollen wir die Matrix A in eine linke untere und eine rechte obere Dreiecksmatrix zerlegen.
Idee: Bringe A auf Dreiecksgestalt, indem, ähnlich wie bei der QR-Zerlegung, Nullen unterhalb der Diagonalen in den Spalten erzeugt werden. Dies lässt sich mit einer Folge von *unteren Dreiecksmatrizen* L_k durchführen:

$$\underbrace{L_{m-1}...L_2L_1}_{L^{-1}}A = U,$$

$$A = L_1^{-1}L_2^{-1}\cdots L_{m-1}^{-1}U = LU$$

U obere Dreiecksmatrix (**u**pper)

L untere Dreiecksmatrix (**l**ower)

Im Deutschen wird anstatt der hier gewählten englischen Bezeichnung **LU** häufig die Bezeichnung **LR** gewählt, wobei **L** für die untere **linke** Dreiecksmatrix und **R** für die obere **rechte** Dreiecksmatrix steht.
Schematische Darstellung:

$$\begin{pmatrix} x & x & x & x\\ x & x & x & x\\ x & x & x & x\\ x & x & x & x \end{pmatrix} \overset{L_1}{\to} \begin{pmatrix} x & x & x & x\\ 0 & x & x & x\\ 0 & x & x & x\\ 0 & x & x & x \end{pmatrix} \overset{L_2}{\to} \begin{pmatrix} x & x & x & x\\ 0 & x & x & x\\ 0 & 0 & x & x\\ 0 & 0 & x & x \end{pmatrix} \overset{L_3}{\to} \begin{pmatrix} x & x & x & x\\ 0 & x & x & x\\ 0 & 0 & x & x\\ 0 & 0 & 0 & x \end{pmatrix}$$

Methoden:
Wir kennen die folgenden Methoden, eine Matrix A in eine Dreiecksmatrix umzuwandeln:

Gram-Schmidt:	$A = QR$	Orthogonalisierung
Householder:	$A = QR$	orthogonale Triangulierung
Gauß-Elimination:	$A = LU$	Triangularisierung (Gauß)

6.1.1 LU-Faktorisierung (ohne Pivotisierung)

Im k-ten Schritt des Eliminationsprozesses muss L_k so gewählt werden, dass alle Zeilen von A mit Index $1, \cdots, k$ erhalten bleiben und der k-te Spaltenvektor der Matrix A wie folgt verändert wird:

$$L_k a_k = L_k \begin{pmatrix} a_{1,k} \\ \vdots \\ a_{k,k} \\ a_{k+1,k} \\ \vdots \\ a_{m,k} \end{pmatrix} = \begin{pmatrix} a_{1,k} \\ \vdots \\ a_{k,k} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Dazu erhalten wir die ersten m Zeilen und ziehen für $k < j \leqslant m$ von der j-ten Zeile jeweils das $l_{j,k}$-fache der k-ten Zeile ab, wobei

$$l_{j,k} = \frac{a_{j,k}}{a_{k,k}}, k < j \leqslant m$$

Die Matrix L_k hat damit folgende Form:

$$L_k = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & -l_{m,k} & & 1 \end{pmatrix}$$

Durch diese Konstruktion erhalten wir in der k-ten Spalte Nullen unterhalb der Diagonalen:
Beispiel:

$$L_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -l_{3,2} & 1 & 0 & 0 \\ 0 & -l_{4,2} & 0 & 1 & 0 \\ 0 & -l_{5,2} & 0 & 0 & 1 \end{pmatrix}$$

Definieren wir den Vektor l_k durch

$$l_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{m,k} \end{pmatrix}, \text{wobei } l_{j,k} = \frac{a_{j,k}}{a_{k,k}}, k < j \leqslant m, l_{j,k} = 0, j \leqslant k,$$

so können wir die Matrix L_k schreiben als

$$L_k = I - l_k e_k^*. \tag{6.1}$$

Bei der Berechnung der Matrix L gilt unter anderem Folgendes:

- Da $e_k^* l_k = 0$, gilt:

$$(I - l_k e_k^*)(I + l_k e_k^*) = I - l_k e_k^* l_k e_k^* = I,$$

  d.h.: $$(I - l_k e_k^*)^{-1} = (I + l_k e_k^*).$$

-

$$\begin{aligned} L_k^{-1} L_{k+1}^{-1} &= (I + l_k e_k^*)(I + l_{k+1} e_{k+1}^*) \\ &= I + l_k e_k^* + l_{k+1} e_{k+1}^* + l_k \underbrace{e_k^* l_{k+1}}_{=0} e_{k+1}^* \\ &= I + l_k e_k^* + l_{k+1} e_{k+1}^* \end{aligned}$$

  Also berechnet sich L wie folgt:

$$L = L_1^{-1} L_2^{-1} \cdots L_{m-1}^{-1} = \begin{pmatrix} 1 & 0 & 0 & . & 0 \\ l_{2,1} & 1 & 0 & . & 0 \\ l_{3,1} & l_{3,2} & \ddots & 0 & 0 \\ & & \ddots & 1 & 0 \\ l_{m,1} & l_{m,2} & & l_{m,m-1} & 1 \end{pmatrix}$$

Algorithmus 6.1: LU-Zerlegung

$U = A$
$L = I$
for $k = 1 \to m - 1$ **do**
  for $j = k + 1 \to m$ **do**
    $l_{j,k} = u_{j,k} / u_{k,k}$
    $u_{j,k:m} = u_{j,k:m} - l_{j,k} u_{k,k:m}$
  end for
end for

Theorem 6.1: Komplexität der LU-Zerlegung

Der obige Algorithmus 6.1 benötigt etwa $\frac{2}{3}m^3$ Operationen.

Bemerkung 6.1:

1. Der Aufwand zur Berechnung der LU-Zerlegung mit $\frac{2}{3}m^3$ Operationen ist um den Faktor 2 kleiner als der Aufwand der QR-Zerlegung, die $\frac{4}{3}m^3$ Operationen benötigt. Aber die QR Zerlegung ist stabiler und daher bekannter.

2. $Ax = b$ kann durch eine LU-Zerlegung gelöst werden.

$$\begin{aligned} LUx &= b \\ Ly &= b \qquad \text{Forward Substitution } (\approx m^2 \text{ Operationen}) \\ Ux &= y \qquad \text{Back Substitution } (\approx m^2 \text{ Operationen}) \end{aligned}$$

6.1.2 LU-Faktorisierung (mit Pivotisierung)

Problem:

Der Algorithmus 6.1 (LU-Zerlegung ohne Pivotisierung) funktioniert nicht, falls $x_{k,k} = 0$: Betrachte dazu die Matrix $A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$.

Außerdem wird der Algorithmus instabil, falls $x_{k,k} \ll x_{j,k}$ für ein j mit $k < j \leqslant m$.
Da x_{kk} bei der Berechnung von l_k eine wichtige Rolle spielt, heißt $x_{k,k}$ *Pivotelement*.
Im k-ten Schritt werden Vielfache der k-ten Zeile von den Zeilen $k+1, ..., m$ der aktuellen Matrix abgezogen und so Nullen in der k-ten Spalte dieser Zeilen erzeugt:

$$\begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & x_{k,k} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & x_{k,k} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{pmatrix}$$

Lässt man Permutationen von Zeilen und Spalten zu, so gibt es keinen Grund, die k-te Zeile und k-te Spalte im k-ten Schritt zu bearbeiten.

Beispiel: $k = 2, i = 4$:

$$\begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & x_{i,k} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & x_{i,k} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{pmatrix}$$

... oder $k = 2, i = 4, j = 3$:

$$\begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & x_{i,k} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & 0 & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & 0 & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & x_{i,k} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & 0 & \mathbf{x} & \mathbf{x} \end{pmatrix}$$

Wird vor diesen Schritten das Element x_{ij} an die Position x_{kk} durch Zeilen- und Spaltenpermutation verschoben, so verläuft die Elimination wie im Standardfall. Leider ist der Aufwand für das Suchen eines guten Pivotelements im k-ten Schritt $O((m-k)^2)$, sodass insgesamt $O(m^3)$ Suchoperationen durchgeführt werden müssen (***komplettes Pivotieren***).

In der Praxis wird deshalb ***partielles Pivotieren*** eingesetzt. Dabei werden nur Zeilen vertauscht. Man sucht dazu das betragsgrößte Element in den $(m-k+1)$ Subdiagonleinträgen der k-ten Spalte ($O(m-k)$ Operationen, ergibt $O(m^2)$ für die gesamte LU-Zerlegung).

Sind bei der Spaltenpivotisierung alle Elemente der Spalte 0, so ist die Matrix singulär.

$$\underbrace{\begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & x_{i,k} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{pmatrix}}_{\text{Pivot auswählen}} \xrightarrow{P_1} \underbrace{\begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & x_{i,k} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{pmatrix}}_{\text{Zeilen vertauschen}} \xrightarrow{L_1} \underbrace{\begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & x_{i,k} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{pmatrix}}_{\text{Elimination}}$$

Mit diesem Verfahren erhält man eine *obere Dreiecksmatrix* U nach $(m-1)$ Schritten:

$$L_{m-1}P_{m-1}\cdots L_2P_2L_1P_1A = U,$$

wobei die Matrizen P Permutationsmatrizen sind.

Ein kleiner Exkurs: Permutationsmatrizen:

Eine Permutationsmatrix ist eine binäre Matrix, die in jeder Zeile und in jeder Spalte genau einen 1-Eintrag hat.

Definition 6.1: Permutationsmatrix

Sei π eine Permutation der Länge n. Eine Permutationsmatrix P zur Permutation π ist dann gegeben durch

$$P_\pi := \left(e_{\pi(1)}|...|e_{\pi(n)}\right).$$

Ein Beispiel für eine Permutationsmatrix: Vertauschen der 2-ten und 4-ten Zeile:

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}}_{P} \begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \end{pmatrix} = \begin{pmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \end{pmatrix}$$

Allgemeines Schema für das Vertauschen der Zeilen i und j:

$$P_{i,j} = \begin{array}{c} \\ \\ \\ i \rightarrow \\ \\ \\ \\ j \rightarrow \\ \\ \\ \end{array} \left(\begin{array}{ccccccc} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ \hline & & & 0 & & 1 & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \\ & & & 1 & & 0 & \\ \hline & & & & & & 1 \\ & & & & & & & \ddots \\ & & & & & & & & 1 \end{array}\right)$$

Bemerkung 6.2: Permutationsmatrizen

1. Es gilt: $P_{i,j}^2 = I$.
   (Dies entspricht einem zweimaligen Vertauschen der gleichen Zeilen)
2. Multiplikation einer Matrix A mit einer Permutationsmatrix von links vertauscht Zeilen.

3. Multiplikation einer Matrix A mit einer Permutationsmatrix von rechts vertauscht Spalten.

Betrachten wir nun die LU-Zerlegung unter Berücksichtigung von Permutationsmatrizen:

Lemma 6.2:

Seien $k < i$, P_i und L_k wie oben definiert. Dann ist $P_i L_k = L'_k P_i$, wobei L'_k bis auf eine Vertauschung von $l_{i_i,k}$ und $l_{j_i,k}$ wieder die Form von Gleichung (6.1) hat, d.h.

$$L'_k = I - l'_k e_k^*.$$

Beweis:

$$P_i L_k = P_i L_k P_i^2 = (P_i L_k P_i) P_i$$

$$(P_i L_k P_i) = \begin{pmatrix} \ddots & & & & & & \\ & 1 & & & & & \\ & -l_{k+1,k} & 1 & & & & \\ & \vdots & & \ddots & & & \\ & -l_{j_i,k} & & & 0 & & 1 \\ & \vdots & & & & \ddots & \\ & -l_{i_i,k} & & & 1 & & 0 \\ & \vdots & & & & & \ddots \end{pmatrix} \cdot P_i$$

$$= \begin{pmatrix} \ddots & & & & & & \\ & 1 & & & & & \\ & -l_{k+1,k} & 1 & & & & \\ & \vdots & & \ddots & & & \\ & -l_{j_i,k} & & & 1 & & \\ & \vdots & & & & \ddots & \\ & -l_{i_i,k} & & & & & 1 \\ & \vdots & & & & & \ddots \end{pmatrix} = L'_k$$

□

Setzt man nun

$$L'_k = P_{m-1} \cdots P_{k+1} L_k P_{k+1} \cdots P_{m-1},$$

so ist L'_k eine linke, untere Dreiecksmatrix und es gilt

$$\begin{aligned} & L_{m-1}P_{m-1}\cdots L_2P_2L_1P_1A &&= U \\ \Leftrightarrow \quad & \underbrace{(L'_{m-1}\cdots L'_2L'_1)}_{=:L^{-1}}\underbrace{(P_{m-1}\cdots P_2P_1)}_{=:P}A &&= U. \end{aligned}$$

Beispiel:

$$L'_3 = L_3, \quad L'_2 = P_3L_2P_3, \quad L'_1 = P_3P_2L_1P_2P_3$$

$$L'_3L'_2L'_1P_3P_2P_1 = L_3(P_3L_2\underbrace{P_3)(P_3}_{=I}P_2L_1\underbrace{P_2P_3)P_3P_2}_{=I}P_1 = L_3P_3L_2P_2L_1P_1$$

Damit erhalten wir die allgemeine LU-Faktorisierung

$$PA = LU.$$

Genauere Informationen zur **Stabilitätsanalyse** findet man in [3], Kapitel 22.

Algorithmus 6.2: LU-Faktorisierung mit Pivotisierung

$U = A, L = I, P = I$
for $k = 1 \to m-1$ **do**
  Wähle $i \geqslant k$ mit maximalem $|u_{ik}|$.
  Vertausche die Zeilen $u_{k,k:m}$ und $u_{i,k:m}$.
  Vertausche die Zeilen $l_{k,1:k-1}$ und $l_{i,1:k-1}$.
  Vertausche die Zeilen $p_{k,:}$ und $p_{i,:}$.
  for $j = k+1 \to m$ **do**
    $l_{j,k} = u_{j,k}/u_{k,k}$
    $u_{j,k:m} = u_{j,k:m} - l_{j,k}u_{k,k:m}$
  end for
end for

6.2 Cholesky-Faktorisierung

Die LU-Zerlegung zerlegt eine Matrix $A \in \mathbb{C}^{m\times m}$ in zwei verschiedene Matrizen L und U. Es liegt nahe zu fragen, ob zwischen diesen beiden Matrizen ein Zusammenhang

besteht.
Die Cholesky-Zerlegung zerlegt eine Matrix $A \in \mathbb{C}^{m\times m}$ in zwei Matrizen L und U, wobei gilt, dass $L^* = U$.

Definition 6.2: Cholesky-Zerlegung

Eine Zerlegung einer hermitschen positiv definiten Matrix $A \in \mathbb{C}^{m\times m}$ der Form

$$A = R^*R,$$

wobei R eine rechte obere Dreiecksmatrix ist, nennt man *Cholesky-Zerlegung*.

Im Folgenden wollen wir eine solche Zerlegung herleiten. Sei dazu $A \in \mathbb{C}^{m\times m}$ hermitesch (im rellen Fall symmetrisch) und positiv definit, d.h. $\forall x \neq 0 : x^*Ax > 0$.

Idee: symmetrische Gauß-Elimination

Sei A von der Form

$$A = \begin{pmatrix} 1 & \omega^* \\ \omega & K \end{pmatrix},$$

wobei ω ein $(m-1)$-dimensionaler $\mathbb{C}$-Vektor und K eine hermitesche $(m-1)\times(m-1)$-Matrix ist.
Nun zerlegen wir A wie bei der LU-Zerlegung, d.h. wir erhalten

$$\begin{pmatrix} 1 & \omega^* \\ \omega & K \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \omega & I \end{pmatrix} \begin{pmatrix} 1 & \omega^* \\ 0 & K-\omega\omega^* \end{pmatrix}$$

(Vielfache der ersten Zeile werden von den anderen Zeilen subtrahiert).

Anstatt mit der normalen Gauß-Elimination weiter zu machen, wird bei der Cholesky-Faktorisierung die Matrix symmetrisch gehalten, indem Nullen in der ersten Zeile eingeführt werden:

$$\begin{pmatrix} 1 & \omega^* \\ 0 & K-\omega\omega^* \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & K-\omega\omega^* \end{pmatrix} \underbrace{\begin{pmatrix} 1 & \omega^* \\ 0 & I \end{pmatrix}}_{\text{Adjungierte von } \begin{pmatrix} 1 & 0 \\ \omega & I \end{pmatrix}},$$

d.h.

$$\begin{pmatrix} 1 & \omega^* \\ \omega & K \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \omega & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & K-\omega\omega^* \end{pmatrix} \begin{pmatrix} 1 & \omega^* \\ 0 & I \end{pmatrix}.$$

Nun verallgemeineren wir: Sei A von der Form (ähnlich wie oben)

$$A = \begin{pmatrix} a_{11} & \omega^* \\ \omega & K \end{pmatrix}.$$

Da A positiv definit ist, muss a_{11} größer 0 sein.
Wir setzen nun $\alpha = \sqrt{a_{11}}$ und erhalten

$$A = \begin{pmatrix} \alpha & 0 \\ \frac{\omega}{\alpha} & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & K - \frac{\omega\omega^*}{a_{11}} \end{pmatrix} \begin{pmatrix} \alpha & \frac{\omega^*}{\alpha} \\ 0 & I \end{pmatrix} =: R_1^* A_1 R_1.$$

Rekursive Anwendung auf die (wieder hermitesche und positiv definite) Untermatrix $K - \frac{\omega\omega^*}{a_{11}}$ liefert schließlich

$$A = \underbrace{R_1^* R_2^* ... R_m^*}_{R^*} \underbrace{R_m ... R_2 R_1}_{R} = R^* R, \quad r_{jj} > 0,$$

wobei R eine obere Dreiecksmatrix ist.

Diese Überlegungen führen zu folgendem Algorithmus.

Algorithmus 6.3: Cholesky-Zerlegung

$R = A$
Setze Einträge von R unter der Diagonalen auf 0.
for $k = 1 \to m$ **do**
  for $j = k + 1 \to m$ **do**
    $R_{j,j:m} = R_{j,j:m} - R_{k,j:m}\overline{R}_{k,j}/R_{k,k}$
  end for
  $R_{k,k:m} = R_{k,k:m}/\sqrt{R_{k,k}}$
end for

Satz 6.3: Komplexität der Cholesky-Zerlegung

Die Cholesky-Zerlegung benötigt $\approx \frac{1}{3}m^3$ Operationen und hat somit eine Komplexität von $O(m^3)$.

Satz 6.4: Cholesky-Zerlegung

Jede hermitesche positiv definite Matrix $A \in \mathbb{C}^{m \times m}$ hat eine Cholesky-Zerlegung.

Bemerkung 6.3: Laufzeitvergleich von LU- und Cholesky-Zerlegung

Der Aufwand zur Berechnung einer Cholesky-Zerlegung ist um den Faktor 2 kleiner als der Aufwand einer LU-Zerlegung. Die Cholesky-Zerlegung existiert jedoch nur für hermitesche positiv-definite Matrizen, die LU-Zerlegung (mit Pivotisierung) existiert für jede quadratische Matrix.

Kapitel 7

Eigenwertprobleme

7.1 Eigenwerte und Eigenvektoren

Definition 7.1: Eigenwert und Eigenvektor

Seien $A \in \mathbb{C}^{m\times m}$ und $x \in \mathbb{C}^m \backslash \{0\}$. Dann heißt x *Eigenvektor* (EV) von A und $\lambda \in \mathbb{C}$ der dazugehörige *Eigenwert* (EW), falls

$$Ax = \lambda x.$$

Der dazugehörige *Eigenraum* ist gegeben durch $E_\lambda := \{x \in \mathbb{C}^m | Ax = \lambda x\}$.

Die Eigenwerte einer Matrix (oder einer linearen Abbildung) kann man mittels des *charakteristischen Polynoms* bestimmen:

$$p_A(z) = \det(A - zI).$$

Zur Erinnerung: $g_A(\lambda) = dim E_\lambda$ ist die *geometrische Multiplizität* (oder geometrische Vielfachheit) von λ. Die *algebraische Multiplizität* (oder algebraische Vielfachheit) a_A von λ ist die Potenz des Faktors $(z-\lambda)$ im charakteristischen Polynom. d.h. $a_A(\lambda_i) = l_i$ für $p_A = (z - \lambda_i)^{l_i}$ mit λ_i EW von A.

Beispiel (Charakteristisches Polynom):
Betrachte die Matrix

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

Das charakteristische Polynom von A ist:

$$p_A(z) = \det(A - zI) = (1 - z)^2.$$

Die Matrix hat folgende Vielfachheiten:

$$\begin{aligned} g_A(\lambda) &= 1, \\ a_A(\lambda) &= 2. \end{aligned}$$

7.2 Schur-Faktorisierung

Definition 7.2: Schur-Faktorisierung

Eine *Schur-Faktorisierung* einer Matrix $A \in \mathbb{C}^{m\times m}$ ist eine Faktorisierung der Form

$$A = QTQ^*,$$

wobei Q unitär und T eine obere Dreiecksmatrix ist.

Bemerkung 7.1:

A und T besitzen dasselbe charakteristische Polynom, dieselben Eigenwerte λ und die gleichen algebraischen und geometrischen Multiplizitäten:

$$\begin{aligned} p_A(z) &= \det(QTQ^* - zI) = \det(Q(T - zI)Q^*) \\ &= \det(Q)\det(T - zI)\det(Q^*) = \det(T - zI) = p_T(z) \end{aligned}$$

Damit müssen die Eigenwerte von A notwendigerweise auf der Diagonalen von T erscheinen ($\det(Q)\det(Q^*) = |\det(Q)|^2 = 1$).

Satz 7.1: Schur-Faktorisierung

Jede quadratische Matrix $A \in \mathbb{C}^{m\times m}$ besitzt eine Schur-Faktorisierung.

Beweis: Induktion über die Dimension m von A.

$\underline{m = 1}$: trivial.

Sei daher $m \geqslant 2$:

Sei x ein Eigenvektor von A mit EW λ. Es gilt:

$$Ax = \lambda x \quad \Leftrightarrow \quad A\frac{x}{\|x\|} = \frac{\lambda x}{\|x\|}.$$

x sei also normalisiert und bilde die erste Spalte einer unitären Matrix U:

$$U = [x, y].$$

Dann sind x und y orthogonal und es ist $x * \lambda x = \lambda \|x\|_2^2 = \lambda$. Es folgt:

$$\begin{aligned} U^*AU &= \begin{pmatrix} x, & y \end{pmatrix}^* A \begin{pmatrix} x, & y \end{pmatrix} \\ &= \begin{pmatrix} x, & y \end{pmatrix}^* \begin{pmatrix} \lambda x, & Ay \end{pmatrix} \\ &= \begin{pmatrix} \lambda & x^*Ay \\ 0 & y^*Ay \end{pmatrix} = \begin{pmatrix} \lambda & B \\ 0 & C \end{pmatrix} \end{aligned}$$

Nach Induktionsvoraussetzung existiert eine Schur-Faktorisierung VTV^* von C.

Nun setzen wir $Q = U \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix}$. Dann gilt (Schur-Faktorisierung):

$$\begin{aligned} \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix}^* U^*AU \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix} &= \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix}^* \underbrace{\begin{pmatrix} \lambda & B \\ 0 & C \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix}}_{\begin{pmatrix} \lambda & BV \\ 0 & CV \end{pmatrix}} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & V^* \end{pmatrix} \begin{pmatrix} \lambda & BV \\ 0 & CV \end{pmatrix} \\ &= \begin{pmatrix} \lambda & BV \\ 0 & V^*CV \end{pmatrix} \end{aligned}$$

□

Bemerkung 7.2: Frobenius Begleitmatrix eines Polynoms

Das charakteristische Polynom der $m \times m$ Matrix

$$M = \begin{pmatrix} 0 & \dots & 0 & -a_0 \\ 1 & \ddots & \vdots & \vdots \\ & \ddots & 0 & -a_{m-2} \\ 0 & & 1 & -a_{m-1} \end{pmatrix}$$

ist

$$p(z) = a_0 + a_1 z + ... + a_{m-1} z^{m-1} + z^m$$

Ist umgekehrt p ein beliebiges monisches Polynom vom Grad m mit den Koeffizienten $a_0, \cdots, a_{m-1}$, dann nennt man die Matrix M die **Frobenius Begleitmatrix** von p.

Zum Beweis betrachtet man $M' = M - zI$ und entwickeln $det(M')$ nach der letzten Spalte. Für $m = 4$ ergibt sich beispielsweise:

$$\begin{aligned}
\det(M') &= \det \begin{pmatrix} -z & 0 & 0 & -a_0 \\ 1 & -z & 0 & -a_1 \\ 0 & 1 & -z & -a_2 \\ 0 & 0 & 1 & -z - a_3 \end{pmatrix} \\
&= (-1)^m \left[(-1)(-a_0) \det \begin{pmatrix} 1 & -z & 0 \\ 0 & 1 & -z \\ 0 & 0 & 1 \end{pmatrix} \right. \\
&\quad + (-1)^2(-a_1) \det \begin{pmatrix} -z & 0 & 0 \\ 0 & 1 & -z \\ 0 & 0 & 1 \end{pmatrix} + (-1)^3(-a_2) \det \begin{pmatrix} -z & 0 & 0 \\ 1 & -z & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&\quad \left. + (-1)^4(-z - a_3) \det \begin{pmatrix} -z & 0 & 0 \\ 1 & -z & 0 \\ 0 & 1 & -z \end{pmatrix} \right] \\
&= (-1)^4(a_0 + a_1 z + a_2 z^2 + a_3 z^3 + z^4)
\end{aligned}$$

Problem der Eigenwertberechnung:

Da für jedes $m \geqslant 5$ ein Polynom $p(z)$ mit Grad m existiert, das rationale Koeffizienten besitzt und eine rationale Wurzel v (d.h. $(p(v) = 0)$ mit der Eigenschaft, dass v nicht als Term mit Radikalen, d.h. mit Additionen, Subtraktionen, Multiplikationen, Divisionen und k-ten Wurzeln aufgelöst werden kann (Theorem von Abel-Ruffini, Galoistheorie), können wir die exakten Wurzeln eines Polynoms nicht mit einer endlichen Zahl von Rechenschritten berechnen, d.h. wir sind auch nicht in der Lage, die entsprechenden Eigenwerte der Matrix exakt zu berechnen. Das bedeutet:

Jeder Eigenwertlöser muss iterativ sein!

Das beste, das wir erzielen können, ist eine hohe *Konvergenzrate*.

Bemerkung: Zur Erinnerung:

Eine Matrix A heißt *hermitesch*, wenn sie gleich ihrer Adjungierten A^* ist, also gleich ihrer komplex konjugierten Transponierten: $A = A^* = \bar{A}^T$

Bemerkung 7.3: Schur-Faktorisierung

Die meisten Algorithmen zur Eigenwertberechnung versuchen eine Schur-Faktorisierung zu bestimmen, indem eine Folge elementarer unitärer Ähnlichkeitstransformationen

$$X \mapsto Q_j^* X Q_j$$

durchgeführt wird, s.d.

$$Q_j^* ... Q_2^* Q_1^* A Q_1 Q_2 ... Q_j$$

gegen eine obere Dreiecksmatrix konvergiert.
Falls A hermitesch ist, dann ist $...Q_j^*...Q_2^*Q_1^*AQ_1Q_2...Q_j...$ ebenfalls hermitesch und daher diagonal. Man kann also dieselben Algorithmen anwenden.

7.3 Iterationsverfahren

Obwohl sich viele der folgenden Konzepte auf allgemeine Matrizen verallgemeinern lassen, beschränken wir uns im folgenden der Einfachheit halber auf relle symmetrische Matrizen, d.h.

$$A = A^T \in \mathbb{R}^{m \times m}, x \in \mathbb{R}^m, x^* = x^T \text{ und } ||x|| = \sqrt{x^T x}$$

Insbesondere besitzt A reelle Eigenwerte and eine vollständige Menge orthogonaler Eigenwerte.
Für den allgemeineren Fall veweisen wir auf [1], Kap. 25-27.

7.3.1 Rayleigh-Quotient

Bemerkung: (vgl. Kapitel 2) **Gesucht:** Verfahren zur Schätzung von Eigenwerten
Ansatz: Zu gegebenem $x \in \mathbb{R}^m \backslash \{0\}$ wird $\alpha \in \mathbb{R}$ gesucht, das einem Eigenwert am ähnlichsten ist, d.h. das den folgenden Ausdruck minimiert:

$$\|Ax - \alpha x\|_2 \xrightarrow{\alpha} \min$$

Dies ist ein $m \times 1$-Least-Squares-Problem der Form

$$x\alpha = Ax,$$

wobei x die Matrix, α der Vektor und Ax die rechte Seite ist. Es gilt:

$$\begin{aligned} \alpha \text{ Lösung des Problems} &\Leftrightarrow x^*x\alpha = x^*Ax \\ &\Leftrightarrow \alpha = \frac{x^*Ax}{x^*x} \end{aligned}$$

Damit ist $r(x) := \alpha$ eine natürliche Schätzung für einen Eigenwert, falls x kein exakter Eigenvektor ist.

Definition 7.3: Rayleigh-Quotient

Für einen Vektor $x \in \mathbb{R}^m$ ist der Rayleigh-Quotient

$$r(x) := \frac{x^T A x}{x^T x}. \tag{7.1}$$

Bemerkung 7.4:

Falls x ein Eigenvektor ist, dann gilt

$$r(x) = \frac{x^T A x}{x^T x} = \frac{\lambda x^T x}{x^T x} = \lambda$$

Wir betrachten nun den Gradienten von $r(x)$ (siehe [2] S. 54) da wir so eine Charakterisierung von Eigenvektoren erhalten können: **Bemerkung:** Untersuchen des Gradienten ermöglicht, das lokale Verhalten von $r(x)$ zu betrachten.

$$\begin{aligned}
\frac{\partial r(x)}{\partial x_j} &= \frac{f'g}{g^2} - \frac{g'f}{g^2} \qquad \text{Quotientenregel} \\
&= \frac{\frac{\partial}{\partial x_j}(x^T A x)}{x^T x} - \frac{(x^T A x)\frac{\partial}{\partial x_j}(x^T x)}{(x^T x)^2} \\
&= \frac{2(Ax)_j}{x^T x} - \frac{(x^T A x)(2x_j)}{(x^T x)^2} \\
&= \frac{2}{x^T x}(Ax - r(x)x)_j \\
\Rightarrow \nabla r(x) &= \frac{2}{x^T x}(Ax - r(x)x)
\end{aligned}$$

Bemerkung 7.5: Eigenwertapproximation durch Rayleigh-Quotient

1. Ist $\nabla r(x) = 0$ für $x \neq 0$, dann ist x ein Eigenvektor und $r(x)$ der entsprechende Eigenwert, d.h. die Eigenvektoren von A sind *stationäre Punkte* von $r(x)$ und die Eigenwerte von A die Werte von $r(x)$ an diesen Stellen.

2. Da $\nabla r(q_j) = 0$ für alle Eigenvektoren q_j, $j = 1, ..., m$, von A gilt, folgt aus der Taylorapproximation

$$r(x) = r(q_j) + \nabla r(q_j)(q_j - x) + O(\|q_j - x\|^2),$$

dass

$$r(x) - r(q_j) = O(\|q_j - x\|^2),$$

d.h. der Rayleigh-Koeffizient ist eine quadratisch genaue Approximation eines Eigenwertes.

7.3.2 Potenziteration (Power Iteration)

Grundidee: Sei A eine reelle symmetrische $n \times n$ Matrix mit n betragsmäßig verschiedenen (daher reellen) Eigenwerten λ_i, die wir folgt indiziert sind:

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| \geqslant 0.$$

Wir betrachte die Folge

$$\frac{x}{\|x\|}, \frac{Ax}{\|Ax\|}, \frac{A^2x}{\|A^2x\|}, \frac{A^3x}{\|A^3x\|}, \ldots$$

Dann ist der Grenzwert dieser Folge der Eigenvektor zum betragsmäßig größte Eigenwert.

Um das zu einzusehen sei q_i, $\|q_i\| = 1$, $i = 1, \cdots, n$ jeweils der zu λ_i gehörende Eigenqektor von A.

Dann kann jeder Vektor $v \in \mathbb{R}^n$ in diese Eigenbasis entwickelt werden:

$$v = \sum_{i=1}^{n} a_i q_i \text{ und es gilt } A^k v = \sum_{i=1}^{n} \lambda_i^k a_i q_i \quad \forall k >= 1.$$

Konkret ergibt sich

$$\begin{aligned}
v^{(0)} &= a_1 q_1 + a_2 q_2 + \ldots + a_n q_n \qquad q_i \text{ Eigenvektor} \\
v^{(k)} &= c_k A^k v^{(0)} \qquad\qquad\qquad \text{(wegen der Normierung)} \\
&= c_k (a_1 \lambda_1^k q_1 + a_2 \lambda_2^k q_2 + \ldots + a_m \lambda_m^k q_m) \\
&= c_k \lambda_1^k \left(a_1 q_1 + a_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k q_2 + \ldots + a_m \left(\frac{\lambda_m}{\lambda_1}\right)^k q_m \right),
\end{aligned}$$

d.h. für große k gilt

$$A^k v \approx \lambda_1^k a_1 q_1.$$

Verwenden wir den Rayleigh-Quotienten zur Schätzung des Eigenwerts von $v^{(k)}$ so erhalten wir folgenden Algorithmus zur Eigenvektorschätzung zum betragsmäßig größten Eigenwert:

Algorithmus 7.1: Potenziteration

Sei $v^{(0)}$ ein vektor mit $\|v^{(0)}\|_2 = 1$.
for k = 1,2,... **do**
$w = Av^{(k-1)}$
$v^{(k)} = w/\|w\|$ $(Normalisierung)$
$\lambda^{(k)} = (v^{(k)})^* Av^{(k)}$ $(Rayleigh - quotient)$
end for

Theorem 7.2: Konvergenz der Potenziteration

Angenommen $|\lambda_1| > |\lambda_2| \geqslant ... \geqslant |\lambda_m| \geqslant 0$ sind sortierte Eigenwerte, $q_1 \in E_{\lambda_1}$ mit $\|q_1\| = 1$ und $q_1^T v^{(0)} \neq 0$. Dann gilt

$$\|v^{(k)} - \pm q_1\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \qquad |\lambda^{(k)} - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right) \quad \text{für } k \to \infty$$

Das Vorzeichen bei $\pm$ wird hier in jedem Schritt so gewählt, dass das Ergebnis kleiner ist.

7.3.3 Inverse Iteration (Iteration der Potenzmethode)

Von der Potenziteration zur Potenzmethode:

Mit der Potenziteration kann der betragsmäßig größte Eigenwert berechnet werden. Wie berechnet man jedoch andere Eigenwerte?

Sei $\mu \in \mathbb{R}$, wobei μ kein Eigenwert von A ist. Dann besitzt $(A - \mu I)^{-1}$ dieselben Eigenvektoren wie A, denn:

$$Av = \lambda v \;\Leftrightarrow\; (A - \mu I)v = (\lambda - \mu)v \;\Leftrightarrow\; (\lambda - \mu)^{-1} v = (A - \mu I)^{-1} v$$

Die Eigenwerte von $(A - \mu I)^{-1}$ sind also $(\lambda_j - \mu)^{-1}$, wobei die $\{\lambda_j\}_j$ Eigenwerte von A sind. Ist nun μ bereits nahe eines Eigenwerts λ_J von A, dann ist $(\lambda_J - \mu)^{-1}$ größer als $(\lambda_j - \mu)^{-1}$ $\forall j \neq J$. Daher führt die Anwendung der Potenzmethode auf $(A - \mu I)^{-1}$ zu einer Konvergenz gegen q_J. Diese Idee heißt *inverse Iteration*.

Theorem 7.3: Potenzmethode (Inverse Iteration)

Sei λ_J der Eigenwert, der am nächsten zu μ liegt, λ_K der zweitnächste, d.h.

$$|\mu - \lambda_J| < |\mu - \lambda_K| \leqslant |\mu - \lambda_j| \quad \forall j \neq J.$$

Des Weiteren sei $q_J^T v^{(0)} \neq 0$. Dann gilt für die Iteration der Potenzmethode

$$\|v^{(k)} - (\pm q_J)\| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^k\right), \qquad |\lambda^{(k)} - \lambda_J| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^{2k}\right)$$

(für $k \to \infty$ und gleicher Bedeutung des $\pm$-Zeichens wie oben.)

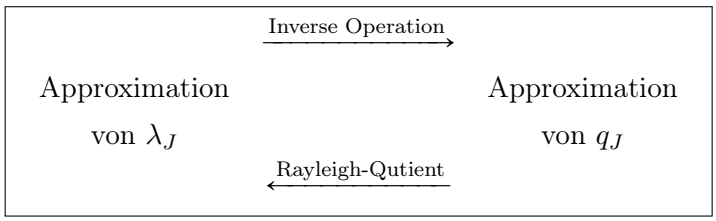
Beweis: Folgt aus der Konvergenz der Potenziteration indem man A durch $(A - \mu I)^{-1}$ ersetzt. □

7.3.4 Rayleigh-Quotient-Iteration

Bisher haben wir folgende Methoden betrachtet:

1. Eigenwert-Schätzung aus einer Eigenvektor-Schätzung (Rayleigh-Quotient)
2. Eigenvektor-Schätzung aus Eigenwert-Schätzung (Inverse Iteration)

Jetzt kombinieren wir die beiden Ideen zur *Rayleigh-Quotient-Iteration*:



Algorithmus 7.2: Rayleigh-Quotient-Iteration

Sei $v^{(0)}$ ein Vektor mit $\|v^{(0)}\|_2 = 1$.
$\lambda^{(0)} = \left(v^{(0)}\right)^* A v^{(0)}$ (Rayleigh-Quotient)
for k = 1,2,... **do**
  solve $(A - \lambda^{(k-1)} I) w = v^{(k-1)}$
  $v^{(k)} = w/\|w\|$ $(Normalisierung)$
  $\lambda^{(k)} = (v^{(k)})^* A v^{(k)}$ $(Rayleigh - Quotient)$

end for

Theorem 7.4: Konvergenz der Rayleigh-Quotient-Iteration

Die Rayleigh-Quotient-Iteration konvergiert gegen ein Eigenwert/Eigenvektor-Paar für alle Startvektoren $v^{(0)} \neq 0$. Falls sie konvergiert, ist die Konvergenz kubisch, d.h. falls λ_J ein Eigenwert von A und $v^{(0)}$ hinreichend dicht zum Eigenvektor q_J liegt, dann gilt

$$\|v^{(k+1)} - (\pm q_J)\| = O(\|v^{(k)} - (\pm q_J)\|^3)$$

und

$$|\lambda^{(k+1)} - \lambda_J| = O(|\lambda^{(k)} - \lambda_J)|^3)$$

(Hier ohne Beweis. Siehe [3], Kapitel 27.3.)

Tabelle 7.1: Aufwand der Verfahren im Vergleich

Power + Inverse Iteration	**Rayleigh-Quotient-Iteration**
• Konvergenz: Quadratisch	• Konvergenz: Kubisch
• Jeder Schritt: $O(m^2)$ Operationen	• Jeder Schritt: Lösen eines Gleichungssystems: $O(m^3)$ Operationen
• $\Rightarrow O(m^2)$ Aufwand	• $\Rightarrow O(m^3)$ Aufwand
	• Lösen des Gleichungssystems kann auf $O(m^2)$ reduziert werden (QR- oder LU-Zerlegung)

7.4 QR-Verfahren

Sei $A \in \mathbb{R}^{n\times n}$ eine quadratische Matrix und $(\mu_k)_{k\geqslant 0}$ eine Folge reeller Zahlen, sogenannte *'Shifts'*. Dann berechnet man im k-ten Iterationsschritt eine QR-Zerlegung (vgl. Kapitel

4)

$$A_k - \mu_k I \quad = \quad Q_k R_k, \tag{7.2}$$

wobei Q_k unitär und R_k eine reelle obere Dreiecksmatrix ist und setzt

$$A_{k+1} \quad := \quad R_k Q_k + \mu_k I.$$

Lemma 7.5: QR-Verfahren

Seien μ_k, Q_k, R_k wie oben definiert und sei $A_0 := A$. Dann gelten die folgenden Identitäten:

(a) $A_{k+1} = Q_k^* A_k Q_k$ (d.h. insbesondere, dass die Matrizen A_k dieselben Eigenwerte wie A besitzen)

(b) $A_{k+1} = (Q_0 Q_1 Q_2 \dots Q_k)^* A (Q_0 Q_1 Q_2 \dots Q_k)$

(c) $\prod_{j=0}^{k} (A - \mu_j I) = \underbrace{(Q_0 Q_1 Q_2 \dots Q_k)}_{Q} \underbrace{(R_k R_{k-1} \dots R_0)}_{R}$

Beweis:

(a) Für A_{k+1} gilt:

$$\begin{aligned} A_{k+1} &= R_k Q_k + \mu_k I \\ &= \underbrace{Q_k^* Q_k}_{id} R_k Q_k + \mu_k \underbrace{Q_k^* Q_k}_{id} \\ &= Q_k^* \underbrace{(Q_k R_k + \mu_k I)}_{A_k} Q_k \\ &= Q_k^* A_k Q_k \end{aligned}$$

(b) Folgt direkt aus (a) mittels vollständiger Induktion.

(c) Induktion über k:

$\underline{k = 0}$: $\underbrace{A}_{A_0} - \mu I = Q_0 R_0$

$\underline{k \to k+1}$:

$\underline{k \to k+1}$:

$$\begin{aligned} Q_{k+1} R_{k+1} &= A_{k+1} - \mu_{k+1} I \\ &\overset{(b)}{=} (Q_0 \dots Q_k)^* A (Q_0 \dots Q_k) - \mu_{k+1} (Q_0 \dots Q_k)^* (Q_0 \dots Q_k) \\ &= (Q_0 \dots Q_k)^* (A - \mu_{k+1} I)(Q_0 \dots Q_k) \end{aligned}$$

$$\Rightarrow Q_0 \dots Q_k Q_{k+1} R_{k+1} = (A - \mu_{k+1} I)(Q_0 \dots Q_k)$$
$$\Rightarrow Q_0 \dots Q_k Q_{k+1} R_{k+1} R_k \dots R_0 = (A - \mu_{k+1} I) \underbrace{(Q_0 \dots Q_k)(R_k \dots R_0)}_{\text{Ind.-Ann. } \prod_{j=0}^{k} (A - \mu_j I)}$$
$$= \prod_{j=0}^{k+1} (A - \mu_j I)$$

□

Satz 7.6: Konvergenz des QR-Verfahrens

Die Matrizen A_k werden im Verlauf der Iteration zu oberen Dreiecksmatrizen.

1. Wir betrachten zunächst den Fall ohne *Shifts*: $\mu_k = 0 \ \forall k \in \mathbb{N}_0$. Es ist

$$A_{k+1} = \underbrace{(Q_0 ... Q_k)}_{Q_k} \underbrace{(R_k ... R_0)}_{R_k} = A^{k+1} \tag{7.3}$$

eine QR-Zerlegung von A_{k+1}. Vergleicht man die jeweils erste Spalte dieser Matrizen gleichzeitig, so sieht man

$$\underbrace{A_{k+1} e_1}_{\text{1. Spalte}} = Q_k r_{11}^{(k)} e_1 = r_{11}^{(k)} q_1^{(k)}.$$

Nach der Potenzmethode (Algorithmus **??**) konvergiert $A^{k+1} e_1$ gegen den Eigenvektor zum dominanten Eigenwert. Nach (b) ist

$$\begin{aligned} A_{k+1} &= Q_k^* A Q_k \\ A_{k+1} e_1 &= Q_k^* \underbrace{A q_1^{(k)}}_{\lambda_1 q_1^{(k)}} \approx \lambda_1 Q_k^* q_1^{(k)} \overset{\text{ausmultiplizieren}}{=} \lambda_1 e_1 \end{aligned}$$

Damit hat A_{k+1} in etwa folgende Gestalt: $A_{k+1} \approx \left(\begin{array}{c|c} \lambda_1 & \dots \\ 0 & \\ \vdots & \\ 0 & \dots \end{array}\right)$.

2. Aus Gleichung (7.3) folgt für eine invertierbare Matrix A wegen der Orthogonalität von Q_k

$$\begin{aligned} Q_k^* &= R_k (A^{k+1})^{-1} \\ q_n^{(k)} = e_n^* Q_k^* &= e_n^* R_k (A^{k+1})^{-1} = r_{nn}^{(k)} e_n^* (A^{k+1})^{-1}. \end{aligned}$$

Der Vektor $q_n^{(k)}$ (letzte Spalte von Q^k) ist also das mit $r_{nn}^{(k)}$ multiplizierte Ergebnis der inversen Iteration, also eine Näherung des linken Eigenvektors zum betragskleinsten Eigenwert λ_n von A. Daher folgt aus (b)

$$e_n^* A_{k+1} = e_n^* Q_k^* A Q_k = q_n^{(k)} A Q_k \approx \lambda_n q_n^{(k)*} Q_k = \lambda_n e_n$$

wobei $A_{k+1} \approx \begin{pmatrix} \lambda_1 & \dots & \dots & \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \lambda_n \end{pmatrix}$.

Wir vermuten daher, dass A_k für $k \to \infty$ gegen eine obere Dreiecksmatrix konvergiert.

7.4.1 Beschleunigung des Verfahrens: Rayleigh-Quotienten-Iteration

3. Nach der vorangehenden Argumentation ist e_n ein Näherungsvektor der letzten Spalte von A_k.

Bilden wir also den Rayleigh-Quotienten $\mu_k = e_n^* A_k e_n$ (rechtes unteres Element von A_k) und führen einen Schritt der inversen Iteration bzgl. des linken Eigenvektors aus, so ergibt dies

$$\underbrace{e_n^* \underbrace{(\underbrace{A_k - \mu_k I}_{Q_k R_k})^{-1}}_{R_k^{-1} \underbrace{Q_k^{-1}}_{Q_k^*}}}_{} = e_n^* R_k^{-1} Q_k^* = \frac{1}{r_{nn}^{(k)}} e_n^* Q_k^* = \frac{1}{r_{nn}^{(k)}} q_n^{(k)*}, \tag{7.4}$$

wobei $r_{nn}^{(k)}$ das rechte untere Element von R_k und $q_n^{(k)}$ die hinterste Spalte von Q_k bezeichnet.

(Beachte: R_k^{-1} ist wieder eine Dreiecksmatrix, deren Diagonaleinträge die Kehrwerte der entsprechenden Diagonaleinträge von R_k sind.)

Mit anderen Worten: Ein Schritt der Rayleigh-Quotienten-Iteration ergibt gerade die hinterste Spalte von Q_k als neue Näherung an den linken Eigenvektor von A_k zu λ_n.

Darüber hinaus erkennt man aus Lemma 7.5 Teil (a) sofort, dass das rechte untere Element von A_{k+1} der zugehöhrige Rayleigh-Quotient, also der nächste *Shift* μ_{k+1} aus der Rayleigh-Quotienten-Iteration ist:

$$\mu_{k+1} = q_n^{(k)*} A_k q_n^{(k)} = e_n^* \underbrace{Q_k^* A_k Q_k}_{A_{k+1}} e_n = e_n^* A_{k+1} e_n \tag{7.5}$$

Wählt man also als *Shift* μ_k in Gleichung (7.2) das (n,n)-te Element von A_k, dann darf man wie bei der Rayleigh-Quotienten-Iteration quadratische oder gar kubische

Konvergenz dieser Elemente gegen den kleinsten Eigenwert λ_n von A erwarten.
Shifts dienen also der Konvergenz-Beschleunigung des QR-Verfahrens.

7.4.2 Zusammenfassung: QR-Verfahren mit und ohne Shifts

Algorithmus 7.3: QR-Verfahren ohne Shifts

$A^{(0)} = A$
for $k = 1, 2, \ldots$ **do**
  $Q^{(k)}R^{(k)} = A^{(k-1)}$
  $A^{(k)} = R^{(k)}Q^{(k)}$
end for

Die Matrix $A \in \mathbb{C}^{m\times m}$ wird zunächst mittels QR-Faktorisierung in Q und R zerlegt. Dann werden R und Q in umgekehrter Reihenfolge multipliziert.

Algorithmus 7.4: QR-Verfahren mit einfachen Shifts

$A^{(0)} = A$
for $k = 1, 2, \ldots$ **do**
  Wähle einen Shift $\mu^{(k)}$
  $Q^{(k)}R^{(k)} = A^{(k-1)} - \mu^{(k)}I$
  $A^{(k)} = R^{(k)}Q^{(k)} + \mu^{(k)}I$
end for

Statt die QR-Zerlegung von A zu berechnen, wird nun eine QR-Zerlegung von $A - \mu^{(k)}I$ berechnet.

Algorithmus 7.5: Simultane Iteration

Wähle $\bar{Q}^{(0)} = I$
for $k = 1, 2, \ldots$ **do**
  $Z = A\bar{Q}^{(k-1)}$
  $Z = \bar{Q}^{(k)}R^{(k)}$
  $A^{(k)} = (\bar{Q}^{(k)})^T A\bar{Q}^{(k)}$
end for

Algorithmus 7.6: QR-Verfahren ohne Shifts

$A^{(0)} = A$
for $k = 1, 2, \ldots$ **do**
  $A^{(k-1)} = Q^{(k)}R^{(k)}$
  $A^{(k)} = R^{(k)}Q^{(k)}$
  $\bar{Q}^{(k)} = Q^{(1)}Q^{(2)}\cdots Q^{(k)}$
end for

Definiert man sich für beide Algorithmen eine weitere Matrix $\bar{R}^{(k)} := R^{(k)}R^{(k-1)}\cdots R^{(1)}$, so gilt folgender Satz:

Satz 7.7: Simultane Iteration und QR-Verfahren ohne Shifts

Die beiden obigen Algorithmen erzeugen eine identische Folge von Matrizen $\bar{R}^{(k)}, \bar{Q}^{(k)}, A^{(k)}$ mit

$$A^k = \bar{Q}^{(k)}\bar{R}^{(k)} \text{ mit der Projektion } A^{(k)} = (\bar{Q}^{(k)})^T A \bar{Q}^{(k)}.$$

(Hier ohne Beweis. Siehe [3], Kapitel 28.3.)

7.4.3 Hessenberg: Zweistufiges Verfahren

Für beliebige Matrizen $A \in \mathbb{K}^{n\times n}$ ist das QR-Verfahren sehr aufwendig, denn jede Iteration benötigt $O(n^3)$ Operationen. Daher wird die Matix A zunächst auf eine Obere Hessenberg-Form transformiert.

Definition 7.4: (Obere) Hessenbergmatrix

Eine (obere) Hessenbergmatrix ist eine quadratische Matrix $H \in \mathbb{C}^{n\times n}$, deren Einträge unterhalb der ersten Nebendiagonalen gleich Null sind:

$$\begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \dots & h_{1,n} \\ h_{2,1} & h_{2,2} & h_{2,3} & \dots & h_{2,n} \\ 0 & h_{3,2} & h_{3,3} & \dots & h_{3,n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & h_{n,n-1} & h_{n,n} \end{pmatrix}$$

Das zweistufige Hessenberg-Verfahren:

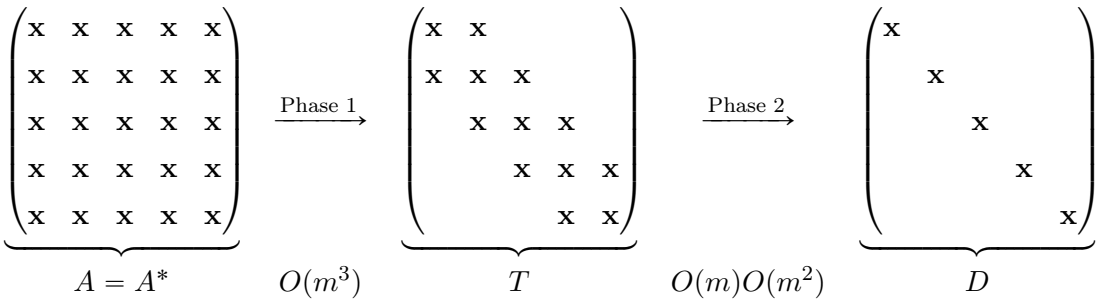
1. Berechnung einer oberen Hessenbergmatrix ($O(m^3)$ Operationen)
2. Iteration zur Berechnung einer oberen Dreiecksmatrix. ($O(m)$ Iterationen, um Maschinengenauigkeit zu erreichen, jede Iteration mit $O(m^2)$ Aufwand.)

Das Verfahren lässt sich schematisch folgendermaßen darstellen:

$$\underbrace{\begin{pmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{pmatrix}}_{A \neq A^*} \xrightarrow{\text{Phase 1}} \underbrace{\begin{pmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \end{pmatrix}}_{H} \xrightarrow{\text{Phase 2}} \underbrace{\begin{pmatrix} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \\ & & & & x \end{pmatrix}}_{T}$$

$$A \neq A^* \qquad O(m^3) \qquad H \qquad O(m)O(m^2) \qquad T$$

Falls A hermitesch ist, so ist die Hessenbergmatrix *tridiagonal*:



Die Hessenberg-Form lässt sich analog zur Dreiecksform in der QR-Zerlegung durch Householder-Spiegelungen herstellen (vgl. Algorithmus 4.5). Die k-te Spiegelung wird dabei so modifiziert, dass sie das k-te Diagonalelement unberührt lässt. Um eine Ähnlichkeitstransformation zu erhalten, wird die unitäre Spiegelungsmatrix anders als bei der QR-Zerlegung sowohl von links als auch von rechts angewandt. Daraus resultiert folgender Algorithmus:

Algorithmus 7.7: Householder-Reduzierung auf Hessenberg-Form

for $k = 1 \to m-2$ **do**

$x = A_{k+1:m,k}$

$v_k = \operatorname{sign}(x_1)\|x\|_2 e_1 + x$

$v_k = \dfrac{v_k}{\|v_k\|_2}$

$A_{k+1:m,k:m} = A_{k+1:m,k:m} - 2v_k v_k^* A_{k+1:m,k:m}$

$A_{1:m,k+1:m} = A_{1:m,k+1:m} - 2A_{1:m,k+1:m} v_k v_k^*$

end for

Satz 7.8: Komplexität des Hessenbergverfahren

Das Hessenbergverfahren benötigt $\frac{10}{3}m^3$ Operationen.
Die Komplexitätsklasse des Verfahrens ist also $O(m^3)$

Satz 7.9: Stabilität des Hessenbergverfahren

Die Berechnung der Hessenbergform einer Matrix ist wie die Berechnung einer QR-Faktorisierung rückwärtsstabil.

7.4.4 Berechnung einer Singulärwertzerlegung (SVD)

Die SVD $A = U\Sigma V^*$ einer $(m \times n)$-Matrix A, $m \geqslant n$, lässt sich aus einer Eigenwertzerlegung von A^*A berechnen:

$$A^*A = V\Sigma^*\Sigma V^* \tag{7.6}$$

Man kann also die SVD einer Matrix wie folgt bestimmen:

1. Berechne A^*A.
2. Berechne die Eigenwertzerlegung von $A^*A = V\Lambda V^*$.
3. Sei Σ die $(m \times n)$ nicht-diagonale Wurzel von Λ.
4. Löse das LGS $U\Sigma = AV$ für eine unitäre Matrix U (z.B. durch eine QR-Faktorisierung).

Problem: Instabilität des Algorithmus:

1. Falls A^*A und δB gestört werden, so sind Änderungen der Eigenwerte durch die 2-Norm von δB beschränkt:

$$|\lambda_k(A^*A + \delta B) - \lambda_k(A^*A)| \leqslant \|\delta B\|_2.$$

2. Ähnliches gilt für die Singulärwerte:

$$|\sigma_k(A + \delta A) - \sigma_k A| \leqslant \|\delta A\|_2.$$

Ein rückwärts-stabiler Algorithmus zur Berechnung der Singulärwerte $\tilde{\sigma}_k$ erfüllt **Bemerkung:** ε_M steht für $\varepsilon_{machine}$ (siehe Kapitel 5: Floating Point Arithmetik):

$$\begin{aligned} \tilde{\sigma}_k &= \sigma_k(A + \delta A), \qquad \frac{\|\delta A\|}{\|A\|} = O(\varepsilon_M) \\ \Rightarrow \quad |\tilde{\sigma}_k - \sigma_k| &= O(\varepsilon_M \cdot \|A\|) \end{aligned}$$

Es gilt für die Berechnung von $\lambda_k(A^*A)$:

$$\begin{aligned} |\tilde{\lambda}_k - \lambda_k| = \quad & O(\varepsilon_M \cdot \|A^*A\|) \quad = O(\varepsilon_M \cdot \|A\|^2) \\ \Rightarrow \qquad |\tilde{\sigma}_k - \sigma_k| \quad & = O\left(\tfrac{|\tilde{\lambda}_k - \lambda_k|}{\sqrt{\lambda_k}}\right) = \quad O\left(\varepsilon_M \cdot \frac{\|A\|^2}{\sigma_k}\right) \end{aligned}$$

Dies stellt kein Problem dar für dominante Singulärwerte mit $\sigma_1 \approx \|A\|$, ist jedoch problematisch für $\sigma_k \ll \|A\|$.

Angenommen, A ist quadratisch $(m = n)$. Betrachte die hermitesche Matrix

$$H = \begin{pmatrix} 0 & A^* \\ A & 0 \end{pmatrix}.$$

Wegen $A = U\Sigma V^* \quad \Leftrightarrow \quad AV = U\Sigma \quad \Leftrightarrow \quad A^*U = V\Sigma^* = V\Sigma$ erhält man folgende Eigenwertzerlegung von H:

$$\begin{pmatrix} 0 & A^* \\ A & 0 \end{pmatrix} \begin{pmatrix} V & V \\ U & -U \end{pmatrix} = \begin{pmatrix} V & V \\ U & -U \end{pmatrix} \begin{pmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{pmatrix}.$$

D.h. die Singulärwerte von A sind die Absolutwerte der Eigenwerte von H und die Singulärvektoren von A können aus den Eigenvektoren von H entnommen werden.
Im Gegensatz zur Berechnung der Eigenwertzerlegung von A^*A oder AA^* ist dieser Ansatz rückwärts-stabil. Die Standard-Algorithmen zur Berechnung der SVD basieren daher auf dieser Idee.
(Für Beispiele siehe [3], Kapitel 31.)

Teil II

Numerik in der Analysis

Kapitel 8

Differenzierbare Funktionen

8.1 Differenzierbare Funktionen

Vorbemerkung: $f : \mathbb{R} \longrightarrow \mathbb{R}$ heißt *differenzierbar* an einer Stelle a, falls der Grenzwert

$$\lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$$

existiert. Gleichwertig damit ist die Existenz einer (von a abhängigen) linearen Abbildung $L : \mathbb{R} \longrightarrow \mathbb{C}$, für die gilt **Bemerkung:** Prinzip der Approximation von Zuwächsen durch lineare Abbildungen

$$\lim_{h \to 0} \frac{f(a+h) - f(a) - Lh}{\|h\|} = 0.$$

Dabei ist dann $Lh = f'(a)h$. Das heißt aber, dass der Zuwachs $f(a+h) - f(a)$ durch Lh so gut approximiert wird, dass der Fehler schneller als $\|h\|$ gegen 0 geht.

Definition 8.1: differenzierbar

Eine Funktion $f : U \longrightarrow \mathbb{C}$ auf einer offenen Menge $U \subset \mathbb{R}^n$ heißt *differenzierbar im Punkt* $a \in U$, wenn es eine lineare Abbildung $L : \mathbb{R}^n \longrightarrow \mathbb{C}$ gibt derart, dass

$$\lim_{h \to 0} \frac{f(a+h) - f(a) - Lh}{\|h\|} = 0. \tag{8.1}$$

Die Funktion f heißt *differenzierbar auf* U, wenn sie in jedem Punkt $x \in U$ differenzierbar ist.

Bemerkung 8.1:

In (8.1) ist es gleichgültig, welche Norm verwendet wird, da alle Normen auf $\mathbb{R}^n$ zueinander äquivalent sind.

Oft formuliert man (8.1) anhand des durch

$$f(a+h) - f(a) = Lh + R(h) \tag{8.2}$$

definierten Restes $R(h)$; sie lautet dann

$$\lim_{h \to 0} \frac{R(h)}{\|h\|} = 0. \tag{8.3}$$

Lemma 8.1: Eindeutigkeit des Differentials

Die Gleichung (8.1) wird von höchstens einer linearen Abbildung erfüllt.

Beweis: Ist L^* eine weitere lineare Abbildung, so gilt für jeden Vektor v mit $\|v\| = 1$

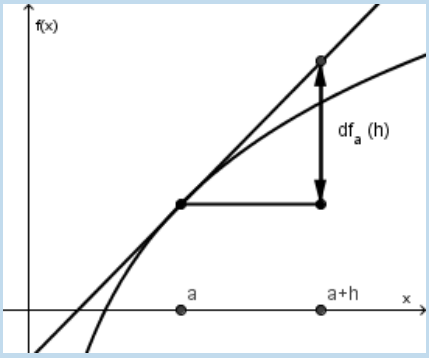
$$\lim_{t \searrow 0} \frac{(f(a+tv) - f(a) - Ltv) - (f(a+tv) - f(a) - L^*tv)}{\|tv\|}$$
$$= \lim_{t \searrow 0} \frac{(L^* - L)(tv)}{|t|} = (L^* - L)(v) = 0$$

Da die Menge der Einheitsvektoren den $\mathbb{R}^n$ aufspannt, folgt $L = L^*$.

□

Bemerkung 8.2: Differential

Die eindeutig bestimmte lineare Abbildung L heißt *Differential* oder *Linearisierung* von f im Punkt a und wird mit $df(a)$ oder dfa bezeichnet (siehe Abbildung). In alten Büchern wird das Differential auch als *totales Differential* bezeichnet.



Sei $\{e_1, ..., e_n\}$ die Standardbasis des $\mathbb{R}^n$. Wegen der Linearität von $df(a)$ gilt dann für jeden Vektor $h = (h_1, ..., h_n)^t \in \mathbb{R}^n$

$$df(a)h = \sum_{\nu=1}^{n} (df(a)e_\nu)h_\nu. \tag{8.4}$$

Dann ergibt sich mit nachfolgender Definition

$$df(a)h = f'(a)h. \tag{8.5}$$

Definition 8.2: Ableitung

Den Zeilenvektor

$$f'(a) := (df(a)e_1, ..., df(a)e_n) \tag{8.6}$$

nennen wir *Ableitung von f in a.*

Die affin-lineare Funktion

$$Tf(x;a) := f(a) + f'(a)(x-a) \tag{8.7}$$

heißt *lineare Approximation* von f in a und bei reellem f heißt deren Graph

$$\{(x, x_{n+1}) \in \mathbb{R}^{n+1} | x_{n+1} = Tf(x;a)\} \tag{8.8}$$

die *Tangentialhyperebene* an den Graphen von f in $(a, f(a))$.
Bemerkung: Zur Erinnerung: Eine Funktion $f : D \longrightarrow \mathbb{R}$ heißt *stetig* im Punkt $a \in D$, falls $\lim\limits_{x \to a} f(x) = f(a)$.

Satz 8.2: Stetigkeit differenzierbarer Funktionen

Eine in a differenzierbare Funktion ist auch in a stetig.

(Zum Beweis: In (8.2) gilt $Lh \to 0$ und $R(h) \to 0$ für $h \to 0$.)

Beispiel (1-zeilige Matrix):
Sei $f(x) := Ax + b$, A eine $1-$zeilige Matrix und $b \in \mathbb{C}$. Die durch $Lh := Ah$ definierte lineare Abbildung erfüllt (8.1):

$$\lim_{h \to 0} \frac{A(x+h) + b - (Ax+b) - Ah}{\|h\|} = 0 \qquad \forall x \in \mathbb{R}^n.$$

f ist also in jedem Punkt a differenzierbar und es gilt

$$\begin{aligned} df_a h &= Ah \\ f'(a) &= A. \end{aligned}$$

Beispiel (symmetrische $(n \times n)$-Matrix):
Sei $f(x) := x^t Ax$, $A = (a_{ik})$ eine symmetrische $(n \times n)$-Matrix. Dann gilt:

$$f(a+h) - f(a) = 2a^t Ah + h^t Ah.$$

$Lh := 2a^t Ah$ definiert eine lineare Abbildung mit $R(h) = h^t Ah$.
Sei $\sigma := \sum_{i,k=1}^{n} |a_{ik}|$. Dann gilt $|R(h)| = |h^t Ah| \leqslant \sigma \|h\|_\infty^2$ und damit $\lim_{h\to 0} \frac{R(h)}{\|h\|_\infty} = 0$.
f ist also in jedem Punkt a differenzierbar und es gilt

$$df(a)h = 2a^t Ah, \qquad f'(a) = 2a^t A.$$

8.1.1 Darstellung des Differentials durch Richtungsableitungen

Sei f eine differenzierbare Funktion. Die Werte $df(a)h$, $h \in \mathbb{R}^n$ sollen mit Hilfe von *Richtungsableitungen* ermittelt werden.
Für alle $t \in \mathbb{R}$ mit hinreichend kleinem Betrag gilt zunächst

$$f(a+th) = f(a) + df(a)th + R(th).$$

Da R die Bedingung (8.3) erfüllt, gilt:

$$df(a)h = \lim_{t\to 0} \frac{f(a+th) - f(a)}{t}. \tag{8.9}$$

Definition 8.3: Partielle Ableitungen

Sei $f : U \longrightarrow \mathbb{C}$ eine (nicht notwendig differenzierbare) Funktion in einer Umgebung U von a. Dann versteht man unter der Ableitung von f im Punkt a in Richtung des Vektors $h \in \mathbb{R}^n$ im Existenzfall den Grenzwert

$$D_h f(a) = \lim_{t\to 0} \frac{f(a+th) - f(a)}{t}.$$

Die Ableitungen in den Richtungen $e_1, ..., e_n$ der Standardbasis heißen *partielle Ableitungen* von f und f heißt *partiell differenzierbar* in a, wenn alle partiellen Ableitungen $D_{e_1} f(a), ..., D_{e_n} f(a)$ existieren.

Weitere Bezeichnungen für die partiellen Ableitungen sind:

$$D_{e_1} f(a) = \partial_1 f(a) = \frac{\partial f}{\partial x_1}(a) = f_{x_1}(a)$$

Satz 8.3: Richtungsableitungen

Eine in a differenzierbare Funktion f hat dort Richtungsableitungen in jeder Richtung. Sie ist dort insbesondere partiell differenzierbar. Ihr Differential in a hat für jeden Vektor $h = (h_1, ..., h_n)^t \in \mathbb{R}^n$ den Wert

$$df(a)h = f'(a)h = \partial_h f(a) = \sum_{\nu=1}^{n} \partial_\nu f(a) h_\nu \tag{8.10}$$

und ihre Ableitung $f'(a)$ ist die $1-$zeilige Matrix

$$f'(a) = (\partial_1 f(a), ..., \partial_n f(a))$$

Beweis: Die Existenz ist mit der Herleitung von (8.9) gezeigt. Die Formeln sind wegen $df(a)e_\nu = \partial_\nu f(a)$ identisch mit (8.4), (8.6) und (8.5).
Berechnung der partiellen Ableitungen:
Die Definition $\partial_\nu f(a) = \lim_{t\to 0} \frac{f(a+te_\nu)-f(a)}{t}$ mit $a = (a_1, ..., a_n)^t$ läuft darauf hinaus, in $f(x_1, ..., x_n)$ alle Variablen x_k bis auf die ν-te konstant $= a_k$ zu setzen und die dann nur von x_ν abhängige Funktion als Funktion *einer* Variablen zu differenzieren. □

Beispiel ($f(x, y) = x^2 + y^2$):

Die Richtungsableitungen der Funktion $f(x, y) = x^2 + y^2$ sind $\partial f_x(a, b) = 2a$ und $\partial f_y(a, b) = 2b$.

8.1.2 Hauptkriterium für Differenzierbarkeit

Um eine Funktion f auf Differenzierbarkeit in a zu untersuchen, klärt man zunächst, ob sie partiell differenzierbar ist. Im positiven Fall prüft man weiter, ob die einzige als Differential in Frage kommende lineare Abbildung

$$L : \mathbb{R}^n \longrightarrow \mathbb{C},$$
$$Lh = \sum_{\nu=1}^{n} \partial_\nu f(a) h$$

die Bedingung (8.1) erfüllt.

Bemerkung 8.3: Hinreichende Bedingung

Die bloße Existenz der partiellen Ableitungen impliziert i.a. nicht die Differenzierbarkeit.
Betrachte die Funktion $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$, definiert durch

$$f(x,y) = \begin{cases} 0, & \text{für } (x,y) = (0,0) \\ \dfrac{xy}{x^2+y^2}, & \text{sonst} \end{cases}$$

f ist im Nullpunkt nicht stetig, also erst recht nicht differenzierbar.
Speziell in $(0,0)$ hat f wegen $f(x,0) = 0$ und $f(0,y) = 0$ die partiellen Ableitungen $\partial_x f(0,0) = 0$ und $\partial_y f(0,0) = 0$. In den Punkten $(0,y)$, $y \neq 0$ gilt $\partial_x f(0,y) = 1/y$. $\partial_x f$ ist im Nullpunkt also unstetig, $\partial_y f$ ebenso.

Bemerkung 8.4:

Auch die Existenz aller Richtungsableitungen hat nicht die Differenzierbarkeit zur Folge.
Betrachte dazu die Funktion $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$, definiert durch

$$f(x,y) = \begin{cases} 0, & \text{für } (x,y) = (0,0) \\ \dfrac{x^2y}{x^2+y^2}, & \text{sonst.} \end{cases}$$

Es ist

$$f(tx,ty) = \frac{t^2x^2ty}{t^2x^2+t^2y^2} = tf(x,y).$$

Diese Geraden haben im Nullpunkt Ableitungen in jede Richtung $h = (h_1,h_2)$:

$$\partial_h f(0,0) = \lim_{t\to 0} \frac{f(th_1,th_2) - f(0,0)}{t} = f(h_1,h_2).$$

Insbesondere sind die partiellen Ableitungen $\partial_x f(0,0) = 0$ und $\partial_y f(0,0) = 0$. Als Differential kommt also höchstens $L = (0,0)$ in Frage. Damit ist (8.1) aber nicht erfüllt, da $\forall (h_1,h_1)$:

$$\frac{f(h_1,h_1) - f(0,0) - L(h_1,h_1)}{\|(h_1,h_1)\|_\infty} = \frac{h_1^3}{2h_1^2|h_1|} = \pm\frac{1}{2} \neq 0$$

gilt. Folglich ist f im Nullpunkt nicht differenzierbar. Man prüft leicht nach, dass die partielle Ableitung im Nullpunkt unstetig ist. (In den Punkten $(0,y)$, $y \neq 0$ gilt

$\frac{\partial f}{\partial x}(0,y) = \frac{1}{y}$. $\frac{\partial f}{\partial x}$ ist im Nullpunkt also unstetig.)

Theorem 8.4: Differenzierbarkeitskriterium

Existieren in einer Umgebung U von $a \in \mathbb{R}^n$ alle partiellen Ableitungen $\partial_1 f, ..., \partial_n f$ und sind diese im Punkt a stetig, so ist f in a differenzierbar.

Beweis: Wir dürfen f als reell voraussetzen, da ein komplexes f genau dann differenzierbar ist, wenn Ref und Imf differenzierbar sind.
Wir zeigen dann, dass die Linearform $L : \mathbb{R}^n \longrightarrow \mathbb{R}$ mit $Lh = \sum_{\nu=1}^{n} \partial_\nu f(a) h_\nu$ die Bedingung (8.1) erfüllt.
Sei Q ein offener achsenparalleler Quader in U mit $a \in Q$. Jeder Punkt $a + h \in Q$ kann mit a durch stückweise achsenparallelen Streckung in Q verbunden werden. Man setze dazu $a_0 := a$, $a_\nu = a_{\nu-1} + h_\nu e_\nu$, $\nu = 1, ..., n$. Insbesondere ist dann $a_n = a + h$ und

$$f(a+h) - f(a) = \sum_{\nu=1}^{n} \left(f(a_\nu) - f(a_{\nu-1}) \right).$$

Die Differenzen der Summe werden gemäß dem Mittelwertsatz der Differenzialrechnung
Bemerkung: MWS: $f : [a,b] \mapsto \mathbb{R}$, $a < b$, stetig auf $[a,b]$ und diffbar in (a,b)
$\Rightarrow \exists \xi \in (a,b)$:
$f'(\xi) = \frac{f(b)-f(a)}{b-a}$ umgeformt. Betrachte dazu die Funktionen $\varphi_\nu : [0, h_\nu] \longrightarrow \mathbb{R}$, $\varphi_\nu(t) := f(a_{\nu-1} + te_\nu)$. Mit diesen Funktionen gilt

$$f(a_\nu) - f(a_{\nu-1}) = \varphi_\nu(h_\nu) - \varphi_\nu(0).$$

Da f partiell differenzierbar ist, sind die Funktionen φ_ν differenzierbar und es gilt

$$\varphi_\nu'(t) = \partial_\nu f(a_{\nu-1} + te_\nu).$$

Nach dem Mittelwertsatz gibt es ein $\tau_\nu \in [0, h_\nu]$, sodass $\varphi_\nu(h_\nu) - \varphi_\nu(0) = h_\nu \varphi'(\tau_\nu)$. Mit $\xi_\nu := a_{\nu-1} + \tau_\nu e_\nu$ folgt nun $f(a_\nu) - f(a_{\nu-1}) = h_\nu \partial_\nu f(\xi_\nu)$. Damit ergibt sich

$$f(a+h) - f(a) - Lh = \sum_{\nu=1}^{n} (\partial_\nu f(\xi_\nu) - \partial_\nu f(a)) h_\nu$$

und weiter

$$|f(a+h) - f(a) - Lh| \leqslant \|h\|_\infty \sum_{\nu=1}^{n} |\partial_\nu f(\xi_\nu) - \partial_\nu f(a)|.$$

Für $h \to 0$ gilt $\xi_\nu \to a$, $\nu = 1, ..., n$ und wegen der Stetigkeit der partiellen Ableitungen in a erhält man also

$$\lim_{h \to 0} \frac{f(a+h) - f(a) - Lh}{\|h\|_\infty} = \lim_{h \to 0} \sum_{\nu=1}^{n} |\partial_\nu f(\xi_\nu) - \partial_\nu f(a)| = 0.$$

□

Beispiel (Differentiation rotationssymmetrischer Funktionen):
Es sei $F : I \longrightarrow \mathbb{C}$ eine Funktion auf einem Intervall $I \subset [o, \infty)$.
Mit F erhält man auf der Kugelschale $K(I) := \left\{ x \in \mathbb{R}^n \;\middle|\; \|x\|_2 = \sqrt{\sum_{\nu=1}^{n} x_\nu^2} \in I \right\}$ eine Funktion f durch $f(x) := F(\|x\|_2)$.
Es sei nun I offen und F stetig differenzierbar. Damit ist auch $K(I)$ offen und f hat an jeder Stelle $x \in K(I)$, $x \neq 0$, die partiellen Ableitungen

$$\partial_\nu f(x) = F'(\|x\|_2) \cdot \frac{x_\nu}{\|x\|_2}, \qquad \nu = 1, ..., n.$$

Diese sind offensichtlich stetig. Somit ist f an jeder von 0 verschiedenen Stelle $x \in K(I)$ differenzierbar und hat dort die Ableitung

$$f'(x) = \frac{F'(\|x\|_2)}{\|x\|_2} \cdot x^t \tag{8.11}$$

Definition 8.4: stetig differenzierbar

Eine differenzierbare Funktion $f : U \longrightarrow \mathbb{C}$ auf einer offenen Menge $U \subset \mathbb{R}^n$ heißt *stetig differenzierbar auf* U, wenn $df : U \longrightarrow L(\mathbb{R}^n, c)$ stetig ist. Dies ist gleichwertig zur Stetigkeit der Ableitung

$$f' : U \longrightarrow \mathbb{C}^n, \qquad x \mapsto (\partial_1 f(x), ..., \partial_n f(x)).$$

Mit dem Differenzierbarkeitskriterium folgt, dass eine Funktion $f : U \longrightarrow \mathbb{C}$ genau dann stetig differenzierbar ist, wenn alle n partiellen Ableitungen $\partial_1 f, ..., \partial_n f$ auf U existieren und stetig sind.
Den Vektorraum der stetig differenzierbaren Funktionen auf U bezeichnet man mit $\mathcal{C}^1(U)$.

Definition 8.5: Gradient

Auf dem $\mathbb{R}^n$ sei ein Skalarprodukt gegeben. Dann kann man jede Linearform (lineare Abbildung) $L : \mathbb{R}^n \longrightarrow \mathbb{R}$ mit Hilfe eines eindeutig bestimmten Vektors $g \in \mathbb{R}^n$ darstellen:

$$Lh = \langle g, h \rangle \quad \forall h \in \mathbb{R}^n.$$

Ist L das Differential einer in a differenzierbaren reellwertigen Funktion f, so heißt g *Gradient von* f *in* a *bzgl.* $\langle \cdot, \cdot \rangle$. Er wird mit $\operatorname{grad} f(a)$ bezeichnet. Bzgl. $\langle \cdot, \cdot \rangle$ ist

er der durch

$$df(a)h = \partial_h f(a) = \langle \text{grad} f(a), h \rangle$$

eindeutig bestimmte Vektor im $\mathbb{R}^n$.

Bemerkung 8.5:

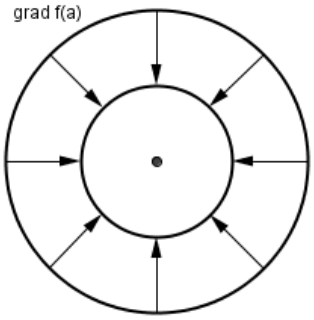
Im Fall des Standardskalarprodukts ist $\text{grad} f(a)$ nach (8.10) der Spaltenvektor

$$\text{grad} f(a) = \begin{pmatrix} \partial_1 f(a) \\ \vdots \\ \partial_n f(a) \end{pmatrix} =: \nabla f(a) \in \mathbb{R}^n.$$

($\nabla f(a)$ wird gesprochen "nabla f von a".)

Beispiel (Gradient einer rotationssymmetrischen Funktionen):

Die in (8.11) definierte rotationssymmetrische Funktion f hat im Fall einer reellen $\mathcal{C}^1$-Funktion F im Punkt $a \neq 0$ den Gradienten



$$\text{grad} f(a) = \frac{F'(\|a\|_2)}{\|a\|_2} \cdot a.$$

Dieser ist im Fall $F'(\|a\|_2) > 0$ zum Ortsvektor $\overrightarrow{0a}$ parallel und im Fall $F'(\|a\|_2) < 0$ antiparallel.
$\|\cdot\|$ bezeichne die zum Skalarprodukt gehörige Norm. Aufgrund der Cauchy-Schwarz-Ungleichung gibt es einen Winkel φ zwischen den Vektoren $\text{grad} f(a)$ und h derart, dass

$$df(a)h = \partial_h f(a) = \langle \text{grad} f(a), h \rangle = \|\text{grad} f(a)\| \cdot \|h\| \cdot \cos\varphi. \tag{8.12}$$

Bemerkung: Cauchy-Schwarz-Ungleichung: $|\langle g, h\rangle|^2 \leqslant \langle g, g\rangle\langle h, h\rangle$. Gleichheit, wenn g und h linear abhängig, $\cos\varphi = \frac{\langle x,y\rangle}{\|x\|\cdot\|y\|}$
Nach dieser Darstellung zeichnet sich der Gradient durch folgende Maximalitätseigenschaft aus:

1. Seine Länge $\|\text{grad} f(a)\|$ ist das Maximum aller Richtungsableitungen $\partial_h f(a)$ nach den Einheitsvektoren

$$\|\text{grad} f(a)\| = \max\{\partial_h f(a) \quad | \quad \|h\| = 1\} =: M$$

2. Im Fall $M \neq 0$ gibt es genau einen Einheitsvektor $\hat{h}$ mit $\partial_{\hat{h}} f(a) = M$ und mit diesem ist $\text{grad} f(a) = M\hat{h}$. Der Gradient zeigt also in die Richtung des stärksten Anstiegs der Funktion im Punkt a.

Satz 8.5: Rechenregeln

Es gelten folgende algebraische Regeln:
Seien $f, g : U \longrightarrow \mathbb{C}$ differenzierbar in $a \in U$. Dann sind auch $f + g$ und $f \cdot g$ in a differenzierbar und es gilt

$$\begin{aligned} d(f+g)(a) &= df(a) + dg(a) \\ d(f \cdot g)(a) &= df(a) \cdot g(a) + f(a) \cdot dg(a) \end{aligned}$$

Ist zusätzlich $f(a) \neq 0$, so ist auch $\frac{1}{f}$ in a differenzierbar mit

$$d\left(\frac{1}{f}\right)(a) = -\frac{df(a)}{f^2(a)} \qquad \text{(Qutotientenregel).}$$

Bemerkung 8.6:

Für die Ableitungen in Satz (8.1.2) gelten also dieselben Regeln wie im Fall $n = 1$:

$$\begin{aligned} (f+g)'(a) &= f'(a) + g'(a) \\ (f \cdot g)'(a) &= f'(a) \cdot g(a) + g'(a) \cdot f(a) \\ \left(\frac{1}{f}\right)'(a) &= -\frac{f'(a)}{f^2(a)} \end{aligned}$$

Lemma 8.6: Rechenregeln für Differenzierbarkeit auf U

Sind f und g in U stetig differenzierbar, dann sind auch $f + g$, $f \cdot g$ und f/g in $\{x \in U \mid g(x) \neq 0\}$ differenzierbar.

Beweis: (Quotientenregel) Zeige, dass die Linearform $-df(a)/f^2(a)$ die Bedingung (8.1) erfüllt, d.h.

$$\lim_{h \to 0} \frac{f(a+h) - f(a) - Lh}{\|h\|} = 0.$$

Für hinreichend kurze Vektoren $h \in \mathbb{R}^n$ ist auch $f(a+h) \neq 0$ und es gilt

$$\begin{aligned} &\tfrac{1}{\|h\|}\left(\tfrac{1}{f(a+h)} - \tfrac{1}{f(a)} + \tfrac{df(a)h}{f^2(a)}\right) \\ = \tfrac{-1}{f(a)f(a+h)}\Bigg(&\underbrace{\frac{f(a+h) - f(a) - df(a)h}{\|h\|}}_{\xrightarrow{h \to 0} 0 \text{ Diffbarkeit von } f \text{ in } a} + \underbrace{\frac{f(a) - f(a+h)}{f(a)} \cdot \frac{df(a)h}{\|h\|}}_{\frac{df(a)h}{\|h\|} \text{ bleibt beschränkt für } \|h\| \to 0} \Bigg) \end{aligned}$$

□

Folgerung 8.7: Rationale Funktionen

Jede rationale Funktion ist in ihrem Definitionsbereich stetig differenzierbar.

Satz 8.8: Kettenregel (1. Version)

Es sei $\gamma = \gamma_1, ..., \gamma_n : I \longrightarrow U$ differenzierbar in t_0 und $f : U \longrightarrow \mathbb{C}$ differenzierbar in $a = \gamma(t_0)$. Dann ist $f \circ \gamma$ differenzierbar in t_0 und hat dort die Ableitung

$$\frac{d(f \circ \gamma)}{dt}(t_0) = df(a)\dot{\gamma}(t_0) = f'(a)\dot{\gamma}(t_0) = \sum_{i=1}^{n} \partial_i f(a) \cdot \dot{\gamma}_i(t_0).$$

Mit Hilfe des Gradienten lautet die Formel nach Definition 8.1.2

$$\frac{d(f \circ \gamma)}{dt}(t_0) = \langle \text{grad} f(a), \dot{\gamma}(t_0) \rangle$$

Beweis: Für $k \in \mathbb{R}$, $h \in \mathbb{R}^n$ mit hinreichend kleinen Beträgen gilt nach Voraussetzung

$$\begin{aligned} \gamma(t_0 + k) = &\quad \gamma(t_0) + \dot{\gamma}(t_0)k + r_1(k)|k|, && \lim_{k\to 0} r_1(k) = 0 \\ f(a + h) = &\quad f(a) + df(a)h + r_2(h)\|h\|, && \lim_{h\to 0} r_2(h) = 0 \end{aligned}$$

Setzt man $h := \gamma(t_0 + k) - \gamma(t_0)$, so folgt

$$f(\gamma(t_0 + h)) = f(\gamma(t_0)) + df\gamma(t_0)\dot{\gamma}(t_0)k + R(k), \tag{8.13}$$

wobei

$$R(k) := df(a)r_1(k)|k| + r_2(\gamma(t_0 + k) - \gamma(t_0))\|\dot{\gamma}(t_0)k + r_1(k)|k|\|.$$

Offensichtlich gilt $\lim_{k\to 0} \frac{R(k)}{k} = 0$. Damit folgt die Behauptung aus (8.13). □

Beispiel (Kettenregel):

Sei f eine differenzierbare Funktion auf $\mathbb{R}^2$. Wir betrachten ihre Komposition $F := f \circ \rho_2$ mit der Polarkoordinatenabbildung $F(r, \varphi) = f(r\cos\varphi, r\sin\varphi)$.

Differenziert man F bei festgehaltenem φ nach r, erhält man die partielle Ableitung nach r (nach φ entsprechend). Es ergibt sich

$$\begin{aligned} F_r(r, \varphi) &= f_x(r\cos\varphi, r\sin\varphi) \cdot \cos\varphi + f_y(r\cos\varphi, r\sin\varphi) \cdot \sin\varphi \\ F_\varphi(r, \varphi) &= f_x(r\cos\varphi, r\sin\varphi) \cdot (-r\sin\varphi) + f_y(r\cos\varphi, r\sin\varphi) \cdot r\cos\varphi \end{aligned}$$

8.1.3 Orthogonalität von Gradient und Nullmenge

Sei $f : U \longrightarrow \mathbb{R}$, $U \subset \mathbb{R}^n$ eine differenzierbare Funktion und $\gamma : I \longrightarrow U$ eine differenzierbare Kurve, die in einer Niveaumenge **Bemerkung:** Sei $f :\longrightarrow \mathbb{R}$, $D \subset \mathbb{R}^n$ und $c \in \mathbb{R}$.
Dann bezeichnet $N_c = \{x \in D | f(x) = c\}$, $N_c \subset \mathbb{R}^n$ die *Niveaumenge* von f zum Niveau c von f verläuft, d.h. es ist $f(\gamma(t)) = c$ für eine Konstante c und $\forall t \in I$.
Dann steht der Gradient von f im Punkt $\gamma(t)$ senkrecht auf dem Tangentialvektor $\dot{\gamma}(t)$:

$$\text{grad} f(\gamma(t)) \quad \perp \quad \dot{\gamma}(t), \qquad t \in I.$$

Beweis: Wegen $c = f \circ \gamma(t)$ gilt

$$0 = \frac{d}{dt}(f \circ \gamma)(t) = \langle \text{grad} f(\gamma(t)), \dot{\gamma}(t) \rangle = 0$$

□

Theorem 8.9: Mittelwertsatz

Sei f eine reelle differenzierbare Funktion in einer offenen Menge $U \subset \mathbb{R}^n$. Ferner seien $a, b \in U$ Punkte, deren Verbindungsstrecke in U liegt. Dann gibt es einen Punkt $\xi \in [a, b]$ mit

$$f(a) - f(b) = df(\xi)(b - a) = f'(\xi)(b - a)$$

Beweis: Setze $\gamma(t) := a + t(b - a)$, $t \in [0, 1]$ und betrachte $F := F \circ \gamma : [0, 1] \longrightarrow \mathbb{R}$. Dann gilt $F(1) - F(0) = f(b) - f(a)$. Nach der Kettenregel ist F differenzierbar und nach dem Mittelwertsatz der Differentialrechnung in *einer* Variablen $\exists \tau \in [0, 1]$, sodass

$$F(1) - F(0) = F(\tau) = df(\gamma(\tau))(b - a).$$

Somit leistet der Punkt $\xi := \gamma(\tau)$ das Gewünschte. □

Korollar 8.10: Folgerung aus dem Mittelwertsatz

Sei $U \subset \mathbb{R}^n$ eine zusammenhängende offene Menge. Hat eine Funktion $f : U \longrightarrow \mathbb{C}$ überall die Ableitung 0, so ist sie konstant.

Beweis: Es genügt, die Behauptung für ein reelles f zu zeigen. Seien a, b beliebige Punkte in U. Dazu wähle nun Punkte $a_0 := a, a_1, ..., a_k = b$ derart, dass die Strecken $[a_{i-1}, a_i]$ in U liegen. Anwendung des MWS bei jeder Strecke ergibt wegen $f' = 0$: $f(a) = f(a_1), ..., f(a_{k-1}) = f(b)$. □

Theorem 8.11: Schrankensatz

Eine $\mathcal{C}^1$-Funktion $f : U \longrightarrow \mathbb{C}$ auf einer offenen Menge U ist auf jeder kompakten konvexen Teilmenge $K \subset U$ Lipschitz-stetig. D.h. mit

$$\|f'\|_K := \max_{\xi \in K} \|f'(\xi)\|_{1,K} = \max_{\xi \in K}(|\partial_1 f(\xi)| + ... + |\partial_n f(\xi)|)$$

gilt für beliebige $x, y \in K$

$$|f(x) - f(y)| \leqslant \|f'\|_K \cdot \|y - x\|_\infty.$$

Beweis: Mit $x, y \in K$ liegt auch die Strecke $[x, y]$ in K. Folglich ist die Funktion $F := f \circ \gamma : [0, 1] \longrightarrow \mathbb{C}$ mit $\gamma(t) = x + t(y - x)$ definiert.
Nach dem Schrankensatz für Funktionen *einer* Veränderlichen gilt daher

$$|f(x) - f(y)| = |F(1) - F(0)| \leqslant \|\dot{F}\|_{[0,1]}.$$

Die Kettenregel ergibt die Ableitung

$$|\dot{F}(t)| \leqslant \sum_{i=1}^{n} |\partial_i f(\gamma(t))| \cdot |y_i - x_i|.$$

Damit folgt die Behauptung. □

Definition 8.6: Höhere Ableitungen

Die partiellen Ableitungen $\partial_1 f, ..., \partial_n f$ einer Funktion können ihrerseits partiell differenzierbar sein. Darum heißen die Funktionen

$$\partial_{ij} f := \partial_i(\partial_j f)$$

partielle Ableitungen 2. Ordnung von f.
Weitere Bezeichnungen sind $f_{x_j x_i}$ und $\dfrac{\partial^2 f}{\partial x_j \partial x_i}$.

Beispiel ($f(x,y) = x^y$):
Betrachte die Funktion $f(x,y) := x^y$ auf $\mathbb{R}_+ \times \mathbb{R}$. Die partiellen Ableitungen 1. Ordnung sind **Bemerkung:** wobei wir zum Ableiten nutzen, dass $x^y = e^{\ln x^y}$

$$f_x(x,y) = yx^{y-1}, \qquad f_y(x,y) = x^y \ln x$$

und die partiellen Ableitungen 2. Ordnung

$$f_{xx}(x,y) = y(y-1)x^{y-2} \qquad f_{xy}(x,y) = x^{y-1}(1 + y\ln x)$$
$$f_{yx}(x,y) = x^{y-1}(1 + y\ln x) \qquad f_{yy}(x,y) = x^y(\ln x)^2.$$

Bemerkung 8.7:

In diesem Beispiel ist $f_{xy} = f_{yx}$. Im Allgemeinen ist jedoch $\partial_{ij} f \neq \partial_{ji} f$. Es kann sogar vorkommen, dass nur eine der partiellen Ableitungen $\partial_{ij} f$ oder $\partial_{ji} f$ existiert. Dies ist jedoch nicht der Fall, wenn eine der partiellen Ableitungen $\partial_{ij} f$ oder $\partial_{ji} f$ stetig ist.

Satz 8.12: Schwarz

Die Funktion f besitze in einer Umgebung von $a \in \mathbb{R}^n$ die partiellen Ableitungen $\partial_i f, \partial_j f$ und $\partial_{ji} f$. Ferner sei $\partial_{ji} f$ in a stetig. Dann existiert auch $\partial_{ij} f(a)$ und es gilt

$$\partial_{ij} f(a) = \partial_{ji} f(a)$$

Für den Beweis des Satzes benötigen wir das folgende Lemma. (Wir verwenden im Beweis ein Analogon des MWS.)
Sei $Q \subset \mathbb{R}^2$ ein Rechteck mit den Ecken $(a,b), (a+h, b+k)$, $h, k \neq 0$, $\varphi : Q \longrightarrow \mathbb{R}$. Für φ sei

$$D_{Q\varphi} := \varphi(a+h, b+k) - \varphi(a+h, b) - \varphi(a, b+k) + \varphi(a,b).$$

Lemma 8.13:

Sei φ reell und besitze auf Q die partiellen Ableitungen $Q_1\varphi$ und $Q_2\varphi$. Dann existiert ein Tupel $(\xi, \eta) \in Q$ mit

$$D_{Q\varphi} = hk \cdot \partial_{21}\varphi(\xi, \eta).$$

Beweis: (des Lemmas)

$$\begin{aligned} u(x) &:= \varphi(x, b+k) - \varphi(x, b) \\ D_{Q\varphi} &= u(a+h) - u(a) = hu'(\xi) \\ &= h\left(\partial_1\varphi(\xi, b+k) - \partial_1\varphi(\xi, b)\right) = hk\partial_{21}\varphi(\xi, \eta) \end{aligned}$$

□

Nun können wir den Satz von Schwarz beweisen:

Beweis: Es genügt, ein reelles f zu betrachten. Man setze für (x, y) aus einer Umgebung $V \subset \mathbb{R}^2$ von $(0, 0)$

$$\varphi(x, y) := f(a + xe_i + ye_j).$$

Bei geeigneter Wahl von V existieren die partiellen Ableitungen $\partial_1\varphi, \partial_2\varphi$ und $\partial_{21}\varphi$. Ferner ist $\partial_{21}\varphi$ im Punkt $(0, 0)$ stetig.
Es ist zu zeigen: $\partial_{12}\varphi$ existiert in $(0, 0)$ und $\partial_{12}\varphi(0, 0) = \partial_{21}\varphi(0, 0)$ $(*)$.
Sei dazu $\varepsilon > 0$. Man wähle eine Umgebung $V' \subset V$ von $(0, 0)$, sodass für $(x, y) \in V'$ die Abschätzung $|\partial_{21}\varphi(x, y) - \partial_{21}\varphi(0, 0)| < \varepsilon$ gilt und weiter wähle man ein achsenparalleles Rechteck $Q \subset V'$ mit den gegenüberliegenden Ecken $(0, 0)$ und (h, k), $(h, k) \neq 0$.
Nach dem obigen Lemma ist dann

$$\left|\frac{D_{Q\varphi}}{h \cdot k} - \partial_{21}\varphi(0, 0)\right| < \varepsilon.$$

Wegen

$$\frac{D_{Q\varphi}}{h \cdot k} = \frac{1}{h}\left(\frac{\varphi(h, k) - \varphi(h, 0)}{k} - \frac{\varphi(0, k) - \varphi(0, 0)}{k}\right)$$

folgt mit $k \to 0$

$$\left|\frac{\partial_2\varphi(h, 0) - \partial_2\varphi(0, 0)}{h} - \partial_{21}\varphi(0, 0)\right| \leqslant \varepsilon$$

für alle hinreichend kleinen $|h| \neq 0$. Damit ist $(*)$ bewiesen. □

Definition 8.7: k-mal stetig differenzierbar

Sei $U \subset \mathbb{R}^n$ offen. Eine Funktion $f : U \longrightarrow \mathbb{C}$ heißt *k-mal stetig differenzierbar* oder auch $\mathcal{C}^k$-Funktion, $k \geqslant 1$, wenn alle partiellen Ableitungen $\partial_{i1} f, ..., \partial_{ik} f$ k-ter Ordnung auf U existieren und stetig sind. Den Vektorraum der $\mathcal{C}^k$-Funktionen auf U bezeichnet man mit

$$\mathcal{C}^k(U).$$

Auf Grund des Satzes von Schwarz (Satz 8.1.3) spielt bei einer $\mathcal{C}^k$-Funktion die Reihenfolge der partiellen Ableitungen $\partial_{i1} f, ..., \partial_{ik} f$ keine Rolle.
Schließlich definiert man

$$\mathcal{C}^\infty(U) := \bigcap_k^\infty \mathcal{C}^k(U).$$

8.1.4 Differentiale höherer Ordnung

Auf Grund des Satzes von Schwarz (Satz 8.1.3) kann man einer in einer Umgebung eines Punktes $a \in \mathbb{R}^n$ p-mal stetig differenzierbaren Funktion f in Verallgemeinerung des Differentials eine symmetrische, p-fach lineare Abbildung

$$d^p f(a) : \underbrace{\mathbb{R}^n \times ... \times \mathbb{R}^n}_{p} \longrightarrow \mathbb{C}$$

zuordnen.
Wir betrachten zunächst die Fälle $p = 1$ und $p = 2$:

$p = 1 :$ $\quad a \mapsto df(a) : \mathbb{R}^n \longrightarrow \mathbb{C}$

$$df(a)u = (\partial_1 f(a), ..., \partial_n f(a)) \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}$$

$p = 2 :$ $\quad (u, v) \in \mathbb{R}^n \times \mathbb{R}^n$

$$d^{(2)} f(a)(u, v) := \partial_u(\partial_v f)(a) \tag{8.14}$$

Diese Definition ist aus folgenden Grüden sinnvoll: Es gilt $v = (v_1, ..., v_n) \Rightarrow \partial_v f(x) = \sum_{i=1}^n \partial_i f(x) v_i$. Die Funktion $\partial_v f$ ist in einer Umgebung von a stetig differenzierbar, da die Summanden $\partial_1 f, ..., \partial_n f$ diese Eigenschaft haben.
$\partial_v f$ besitzt also Richtungsableitungen und es gilt

$$\partial_u(\partial_v f(a)) = \sum_{i,j=1}^n \partial_{ij} f(a) u_i v_j. \tag{8.15}$$

$(u, v) \mapsto \partial_u \partial_v f(a)$ ist linear in jeder Variablen u, v und nach dem Satz von Schwarz (Satz 8.1.3) symmetrisch.

Definition 8.8: Differential zweiter Ordnung

Der Ausdruck in (8.14) bzw. in seiner Variante (8.15) heißt *Differential zweiter Ordnung von f in a.*

Bezüglich der Standardbasis des $\mathbb{R}^n$ besitzt das Differential zweiter Ordnung folgende Matrixdarstellung:

$$f''(a) = H_f(a) = \begin{pmatrix} \partial_{11} f(a) & \dots & \partial_{1n} f(a) \\ \vdots & & \vdots \\ \partial_{n1} f(a) & \dots & \partial_{nn} f(a) \end{pmatrix} \tag{8.16}$$

Für diese Matrix gilt $d^2 f(a)(u, v) = u^t f''(a) v$.

Bemerkung: Nach dem Satz von Schwarz ist die Hesse-Matrix, wie sie hier eingeführt wurde, symmetrisch.

Definition 8.9: Hesse-Matrix

Die Matrix in (8.16) heißt *Hesse-Matrix* oder zweite Ableitung von f in a.

Analog zur obigen Darstellung definiert man Differentiale höherer Ordnung:

Definition 8.10: Differentiale höherer Ordnung

Für beliebige $p \geqslant 1$ definiert man $d^p f(a)$ wie folgt:

$$\partial^p f(a)(v^1, ..., v^p) := \partial_{v^1}, ..., \partial_{v^p} f(a). \tag{8.17}$$

Die dadurch erklärte Abbildung $d^p f(a)$ ist invariant gegen Vertauschung der Variablen $v^1, ..., v^p$ und linear in jeder einzelnen Variablen. Sie hat die Darstellung

$$d^p f(a)(v^1, ..., v^p) = \sum_{i_1=1}^{n} ... \sum_{i_p=1}^{n} \partial_{i_1} ... \partial_{i_p} f(a) v^1_{i_1} ... v^p_{i_p}. \tag{8.18}$$

8.1.5 Die Taylor-Approximation

Sei $f : U \longrightarrow \mathbb{R}$ eine $\mathcal{C}^{p+1}$-Funktion auf einer offenen Menge $U \subset \mathbb{R}^n$. Weiter seien $a, x \in U$ Punkte, deren Verbindungsstrecke in U liegt. (Wir führen dies auf den 1-dimensionalen Fall zurück.)
Wir betrachten die Funktion $F : [0,1] \longrightarrow \mathbb{R}$,

$$F(t) := f(a + th), \qquad h := x - a.$$

Es gilt $f(a) = F(0)$, $f(x) = F(1)$. F ist eine $\mathcal{C}^{p+1}$-Funktion auf $[0,1]$. Nach der Taylorformel für Funktionen einer Veränderlichen gilt somit

$$F(1) = F(0) + F'(0) + \frac{1}{2!}F''(0) + ... + \frac{1}{p!}F^{(p)}(0) + R_{p+1},$$

wobei das Restglied nach Lagrange mit einem $\tau \in [0,1]$ in der Form

$$R_{p+1} = \frac{1}{(p+1)!}F^{(p+1)}(\tau)$$

dargestellt werden kann.

Die Ableitungen $F^{(k)}$ berechnen wir durch wiederholte Anwendung der Kettenregel:

$$\begin{aligned}
F'(t) &= \sum_{i=1}^{n} \partial_i f(a+th) \cdot h_i \\
F''(t) &= \sum_{i=1}^{n}\sum_{j=1}^{n} \partial_j\partial_i f(a+th) \cdot h_i h_j \\
&\vdots \\
F^{(p)}(t) &= \sum_{i_1=1}^{n} ... \sum_{i_p=1}^{n} \partial_{i_1}...\partial_{i_p} f(a+th) \cdot h_{i_1}...h_{i_p}
\end{aligned}$$

Wir stellen $F^{(k)}(t)$ mit Hilfe des Differentials $d^{(k)}f(a)$ dar. Dazu führen wir allgemein für einen Vektor $x \in \mathbb{R}^n$ folgende Bezeichnung ein:

$$d^{(k)}f(a)x^k := d^{(k)}f(a)(\underbrace{x, ..., x}_{k-\text{mal}}),$$

Dies wird nach (8.18) komponentenweise zu

$$d^{(k)}f(a)x^k = \sum_{i_1=1}^{n} ... \sum_{i_k=1}^{n} \partial_{i_1}...\partial_{i_k} f(a) x_{i_1}...x_{i_k}.$$

$d^{(k)}f(a)x^k$ ist ein homogenes Polynom vom Grad k. Damit gilt

$$F^{(k)}(t) = d^{(k)}f(a+th)h^k.$$

Schließlich setzen wir noch $d^{(0)}f(a)x^0 := f(a)$ und wir können die Taylorapproximation wie folgt definieren.

Definition 8.11: Taylorapproximation

Die *Taylorapproximation der Ordnung p von f in a* ist

$$T_p f(x;a) := \sum_{k=0}^{p} \frac{1}{k!} d^{(k)} f(a)(x-a)^k.$$

Bemerkung 8.8:

$T_1 f(x;a)$ ist die bereits in (8.7) eingeführte lineare Approximation.

Satz 8.14: Taylorformel mit Rest

Es sei $f : U \longrightarrow \mathbb{R}$ eine $\mathcal{C}^{p+1}$-Funktion. Sind $a, x \in U$ Punkte, deren Verbindungsstrecke in U liegt, so gilt

$$f(x) = T_p f(x;a) + R_{p+1}(x;a),$$

wobei das Restglied mit einem geeigneten Punkt $\xi \in [a,x]$ in der Form

$$R_{p+1}(x;a) = \frac{1}{(p+1)!} d^{p+1} f(\xi)(x-a)^{p+1}$$

dargestellt werden kann.

Korollar 8.15: Qualitative Taylorformel

Ist $f : U \longrightarrow \mathbb{R}$ eine $\mathcal{C}^p$-Funktion, so gilt an jeder Stelle $a \in U$ für $x \to a$

$$f(x) = T_p f(x;a) + o(\|x-a\|^p),$$

d.h. es gilt

$$\lim_{x \to a} \frac{f(x) - T_p f(x;a)}{\|x-a\|^p} = 0.$$

Bemerkung 8.9:

$T_pf(x;a)$ stellt also ein Polynom eines Grades $\leqslant p$ dar, welches f in der Nähe von a derart gut approximiert, dass der Fehler $f(x) - T_pf(x;a)$ für $x \to a$ schneller gegen Null geht als $\|x-a\|^p$.

Beweis: Zu $\varepsilon > 0$ wähle man eine Kugel $K_r(a) \subset U$ so, dass für $y \in K_r(a)$

$$\frac{1}{p!} \sum_{i_1=1}^{n} \dots \sum_{i_p=1}^{n} \left| \partial_{i_1} \dots \partial_{i_p} f(y) - \partial_{i_1} \dots \partial_{i_p} f(a) \right| < \varepsilon$$

gilt (Stetigkeit von $d^p f$). Für jeden Vektor $h \in \mathbb{R}^n$ erhält man dann wegen $|h_{i_1} \cdot \dots \cdot h_{i_p}| \leqslant \|h\|_\infty^p$

$$\left| \frac{1}{p!} \left(d^{(p)} f(y) - d^{(p)} f(a) \right) h^p \right| \leqslant \varepsilon \|h\|_\infty^p.$$

Zu jedem $x \in K_r(a)$ wähle man nun weiter einen Punkt $\xi \in [a, x[$, mit dem die Taylorformel mit Rest gilt:

$$\begin{aligned} f(x) &= T_{p-1}f(x;a) + \frac{1}{p!} d^p f(\xi)(x-a)^p \\ &= T_pf(x;a) + \frac{1}{p!} \left(d^p f(\xi) - d^p f(a) \right)(x-a)^p, \end{aligned}$$

d.h. $|f(x) - T_pf(x;a)| = \left| \frac{1}{p!} \left(d^p f(\xi) - d^p f(a) \right)(x-a) \right| \leqslant \varepsilon \|(x-a)\|_\infty^p$. □

Beispiel (Taylorpolynom 2. Ordnung):
Mit $df(a)h = f'(a)h$ und $d^2f(a)h = h^t f''(a)h$ ergibt sich:

$$\begin{aligned} T_2f(x;a) &= f(a) + f'(a)(x-a) + \frac{1}{2}(x-a)^t f''(a)(x-a) \\ &= f(a) + \sum_{i=1}^{n} \partial_i f(a)(x_i - a_i) + \frac{1}{2} \sum_{i,j=1}^{n} \partial_{ij} f(a)(x_i - a_i)(x_j - a_j) \end{aligned}$$

Beispiel ($f(x,y) = x^y$):
Sei $f(x,y) = x^y$. Gesucht: $T_pf((x,y);(1,1))$.
Dazu bestimmen wir zunächst die partiellen Ableitungen erster und zweiter Ordnung:

$$\begin{aligned} f_x(x,y) &= yx^{y-1} & f_y(x,y) &= x^y \ln x \\ f_{xx}(x,y) &= y(y-1)x^{y-2} & f_{yy}(x,y) &= x^y (\ln x)^2 \\ f_{xy}(x,y) &= f_{yx}(x,y) = x^{y-1}(1 + \ln x) \end{aligned}$$

Auswertungen der Ableitungen im Punkt $(1,1)$ ergeben:

$$f(1,1) = 1, \qquad f'(1,1)\left(f_x(1,1), f_y(1,1)\right) = (1,0), \qquad f''(1,1) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Damit ergibt sich:

$$\begin{aligned} T_p f\left((x,y);(1,1)\right) &= 1 + (1,0)\begin{pmatrix} x-1 \\ y-1 \end{pmatrix} + \underbrace{(x-1, y-1)\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{(y-1,x-1)}\begin{pmatrix} x-1 \\ y-1 \end{pmatrix} \\ &= 1 + (x-1) + (y-1)(x-1) \end{aligned}$$

Definition 8.12: Taylorreihen

Es sei $f \in \mathcal{C}^\infty(U)$. Dann heißt die Reihe

$$\sum_{k=0}^{n} \frac{1}{k!} d^{(k)} f(a)(x-a)^k$$

Taylorreihe von f im Punkt $a \in U$.
Die Reihe konvergiert genau dann gegen $f(x)$, wenn $\lim\limits_{k\to\infty} R_k(x;a) = 0$.
f heißt *reell-analytisch* in U, wenn jeder Punkt $a \in U$ eine Umgebung hat, in der f durch die Taylorreihe in a dargestellt wird.

8.1.6 (Geometrische) Bedeutung der zweiten Ableitung

Definition 8.13: Schmiegequadrik

Sei f eine reelle $\mathcal{C}^2$-Funktion in einer Umgebung von $a \in \mathbb{R}^n$. Ist $f''(a)$ nicht die Nullmatrix, so beschreibt die quadratische Gleichung

$$x_{n+1} = T_2 f(x;a) = f(a) + f'(a)(x-a) + \frac{1}{2}(x-a)^t f''(a)(x-a)$$

eine Quadrik im $\mathbb{R}^{n+1}$. Diese heißt wegen $f(x) - T_2 f(x;a) = o(\|x-a\|^2)$ *Schmiegequadrik* an den Graphen von f in $(a, f(a))$.

Die Schmiegequadrik hat im Punkt $(a, f(a))$ dieselbe Tangentialhyperebene wie der Graph und auch dieselbe Krümmung, wie in der Differentialgeometrie gezeigt wird.
Im Fall $n = 2$ kann man durch Koordinatentransformation jede Schmiegequadrik in eine

der folgenden Normalformen bringen:

(E)	$z = \pm(x^2 + y^2)$	elliptisches Paraboloid
(H)	$z = x^2 - y^2$	hyperbolisches Paraboloid
(P)	$z = \pm x^2$	parabolischer Zylinder

Eine Transformation in eine der Formen $(E), (H), (P)$ ist genau dann möglich, wenn die Hesse-Matrix $f''(a)$ definit, indefinit bzw. singulär, aber $\neq 0$ ist.

Bemerkung 8.10: Definitheitskriterium

Eine quadratische Form $Q : \mathbb{R}^n \longrightarrow \mathbb{R}$, $Q(x) = x^t Ax$, und ihre darstellende Matrix

A heißen	positiv definit, wenn	$Q(x) > 0 \quad \forall x \neq 0$	$Q > 0$
	negativ definit, wenn	$Q(x) < 0 \quad \forall x \neq 0$	$Q < 0$
	positiv semidefinit, wenn	$Q(x) \geqslant 0$	$Q \geqslant 0$
	negativ semidefinit, wenn	$Q(x) \leqslant 0$	$Q \leqslant 0$
	indefinit, wenn	Q sowohl positive als auch negative Werte annimmt.	$Q \gtrless 0$

Diese Fälle lassen sich wie folgt charakterisieren:

$$
\begin{aligned}
Q > 0 &\Leftrightarrow \text{alle EW sind} > 0 \\
Q < 0 &\Leftrightarrow \text{alle EW sind} < 0 \\
Q \geqslant 0 &\Leftrightarrow \text{alle EW sind} \geqslant 0 \\
Q \leqslant 0 &\Leftrightarrow \text{alle EW sind} \leqslant 0 \\
Q \gtrless 0 &\Leftrightarrow Q \text{ hat positive und negative EW}
\end{aligned}
$$

Bemerkung 8.11:

Im Fall $n = 2$ hat man folgendes Kriterium:

$$
A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \text{ ist } \begin{array}{lll}
\text{positiv definit} & \Leftrightarrow & \det A > 0 \text{ und } a > 0 \\
\text{negativ definit} & \Leftrightarrow & \det A > 0 \text{ und } a < 0 \\
\text{semidefinit} & \Leftrightarrow & \det A \geqslant 0 \\
\text{indefinit} & \Leftrightarrow & \det A < 0
\end{array}
$$

Bemerkung 8.12:

Sei f wieder eine reelle $\mathcal{C}^2$-Funktion in einer Umgebung von $a \in \mathbb{R}^n$. Ihr Graph heißt im Punkt $(a, f(a))$

$$\begin{aligned} \text{elliptisch} &\Leftrightarrow f''(a) \text{ ist (positiv oder negativ) definit} \\ \text{hyperbolisch} &\Leftrightarrow f''(a) \text{ ist nicht singulär und indefinit} \\ \text{parabolisch} &\Leftrightarrow f''(a) \text{ ist singulär und } \neq 0 \end{aligned}$$

Ein hyperbolischer Punkt heißt auch *Sattelpunkt*.

8.1.7 Lokale Minima und Maxima

Definition 8.14: Lokale Extrema

Sei f eine reelle Funktion auf $X \subset \mathbb{R}^n$. Man sagt f habe in $a \in X$ ein *lokales Maximum* bzw. *Minimum*, wenn es in X eine Umgebung V von a gibt, sodass

$$f(x) \leqslant f(a) \qquad \text{bzw.} \qquad f(x) \geqslant f(a) \qquad \forall x \in U$$

gilt. Kann V so gewählt werden, dass sogar

$$f(x) < f(a) \qquad \text{bzw.} \qquad f(x) > f(a) \qquad \forall x \in U \backslash \{a\}$$

gilt, so heißt a *isoliertes Maximum* bzw. *Minimum*.

Satz 8.16: Notwendiges Kriterium

Sei $U \subset \mathbb{R}^n$ offen. Hat $f : U \longrightarrow \mathbb{R}$ in a ein lokales Extremum und ist f in a partiell differenzierbar, so gilt

$$\partial_1 f(a) = \ldots = \partial_n f(a) = 0. \tag{8.19}$$

Ist f in a differenzierbar, so besagt (8.19) $df(a) = 0$.

Beweis: Die durch $F(t) = f(a + te_k)$ in einem hinreichend kleinem Intervall um $0 \in \mathbb{R}$ erklärte Funktion hat in $t = 0$ ein lokales Extremum. Also ist $F'(0) = 0$ und damit folgt $\partial_k f(a) = F'(0) = 0$. □

Definition 8.15: stationär

Eine in a differenzierbare Funktion heißt *stationär in* a, wenn $df(a) = 0$. Nach dem soeben bewiesenen Satz hat eine differenzierbare Funktion auf einer offenen Menge höchstens an stationären Stellen lokale Extrema.

Satz 8.17: Hinreichendes Kriterium

Es sei $U \subset \mathbb{R}^n$ eine offene Menge und $f : U \longrightarrow \mathbb{R}$ eine $\mathcal{C}^2$-Funktion mit $f'(a) = 0$. Dann gilt:

$$\begin{aligned} f''(a) > 0 &\quad\Leftrightarrow\quad f \text{ hat in } a \text{ ein isoliertes Minimum} \\ f''(a) < 0 &\quad\Leftrightarrow\quad f \text{ hat in } a \text{ ein isoliertes Maximum} \\ f''(a) \gtrless 0 &\quad\Leftrightarrow\quad f \text{ hat in } a \text{ kein lokales Extremum.} \end{aligned}$$

Im indefiniten Fall gibt es Geraden g_1, g_2 durch den Punkt a derart, dass $f\big|_{U\cap g_1}$ in a ein isoliertes Maximum und $f\big|_{U\cap g_2}$ in a ein isoliertes Minimum besitzt.

Beweis: Sei zunächst $f''(a) > 0$. Wegen $f'(a) = 0$ gilt nach der qualitativen Taylorformel (Korollar 8.1.5) für hinreichend kleine Vektoren h

$$f(a+h) = f(a) + \frac{1}{2}h^t f''(a)h + R(h),$$

wobei $R(h)/\|h\|^2 \to 0$ für $h \to 0$. Die Funktion $h \mapsto h^t f''(a)h$ hat auf der Einheitssphäre $\{x \mid \|x\| = 1\}$ wegen $f''(a) > 0$ ein positives Minimum m. Da jeder Vektor h das $\|h\|$-fache eines Einheitsvektors ist, folgt für alle h

$$h^t f''(a)h \geqslant m\|h\|^2.$$

Wir wählen nun eine Kugel $K_\varepsilon(a) \subset U$ so klein, dass $|R(h)| \leqslant 1/4m\|h\|^2$ für $\|h\| < \varepsilon$ gilt. Für $a + h \in K_\varepsilon(a)$ erhalten wir dann

$$f(a+h) \geqslant f(a) + \frac{m}{4}\|h\|^2.$$

Danach nimmt f innerhalb $K_\varepsilon(a)$ genau im Punkt a ein Minimum an. Im Fall $f''(a) > 0$ ist die Behauptung damit bewiesen. $f''(a) < 0$ wird durch den Übergang zu $-f$ analog behandelt.

Sei schließlich $f''(a)$ indefinit. Wir wählen Vektoren v und w mit $v^t f''(a)v > 0$ bzw. $w^t f''(a)w < 0$ und betrachten die Funktionen

$$\begin{aligned} F_v(t) &= f(a+tv), \\ F_w(t) &= f(a+tw), \end{aligned}$$

die in geeigneten Intervallen um $0 \in \mathbb{R}$ definiert sind. Ihre ersten und zweiten Ableitungen in 0 sind

$$F_v'(0) = f'(a)v = 0 \qquad F_v''(0) = v^t F''(a)v > 0,$$
$$F_w'(0) = f'(a)w = 0 \qquad F_w''(0) = w^t F''(a)w < 0.$$

Somit hat F_v in 0 ein isoliertes lokales Minimum und F_w ein isoliertes lokales Maximum. f hat daher in a kein lokales Extremum. □

Kapitel 9

Differenzierbare Abbildungen

9.1 Differenzierbare Abbildungen

Es seien X, Y endlich-dimensionale normierte Vektorräume über einem Körper $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$ und $f : U \longrightarrow Y$ eine Abbildung auf einer offenen Menge $U \subset X$. Besonders wichtig ist dabei der Fall, in dem $X = \mathbb{K}^n$ und $Y = \mathbb{K}^m$:

$$f = \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix} : \mathbb{K}^n \supset U \longrightarrow \mathbb{K}^m, \qquad \text{(Standardfall)} \tag{9.1}$$

Weiter verwenden wir auf dem Vektorraum $L(X, Y)$ der $\mathbb{K}$-linearen Abbildungen von X und Y die induzierte Operatornorm.
Die Endlichkeit der Dimensionen der Vektorräume impliziert, dass jede lineare Abbildung von X nach Y stetig ist und dass X, Y und $L(X, Y)$ vollständig normierte Räume sind.

Definition 9.1: Differenzierbarkeit

$f : U \longrightarrow Y$ heißt *differenzierbar im Punkt* $a \in U$, genauer $\mathbb{K}$-*differenzierbar*, wenn es eine $\mathbb{K}$-lineare Abbildung $L : X \longrightarrow Y$ gibt derart, dass der durch

$$f(a + h) = f(a) + L(h) + R(h)$$

erklärte Rest R die Bedingung

$$\lim_{h \to 0} \frac{R(h)}{\|h\|} = 0 \tag{9.2}$$

erfüllt.

Bemerkung 9.1: Analogon zu differenzierbaren Funktionen

Wie für Funktionen zeigt man, dass es höchstens eine solche Abbildung L gibt. Diese heißt *Differential* oder *Linearisierung von f in a* und wird mit $df(a)$ bezeichnet. Bzgl. Basen in X und Y kann $df(a)$ als Matrix dargestellt werden. Diese heißt *Funktionalmatrix* oder auch *Ableitung von f in a* (bzgl. der Basen) und wird mit $f'(a)$ bezeichnet. Im Fall $\dim X = \dim Y$ heißt die Determinante von $f'(a)$ *Funktionaldeterminante* von f in a.

Beispiel ($f(x) := Ax + b$):
Eine affine Abbildung $f : \mathbb{K}^n \longrightarrow \mathbb{K}^m$

$$f(x) := Ax + b, \quad A \in \mathbb{K}^{m\times n}, b \in \mathbb{K}^m$$

ist in jedem Punkt differenzierbar. Ihre Ableitung ist die Matrix A und das Differential die durch $h \mapsto Ah$ gegebene lineare Abbildung $\mathbb{K}^n \longrightarrow \mathbb{K}^m$:

$$f'(a) = A, \qquad df(a)h = Ah \quad \forall h \in \mathbb{K}^n.$$

Lemma 9.1: Reduktionslemma

Eine Abbildung $f = (f_1, f_2) : U \longrightarrow Y_1 \times Y_2$ in eine direkte Summe ist genau dann differenzierbar im Punkt $a \in U$, wenn dort $f_1 : U \longrightarrow Y_1$ und $f_2 : U \longrightarrow Y_2$ differenzierbar sind. Gegebenenfalls ist

$$df(a) = (d_1 f(a), d_2 f(a)) . \tag{9.3}$$

Beweis: f_1, f_2 seien in a differenzierbar. Dann gilt für $i = 1, 2$:

$$f_i(a + h) = f_i(a) + df_i(a)h + R_i(h),$$

wobei R_i die Bedingung (9.2) erfüllt. Wir setzen $Lh := (df_1(a)h, df_2(a)h)$. L ist eine lineare Abbildung $X \longrightarrow Y_1 \times Y_2$ und mit ihr gilt

$$f(a + h) = f(a) + L(h) + (R_1(h), R_2(h)).$$

$R(h) := (R_1(h), R_2(h))$ erfüllt die Bedingung (9.2). Also ist f differenzierbar und hat das Differential L. Analog zeigt man die Umkehrung. □

9.1.1 Funktionalmatrix

Korollar 9.2: Funktionalmatrix

Die Abbildung (9.1) ist genau dann in $a \in U$ differenzierbar, wenn dort jede der Komponentenfunktionen $f_1, ..., f_m$ differenzierbar ist. Gegebenenfalls gilt für $h \in \mathbb{K}^n$

$$df(a)h = f'(a)h,$$

wobei die Funktionalmatrix $f'(a)$ folgende Gestalt hat:

$$f'(a) = \begin{pmatrix} f_1'(a) \\ \vdots \\ f_m'(a) \end{pmatrix} = \begin{pmatrix} \partial_1 f_1(a) & \dots & \partial_n f_1(a) \\ \vdots & & \vdots \\ \partial_1 f_m(a) & \dots & \partial_n f_m(a) \end{pmatrix} \tag{9.4}$$

9.1.2 Differenzierbarkeitskriterium für Abbildungen

Im Folgenden werden wir für allgemeine Abbildungen eine Analogon zu differenzierbaren Funktionen schaffen.
Dazu werden die meisten Aussagen in ähnlicher Weise hergeleitet, wie dies für differenzierbare Funktionen gemacht wird.

Theorem 9.3: Differenzierbarkeitskriterium

Eine Abbildung $f = (f_1, ..., f_m) : U \longrightarrow \mathbb{R}^m, U \subset \mathbb{R}^n$, ist in $a \in U$ $\mathbb{R}$-differenzierbar, wenn alle partiellen Ableitungen $\partial_\nu f_\mu$, $\nu = 1, ..., n$, $\mu = 1, ..., m$ in einer Umgebung von a existieren und im Punkt a stetig sind.

Definition 9.2: Richtungsableitungen

Das Differential einer in a differenzierbaren Abbildung kann wie für Funktionen mit Hilfe von *Richtungsableitungen* berechnet werden. In Verallgemeinerung von (8.9) gilt

$$df(a)h = \lim_{t \to 0} \frac{f(a+th) - f(a)}{t} =: \partial_h f(a).$$

$\partial_h f(a)$ heißt *Ableitung von f in Richtung h im Punkt a*. Die Ableitungen in Rich-

tung einer fest gewählten Basis $e_1, ..., e_n$ für X heißen die partiellen Ableitungen bzgl. der Basis und werden wieder mit $\partial_1 f(a), ..., \partial_n f(a)$ bezeichnet.

Definition 9.3: Funktionalmatrix und Richtungsableitungen

Mit der Funktionalmatrix $f'(a)$ bzgl. der Basen in X und Y hat man für die Richtungsableitungen die Darstellung

$$\partial_h f(a) = f'(a)h.$$

Insbesondere ist im Standardfall $\partial_\nu f(a) = f'(a)e_\nu$ gleich der ν-ten Spalte in der Funktionalmatrix:

$$f'(a) = (\partial_1 f(a), ..., \partial_n f(a)) \qquad \text{mit} \qquad \partial_\nu f(a) = \begin{pmatrix} \partial_\nu f_1(a) \\ \vdots \\ \partial_\nu f_m(a) \end{pmatrix}.$$

Definition 9.4: stetig differenzierbar

Eine differenzierbare Abbildung $f : U \longrightarrow Y$ auf einer offenen Menge $U \subset X$ heißt *stetig differenzierbar in* U, wenn ihr Differential $df : U \longrightarrow L(X, Y)$, $x \mapsto df(x)$, stetig ist.

Bemerkung 9.2: Stetigkeitstest

$df : U \longrightarrow L(X, Y)$ ist genau dann stetig, wenn für jeden Vektor $h \in X$ die Abbildung $U \longrightarrow Y$, $x \mapsto df(x)h$, stetig ist.

Bemerkung 9.3: Ergänzung zum Reduktionslemma

Das Reduktionslemma kann wie folgt ergänzt werden: Eine Abbildung $f : U \longrightarrow Y_1 \times Y_2$ in eine direkte Summe ist genau dann stetig differenzierbar, wenn ihre beiden Komponenten $f_i : U \longrightarrow Y_i$, $i = 1, 2$, stetig differenzierbar sind.
Für den Standardfall impliziert diese Ergänzung: Die Abbildung (9.1) ist genau dann stetig differenzierbar, wenn alle Komponentenfunktionen $f_1, ..., f_m$ stetig dif-

ferenzierbar sind.

Beispiel (Polarkoordinatenabbildung):

Wir betrachten die Polarkoordinatenabbildung $P_2 : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$, $(r, \varphi) = \begin{pmatrix} r\cos\varphi \\ r\sin\varphi \end{pmatrix}$:

$$P_2'(r,\varphi) = \begin{pmatrix} \partial_1 f_1 & \partial_2 f_1 \\ \partial_1 f_2 & \partial_2 f_2 \end{pmatrix} = \begin{pmatrix} \cos\varphi & -r\sin\varphi \\ \sin\varphi & r\cos\varphi \end{pmatrix}.$$

Definition 9.5: k-mal stetig differenzierbar

Eine Abbildung $f = f_1, ..., f_m : U \longrightarrow \mathbb{K}^n$, $U \subset \mathbb{K}^m$, heißt *k-mal stetig differenzierbar in U*, wenn alle Komponentenfunktionen $f_1, ..., f_m$ k-mal stetig differenzierbar sind.
Den Raum der k-mal stetig differenzierbaren Abbildungen $U \longrightarrow \mathbb{K}^m$ bezeichnet man mit

$$\mathcal{C}^k(U, \mathbb{K}^m).$$

Man setzt außerdem

$$\mathcal{C}^\infty(U, \mathbb{K}^m) := \bigcap_{k=1}^{\infty} \mathcal{C}^k(U, \mathbb{K}^m).$$

Satz 9.4: Kettenregel

X, Y, Z seien normierte Vektorräume und V offen in X, U offen in Y. In $V \xrightarrow{g} U \xrightarrow{f} Z$ sei g differenzierbar in a und f differenzierbar in $b := g(a)$. Dann ist $f \circ g$ differenzierbar in a und es gilt

$$d(f \circ g)(a) = df(b) \circ dg(a). \tag{9.5}$$

Für Ableitungen bedeutet das

$$(f \circ g)'(a) = f'(b) \cdot g'(a). \tag{9.6}$$

Sind f, g stetig differenzierbar, dann auch $f \circ g$.

Beweis: (Siehe Kapitel 3 in [2]) □

Ein Beispiel: Tangentialvektoren

Eine differenzierbare Abbildung $f : U \longrightarrow \mathbb{K}^m$, $U \subset \mathbb{K}^n$ offen, ordnet einer differenzierbaren Kurve $\gamma : I \longrightarrow U$ die Bildkurve

$$f \circ \gamma : I \longrightarrow \mathbb{K}^m$$

zu. Diese ist nach der Kettenregel ebenfalls differenzierbar und hat für $t_0 \in I$ den Tangentialvektor

$$\frac{d}{dt}(f \circ g)(t) = df(\gamma(t_0))\dot{\gamma}(t_0) = f'(\gamma(t_0))\dot{\gamma}(t_0). \tag{9.7}$$

Tangentialvektoren werden also durch das Differential bzw. mittels Funktionalmatrix abgebildet.

Anwendung auf zu den Basisvektoren parallele Koordinaten:
Für $a \in U$ sei $\varepsilon_i(t) = a + te_i$, t aus einem Intervall um 0, sodass $\varepsilon_i(t) \in U$. Die Bildkurve $f \circ \varepsilon_i$ hat für $t = 0$ im Punkt $f(a)$ den Tangentialvektor $f'(a) \cdot e_i$. Dies ist gerade der i-te Spaltenvektor von $f'(a)$. Die Kurven $f \circ \varepsilon_1, ..., f \circ \varepsilon_n$ heißen die von f erzeugten Koordinatenlinien durch $f(a)$.

Betrachte die Polarkoordinatenabbildung $P_2 : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ $(r, \varphi) \mapsto \begin{pmatrix} r\cos\varphi \\ r\sin\varphi \end{pmatrix}$. Diese bildet die Gerade $g_{\varphi_0} : r \mapsto (r, \varphi_0)$ auf die Gerade durch den Nullpunkt ab und die Gerade $\tilde{g}_{r_0} : \varphi \mapsto (r_0, \varphi)$ auf Kreise durch den Nullpunkt. Die Spalten der Funktionalmatrix

$$P_2'(r_0, \varphi_0) = \begin{pmatrix} \cos\varphi_0 & -r_0\sin\varphi_0 \\ \sin\varphi_0 & r_0\cos\varphi_0 \end{pmatrix}$$

sind im Bildpunkt $P_2(r_0, \varphi_0)$ Tangentialvektoren an $P_2 \circ g_{\varphi_0}$ bzw. $P_2 \circ \tilde{g_{r_0}}$.

9.1.3 Extrema unter Nebenbedingungen

Gegeben sei eine Funktion $f : U \longrightarrow \mathbb{R}$ und weitere Funktionen $\varphi_1, ..., \varphi_k : U \longrightarrow \mathbb{R}$ auf einer Menge $U \subset \mathbb{R}^n$. Sei M die Nullstellenmenge von $\varphi = (\varphi_1, ..., \varphi_k) : U \longrightarrow \mathbb{R}^k$:

$$M = \{x \in U | \varphi(x) = 0\}. \tag{9.8}$$

Gesucht werden Punkte $x_0 \in M$ mit $f(x) \leqslant f(x_0)\ \forall x \in M$ oder $f(x) \geqslant f(x_0)\ \forall x \in M$. Solche Punkte heißen *Maximal- bzw. Minimalpunkte von f auf M* oder auch *unter der Nebenbedingung $\varphi = 0$*.

Das führt uns zu folgender notwendigen Bedingung für Maxima und Minima, falls M eine Mannigfaltigkeit ist:

Satz 9.5: Multiplikatorenregel von Lagrange

f und $\varphi = (\varphi_1, ..., \varphi_k)$ seien stetig differenzierbar auf einer offenen Menge $U \subset \mathbb{R}^n$. $\varphi'(x)$ habe in jedem Punkt $x \in M$ den Rang k. Dann gilt:
Ist $x_0 \in M$ ein Extremalpunkt von f auf M, so ist $f'(x_0)$ eine Linearkombination von $\varphi_1'(x_0), ..., \varphi_k'(x_0)$: Es gibt Zahlen $\lambda_1, ..., \lambda_k \in \mathbb{R}$, sogenannte *Lagrange-Multiplikatoren*, mit

$$f'(x_0) = \sum_{i=1}^{k} \lambda_i \varphi_i'(x_0). \tag{9.9}$$

Im euklidischen $\mathbb{R}^n$ bedeutet (9.9)

$$\text{grad} f(x_0) = \sum_{i=1}^{k} \lambda_i \text{grad} \varphi_i(x_0). \tag{9.10}$$

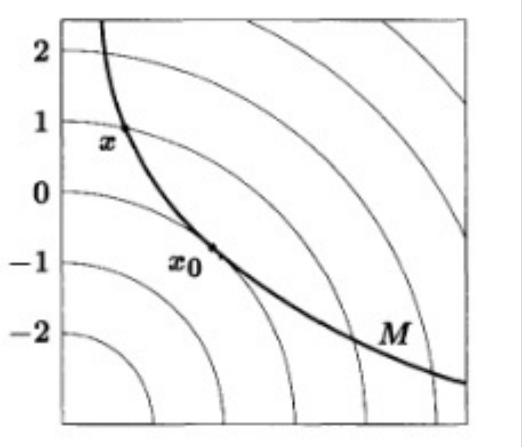
Beweis: (Hier skizzieren wir nur die Idee für den Beweis. Einen ausführlichen Beweis findet man im Kapitel 3.6 in [2])
Wir zeigen (9.10) und dass jeder Tangentialvektor $v \in T_{x_0} M_0$ auf $\text{grad} f(x)$ senkrecht steht. Zu $v \in T_{x_0} M\ \exists$ eine stetig differenzierbare Kurve $\alpha : (-\varepsilon, \varepsilon) \longrightarrow M$ mit $\alpha(0) = x_0$ und $\dot{\alpha}(0) = v$. Die durch $F(t) = f(\alpha(t))$ definierte Funktion $F : (-\varepsilon, \varepsilon) \longrightarrow \mathbb{R}$ hat in $t = 0$ ein lokales Extremum, d.h.

$$0 = \dot{F}(0) = f'(\alpha(0)) \cdot \dot{\alpha}(0) = \langle \text{grad} f(x_0), v \rangle = 0.$$

□

Bemerkung 9.4: Niveaulinien

Betrachte den Fall $f'(x_0) \neq 0$ und $k = 1$. In diesem Fall besagt die Multiplikatorenregel, dass sich die linearen Approximationen von f und φ im Punkt x_0 berühren. Die Notwendigkeit ist leicht einzusehen (siehe Abbildung rechts):



Beispiel (Punkt mit minimalem Abstand zu $(1,0,0)$):
Gesucht ist der Punkt der Ebene $z = x + y$, der vom Punkt $(1,0,0)$ den kleinsten euklidischen Abstand hat.

Zielfunktion: $f(x,y,z) := (x-1)^2 + y^2 + z^2$
Nebenbedingung: $\varphi(x,y,z) = x + y - z = 0$

$$f'(x,y,z) = \lambda\varphi'(x,y,z)$$

$(\partial_1 f, \partial_s f, \partial_3 f)(x,y,z) = (\partial_1\varphi, \partial_2\varphi, \partial_3\varphi)(x,y,z)$
φ hat vollen Rang $\forall x$.
Wir lösen nun das Gleichungssystem (siehe unten) und erhalten $(x,y,z) = (\frac{2}{3}, -\frac{1}{3}, \frac{1}{3})$.
$(\frac{2}{3}, -\frac{1}{3}, \frac{1}{3})$ ist der einzige Punkt, der für den minimalen Abstand in Frage kommt. Somit ist $(\frac{2}{3}, -\frac{1}{3}, \frac{1}{3})$ die Lösung dieser Aufgabe.

Lösen des Gleichungssystems:

$$
\begin{array}{lrcl}
① & 2(x-1) & = & \lambda \\
② & 2y & = & \lambda \\
③ & 2z & = & -\lambda \\
④ & x+y-z & = & 0 \\
\hline
② + ③ = ⑤: & z & = & -y \\
⑤ + ④ = ⑥: & y & = & -\dfrac{x}{2} \\
① + ② = ⑦: & -x & = & 2x-2 \\
 & \dfrac{2}{3} & = & x \quad \Rightarrow \quad y = -\dfrac{1}{3} \quad \Rightarrow \quad z = \dfrac{1}{3}
\end{array}
$$

Bemerkung:

Der Vektor $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 2/3 \\ -1/3 \\ 1/3 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1/3 \\ -1/3 \end{pmatrix}$ steht senkrecht auf der Ebene $x + y - z = 0$.

Kapitel 10

Nichtlineare Gleichungen

10.1 Nichtlineare Gleichungen

$$F, \sigma : D(F) \subset \mathbb{C}^n \longrightarrow \mathbb{C}^n$$

- Durch die Transformation $F(x) := \sigma(x) - y$ kann jede nichtlineare Gleichung $\sigma(x) = y$ in eine Nullstellenaufgabe $F(x) = 0$ überführt werden.
- Da nichtlineare Gleichungen in der Regel nicht geschlossen gelöst werden können, ihre Lösungen also nicht in endlich vielen Schritten berechenbar sind, kommen fast ausschließlich *Iterationsverfahren* zur Approximation der Lösung zur Anwendung.
- Für Iterationsverfahren im $\mathbb{C}^n$ verwenden wir wieder die Notation $x^{(k)}$ mit hochgestelltem, geklammerten Iterationsindex. Im Eindimensionalen verwenden wir einen tiefgestellten Iterationsindex.

10.1.1 Konvergenzbegriffe

Iterationsverfahren zur Berechnung der ν-ten Wurzel einer Zahl $a \in \mathbb{C}\backslash\{0\}$:

$$x_{k+1} = \frac{1}{\nu}\left((\nu-1)x_k + \frac{a}{x_k^{\nu-1}}\right), \qquad k = 0, 1, ... \tag{10.1}$$

Wenn die Folge konvergiert, erfüllt der Grenzwert die Fixpunktgleichung **Bemerkung:**
Sei $f : X \longrightarrow X$, $X \subset \mathbb{R}^n$ stetig. x heißt Fixpunkt, wenn gilt $f(x) = x$.

$$\nu x = (\nu-1)x + \frac{a}{x^{\nu-1}} \qquad \Leftrightarrow \qquad x^\nu = a.$$

Unklar ist, ob diese Folge überhaupt konvergiert und wenn ja, gegen welche Wurzel sie konvergiert.

Für $\nu = 2$ (*klassisches Heronverfahren*) sieht dies folgendermaßen aus:

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right)$$

Nun führt die Transformation

$$z_k = \frac{x_k - \sqrt{a}}{x_k + \sqrt{a}} \tag{10.2}$$

auf die Rekursion

$$z_{k+1} = \frac{\frac{1}{2}x_k + \frac{a}{2x_k} - \sqrt{a}}{\frac{1}{2}x_k + \frac{a}{2x_k} + \sqrt{a}} = \frac{x_k^2 + a - 2\sqrt{a}x_k}{x_k^2 + a + 2\sqrt{a}x_k} = \frac{(x_k - \sqrt{a})^2}{(x_k + \sqrt{a})^2} = z_k^2. \tag{10.3}$$

Man bestimmt den folgenden Grenzwert:

$$\lim_{k\to\infty} z_k = \lim_{k\to\infty} z_0^{(2^k)} = \begin{cases} 0 & |z_0| < 1 \\ 1 & |z_0| = 1 \\ \infty & |z_0| > 1 \end{cases}$$

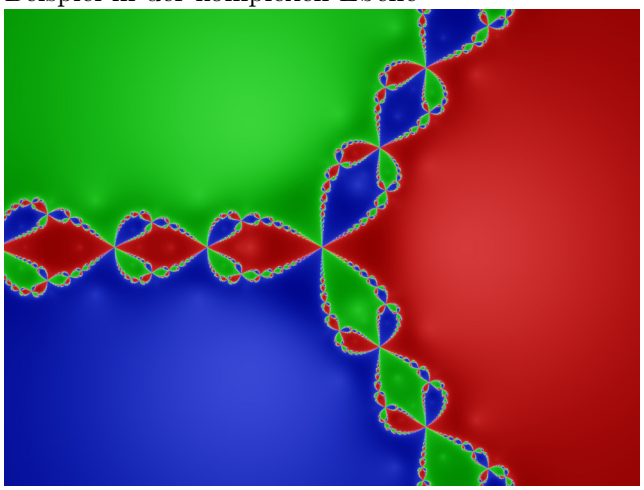
D.h. x_k konvergiert gegen $\sqrt{a}$, falls $|z_0| < 1$ und gegen $-\sqrt{a}$, falls $|z_0| > 1$. Für $|z_0| = 1$ liegt keine Konvergenz vor:

$$\begin{aligned} |z| = \left|\frac{x_0 - \sqrt{a}}{x_0 + \sqrt{a}}\right| \gtrless 1 \quad &\Leftrightarrow \quad |x_0|^2 - 2\mathrm{Re}(\bar{x}_0)\sqrt{a} + |a| = |x_0 - \sqrt{a}|^2 \\ &\gtrless |x_0 + \sqrt{a}|^2 = |x_0|^2 + 2\mathrm{Re}(\bar{x}_0)\sqrt{a} + |a| \\ &\Leftrightarrow \quad 0 \gtrless \mathrm{Re}(\bar{x}_0)\sqrt{a} \end{aligned}$$

Für positives a ergibt sich somit Konvergenz gegen diejenige Wurzel von a, die dasselbe Vorzeichen hat wie der Realteil von x_0.

Bereits für den Fall $\nu = 3$ ergibt sich eine scheinbar chaotische Abhängigkeit vom Startwert x_0.

Beispiel in der komplexen Ebene



$f(z) = z^3 - 1$

Definition 10.1: lokal konvergent

Ein Iterationsverfahren $x^{(k+1)} = \Phi(x^{(k)})$ mit $\Phi : D(\Phi) \subset \mathbb{C}^n \longrightarrow \mathbb{C}^n$ heißt *lokal konvergent gegen* $\hat{x} \in \mathbb{C}^n$, falls eine Umgebung $U \subset D(\Phi)$ um $\hat{x} \in U$ existiert, sodass $\forall\, x^{(0)} \in U$ die resultierende Folge $\{x^{(k)}\}$ gegen $\hat{x}$ konvergiert. In diesem Fall spricht man von einem *anziehenden Fixpunkt* $\hat{x}$ *von* Φ. Das Iterationsverfahren heißt *global konvergent*, wenn U der gesamte Raum $\mathbb{C}^n$ ist.

Bemerkung 10.1: Konvergenz des Heronverfahrens

Das Heronverfahren ($\nu = 2, \nu = 3$) ist für jede ν-te Wurzel von $a \in \mathbb{C}$ lokal konvergent.

Für allgemeinere Probleme kann man so genannte Kontraktionen betrachten. Diese sind interessant, weil sie genau einen Fixpunkt haben, der sich iterativ approximieren lässt.

Definition 10.2: Kontraktion

$\Phi : D \subset \mathbb{C}^n \to \mathbb{C}^n$ heißt *Kontraktion* wenn für alle $x, y \in D$ gilt

$$\|\Phi(x) - \Phi(y)\| \leqslant q \cdot \|x - y\|$$

für ein festes $q \in \mathbb{R}$ mit $0 < q < 1$. Der Faktor q heißt *Lipschitz-Konstante*.

Beispiel (Affin lineare Kontraktion):
Die Abbildung $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ mit $\Phi(x) := A \cdot x + b$ ist genau dann eine Kontraktion wenn für die induzierte Matrixnorm gilt $\|A\| < 1$, denn

$$\|\Phi(x) - \Phi(y)\| = \|A \cdot (x - y)\| \leqslant \|A\| \cdot \|x - y\|.$$

Kontraktionen lassen nützliche Aussagen über Fixpunkte zu.

Satz 10.1: Banachscher Fixpunktsatz

Sei $D \subset \mathbb{C}^n$ und $\Phi : D \to D$ eine Kontraktion mit Lipschitz-Konstante q. Dann hat Φ genau einen Fixpunkt $\hat{x} \in D$.
Für $x^{(0)} \in D$ konvergiert die *Fixpunktiteration* $x^{(k+1)} := \Phi(x^{(k)})$ gegen diesen Fixpunkt, d.h.

$$\hat{x} = \lim_{k\to\infty} x^{(k)}.$$

Für alle $k \in \mathbb{N}$ gelten die folgenden Fehlerabschätzungen:

1. Monotonie: $\|x^{(k)} - \hat{x}\| \leqslant q \cdot \|x^{(k-1)} - \hat{x}\|$
2. A-priori Schranke: $\|x^{(k)} - \hat{x}\| \leqslant \frac{q^k}{1-q} \cdot \|x^{(1)} - x^{(0)}\|$
3. A-posteriori Schranke: $\|x^{(k)} - \hat{x}\| \leqslant \frac{q}{1-q} \cdot \|x^{(k)} - x^{(k-1)}\|$

Beweis: Per Induktion sieht man:

$$\|x^{(n+1)} - x^{(n)}\| \leqslant q^n \cdot \|x^{(1)} - x^{(0)}\|$$

Es folgt für $k, p \in \mathbb{N}$:

$$\begin{aligned}\|x^{(k+p)} - x^{(k)}\| &\leqslant \sum_{i=1}^{p} \|x^{(k+i)} - x^{(k+i-1)}\| \leqslant q^k \cdot \sum_{i=1}^{p} q^{i-1} \cdot \|x^{(1)} - x^{(0)}\| \\ &\leqslant \frac{q^k}{1-q} \cdot \|x^{(1)} - x^{(0)}\| \qquad (10.4)\end{aligned}$$

Dies ist eine Nullfolge, d.h. $x^{(k)}$ ist eine Cauchy-Folge. Somit existiert ein Grenzwert $\hat{x} = \lim_{k\to\infty} x^{(k)}$. Dieser ist ein Fixpunkt, denn

$$\Phi(\hat{x}) = \Phi\left(\lim_{k\to\infty} x^{(k)}\right) = \lim_{k\to\infty} \Phi\left(x^{(k)}\right) = \lim_{k\to\infty} x^{(k+1)} = \hat{x}.$$

Für einen weiteren Fixpunkt $\tilde{x} \in D$ gilt

$$\|\tilde{x} - \hat{x}\| = \|\Phi(\tilde{x}) - \Phi(\hat{x})\| \leqslant q \cdot \|\tilde{x} - \hat{x}\|,$$

d.h. es ist $\|\tilde{x} - \hat{x}\| = 0$ und somit ist $\hat{x} = \tilde{x}$ der einzige Fixpunkt von Φ.
Es bleibt die Abschätzungen zu zeigen:

$$\begin{aligned}\|x^{(k)} - \hat{x}\| &= \|\Phi(x^{(k-1)}) - \Phi(\hat{x})\| \leqslant q \cdot \|x^{(k-1)} - \hat{x}\| \\ \|x^{(k)} - \hat{x}\| &= \lim_{p\to\infty} \|x^{(k)} - x^{(k+p)}\| \overset{(10.4)}{\leqslant} \frac{q^k}{1-q} \cdot \|x^{(1)} - x^{(0)}\| \\ (1-q) \cdot \|x^{(k)} - \hat{x}\| &\leqslant \|x^{(k)} - \hat{x}\| - \|x^{(k+1)} - \hat{x}\| \\ &\leqslant \|x^{(k+1)} - x^{(k)}\| \leqslant q \cdot \|x^{(k)} - x^{(k-1)}\|\end{aligned}$$

□

Iterative Verfahren zur Lösung linearer Gleichungssysteme

Nun können wir zum Beispiel lineare Gleichungssysteme $A \cdot x = b$ mit $A \in \mathbb{C}^{n\times n}$ und $b \in \mathbb{C}^n$ durch eine Fixpunktiteration lösen. Wir zerlegen A in drei Matrizen $L, U, D \in \mathbb{C}^{n\times n}$:

$$A = \underbrace{L}_{\text{Einträge unter,}} + \underbrace{U}_{\text{über}} + \underbrace{D}_{\text{und auf der Diagonalen.}}$$

Beispiel (Gesamtschrittverfahren (alias Jacobi-Verfahren)):
Ein Fixpunkt von $\Phi(x) := D^{-1} \cdot (b - (L + U) \cdot x)$ löst $A \cdot x = b$:

$$x = \Phi(x) \;\Leftrightarrow\; D \cdot x = b - (L + U) \cdot x \;\Leftrightarrow\; A \cdot x = b$$

Wenn $\|D^{-1} \cdot (L + U)\| < 1$ ist diese Abbildung eine Kontraktion (siehe obiges Beispiel).
Es ergibt sich der folgende Algorithmus:
for $k = 0, 1, ...$ (k Iterationsindex)
  for $i = 1$ **to** n

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \cdot \left(b_i - \sum_{\substack{j=1 \\ i\neq j}}^{n} a_{i,j} \cdot x_j^{(k)} \right)$$

Beispiel (Einzelschrittverfahren (alias Gauß-Seidel-Verfahren)):
Ein Fixpunkt von $\Phi(x) := (D + L)^{-1} \cdot (b - U \cdot x)$ löst $A \cdot x = b$:

$$x = \Phi(x) \;\Leftrightarrow\; (D + L) \cdot x = b - U \cdot x \;\Leftrightarrow\; A \cdot x = b$$

Für $\|(D + L)^{-1} \cdot U\| < 1$ ist Φ eine Kontraktion. Ein entsprechender Algorithmus lässt sich mit Vorwärtssubstitution implementieren:
for $k = 0, 1, ...$ (k Iterationsindex)
  for $i = 1$ **to** n

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \cdot \left(b_i - \sum_{j=1}^{i-1} a_{i,j} \cdot x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j} \cdot x_j^{(k)} \right)$$

Iterative Verfahren zum Lösen von Gleichungssystemen sind nicht immer anwendbar, aber sie sind:

- kaum anfällig für Rundungsfehler,
- effektiv wenn nur ein Vektor $b \in \mathbb{C}^n$ genutzt wird,
- sehr schnell, wenn A eine dünn besetzte Matrix ist.

Der Begriff der Kontraktion ist also für affin lineare Abbildungen nützlich. Man kann ihn aber auch auf nichtlineare Abbildungen anwenden. Dazu prüft man ob die Taylorapproximation erster Ordnung eine Kontraktion ist.

Satz 10.2: Kriterium für Kontraktionen

Sei $\Phi : D \to \mathbb{C}^n$ stetig differenzierbar und sei $\hat{x} \in D$ ein Fixpunkt von Φ mit $\|\Phi'(\hat{x})\| < 1$. Dann existiert ein $r > 0$ so dass Φ auf dem Ball $B_r(\hat{x})$ vom Radius r um $\hat{x}$ eine Kontraktion ist.

Beweis: Da $\|\Phi'(x)\|$ stetig ist, existiert ein $r > 0$ und ein $q < 1$ so dass $\|\Phi'(x)\| \leqslant q$ für alle $x \in B_r(\hat{x})$. Nach dem Mittelwertsatz der Differentialrechnung im $\mathbb{C}^n$ wissen wir für alle $x, y \in B_r(\hat{x})$:

$$\Phi(y) - \Phi(x) = \int_0^1 \Phi'(x + t \cdot (y - x)) \cdot (y - x) dt$$

$$\Rightarrow \|\Phi(x) - \Phi(y)\| \leqslant \int_0^1 \|\Phi'(x + t \cdot (y - x))\| \cdot \|y - x\| dt \leqslant q \cdot \|x - y\|$$

Insbesondere ist $\|\Phi(x) - \hat{x}\| \leqslant q \cdot \|x - \hat{x}\| \leqslant r$, d.h. $\Phi(x) \in B_r(\hat{x})$. □

10.1.2 Konvergenzgeschwindigkeit einer Folge

Definition 10.3: Konvergenzordnung

Für eine reelle nichtnegative Nullfolge $\{\varepsilon_k\}_{k \in \mathbb{N}}$ wird

$$K = \limsup_{k \to \infty} \varepsilon_k^{1/k} \tag{10.5}$$

als *asymtotischer Konvergenzfaktor* definiert. Die Folge $\{\varepsilon_k\}$ heißt *sublinear*, *linear*, bzw. *superlinear*, je nachdem ob $K = 1$, $0 < K < 1$ oder $K = 0$ ist. Gilt im superlinear konvergenten Fall zudem

$$\varepsilon_{k+1} \leqslant c \cdot \varepsilon_k^p \qquad \text{für ein } p > 1, c > 0 \text{ und fast alle } k \in \mathbb{N},$$

dann hat die Folge die *Konvergenzordnung* p.
Entsprechend wird die Terminologie für konvergente Folgen $\{x^{(k)}\} \subset \mathbb{K}^n$ mit Grenzwert $\hat{x}$ über $\varepsilon_k := \|x^{(k)} - \hat{x}\|$ eingeführt.

Bemerkung 10.2: Konvergenzgeschwindigkeit

Als Faustregel erwartet man, dass sich bei einem Iterationsverfahren mit Konvergenzordnung p die Anzahl der korrekten Dezimalstellen bei jeder Iteration ver-p-facht.

Beispiel (Konvergenzordnung $p = 2$ (quadratische Konvergenz)):
Heronverfahren mit $\nu = 2, a > 0$:

$$z_k = \frac{x_k - \sqrt{a}}{x_k + \sqrt{a}}, \qquad z_{k+1} = \left(\frac{x_k - \sqrt{a}}{x_k + \sqrt{a}}\right)^2 = z_k^2 \tag{10.6}$$

Für $\{x_k\}$ des Heronverfahrens ergibt sich unter der Voraussetzung $\sqrt{a}/2 \leqslant x_k \leqslant 2\sqrt{a}$, dass auch $x_{k+1} \in \left[\frac{\sqrt{a}}{2}, 2\sqrt{a}\right]$ und es folgt aus (10.6):

$$|x_{k+1} - \sqrt{a}| = |z_{k+1}(x_{k+1} + \sqrt{a})| = \frac{|x_{k+1} + \sqrt{a}|}{|x_k + \sqrt{a}|^2}|x_k - \sqrt{a}|^2 \leqslant \frac{\frac{3}{2}\sqrt{a}}{\frac{9}{4}a}|x_k - \sqrt{a}|^2$$

Satz 10.3: Lokale Konvergenz der Fixpunktiteration

Die Funktion $\Phi : D(\Phi) \subset \mathbb{C}^n \longrightarrow \mathbb{C}^n$ sei stetig differenzierbar und habe einen Fixpunkt $\hat{x}$ in $D(\Phi)$. Ferner sei $\|\cdot\|$ eine Norm in $\mathbb{C}^n$ und $\|\cdot\|_M$ eine verträgliche Matrixnorm in $\mathbb{C}^{n\times n}$ mit $\|\Phi'(\hat{x})\|_M < 1$.
Dann ist Φ in einer Umgebung U von $\hat{x}$ eine Kontraktion und die Fixpunktiteration

$$x^{(k+1)} = \Phi(x^{(k)}), \qquad k = 0, 1, 2, ...$$

ist lokal konvergent gegen $\hat{x}$.

Beweis: Dass Φ lokal eine Kontraktion ist haben wir bereits bewiesen. Mit dem Banachschen Fixpunktsatz folgt die Konvergenz. □

Satz 10.4: Lokal superlineare Konvergenz

Die Funktion $\Phi : D(\Phi) \subset \mathbb{C}^n \longrightarrow \mathbb{C}^n$ sei p-mal stetig differenzierbar und habe einen Fixpunkt $\hat{x} \in D(\Phi)$. Ferner sei $p \geqslant 2$,

$$d^{(1)}\Phi(\hat{x}) = 0, ..., d^{(p-1)}\Phi(\hat{x}) = 0$$
$$\text{und} \quad d^{(p)}\Phi(\hat{x}) \neq 0.$$

Dann ist die Fixpunktiteration $x_{k+1} = \Phi(x_k)$ lokal superlinear konvergent gegen $\hat{x}$

und die Konvergenzordnung ist genau p.

Beweis: Es gilt $\|\Phi'(\hat{x})\| = 0 < 1$ und somit folgt die lokale Konvergenz aus dem vorigen Satz. Nun entwickeln wir Φ um $\hat{x}$ in ein Taylorpolynom:

$$\begin{aligned}\Phi(x) &= \Phi(\hat{x}) + \sum_{i=1}^{p} \frac{1}{i!} \cdot d^{(i)}\Phi(\hat{x})(x-\hat{x})^i + o(\|x-\hat{x}\|^p) \\ &= \Phi(\hat{x}) + \frac{1}{p!} \cdot d^{(p)}\Phi(\hat{x})(x-\hat{x})^p + o(\|x-\hat{x}\|^p)\end{aligned}$$

Dann existiert eine von $d^{(p)}\Phi(\hat{x})$ abhängige Konstante $c > 0$ so dass:

$$\begin{aligned}\|x^{(k+1)} - \hat{x}\| &= \|\Phi(x^{(k)}) - \Phi(\hat{x})\| \\ &= \frac{1}{p!} \cdot \|d^p\Phi(\hat{x})(x^{(k)} - \hat{x})^p\| + o(\|x^{(k)} - \hat{x}\|^p) \\ &\leqslant \frac{1}{p!} \cdot c \cdot \|x^{(k)} - \hat{x}\|^p + o(\|x^{(k)} - \hat{x}\|^p)\end{aligned}$$

□

10.1.3 Nullstellenbestimmung reeller Funktionen

Wir erinnern uns, dass jede Gleichung als Nullstellenproblem aufgefasst werden kann:

$$f_1(x) = f_2(x) \quad \Leftrightarrow \quad f_1(x) - f_2(x) = 0$$

Entsprechende Lösungsalgorithmen sind daher auf sehr viele Probleme anwendbar. Gleichzeitig sind sie bei hinreichender Differenzierbarkeit und niedriger Dimension sehr effizient. Bei vielen wichtigen Problemen existieren allerdings auch spezialisierte Algorithmen, die effektiver arbeiten. So ist, z.B. Eigenwertberechnung ein Nullstellenproblem:

$$A \cdot v = \lambda \cdot v \quad \Leftrightarrow \quad (A - \lambda \cdot I) \cdot v = 0$$

Allerdings sind hier spezialisierte Algorithmen effektiver.

Das eindimensionale Newtonverfahren

Sei nun $D(f) \subset \mathbb{C}$ offen und $f : D(f) \to \mathbb{C}$ stetig differenzierbar. Wir betrachten die Nullstellenaufgabe $f(x) = 0$. Um diese zu lösen formulieren wir ein Fixpunktproblem:

$$x = x + g(x) \cdot f(x) =: \Phi(x)$$

Die Funktion $g: D(f) \to \mathbb{C}\backslash\{0\}$ werden wir gleich genauer bestimmen. Zunächst stellen wir fest, dass unabhängig von der Wahl gilt:

$$x = \Phi(x) \Leftrightarrow f(x) = 0$$

Wir haben also in der Tat die Nullstellenaufgabe in ein Fixpunktproblem überführt. Nun wollen wir aber noch für lokal quadratische Konvergenz sorgen, d.h. wir wollen g so wählen, dass für eine Nullstelle $\hat{x} \in D(f)$ gilt $\Phi'(\hat{x}) = 0$. Daraus ergibt sich:

$$\Phi'(\hat{x}) = 1 + g'(\hat{x}) \cdot \underbrace{f(\hat{x})}_{=0} + g(\hat{x}) \cdot f'(\hat{x}) = 0$$

$$\Rightarrow g(\hat{x}) = \frac{-1}{f'(\hat{x})}$$

Wählen wir speziell $g := \frac{-1}{f'}$, so erhalten wir das *Newtonverfahren*:

$$x_{k+1} = \Phi(x_k) = x_k - \frac{1}{f'(x_k)} \cdot f(x_k)$$

Bemerkung 10.3: Geometrische Anschauung

x_{k+1} ist der Schnitt der Tangente an den Graph von f im Punkt $(x_k, f(x_k))$

$$y(x) = f(x_k) + f'(x_k) \cdot (x - x_k)$$

mit der x-Achse.

Das Newton-Verfahren alterniert also zwischen zwei Schritten:

1. Approxiere f durch seine Taylorapproximation erster Ordnung.
2. Berechne die Nullstelle der Taylorapproximation.

Beispiel ($f(x) = x^\nu - a$):
Die Nullstellen der Funktion

$$f(x) = x^\nu - a, \qquad \nu \in \mathbb{N}\backslash\{1\}, a \in \mathbb{R}^+$$

sind die ν-ten Wurzeln der Zahl a. Das Newtonverfahren ergibt

$$x_{k+1} = x_k - \frac{x_k^\nu - a}{\nu x_k^{\nu-1}} = \frac{\nu - 1}{\nu} x_k + \frac{a}{\nu} x_k^{1-\nu}, \qquad k = 0, 1, ...$$

Das entspricht dem Heronverfahren.

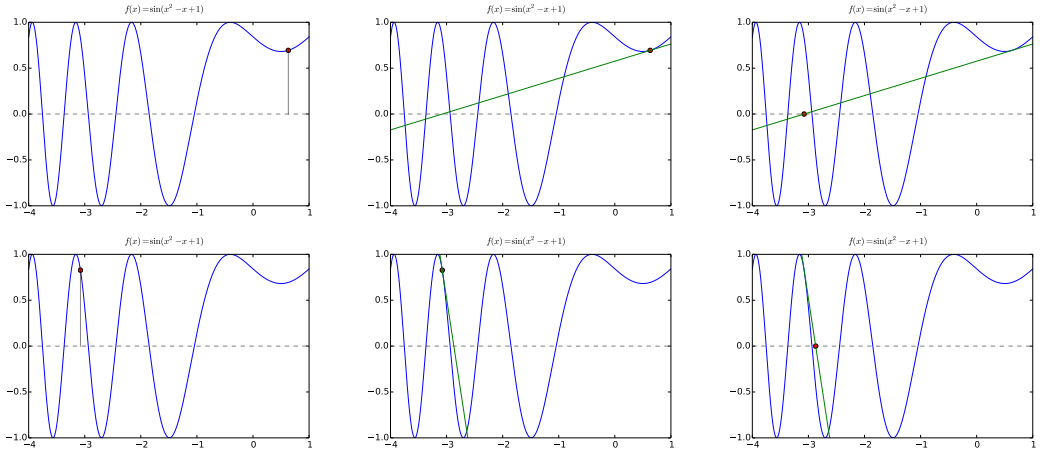


Abbildung 10.1: Veranschaulichung von zwei Iterationen des Newtonverfahrens für die Funktion $f(x) = \sin(x^2 - x + 1)$.

Satz 10.5: Lokal quadratische Konvergenz des Newtonverfahrens

Sei $f : D(f) \to \mathbb{C}$ zwei mal stetig differenzierbar und $\hat{x} \in D(f)$ mit $f(\hat{x}) = 0$ und $f'(\hat{x}) \neq 0$.
Dann konvergiert das Newtonverfahren (mindestens) lokal quadratisch gegen $\hat{x}$.

Beweis: Nach Konstruktion gilt $\Phi(\hat{x}) = \hat{x}$ und $\Phi'(\hat{x}) = 0$. Die Behauptung folgt damit aus dem Satz über lokal superlineare Konvergenz. □

Nun betrachten wir Abbildung 10.2. Wie man sehen kann hängt das Ergebnis des Newtonverfahrens auf chaotische Weise von der Initialisierung ab. Wenn die Initialisierung nahe an einer der Nullstellen liegt erhält man quadratische Konvergenz zu der naheliegenden Nullstelle. Bei einer schlechter gewählten Initialisierung kann die Konvergenz dagegen deutlich langsamer sein und man erhält evtl. nicht die gewünschte Nullstelle.
Daher ist es ratsam zur Initialisierung des Newtonverfahrens und ähnlicher Algorithmen möglichst viel Vorwissen zu nutzen. Man kann z.B. einen heuristischen Algorithmus implementieren um zunächst eine Näherungslösung zu berechnen und diese dann als Initialisierung für das Newtonverfahren verwenden. Falls man mit dem Newtonverfahren zunächst nicht das gewünschte Ergebnis erhält kann man auch systematisch unterschiedliche Initialisierungen ausprobieren.
Bei einer guten Initialisierung und einer zwei mal stetig differenzierbaren Funktion ist die Konvergenz des Newtonverfahrens außerordentlich schnell (nämlich quadratisch). Wichtig ist dabei aber auch, dass die Ableitung f' tatsächlich korrekt berechnet und implementiert wurde. Ansonsten können die Ergebnisse unnachvollziehbar sein.

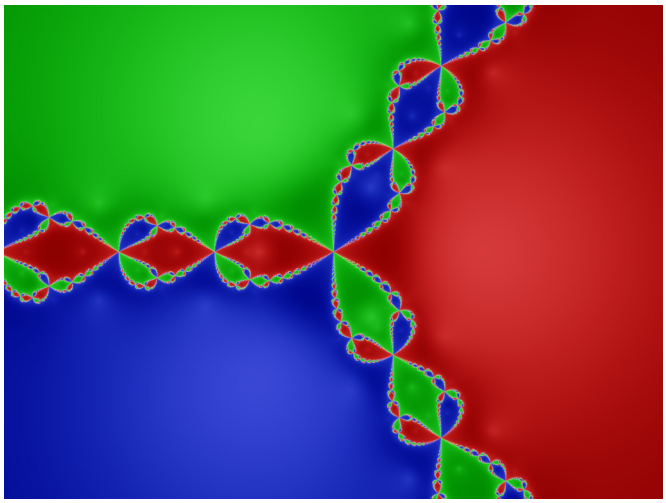


Abbildung 10.2: Ergebnisse des Newtonverfahrens für $f(z) := z^3 - 1$ in Abhängigkeit vom Startwert z_0. Das Bild zeigt die komplexe Ebene. Die Farben zeigen an zu welcher der drei Nullstellen das Newtonverfahren für die Initialisierung z_0 an dieser Stelle konvergiert. Man beachte die fraktale Struktur.

Das Newtonverfahren im $\mathbb{C}^n$

Sei nun $n \in \mathbb{N}$, $D(f) \subset \mathbb{C}^n$ offen und $f : D(f) \to \mathbb{C}^n$ stetig differenzierbar. Wir betrachten die Nullstellenaufgabe $f(x) = 0 \in \mathbb{C}^n$. Der Ansatz ist der gleiche wie im eindimensionalen Fall:

- Approximiere f durch die Taylorapproximation erster Ordnung.
- Löse die Nullstellenaufgabe für die Taylorapproximation durch Lösen eines linearen Gleichungssystems:

$$f(x^{(k)}) + f'(x^{(k)}) \cdot (x^{(k+1)} - x^{(k)}) = 0$$

Anstatt $f'(x^{(k)}) \neq 0$ zu fordern muss nun gefordert werden, dass $f'(x^{(k)})$ regulär ist. Ansonsten übertragen sich alle Berechnungen vom eindimensionalen Fall. Wir definieren

$$x^{(k+1)} = \Phi(x^{(k)}) = x^{(k)} - (f'(x^{(k)}))^{-1} \cdot f(x^{(k)})$$

und erhalten so für eine Nullstelle $\hat{x} \in D(f)$:

$$\Phi(\hat{x}) = \hat{x} - 0 = \hat{x}$$
$$\Phi'(\hat{x}) = I - 0 - (f'(\hat{x}))^{-1} \cdot f'(\hat{x}) = 0$$

Satz 10.6: Lokal quadratische Konvergenz des Newtonverfahrens

Sei $f : D(f) \to \mathbb{C}^n$ zwei mal stetig differenzierbar und $\hat{x} \in D(f)$ mit $f(\hat{x}) = 0$ und $f'(\hat{x})$ regulär.
Dann konvergiert das Newtonverfahren (mindestens) lokal quadratisch gegen $\hat{x}$.

Beweis: Nach Konstruktion gilt $\Phi(\hat{x}) = \hat{x}$ und $\Phi'(\hat{x}) = 0$. Die Behauptung folgt damit aus dem Satz über lokal superlineare Konvergenz. □

Kapitel 11

Nichtlineare Ausgleichsprobleme

11.1 Nichtlineare Ausgleichsprobleme

Sei nun $m > n$, $D(f) \subset \mathbb{R}^n$ und $f : D(f) \to \mathbb{R}^m$ eine stetig differenzierbare Abbildung. Das Gleichungssystem $f(x) = 0$ enthält m Gleichungen aber nur n Unbekannte, es ist also überbestimmt. Analog zu linearen Ausgleichsproblemen betrachten wir daher nun nichtlineare Ausgleichsprobleme:

$$\Phi(x) = \frac{1}{2}\|f(x)\|_2^2 = \frac{1}{2} f(x) \cdot f(x) \to \min$$

Wir suchen also $x \in D(f)$ so, dass die 2-Norm von $f(x)$ minimal wird. Der Vorfaktor $\frac{1}{2}$ ändert in der Definition von Φ verändert dieses Minimum nicht und wird nur zur Vereinfachung des Gradienten von Φ eingeführt.

Erinnerung Die Bedingungen

$$\Phi'(\hat{x}) = 0 \text{ und} \tag{11.1}$$

$$\Phi''(\hat{x}) \text{ positiv definit} \tag{11.2}$$

sind gemäß (8.19) und (8.1.7) notwendig und hinreichend dafür, daßin $\hat{x}$ in lokales Minimum von Φ vorliegt. Dabei ist $\Phi'(x) = \nabla\Phi \in \mathbb{R}^n$ der Gradient von Φ und

$$\Phi''(x) = \left[\frac{\partial^2}{\partial x_i \partial x_j}\Phi(x)\right]\Big]_{ij} \in \mathbb{R}^{n\times n}$$

die (symmetrische) Hessematrix. Ist zumindest die erste Gleichung erfüllt, so heißt $\hat{x}$ stationärer Punkt.

Zur Lösung des nichtlinearen Ausgleichsproblems gibt es grob zwei Verfahrensklassen

- **Gradientenverfahren (Abstiegsverfahren)**, die in jedem Iterationsschritt das Funktional $\Phi(x)$ in einem eindimensionalen affinen Raum minimieren, und
- **Newton-artige Verfahren**, bei denen Φ durch eine lokale Linearisierung ersetzt wird.

Das Newton-Verfahren kann für diesen Fall zum *Gauß-Newton-Verfahren* erweitert werden. Wie beim Newton-Verfahren wird f in jeder Iteration durch seine Taylorapproximation erster Ordnung an der momentanen Iterierten $x^{(k)} \in D(f)$ approximiert. Die nächste Iterierte $x^{(k+1)} \in D(f)$ wird dann durch ein lineares Ausgleichsproblems bestimmt:

$$\|f(x^{(k)}) + f'(x^{(k)}) \cdot (x^{(k+1)} - x^{(k)})\|_2^2 \to \min$$

Leider garantiert das Gauß-Newton-Verfahren keine lokale Konvergenz. Grund dafür ist, dass die Lösung des Ausgleichsproblems $x^{(k+1)}$ bei schlecht konditionierter Matrix $f'(x^{(k)})$ sehr weit von $x^{(k)}$ entfernt sein kann. Für weit entfernte Punkte ist die Taylorapproximation erster Ordnung aber nicht mehr Aussagekräftig. Die Verwendung des Gauß-Newton-Verfahrens ist daher *nicht* empfehlenswert.

Das *Levenberg-Marquardt-Verfahren* überwindet dieses Problem. Die maximale Schrittlänge $\|x^{(k+1)} - x^{(k)}\|$ wird nach oben durch eine Konstante $\rho_k \in \mathbb{R}_+$ beschränkt. Wir definieren also $h^{(k)} := x^{(k+1)} - x^{(k)}$ und fordern $\|h^{(k)}\|_2 < \rho_k$. In der Kugel mit Radius ρ_k um $x^{(k)}$ vertrauen wir darauf, dass die Taylorapproximation sinnvoll ist. Man spricht daher von der sogenannten *Trust-Region*. Um nun $x^{(k+1)} = x^{(k)} + h^{(k)}$ zu berechnen müssen wir

$$\|f(x^{(k)}) + f'(x^{(k)}) \cdot h^{(k)}\|_2^2$$

minimieren unter der Nebenbedingung $\|h^{(k)}\|_2 \leqslant \rho_k \in \mathbb{R}_+$.

Satz 11.1: Optimierung in Trust-Region

Wenn der Vektor $h^{(k)}$

$$\|f(x^{(k)}) + f'(x^{(k)}) \cdot h^{(k)}\|_2^2 \tag{11.3}$$

unter der Nebenbedingung $\|h^{(k)}\|_2 \leqslant \rho_k$ minimiert, existiert ein $\lambda \geqslant 0$ mit

$$(f'(x^{(k)})^T \cdot f'(x^{(k)}) + \lambda \cdot I) \cdot h^{(k)} = -(f'(x^{(k)}))^T \cdot f(x^{(k)}). \tag{11.4}$$

Falls $\|h^{(k)}\|_2 < \rho_k$ gilt $\lambda = 0$.

Beweis: Fall 1, für die optimale Lösung gilt $\|h^{(k)}\|_2 < \rho_k$:
Dann ist $h^{(k)}$ lokales Minimum von (11.3) und somit Lösung des linearen Ausgleichsproblems $\|f(x^{(k)}) + f'(x^{(k)}) \cdot h^{(k)}\|_2^2 \to \min$. Setzen wir $\lambda = 0$, so wird (11.4) zur Normalengleichung für (11.3) und ist somit erfüllt:

$$(f'(x^{(k)}))^T \cdot f'(x^{(k)}) \cdot h^{(k)} = -(f'(x^{(k)}))^T \cdot f(x^{(k)})$$

Fall 2, für die optimale Lösung gilt $\|h^{(k)}\|_2 = \rho_k$:

Wir verwenden die Multiplikatorenregel von Lagrange.

$$\begin{aligned} g : \mathbb{R}^n &\to \mathbb{R} & \varphi : \mathbb{R}^n &\to \mathbb{R} \\ h &\mapsto \|f(x^{(k)}) + f'(x^{(k)}) \cdot h\|_2^2 & h &\mapsto \|h\|_2^2 \end{aligned}$$

$h^{(k)}$ minimiert $g(h^{(k)})$ unter der Nebenbedingung $\varphi(h^{(k)}) = \rho_k^2$.

$$\begin{aligned} \partial_i \varphi(h) &= \partial_i \sum_{j=1}^{n} h_j^2 = 2 \cdot h_i \\ \partial_i g(h) &= \sum_{j=1}^{n} \partial_i (f_j(x^{(k)}) + f_j'(x^{(k)}) \cdot h)^2 \\ &= \sum_{j=1}^{n} \partial_i f_j(x^{(k)}) \cdot 2 \cdot (f_j(x^{(k)}) + f_j'(x^{(k)}) \cdot h) \\ &= 2 \cdot (f'(x^{(k)})^T \cdot (f(x^{(k)}) + f'(x^{(k)}) \cdot h))_i \end{aligned}$$

Nach der Multiplikatorenregel von Lagrange existiert dann ein $\lambda \in \mathbb{R}$ mit:

$$\begin{aligned} & \text{grad} g(h^{(k)}) = -\lambda \cdot \text{grad} \varphi(h^{(k)}) \\ \Leftrightarrow\ & f'(x^{(k)})^T \cdot (f(x^{(k)}) + f'(x^{(k)}) \cdot h^{(k)}) = -\lambda \cdot h^{(k)} \\ \Leftrightarrow\ & (f'(x^{(k)})^T \cdot f'(x^{(k)}) + \lambda \cdot I) \cdot h^{(k)} = -f'(x^{(k)})^T \cdot f(x^{(k)}) \end{aligned}$$

Um $\lambda \geqslant 0$ zu zeigen betrachten wir die Eigenwertzerlegung $V \cdot \Lambda \cdot V^T$ der symmetrischen, positiv semi-definiten Matrix $f'(x^{(k)})^T \cdot f'(x^{(k)})$. Dabei ist $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ und V orthogonal. Es gilt:

$$\begin{aligned} \|h^{(k)}\|_2^2 &= \| - (f'(x^{(k)})^T \cdot f'(x^{(k)}) + \lambda \cdot I)^{-1} \cdot f'(x^{(k)})^T \cdot f(x^{(k)})\|_2^2 \\ &= \|V \cdot (\Lambda + \lambda \cdot I)^{-1} \cdot V^T \cdot f'(x^{(k)})^T \cdot f(x^{(k)})\|_2^2 \\ &= \sum_{i=1}^{n} \frac{1}{(\lambda_i + \lambda)^2} \cdot (V^T \cdot f'(x^{(k)})^T \cdot f(x^{(k)}))_i^2 \end{aligned}$$

Da $\lambda_i \geqslant 0$ ist dieser Ausdruck streng monoton fallend in λ. Wenn die Gleichung also für $\lambda < 0$ erfüllt wäre, so würde eine Lösung mit $\|h^{(k)}\| < \rho_k$ existieren und Fall 1 greift. □

Wir können $h^{(k)}$ also durch die folgenden Schritten berechnen:

- Löse $\|f(x^{(k)}) + f'(x^{(k)}) \cdot h^{(k)}\|_2^2 \to \min$ ohne Nebenbedingung.
- Falls $\|h^{(k)}\|_2 > \rho_k$:
  - Berechne λ durch Lösen der nichtlinearen Gleichung $\|h^{(k)}\|_2 = \rho_k$ mit $h^{(k)}$ wie in (11.4).
    - z.B. mit dem Newton-Verfahren.

– Ermittle neues $h^{(k)}$ mit $\|h^{(k)}\|_2 = \rho_k$ durch Lösen des linearen Gleichungssystems (11.4).

Es bleibt die größe der Trust-Region ρ_k zu wählen. Diese sollte möglichst groß sein damit der Algorithmus in großen Schritten der Lösung entgegengehen kann. Andererseits kann ein zu großer Radius dazu führen, dass Iterationsschritte das Ergebnis verschlechtern. Sei $\mathcal{E}(x) := \|f(x)\|_2^2$ das betrachtete Fehlerfunktional. Gemäß der Taylorapproximation gilt

$$\begin{aligned}\mathcal{E}(x^{(k)} + h) - \mathcal{E}(x^{(k)}) &= \mathcal{E}'(x^{(k)}) \cdot h + o(\|h\|_2) \\ &= 2 \cdot f(x^{(k)})^T \cdot f'(x^{(k)}) \cdot h + o(\|h\|_2).\end{aligned}$$

Um also zu bewerten wie angemessen unsere Trust-Region ist betrachten wir den Quotient aus tatsächlicher und geschätzter Verbesserung:

$$\mu_k := \frac{\|f(x^{(k)} + h^{(k)})\|_2^2 - \|f(x^{(k)})\|_2^2}{2 \cdot f(x^{(k)})^T \cdot f'(x^{(k)}) \cdot h^{(k)}}$$

Wir wählen feste Konstanten $0 < \mu_- < \mu_+ < 1$. Falls $\mu_k < \mu_-$ ist, verwerfen wir $h^{(k)}$ und verwenden in der nächsten Iteration eine kleinere Trust-Region verwendet. Falls $\mu_k > \mu_+$ ist, wird in der nächsten Iteration eine größere Trust-Region verwendet. Diese weit verbreitete Forderung an die Verbesserung in einer Iteration bezeichnet man als *Armijo-Goldstein-Kriterium*.

Algorithmus 11.1: Levenberg-Marquardt-Verfahren

Gegeben: $0 < \mu_- < \mu_+ < 1$, $\rho_0 \in \mathbb{R}_+$, $x^{(0)} \in D(f)$
for $k = 0, 1, \ldots$ **do**
  Minimiere $\|f(x^{(k)}) + f'(x^{(k)}) \cdot h^{(k)}\|_2^2$ mit $\|h^{(k)}\| \leqslant \rho_k$
  $\mu_k := \frac{\|f(x^{(k)}+h^{(k)})\|_2^2 - \|f(x^{(k)})\|_2^2}{2 \cdot f(x^{(k)})^T \cdot f'(x^{(k)}) \cdot h^{(k)}}$
  if $\mu_k < \mu_-$ **then**
    $x^{(k+1)} := x^{(k)}$
    $\rho_{k+1} := \frac{\rho_k}{2}$
  else
    $x^{(k+1)} := x^{(k)} + h^{(k)}$
    if $\mu_k > \mu_+$ **then**
      $\rho_{k+1} := 2 \cdot \rho_k$
    end if
  end if
end for

Es ergibt sich Algorithmus 11.1. Dieser garantiert zumindest in gewissem Sinne Konvergenz:

Satz 11.2: Konvergenz von Levenberg-Marquardt

Sei $f : D(f) \to \mathbb{R}^m$ stetig differenzierbar. Sei $x^{(0)} \in D(f)$ und sei $U \subset D(f)$ kompakt mit

$$\{x \in D(f) \mid \|f(x)\|_2 \leqslant \|f(x^{(0)})\|_2\} \subset U.$$

Dann gilt für die Iterierten des Levenberg-Marquardt-Verfahrens

$$\lim_{k\to\infty} \mathcal{E}'(x^{(k)}) = 0$$

wobei wieder $\mathcal{E}(x) := \|f(x)\|_2^2$ das Fehlerfunktional bezeichnet.

Beweis: Siehe "Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens", 3. Auflage von Martin Hanke-Bourgeois, S. 191 ff. □

Dieses Ergebnis ist allerdings mit Vorsicht zu genießen. Idealerweise würde das Verfahren garantiert zu einem globalen Minimum von $\|f(x)\|_2$ konvergieren. Das ist aber nicht der Fall. Der obige Satz garantiert noch nichtmal, dass die Folge $x^{(k)}$ überhaupt konvergiert. Wenn sie konvergiert garantiert der Satz lediglich, dass der Grenzwert ein kritischer Punkt ist. In vielen (aber nicht allen) Fällen wird der Grenzwert ein lokales Minimum sein. Auch hier sollte man also nach Möglichkeit eine Initialisierung $x^{(0)}$ wählen, die nahe an dem gesuchten globalen Minimium liegt.
Darüber hinaus sagt der Satz nichts über die Konvergenzordnung des Verfahrens aus. In der Praxis erweist sich das Levenberg-Marquardt-Verfahren in dieser Hinischt aber als vergleichsweise gut.
Gradientenverfahren approximieren das Minimum des nichtlinearen Ausgleichsproblems $\hat{x}$ ausgehend von einem Startwert $x^{(0)}$ durch eine Iterationsfolge $x^{(k)}$, bei der sich $x^{(k+1)}$ aus $x^{(k)}$ durch die Wahl einer **Suchrichtung** $d^{(k)}$ und einer **Schrittweite** $\alpha_k > 0$ ergibt:

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}.$$

Suchrichtung und Schrittweite werden dabei so bestimmt, dass eine Abstiegsbedingung $\Phi(x^{(k+1)}) < \Phi(x^{(k)})$ erfüllt ist.
Als Suchrichtung an der Stelle $x^{(k)}$ bietet sich $d^{(k)} = -\nabla\Phi(x^{(k)})$ an. Wegen der Cauchy-Schwarz-Ungleichung gilt für ein Suchrichtung $d \in \mathbb{R}^n$ mit $||d|| = 1$

$$\frac{\partial\Phi}{\partial d}(x) = \nabla\Phi(x) \cdot d \leqslant ||\nabla\Phi(x)||||d|| = ||\nabla\Phi(x)||$$

Setzt man $d = \nabla\Phi(x)/||\nabla\Phi(x)||$, so ergibt sich Gleichheit, d.h. die Wahl $d^{(k)} = -\nabla\Phi(x^{(k)})$ ist optimal. Daher nennt man das reusltierende Verfahren auch die **Methode des steilsten Abstiegs**.

Mit $\Phi(x) = \frac{1}{2}f(x) \cdot f(x)$ ergibt sich für das nichtlineare Ausgleichsproblem

$$\nabla\Phi(x) = \sum_{j=1}^{m} f_j(x)\nabla f_j(x) = f'(x)^t f(x)$$

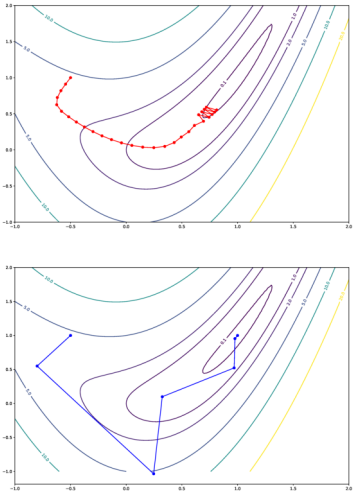
und wir erhalten die Iterationsfolge

$$x^{(k+1)} = x^{(k)} - \alpha_k f'(x^{(k)})^t f(x^{(k)})$$

Beispiel (Rosenbrockfunktion):
Ein klassiches Beispiel zum Testen von Optimierungsverfahren ist die *Rosenbrock-funktion*

$$\Phi(x) = \begin{pmatrix} 4(x_2 - x_1^2) \\ x_2 - 1 \end{pmatrix}$$

Wie an den Niveaulinien zu sehen ist, handelt es sich um ein langes bananenförmiges Tal mit einem Minimum bei $(1, 1)^t$.



Oben: Steilster Abstieg mit konstanter Schrittweite und . Die Gradienten von Φ stehen senkrecht auf den elliptisch geformten Niveaulinien dieser Funktion und weisen nur in Ausnahmefällen in eine effiziente Abstiegsrichtung. Unten: Iterationen des Newton-Verfahrens. Jeweils mit Startwert $x^{(0)} = (-0.5, 1)^t$.

Die pro Schritt optimale Abstiegsrichtung braucht auf lange Sicht nicht optimal sein. In der Praxis werden daher auch alternative Suchrichtungen um den negativen Gradienten verwendet:

$$c||\nabla\Phi(x^{(k)})||_2 \leqslant ||d^{(k)}||_2 \leqslant C||\nabla\Phi(x^{(k)})|| \tag{11.5}$$

mit festen Konstanten $c, C > 0$ und

$$\cos\angle(\nabla\Phi(x^{(k)}), d^{(k)}) = \frac{\nabla\Phi(x^{(k)^t} d^{(k)}}{||\nabla\Phi(x^{(k)}||_2 ||d^{(k)}||_2} \leqslant -\delta \tag{11.6}$$

mit festem $\delta \in (0, 1]$.

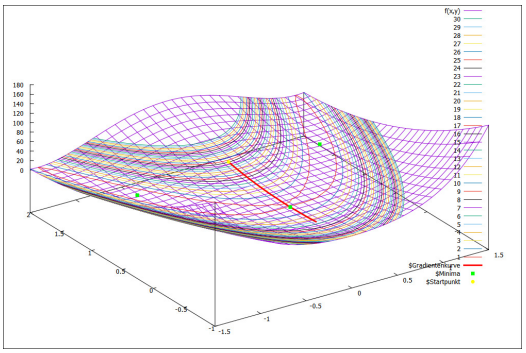
Zur Steuerung der Schrittweite ist es naheliegend, in jedem Schritt das eindimensionale Optimierungsproblem

$$min_{\alpha\in[0,\infty)}\Phi(x^{(k)}+\alpha d^{(k)}) \quad (11.7)$$

exakt zu lösen und dazu eine Nullstelle $\hat{\alpha} \in [0,\infty)$ von

$$\nabla\Phi(x^{(k)}+\alpha d^{(k)})^t \cdot d^{(k)} \quad (11.8)$$

zu suchen.



Ein solches $\hat{\alpha}$ existiert, wenn wir vorraussetzen, dass die Menge $M_0 = \{x \in \mathbb{R}^n | \Phi(x) \leqslant \Phi(x_0)\}$ kompakt ist. Falls Φ zweimal stetig differenzierbar ist, kann man dazu z.B. das Newton-Verfahren einsetzen. Allerdings ist dabei nicht garantiert, dass wir die nächste Nullstelle finden.

Armijo-Schrittweiten

Ein in der Praxis angewandtes Verfahren verwendet *Armijo-Schrittweiten*. Dabei wird ausgehend von einer maximalen Schrittweite $\alpha = 1$ die Schrittweite sukzessive um einen Faktor β, z.B. $\beta = 1/2$, verkleinert, bis die *Armijo-Goldstein-* Abstiegsbedingung erfüllt ist:

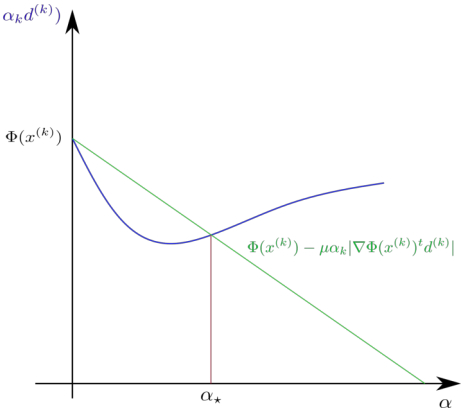


Abbildung 11.20: Armijo-Schrittweiten

$$\Phi(x^{(k)}+\alpha d^{(k)}) \leqslant \Phi(x^{(k)}) - \mu\alpha_k|\nabla\Phi(x^{(k)})^t d^{(k)}| \quad (11.9)$$

Algorithmus 11.2: Allgemeines Abstiegsverfahren

Input: Eine Funktion $\Phi : \mathbb{R}^n \to \mathbb{R}$ und ein $\mu \in (0,1), \mu$ etwa 0.5
wähle $\mathbf{x}^{(0)}$ und Suchrichtung $\mathbf{d}^{(0)}$
for $k = 0, 1, 2, \ldots$ **do**
  $\alpha = 1$ // Schrittweite initialisieren
  while $\Phi(\mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)}) > \Phi(\mathbf{x}^{(k)}) - \mu\alpha\|\text{grad}\Phi(\mathbf{x}^{(k)})^*\mathbf{d}^{(k)}\|$ **do**
    $\alpha = \alpha/2$
  end while
  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha\mathbf{d}^{(k)}$
  wähle neue Suchrichtung $\mathbf{d}^{(k+1)}$ unter Beachtung der Einschränkungen 11.5, 11.6.

end for

Satz 11.3: Konvergenz von Abstiegsverfahren

Die Funktion f sei in einer offenen Menge $\mathcal{U} \subset \mathcal{D}(f)$ stetig differenzierbar mit Lipschitz-stetiger Ableitung f'. Ferner enthalte $\mathcal{U}$ den Startvektor $x^{(0)}$ sowie die gesamte Menge

$$\mathcal{M}(x_0) = \{x \in \mathcal{D}(f) : \Phi(x) \leqslant \Phi(x^{(0)})\}.$$

Falls die Suchrichtung $d^{(k)}$ in jedem Iterationsschritt die Bedingung 11.5, 11.6 erfüllt, dann gilt für die Iterierten v0n Algorithmus 10, daß

$$\nabla\Phi(x^{(k)}) \to 0, \quad k \to \infty.$$

Der Satz besagt nicht, dass die Folge $x^{(k)}$ konvergent ist. Im allgemeinen konvergieren die Iterierten lediglich gegen eine stationären Punkt von Φ. Hat Φ nur einen stationären PUnkte $\hat{x}$ und ist die Menge $\mathcal{M}(x_0)$ zudem beschränkt und f hinreichend glatt, dann konvergiert die Folge $\{x^{(k)}\}$ gegen das Minimum $\hat{x}$.

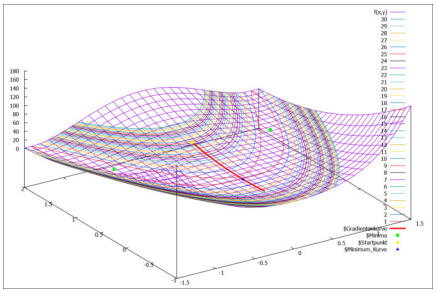
Beweis: Siehe "Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens", 3. Auflage von Martin Hanke-Bourgeois, S. 181 f. □

Gegeben sei die Funktion $f(x_1, x_2) = (x_1, x_2^2 - 1, x_1(x_2^2 - 1))^t$ für das nichtlineare Ausgleichsproblem, s.d.

$$\Phi(x_1, x_2) \quad = \quad \frac{1}{2}||f(x_1, x_2)||_2^2 = \frac{1}{2}\left(x_1^2 + (x_2^2 - 1)^2 + x_1^2(x_2^2 - 1)^2\right)$$

Das Minimum von Φ ist Null und wird für $x_1 = 0$ und $x_2 = \pm 1$ angenommen. Hat eine Iterierte $x^{(k)}$ des Algorithmus 10 die Koordinaten $(x_1, 0)^t$, so ergibt sich

$$d^{(k)} = -(2x_1, 0)^t = -2x^{(k)}.$$



Nach Satz 11.1 konvergiert daher $x^{(k)}$ gegen Null für $k \to \infty$. Der Nullpunkt ist aber nur Sattelpunkt von Φ!.

Literaturverzeichnis

[1] Gerd Fischer. *Lineare Algebra*. Vieweg, Wiesbaden, 2003.

[2] Otto Forster. *Analysis 1*. Vieweg, Wiesbaden, 2006.

[3] Martin Hanke-Bourgeois. *Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens*. Springer, 2009.

[4] Konrad Königsberger. *Analysis 2*. Springer Verlag, 1997.

[5] Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, 1997.