Einführung in die Numerik

Guido Kanschat

2. August 2021

Inhaltsverzeichnis

Kapitel 1

Orthogonale Polynome

1.1 Polynomräume

1.1.1 Lemma: Die Menge der Monome $\{x^0, x^1, \ldots, x^n\}$ ist linear unabhängig.

Beweis. Sei p ein Polynom vom Grad n, also

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \tag{1.1}$$

p ist also gerade eine Linearkombination der Monome. Zu zeigen ist, dass aus der Eigenschaft $p \equiv 0$ folgt, dass alle Koeffizienten verschwinden, also

$$p(x) \equiv 0 \quad \Rightarrow \quad a_n = \cdots = a_0 = 0. \tag{1.2}$$

Zu diesem Zweck berechnen wir die n-te Ableitung von p und erhalten, da mit p auch alle seine Ableitungen identisch verschwinden,

$$n! a_n = 0. \tag{1.3}$$

Daraus schließen wir $a_n = 0$. Nun gilt für die $(n-1)$-te Ableitung

$$n! a_n x + (n-1)! a_{n-1} = (n-1)! a_{n-1} = 0. \tag{1.4}$$

Auf diese Weise schließen wir rekursiv bis a_0, dass alle Koeffizienten verschwinden. Damit ist das Lemma bewiesen. □

1.1.2 Satz: Die Polynome vom maximalen Grad n bilden einen Vektorraum der Dimension $n+1$. Wir bezeichnen ihn mit $\mathbb{P}_n$.

Beweis. Es ist leicht nachzurechnen, dass sowohl die Summe, als auch reelle Vielfache von Polynomen wieder Polynome sind. Insbesondere erhöhen beide Operationen den Grad nicht. Damit ist $\mathbb{P}_n$ ein Vektorraum. Er wird per definitionem von den Monomen vom Grad bis zu n erzeugt. Da diese nach Lemma 1.1.1 linear unabhängig sind, bilden sie eine Basis und die Dimension von $\mathbb{P}_n$ ist $n+1$. □

1.1.3 Quiz: Gegeben beliebige Werte $x_j \in \mathbb{R}$ mit $j = 1, \ldots, n$. Die Menge der Polynome p_i definiert durch

$$p_0(x) = 1$$
$$p_i(x) = \prod_{j=1}^{i} (x - x_j), \qquad i = 1, \ldots, n$$

A ist linear unabhängig

B ist linear abhängig

C ist ein Erzeugendensystem für $\mathbb{P}_n$

D ist eine Basis von $\mathbb{P}_n$

1.2 Skalarprodukt und Orthogonalität

1.2.1 Definition: Sei V ein reeller Vektorraum. Eine Abbildung $a\colon V \times V \to \mathbb{R}$ heißt **Bilinearform**, wenn für $u, v, w \in V$ und $\lambda, \mu \in \mathbb{R}$ gilt

$$a(\lambda u + \mu v, w) = \lambda a(u, w) + \mu a(v, w) \tag{1.5}$$
$$a(w, \lambda u + \mu v) = \lambda a(w, u) + \mu a(w, v). \tag{1.6}$$

Eine Bilinearform heißt **symmetrisch**, wenn für $u, v \in V$ gilt

$$a(u, v) = a(v, u). \tag{1.7}$$

Sie heißt **positiv semi-definit**, wenn $a(u, u) \geq 0$ für alle $u \in V$ und **positiv definit**, wenn zusätzlich

$$a(u, u) = 0 \quad \Longrightarrow \quad u = 0. \tag{1.8}$$

Eine symmetrische, positiv definite Bilinearform heißt **Skalarprodukt**, in der Regel notiert als $\langle \cdot, \cdot \rangle$.

1.2.2 Lemma (Bunjakowski-Cauchy-Schwarzsche Ungleichung): Sei $\langle \cdot, \cdot \rangle$ ein Skalarprodukt auf V. Für zwei beliebige Elemente $u, v \in V$ gilt

$$|\langle u, v \rangle| \leq \sqrt{\langle u, u \rangle} \sqrt{\langle v, v \rangle}. \tag{1.9}$$

Gleichheit gilt genau dann, wenn u und v kollinear sind, also $v = \alpha u$ mit einem skalaren Faktor α.

Beweis. Zunächst zeigen wir nur die Ungleichung: Für $v = 0 \in V$ ist sie offensichtlich.

Seien nun $v, u \in V$ keine Nullvektoren. Für beliebige $\mu, \lambda \in \mathbb{R}$ gilt wegen der Bilinearität

$$0 \leq \langle \lambda u + \mu v, \lambda u + \mu v \rangle = \lambda^2 \langle u, u \rangle + 2\mu\lambda \langle u, v \rangle + \mu^2 \langle v, v \rangle \tag{1.10}$$

Setze $\lambda := \langle v, v \rangle \neq 0$

$$0 \leq \langle v, v \rangle^2 \langle u, u \rangle + 2\mu \langle v, v \rangle \langle u, v \rangle + \mu^2 \langle v, v \rangle \tag{1.11}$$

Dividiere durch$\langle v, v \rangle$

$$0 \leq \langle v, v \rangle \langle u, u \rangle + 2\mu \langle u, v \rangle + \mu^2 \tag{1.12}$$

Setze nun $\mu := -\langle u, v \rangle$

$$0 \leq \langle v, v \rangle \langle u, u \rangle - 2\langle u, v \rangle^2 + \langle u, v \rangle^2 \tag{1.13}$$

Daraus folgt

$$\langle u, v \rangle^2 \leq \langle u, u \rangle \langle v, v \rangle \tag{1.14}$$

und mit der Monotonie der Quadratfunktion die Ungleichung.

Nun bleibt die Äquivalenz für die Gleichheit zu zeigen. Für $v = 0$ ist dies wieder trivial erfüllt. Seien zunächst u, v linear abhängig, also zum Beispiel $u = av$. Dann gilt mit der Abkürzung $f(v) = \sqrt{\langle v, v \rangle}$

$$|\langle u, v \rangle| = |\langle av, v \rangle| = |a| \cdot f(v) \cdot f(v) = f(av) \cdot f(v) = f(u) \cdot f(v). \tag{1.15}$$

Gelte nun umgekehrt $\langle u, v \rangle = \sqrt{\langle u, u \rangle} \sqrt{\langle v, v \rangle}$. Es folgt

$$\langle v, v \rangle \langle u, u \rangle - 2\langle u, v \rangle^2 + \langle u, v \rangle^2 = 0. \tag{1.16}$$

Setze $\mu = \langle u, v \rangle \neq 0$ und $\lambda = \langle v, v \rangle \neq 0$. Dann erhält man

$$\lambda \langle u, u \rangle - 2\mu \langle u, v \rangle + \mu^2 = 0. \tag{1.17}$$

Multipliplikation mit $\langle v, v\rangle$ ergibt

$$0 = \lambda^2\langle u, u\rangle + 2\mu\langle u, v\rangle\langle v, v\rangle + \mu^2\langle v, v\rangle = \langle \lambda u - \mu v, \lambda u - \mu v\rangle. \tag{1.18}$$

Wegen der Definitheit folgt nun $\lambda u - \mu v = 0$ und da μ und λ ungleich Null sind gilt, dass u, v linear abhängig sind □

1.2.3 Lemma: Sei V ein reeller Vektorraum mit Skalarprodukt $\langle \cdot, \cdot\rangle$. Dann ist durch

$$\|u\| = \sqrt{\langle u, u\rangle} \tag{1.19}$$

auf V eine Norm definiert. Ein endlichdimensionaler, reeller Vektorraum V mit Skalarprodukt und zugehöriger Norm heißt **euklidischer Vektorraum**.

Beweis. Das Skalarprodukt ist nicht negativ, daher ist die Abbildung $\|\cdot\| : V \to \mathbb{R}$ wohldefiniert. Wir müssen nun die Normeigenschaften nachrechnen. Sei dazu $u \in V$. Es gilt

1. Nichtnegativität und Definitheit folgen sofort aus den entsprechenden Eigenschaften des Skalarprodukts.
2. Homogenität

$$\|\lambda u\| = \sqrt{\langle \lambda u, \lambda u\rangle} = \sqrt{\lambda^2\langle u, u\rangle} = |\lambda|\sqrt{\langle u, u\rangle} = |\lambda|\|u\| \tag{1.20}$$

3. Deiecksungleichung

$$\begin{aligned} \|u + v\|^2 &= \langle u + v, u + v\rangle \\ &= \langle u, u\rangle + 2\langle u, v\rangle + \langle v, v\rangle \\ &\leq \langle u, u\rangle + 2\|u\|\,\|v\| + \langle v, v\rangle \\ &= \|u\|^2 + 2\|u\|\,\|v\| + \|v\|^2 \\ &= (\|u\| + \|v\|)^2 \end{aligned} \tag{1.21}$$

Daraus folgt durch Wurzelziehen auf beiden Seiten $\|u + v\| \leq \|u\| + \|v\|$. Für die Abschätzung in Zeile (1.21) haben wir die Bunyakovsky-Cauchy-Schwarz-Ungleichung aus Lemma 1.2.2 verwendet.

□

1.2.4 Lemma (L^2-Skalarprodukt): Auf dem Raum $V = \mathbb{P}_n$ der reellen Polynome vom Grad bis zu n ist durch

$$\langle p, q\rangle = \int_{-1}^{1} p(x)q(x)\,\mathrm{d}x \tag{1.22}$$

ein Skalarprodukt definiert. Dieses wird auch L^2-Skalarprodukt genannt.

Beweis. Hier gilt es zu prüfen, ob die Abbildung auch die vier Eigenschaften eines Skalarprodukts erfüllt. Seien dazu im Folgenden $p, q, g \in \mathbb{P}_n$. Zunächst zeigen wir die Symmetrie.

$$\langle p, q\rangle = \int_{-1}^{1} p(x)q(x)\,\mathrm{d}x = \int_{-1}^{1} q(x)p(x)\,\mathrm{d}x = \langle q, p\rangle \tag{1.23}$$

Wenn wir nun Bilinearität zeigen, genügt es wegen der Symmetrieeigenschaft, das erste Argument zu untersuchen.

$$\begin{aligned}
\langle \lambda p + \mu q, g\rangle &= \int_{-1}^{1} (\lambda p(x) + \mu q(x))g(x)\,\mathrm{d}x \\
&= \int_{-1}^{1} \lambda p(x)g(x) + \mu q(x)g(x)\,\mathrm{d}x \\
&= \int_{-1}^{1} \lambda p(x)g(x)\,\mathrm{d}x + \int_{-1}^{1} \mu q(x)g(x)\,\mathrm{d}x \\
&= \lambda \int_{-1}^{1} p(x)g(x)\,\mathrm{d}x + \mu \int_{-1}^{1} q(x)g(x)\,\mathrm{d}x \\
&= \lambda\langle p, g\rangle + \mu\langle q, g\rangle
\end{aligned} \tag{1.24}$$

Als letztes zeigen wir, dass die Abbildung positiv definit ist.

$$0 = \langle p, p\rangle = \int_{-1}^{1} p(x)p(x)\,\mathrm{d}x = \int_{-1}^{1} p(x)^2\,\mathrm{d}x \tag{1.25}$$

Aus den Integraleigenschaften folgt

$$0 = p(x)^2 \quad \forall x \tag{1.26}$$

Dies kann nur der Fall sein, wenn $p \equiv 0$ ist.
Somit haben wir nachgerechnet, dass es sich um Skalarprodukt handelt. □

1.2.5 Definition: Nach dem Lemma 1.2.3 können wir mit diesem Skalarprodukt eine Norm auf $\mathbb{P}_n$ definierten. Diese Norm wird als die L^2-Norm bezeichnet.

$$\|f\|_{L^2} = \sqrt{\langle f, f\rangle_{L^2}} = \sqrt{\int_{-1}^{1} f(x)^2 dx} \tag{1.27}$$

1.2.6 Definition: Zwei Vektoren $u, v \in V$ heißen **orthogonal**, wenn

$$\langle u, v\rangle = 0. \tag{1.28}$$

Ein Vektor $u \in V$ ist orthogonal zum Untervektorraum $W \subset V$, wenn

$$\langle u, v\rangle = 0 \quad \forall v \in W. \tag{1.29}$$

1.2.7 Notation: Von nun an bezeichnet V immer einen endlichdimensionalen, reellen, euklidischen Vektorraum.

1.2.8 Lemma (Pythagoras): Seien zwei Vektoren $u \in V$ und $v \in V$ orthogonal zueinander. Dann gilt

$$\|u + v\|^2 = \|u\|^2 + \|v\|^2 \tag{1.30}$$

Beweis. Seien $u, v \in V$. Es gilt $0 = \langle u, v\rangle$

$$\|u + v\|^2 = \langle u + v, u + v\rangle = \|u\|^2 + \|v\|^2 + 2\langle u, v\rangle = \|u\|^2 + \|v\|^2 \tag{1.31}$$

□

1.3 Bestapproximation und orthogonale Projektion

1.3.1 Definition: Sei $A \subset V$ ein affiner Unterraum eines euklidischen Vektorraums. Dann ist die Bestapproximation $u_b \in A$ eines Vektors $u \in V$ in A definiert durch die Beziehung

$$\|u - u_b\| = \min_{v \in A} \|u - v\|. \tag{1.32}$$

1.3.2 Satz: Sei $w \in V$ und $W \subset V$. Sei $A = w + W$ ein nichtleerer, affiner Unterraum von V. Dann existiert die Bestapproximation nach Definition 1.3.1 und ist eindeutig bestimmt. Es gilt die notwendige und hinreichende Bedingung

$$\langle u - u_b, v\rangle = 0 \quad \forall v \in W. \tag{1.33}$$

Das heißt u_b ist Bestapproximation genau dann wenn $u - u_b$ orthogonal zu W bzgl. des Skalarprodukts $\langle \cdot, \cdot \rangle$ ist.

Beweis. Der Beweis gliedert sich in drei Teile. Zuert wird die Äquivalenz gezeigt danach zeigen wir die Eindeutigkeit und zum Schluss erst die Existenz.

„⇒“: Sei $u_b \in A$ die Bestapproximation des Vektors $u \in V$. Wir definieren nun für beliebiges $v \in W$ die Funktion:

$$\begin{aligned} &F_v \colon \mathbb{R} \to \mathbb{R}, \\ &F_v(x) = \|u - u_b + xv\|^2 \end{aligned} \tag{1.34}$$

Da $u_b \in A$ liegt auch $u_b + xv$ in A und folglich besitzt diese Funktion nach Voraussetzung ein Minimum bei $x = 0$. Wir untersuchen daher die Ableitung

$$\begin{aligned} \frac{d}{dx}F(x) &= \frac{d}{dx}\|u - u_b + xv\|^2 && (1.35)\\ &= \frac{d}{dx}\Big(\|u - u_b\|^2 + 2x\langle u - u_b, v\rangle + x^2\|v\|^2\Big) && (1.36)\\ &= 2\langle u - u_b, v\rangle + 2x\|v\|^2 && (1.37) \end{aligned}$$

und es gilt

$$0 = \frac{d}{dx}F(x)\bigg|_{x=0} = 2\langle u - u_b, v\rangle. \tag{1.38}$$

“⇐“: Erfülle nun $u_b \in A$ die Bedingung $\langle u - u_b, v\rangle = 0$ für alle $v \in W$.

$$\begin{aligned} \|u - u_b\|^2 &= \langle u - u_b, u - u_b\rangle && (1.39)\\ &= \langle u - u_b, u - u_b - v\rangle && (1.40)\\ &\leq \|u - u_b\| \cdot \|u - v\|. && (1.41) \end{aligned}$$

Da nun $v \in W$ beliebig und $u_b \in A$, so wird der gesamte affine Unterraum A durch die Terme der Form $u_b + v$ aufgespannt. Daraus folgt

$$\|u - u_b\| \leq \inf_{v \in A} \|u - v\| \tag{1.42}$$

und u_b ist die Bestapproximation.

Nun zur Eindeutigkeit: Seien u_b und $u_d \in$ A zwei Bestapproximationen. Dann gilt notwendigerweise

$$\langle u - u_b, v\rangle = 0 = \langle u - u_d, v\rangle \quad \forall v \in W \tag{1.43}$$

Dies wird umgeformt zu

$$\langle u - u_d, v\rangle - \langle u - u_b, v\rangle = 0 \qquad \forall v \in W \tag{1.44}$$

$$\Leftrightarrow \qquad \langle u_b - u_d, v\rangle = 0 \qquad \forall v \in W \tag{1.45}$$

Da $u_b, u_d \in A$ folgt $u_b - u_d \in W$. Daher dürfen wir oben $v = u_b - u_d$ einsetzen. Dies ergibt $\|u_b - u_d\|^2 = 0$ und somit folgt $u_b = u_d$.

Schließlich die Existenz: Der endliche dimensionale Teilraum $W \subseteq V$ besitzt eine Basis $(\varphi_1, \ldots, \varphi_n)$ mit $n = \dim W$. Die gesuchte Approximation $u_b \in A$ lässt sich durch die Basis in folgender Form darstellen

$$u_b = w + \sum_{k=1}^{n} x_k \varphi_k \tag{1.46}$$

Dies wird in die notwendige Orthogonalitätsbedingung Satz 1.3.2 eingesetzt.

$$\left\langle u - w - \sum_{k=1}^{n} x_k \varphi_k, v \right\rangle = \langle u - w, v\rangle - \sum_{k=1}^{n} x_k \langle \varphi_k, v\rangle = 0 \quad \forall v \in W. \tag{1.47}$$

Dies ist durch Einsetzen von $v := \varphi_i$ für $i = 1, \ldots, n$ äquivalent zu einem linearen Gleichungssystem $\mathbf{Gx} = \mathbf{b}$ mit Matrix und rechter Seite

$$\mathbf{G} = \big(\langle \varphi_k, \varphi_i\rangle\big)_{i,k=1}^{n} \qquad \mathbf{b} := \big(\langle u - w, \varphi_i\rangle\big)_{i=1}^{n} \tag{1.48}$$

Da die Matrix A quadratisch ist, folgt aus der Eindeutigkeit die Existenz. □

1.3.3 Definition: Sei $W \subset V$ ein Untervektorraum. Dann gilt $V = W \oplus W^{\perp}$, wobei das **orthogonale Komplement** $W^{\perp}$ eindeutig definiert ist durch

$$W^{\perp} = \big\{ v \in V \big| \langle v, w\rangle = 0 \quad \forall w \in W \big\}. \tag{1.49}$$

Die Lösung der Bestapproximationsaufgabe bezeichnen wir mit

$$\Pi_W u = u_b \in W \tag{1.50}$$

und nennen es die **orthogonale Projektion** von $u \in V$ auf W.

1.3.4 Lemma: Das orthogonale Komplement und die orthogonale Projektion sind wohldefiniert.

Beweis. Satz 1.3.2. □

1.3.5 Beispiel: Die Aufgabe der Gaußschen Ausgleichsrechnung (in L^2) lautet: finde zu einer gegebenen Funktion f das Polynom vom Grad höchstens n, das auf dem Intervall $[-1, 1]$ den mittleren quadratischen Abstand minimiert, also $p \in \mathbb{P}_n$ mit

$$\int_{-1}^{1} \big(f(x) - p(x)\big)^2 \,\mathrm{d}x = \min_{q \in \mathbb{P}_n} \int_{-1}^{1} \big(f(x) - q(x)\big)^2 \,\mathrm{d}x. \tag{1.51}$$

Die Lösung erfüllt

$$\int_{-1}^{1} p(x)q(x) \,\mathrm{d}x = \int_{-1}^{1} f(x)q(x) \,\mathrm{d}x \qquad \forall q \in \mathbb{P}_n. \tag{1.52}$$

Bemerkung 1.3.6. Nach Satz 1.3.2 ist die Aufgabe der Gaußschen Ausgleichsrechnung äquivalent zur Minimierungsaufgabe ein $p \in \mathbb{P}_n$ zu finden, so dass

$$\|f - p\|_{L^2}^2 = \min_{q \in \mathbb{P}_n} \|f - q\|_{L^2}^2. \tag{1.53}$$

Bemerkung 1.3.7. Die Formulierung in Beispiel 1.3.5 ist die mathematische Überspitzung einer häufigen Aufgabe der Wissenschaft: gegeben N Messwerte (x_i, f_i), wie kann ich eine Funktion $f(x)$ finden, die die Messwerte im quadratischen Mittel am besten approximiert. Schränken wir die Suche nach der Funktion f auf den Polynomraum $\mathbb{P}_n$ mit $n < N$ ein, so erhalten wir die diskrete Variante der Gaußschen Ausgleichsrechnung: finde $p \in \mathbb{P}_n$, so dass

$$\sum_{i=1}^{N} \big(f_i - p(x_i)\big)^2 = \min_{q \in \mathbb{P}_n} \big(f_i - q(x_i)\big)^2. \tag{1.54}$$

Ein Spezialfall ist $n = N-1$ und wir werden im Kapitel zur Interpolation sehen, dass dort das Minimum zu null wird. Dort werden wir auch sehen, dass

$$\langle p, q \rangle = \sum_{i=1}^{N} p(x_i)q(x_i) \tag{1.55}$$

ein Skalarprodukt auf $\mathbb{P}_n$ mit $n < N$ darstellt.

1.4 Orthogonale Basen

1.4.1 Lemma: Wählt man eine Basis $\{\varphi_i\}$ für W, so transformiert sich die Orthogonalitätsbedingung in Satz 1.3.2 zum linearen Gleichungssystem

$$\mathbf{Gx} = \mathbf{b}. \tag{1.56}$$

Hier sind $\mathbf{x}$ der Koeffizientenvektor der Lösung u_b, $\mathbf{G}$ die **Gramsche Matrix** und $\mathbf{b}$ die rechte Seite gegeben durch

$$g_{ij} = \langle \varphi_i, \varphi_j \rangle, \qquad b_i = \langle u, \varphi_i \rangle. \tag{1.57}$$

Wir haben hier o.B.d.A. den affinen Unterraum A durch den Untervektorraum W ersetzt.

Bemerkung 1.4.2. Das Gleichungssystem hängt nur von der Wahl einer Basis in W ab, nicht in V.

1.4.3 Definition: Eine Menge von Vektoren $\{\varphi_1, \ldots, \varphi_n\} \subset V$ bildet ein **Orthogonalsystem**, wenn

$$\langle \varphi_i, \varphi_j \rangle = 0 \qquad \forall 1 \leq i < j \leq n.$$

Sie ist ein **Orthonormalsystem**, wenn zusätzlich $\|\varphi_i\| = 1$ für alle Elemente gilt. Ein Orthonormalsystem, das eine Basis bildet, heißt **Orthonormalbasis** (**ONB**).

1.4.4 Lemma: Jedes Orthogonalsystem ist linear unabhängig.

1.4.5 Lemma (Parsevalsche Gleichung): Sei $\{\varphi_i\}$ für $i = 1, \ldots, n$ eine ONB von V. dann gilt für jedes $v \in V$ mit der Basisdarstellung

$$v = \sum_{i=1}^{n} x_i \varphi_i \tag{1.58}$$

die Identität

$$\|v\|^2 = \sum_{i=1}^{n} x_i^2. \tag{1.59}$$

1.4.6 Aufgabe: Zeigen Sie Lemma 1.4.4 und die Parsevalsche Gleichung.

1.4.7 Bemerkung: Bezüglich einer ONB ist die Gramsche Matrix die Einheitsmatrix. Damit berechnen sich die Einträge des Koeffizientenvektors $\mathbf{x}$ in Lemma 1.4.1 durch die einfache Formel

$$x_i = b_i = \langle u, \varphi_i \rangle. \tag{1.60}$$

1.4.8 Theorem (Gram-Schmidt-Verfahren): Jede linear unabhängige Menge von Vektoren $\{v_1, \ldots, v_n\} \subset V$ wird mit dem folgenden Verfahren in ein Orthonormalsystem $\{\varphi_1, \ldots, \varphi_n\} \subset V$ umgeformt:

$$\begin{aligned} &\varphi_1 = \tfrac{1}{\|v_1\|}\, v_1 \\ &w_j = v_j - \sum_{i=1}^{j-1} \langle v_j, \varphi_i \rangle\, \varphi_i \qquad \varphi_j = \tfrac{1}{\|w_j\|}\, w_j \qquad j = 2, \ldots, n \end{aligned} \tag{1.61}$$

Für alle $1 \leq k \leq n$ gilt

$$\operatorname{span}\{\varphi_1, \ldots, \varphi_k\} = \operatorname{span}\{v_1, \ldots, v_k\} \tag{1.62}$$

Beweis. Per Induktion über n zeigen wir Orthogonalität und Normierung.

Indukionsanfang: Sei $n = 1$. Wird nur ein Vektor aus dem Raum gewählt, so erfüllt dieser die Orthogonalitätsbedingung, da er der einzige Vektor im System ist. Wird dieser Vektor zusätzlich normiert erhält man ein Orthonormalsystem.

Induktionsschritt: Die Aussage gelte für $\{v_1, \ldots, v_{n-1}\}$ und das Orthonormalsystem $(\varphi_1, \ldots, \varphi_{n-1})$. Zunächst zeigen wir: φ_n ist wohldefiniert, also $w_n \neq 0$. Wäre dies nicht so, so gälte

$$w_n = v_n - \sum_{i=1}^{n-1} \langle v_n, \varphi_i \rangle\, \varphi_i = 0 \tag{1.63}$$

Es ist also $v_n \in \operatorname{span}\{\varphi_1, \ldots, \varphi_{n-1}\}$ und daher ist die Menge $(v_1, \ldots, v_n)$ linear abhängig. Das ist ein Widerspruch zur Voraussetzung.

w_n wird nun normiert zu $\varphi_n = \frac{1}{\|w_n\|} w_n$.

Schließlich gilt

$$\langle \varphi_n, \varphi_j \rangle = \langle v_n, \varphi_j \rangle - \sum_{i=1}^{n-1} \langle v_n, \varphi_i \rangle \underbrace{\langle \varphi_i, \varphi_j \rangle}_{=\delta_{ij}} = 0 \qquad j = 1, \ldots, n-1 \tag{1.64}$$

und damit die Orthogonalität. □

1.4.9 Algorithmus (Gram-Schmidt):

```
1  def gram_schmidt(v):
2    (n,m) = v.shape
3    for j in range(n):
4      delta = np.zeros(m)
5      for i in range(j):
6        r = sprod(v[:,j],v[:,i])
7        delta += r*v[:,i]
8      v[:,j] -= delta
9      norm = np.sqrt(sprod(v[:,j],v[:,j]))
10     v[:,j] /= norm
```

1.4.10. Da dies der erste Algorithmus in dieser Vorlesung ist, erläutern wir das Programm im Detail und gehen etwas auf die Syntax von Python ein. Ebenso sollte der Code mit einem einfachen Beispiel probiert werden, um die Parallelen zum Verfahren besser zu erkennen.

1. Es wird eine Funktion mit dem Namen gram_schmidt. Dieser Funktion wird eine Matrix v übergeben deren Spalten die Vektoren v_1 bis v_n sind, die wir orthogonalisieren wollen.
2. Es wird n die Länge der Zeilen (Anzahl der Vektoren) zugewiesen und m wird die Länge der Spalten (Anzahl der Einträge im Vektor) zugewiesen.
3. Beginn des Gram-Schmidt-Verfahrens und der Schleife über die Vektoren von 1 bis n.
4. Initialsieren eines Vektors δ der Länge m mit Nullen als Einträge
5. Es wird eine weitere Schleife begonnen in der ein Index i über alle bisher orthogonalisierten Vektoren läuft. Dies entspricht der Summe aus dem Verfahren. Beachten Sie, dass diese Schleife für $j = 0$ nicht ausgeführt wird.
6. r ist das Skalarprodukt aus dem Vektor v_j und einem bereits orthogonalisierten Vektor v_i. Die Vektoren befinden sich in der Matrix v und über diesen Befehl wird darauf zugegriffen.
7. Addiere zu δ die Projektion von V_j auf v_i.
8. Hier endet die zweite for-Schleife durch reduktion der Einrückung. Die Summe wird vom Vektor v_j abgezogen, somit v_j orthogonalisiert und wieder in der Matrix v an der richtigen Stelle zugewiesen.
9. Die Norm von v_j wird berechnet.
10. Wie im Verfahren wird in dieser Zeile v_j normiert und in der Matrix v an der Stelle des früheren v_j zugewiesen.

1.4.11 Beispiel: Wir wählen für Polynome das L^2-Skalarprodukt aus Lemma 1.2.4 und die Basis $\{1, x, \ldots, x^{n-1}\}$ für $\mathbb{P}_{n-1}$. Wir verwenden die Iplementation in Algorithmus 1.4.9 und messen den Erfolg nach der Größe der Nebendiagonaleinträge der Gramschen Matrix.

n	$\max_{i\neq j}\lvert g_{ij}\rvert$
5	$8.9 \cdot 10^{-16}$
10	$9.1 \cdot 10^{-12}$
15	$1.2 \cdot 10^{-7}$
20	0.23

1.4.12 Algorithmus (Modifizierter Gram-Schmidt):

```
def modified_gram_schmidt(v):
    (n,m) = v.shape
    for j in range(m):
        for i in range(j):
            r = sprod(v[:,j],v[:,i])
            v[:,j] -= r*v[:,i]
        norm = np.sqrt(sprod(v[:,j],v[:,j]))
        v[:,j] /= norm
```

Bemerkung 1.4.13. In diesem Programm wurde der Zwischenschritt über den Vektor δ ausgelassen.

1.4.14 Beispiel: In dieser Tabelle wiederholen wir die Zahlen $\max_{i\neq j}\lvert g_{ij}\rvert$ aus Beispiel 1.4.11 und stellen sie den entsprechenden Ergebnissen des modifizierten Verfahrens in Algorithmus 1.4.12 gegenüber.

n	Gram-Schmidt	modifiziert
5	$8.9 \cdot 10^{-16}$	$1.3 \cdot 10^{-16}$
10	$9.1 \cdot 10^{-12}$	$2.9 \cdot 10^{-12}$
15	$1.2 \cdot 10^{-7}$	$2.7 \cdot 10^{-9}$
20	0.23	$3.9 \cdot 10^{-5}$

Bemerkung 1.4.15. Wir sehen, dass die Wahl der Implementation eines Rechenverfahrens bei mathematischer Äquivalenz durchaus erheblichen Einfluss auf das Ergebnis haben kann. Dieses Phänomen werden wir in Kapitel 2 näher untersuchen. Zunächst diskutieren wir aber eine weitere Variante der Erzeugung orthogonaler Basen in Polynomräumen.

1.5 Drei-Term-Rekursion

1.5.1 Satz (Dreiterm-Rekursion): Zu jedem Skalarprodukt $\langle \cdot, \cdot \rangle$ auf dem Raum der stetigen Funktionen mit der Eigenschaft, dass für alle Polynome p, q

$$\langle xp, q \rangle = \langle p, xq \rangle \tag{1.65}$$

gilt, gibt es genau eine Folge von orthogonalen Polynomen $\{p_k\}_{k=0,\dots}$ wobei $p_k \in \mathbb{P}_k$ mit führendem Koeffizienten eins. Sie genügen der Dreiterm-Rekursionsformel

$$p_k(x) = (x - a_k)p_{k-1}(x) - b_k p_{k-2}(x), \qquad k = 1, 2, \dots \tag{1.66}$$

mit Startwerten $p_{-1} \equiv 0$ und $p_0 \equiv 1$. Die Koeffizienten sind

$$a_k = \frac{\langle xp_{k-1}, p_{k-1} \rangle}{\langle p_{k-1}, p_{k-1} \rangle} \qquad \text{und} \qquad b_k = \frac{\langle p_{k-1}, p_{k-1} \rangle}{\langle p_{k-2}, p_{k-2} \rangle}. \tag{1.67}$$

Beweis. Siehe auch [Deuflhard and Hohmann, 2008, Satz 6.2] und [Rannacher, 2017, Satz 2.17].

Wir beginnen mit der Eindeutigkeit: wenn p_k orthogonal zu allen vorherigen Folgengliedern ist, so ist es auch orthogonal zu allen ihren Linearkombinationen und daher gilt $p_k \perp \mathbb{P}_{k-1}$. Nach Lemma 1.3.4 ist das orthogonale Komplement von $\mathbb{P}_{k-1}$ in $\mathbb{P}_k$ wohldefiniert und nach der Dimensionsformel ist seine Dimension eins. Daher unterscheiden sich alle Polynome dort nur um einen skalaren Faktor und durch die Normierung wird p_k eindeutig.

Wir zeigen nun per inductionem, dass die erzeugte Folge eine Orthogonalfolge ist. Da alle Vektoren zum Nullvektor orthogonal ist, gilt die Aussage für die beiden Startpolynome. Sei sie nun bis p_{k-1} bewiesen. Dann gilt zunächst mit der Wahl von a_k:

$$\begin{aligned} \langle p_k, p_{k-1} \rangle &= \langle xp_{k-1}, p_{k-1} \rangle - a_k \langle p_{k-1}, p_{k-1} \rangle - b_k \langle p_{k-2}, p_{k-1} \rangle && (1.68)\\ &= \langle xp_{k-1}, p_{k-1} \rangle - \langle xp_{k-1}, p_{k-1} \rangle - 0 && (1.69)\\ &= 0. && (1.70) \end{aligned}$$

Nach der Wahl von b_k erhalten wir:

$$\begin{aligned} \langle p_k, p_{k-2} \rangle &= \langle xp_{k-1}, p_{k-2} \rangle - a_k \langle p_{k-1}, p_{k-2} \rangle - b_k \langle p_{k-2}, p_{k-2} \rangle && (1.71)\\ &= \langle xp_{k-1}, p_{k-2} \rangle - 0 - \langle p_{k-1}, p_{k-1} \rangle && (1.72)\\ &= \langle p_{k-1}, xp_{k-2} - p_{k-1} \rangle. && (1.73) \end{aligned}$$

Da die führenden Koeffizienten beider Polynome zur rechten eins sind, ist die Differenz in $\mathbb{P}_{k-2}$ und damit nach Induktionsannahme orthogonal zu p_{k-1}. Schließlich gilt für $j < k-2$:

$$\langle p_k, p_j \rangle = \langle x p_{k-1}, p_j \rangle - a_k \langle p_{k-1}, p_j \rangle - b_k \langle p_{k-2}, p_j \rangle \tag{1.74}$$
$$= \langle p_{k-1}, x p_j \rangle - 0 - 0. \tag{1.75}$$

Dieser Term verschwindet, da $x p_j \in \mathbb{P}_{k-2}$ und damit orthogonal zu p_{k-1}. □

1.5.2 Bemerkung: Der Beweis ergibt eigentlich die "Eindeutigkeit einer Orthogonalfolge bis auf Normierung". Tatsächlich werden in der Literatur immer wieder veschiedene Normierungen benutzt. Beispiele sind:

1. $p_k(1) = 1$
2. $\|p_k\| = 1$
3. Führender Koeffizient eins, $p_k = x^k + \ldots$

1.5.3 Definition: Die **Legendre-Polynome** L_k sind definiert durch die Dreiterm-Rekursion

$$L_k(x) = \tfrac{2k-1}{k} x L_{k-1}(x) - \tfrac{k-1}{k} L_{k-2}(x) \tag{1.76}$$

mit den Startbedingungen $L_0(x) = 1$ und $L_1(x) = x$. Sie sind orthogonal bezüglich des L^2-Skalarprodukts in Lemma 1.2.4.

1.5.4 Beispiel: Das Problem der Gaußschen Ausgleichsrechnung war: Zu einer gegebenen Funktion f finde $p \in \mathbb{P}_n$, so dass

$$\|f - q\|_{L^2}^2 = \min_{q \in \mathbb{P}_n} \|f - q\|_{L^2}^2. \tag{1.77}$$

Mit Hilfe der Legendre-Polynome können wir nun die Lösung explizit angeben als

$$p(x) = \sum_{i=0}^{n} \alpha_i L_i(x) \qquad \text{mit} \qquad \alpha_i = \frac{1}{\|L_i\|^2} \int_{-1}^{1} f L_i(x)\, \mathrm{d}x. \tag{1.78}$$

1.5.5 Aufgabe: Oft benötigt man Legendre-Polynome nicht auf dem Intervall $[-1, 1]$, sondern auf einem anderen Intervall $[a, b]$. Man spricht dann auch von *shifted Legendre polynomials*. Während die abstrakte Formel in Satz 1.5.1 natürlich weiter gültig bleibt, ändern sich dabei die Skalarprodukte und damit die Koeffizienten in Definition 1.5.3
Berechnen Sie die ersten drei Polynome für das Intervall $[0, 1]$ mit der Normierung $L_k(1) = 1$.

1.5.6 Definition: Die **Tschebyscheff-Polynome** T_k sind definiert durch die Dreiterm-Rekursion

$$T_k = 2xT_{k-1}(x) - T_{k-2}(x). \tag{1.79}$$

Sie sind orthogonal bezüglich des Skalarprodukts

$$\langle p, q\rangle = \int_{-1}^{1} \tfrac{1}{\sqrt{1-x^2}}\, p(x)q(x)\,\mathrm{d}x. \tag{1.80}$$

1.5.7 Aufgabe: Zeigen Sie durch vollständige Induktion, dass die Teschebyscheff-Polynome der Darstellung

$$T_k(x) = \cos(k \arccos x) \tag{1.81}$$

genügen.

Kapitel 2

Konditionierung und Stabilität

2.1 Fließkommazahlen

2.1.1. Die Menge der reellen Zahlen ist nicht abbildbar im Rechner mit endlichem Speicher. Wir betrachten deshalb eine neue Art Zahlen darzustellen, eine die für den Rechner geeignet ist.

2.1.2 Definition: Die Darstellung einer numerischen Größe als **Fließkommazahl** (auch **Gleitkommazahl**) $x \in \mathbb{R}$ im Rechner beruht auf einer Basis $2 \leq b \in \mathbb{N}$. Desweiteren besteht sie aus einer **Mantisse** $1/b \leq m < 1$ und einem Exponenten t, so dass x die folgende Gestalt hat

$$x = \pm m \cdot b^t \tag{2.1}$$

Diese Darstellung ist mit der Normierung von $m \neq 0$ für $x \neq 0$ eindeutig. Für $x = 0$ wird festgesetzt, dass $m = 0$ gilt. Sowohl die Mantisse, als auch der Exponent haben einen endlichen Wertebereich, typischerweise eine feste Anzahl von Stellen. Die endliche Menge der damit darstellbaren Zahlen bezeichnen wir mit $\mathbb{M}$.

2.1.3. Die folgenden drei Beispiele beschreiben Teile der Implementation von Fließkommazahlen nach dem IEE-Standard 754. Die Quelle ist jeweils Wikipedia.

Es scheint jeweils ein Bit zuviel aufgelistet. Das liegt daran, dass das führende Bit der Mantisse immer eins ist und daher nicht gespeichert werden muss.

Aus dem Wertebereich des Exponenten werden zwei Werte reserviert, die Zahlen repräsentieren, die in der normalisierten Darstellung nicht verfügbar sind. Dabei handelt es sich insbesondere um die null, die Wert $\pm\infty$ und den Wert „NaN“, was für „not a number“ steht und das Resultat fehlerhafter Berechnung signalisiert.

2.1.4 Beispiel: Im Fließkommaformat mit 64 Bit (NumPy: `float64`) nach dem Standard IEEE 754, das auf Rechnern sehr weit verbreitet ist, wird die Basis 2 verwendet. Es hat

- 1 Bit Vorzeichen,
- 11 Bit Exponent und
- 53 Bit Mantisse (das erste ist immer 1 und wird nicht gespeichert)

Der Wertebereich ist zunächst

$$\left.\begin{matrix} 2^{-1022} \\ \approx 2.25 \cdot 10^{-308} \end{matrix}\right\} \leq x \leq \left\{\begin{matrix} 2^{1023}(2 - 2^{-52}) \\ \approx 1.8 \cdot 10^{308} \end{matrix}\right. \tag{2.2}$$

Tatsächlich liegt das Minimum durch Verkürzung der Mantisse bei 2^{-1074}. Zusätzlich gibt es Darstellungen für ± 0, unendlich, und illegale Zahlen.

2.1.5 Beispiel: Das Format mit 32 Bit (NumPy: `float32`) nach IEEE 754 hat

- 1 Bit Vorzeichen,
- 8 Bit Exponent und
- 24 Bit Mantisse.

2.1.6 Beispiel: Das Format mit 16 Bit (NumPy: `float16`) nach IEEE 754 hat

- 1 Bit Vorzeichen,
- 5 Bit Exponent und
- 11 Bit Mantisse.

2.1.7 Definition: Zahlen, die durch die endliche Mantisse nicht dargestellt werden können, unterliegen der **Rundung** auf eine benachbarte Fließkommazahl, notiert als $\mathrm{rd}(x)$. Besitzt die Mantisse r Stellen zum Exponenten b, so ist der relative Fehler, der dabei entsteht beschränkt durch b^{1-r} bei Rundung zur nächsten Fließkommazahl sogar durch $\frac{1}{2}b^{1-r}$.
Wir bezeichnen das Maximum des möglichen relativen Rundungsfehlers für ein Fließkommaformat als **Maschinengenauigkeit**, abgekürzt mit `eps`. Es gilt also per definitionem

$$\left|\frac{x - \mathrm{rd}(x)}{x}\right| \leq \texttt{eps}. \tag{2.3}$$

2.1.8 Beispiel: Bei den Fließkommaformaten nach IEEE 754 gilt die Rundung zur nächsten darstellbaren Zahl. Sollte eine Zahl exakt zwischen zwei darstellbaren Zahlen liegen, so wird zur nächsten darstellbaren Zahl mit gerader Mantisse gerundet.
Die Maschinengenauigkeit liegt bei

Format	`eps`	
`float64`	2^{-53}	$\approx 1.11 \cdot 10^{-16}$
`float32`	2^{-24}	$\approx 5.96 \cdot 10^{-8}$
`float16`	2^{-11}	$\approx 4.88 \cdot 10^{-4}$

2.1.9 Definition: Die Implementation der Grundrechenarten für Fließkommazahlen beinhaltet immer eine Rundung, damit das Ergebnis darstellbar ist. Wir kennzeichnen diese **Maschinenoperationen** für $x, y \in \mathbb{M}$ mit den Symbolen

$$x \oplus y = \mathrm{rd}(x + y) \qquad x \odot y = \mathrm{rd}(xy) \tag{2.4}$$
$$x \ominus y = \mathrm{rd}(x - y) \qquad x \oslash y = \mathrm{rd}(x/y). \tag{2.5}$$

2.1.10 Lemma: Die Maschinenoperation $\oplus$ und $\odot$ sind weder assoziativ noch distributiv, wenn auch die Unterschiede nur in der Größenordnung der Maschinengenauigkeit `eps` liegen.

Beweis. Die Rundung am Ende einer Operation kann so verstanden werden, dass jeweils ein unbekannter, relativer Fehler $|\varepsilon| \leq$ `eps` zum Ergebnis addiert

wird. Damit gilt

$$\begin{aligned}(x \oplus y) \oplus z &= \big((x+y)(1+\varepsilon_1)+z\big)(1+\varepsilon_2)\\ &= (x+y+z+\varepsilon_1 x+\varepsilon_1 y)(1+\varepsilon_2)\\ x \oplus (y \oplus z) &= (x+y+z+\varepsilon_3 y+\varepsilon_3 z)(1+\varepsilon_4).\end{aligned} \tag{2.6}$$

Selbst wenn die Werte der ε_i alle etwa gleich groß sind, so differieren doch die Fehler, wenn x und z sehr verschieden sind. Die Rechnungen für die Multiplikation und das Distributivgesetz sind ähnlich. □

Bemerkung 2.1.11. Das vorherige Lemma gibt einen ersten Hinweis, warum die beiden Varianten des Gram-Schmidt-Verfahrens sich so verschieden verhalten.

2.1.12 Beispiel (Harmonische Reihe in Fließkommaarithmetik):

```
1 sum = np.float16(1.0)
2 i = 1.
3 old = np.float16(0.0)
4 while (sum != old):
5     old = sum
6     i += 1.
7     diff = np.float16(1./i)
8     sum += diff
9 print (sum)
```

Bricht das Programm ab? Begründen Sie dies.

Bemerkung 2.1.13. Die Harmonische Reihe ist folgende Summe $1+\frac{1}{2}+\frac{1}{3}+\cdots$
1 Das ist eine Möglichkeit in Python einer Variable eine Zahl zuzuweisen. Hier werden nochmals explizite Angaben zur Art der Zahl gemacht.
2 Hier wird der Variable i auf eine andere Weise die 1 zugewiesen.
3 Wi initialisieren die Variable *old* mit 0
4 Wir springen in eine while- Schleife, die solange den darunterstehenden Code ausführt, bis die Bedingung in den Klammern nicht mehr erfüllt ist.
5 Hier wird *old* der momentane Wert der Summe zugewiesen.
6 Im Code steht eigentlich $i = i + 1$, also i wird um 1 erhöht.
7 Der Variable $diff$ wird das nächste Glied in der Summe zugewiesen.
8 Die Summe wird um ein weiteres Glied erweitert. Die Schleife spring nach oben und testet zuerst ob die Bedingung noch erfüllt ist.

2.1.14 Aufgabe: Schreiben Sie ein Programm, das bis auf 10% Genauigkeit die kleinsten Zahlen a und b ermittelt, so dass $1.0 + a = 1.0$ und $1.9 + b = 1.9$. Bestimmen Sie damit `eps` für mindestens eines der IEEE 754 Fließkommaformate.

2.1.15 Fazit:

1. Fließkommazahlen haben endlichen Wertebereich
2. Die Eingabe reeller Zahlen sowie die Ergebnisse von Rechenoperationen werden durch Rundung verfälscht.
3. Rundungsfehler sind relative Fehler beschränkt durch die Maschinengenauigkeit `eps`
4. Grundrechenarten mit Fließkommazahlen sind nicht assoziativ

2.2 Konditionierung einer Rechenaufgabe

2.2.1. In diesem Abschnitt nehmen wir zunächst an, die Berechnungen seien exakt und nur die Eingabedaten durch Rundung verfälscht. Daraufhin untersuchen wir, wie stark sich die Lösung einer Rechenaufgabe abhängig von Variationen der Eingabedaten verändert.

2.2.1 Einführung der Konditionierung

2.2.2 Definition: Eine **numerische Aufgabe** ist die Berechnung endlich vieler **Ausgabedaten** y_i, $i = 1, \ldots, n$ aus ebenfalls endlich vielen Eingabedaten x_j, $j = 1, \ldots, m$. Wir schreiben

$$y_i = f_i(x_1, \ldots, x_m). \tag{2.7}$$

Zur Lösung der Aufgabe verwenden wir als Rechenvorschrift einen **Algorithmus**, bzw. seine Implementation f auf einem Computer.

2.2.3 Definition: Aus der Verwendung fehlerhafter Eingabedaten $x+\delta x$ ergeben sich fehlerhafte Resultate $y + \delta y$. Mit δx und δy bezeichnen wir die **absoluten Fehler**. Die **relativen Fehler** sind $\delta x/\|x\|$, und $\delta y/\|y\|$ bzw. $\delta x_j/|x_j|$ und $\delta y_i/|y_i|$.
Eine numerische Aufgabe heißt **gut konditioniert**, wenn es eine moderate Konstante κ bzw. Konstanten κ_{ij} gibt, so dass die Abschätzung

$$\frac{\|\delta y\|}{\|y\|} \leq \kappa \frac{\|\delta x\|}{\|x\|} \qquad \text{bzw.} \qquad \frac{|\delta y_i|}{|y_i|} \leq \kappa_{ij} \frac{|\delta x_j|}{|x_j|} \tag{2.8}$$

für den bestmöglichen Algorithmus zur Lösung der Aufgabe gilt. Andernfalls heißt sie **schlecht konditioniert**.

Bemerkung 2.2.4. Die Begriffe „gut“, bzw. „schlecht konditioniert“ sind nicht scharf definiert. In der Tat hängt die Grenze, ab der die Konstante κ nicht mehr als „moderat“ angesehen wird, von außermathematischen Faktoren wie den Ansprüchen der Anwendung oder dem persönlichen Geschmack des Anwenders ab. Dennoch werden wir uns nun um eine Quantifizierung der Konditionierung bemühen, die bei der Entscheidung, ob eine Aufgabe berechenbar ist, helfen kann.

Bemerkung 2.2.5. Von entscheidender Bedeutung ist, dass die Konditionierung einer numerischen Aufgabe das Optimum über alle Algorithmen ist und damit vom konkreten Algorithmus unabhängig. Die ungeschickte Wahl eines Verfahrens führt natürlich zu einer schlechteren Konstanten in der Konditionsabschätzung.

2.2.2 Differenzielle Fehleranalyse

2.2.6. Besonders einfach lassen sich die Relationen zwischen den Fehlern der Eingabe- und Ausgabedaten über Ableitungen der Funktion f in Definition 2.2.2 beschreiben. Für diesen Fall stehen uns alle Rechenregeln wie Ketten- und Produktregel oder der Satz von Taylor zur Verfügung. Natürlich gelten die Aussagen dann nur asymptotisch für `eps` $\to 0$.

Andererseits ist `eps` in der Regel sehr klein, weshalb die asymptotische Analyse oft hinreichend genau ist. Und schließlich bemühen wir uns, wo immer möglich, gesicherte Scharanken einzubauen.

2.2.7 Definition: Zur quantitativen Beschreibung von Grenzprozessen dienen die **Landauschen Symbole** $\mathcal{O}(\cdot)$ und $o(\cdot)$. Für Folgen/Funktionen $f(x)$ und $g(x)$ bedeuten

$$f = o(g) \qquad :\Leftrightarrow \qquad \lim_{x\to a} \frac{|f(x)|}{|g(x)|} = 0 \tag{2.9}$$

$$f = \mathcal{O}(g) \qquad :\Leftrightarrow \qquad \operatorname{limsup}_{x\to a} \frac{|f(x)|}{|g(x)|} < \infty. \tag{2.10}$$

Dabei darf a eine feste Zahl oder den Limes gegen $\pm\infty$ bezeichnen. Zusätzlich definieren wir **gleich in erster Näherung**

$$f \doteq g \qquad :\Leftrightarrow \qquad f(t) = g(t) + o(1), \tag{2.11}$$

sowie analog $\preccurlyeq$ und $\succcurlyeq$.

Bemerkung 2.2.8. Typischerweise wird bei der Schreibweise mit Landauschen Symbolen implizit eine Konvergenz für $t \to 0$, $t \to \infty$ oder zum Beispiel $h \to 0$ und $n \to \infty$ angenommen. Diese erschließt sich aus dem Sinn.

Die Definition von $o(\cdot)$ impliziert, dass die Konvergenz $f(t) \to 0$ durch

$$f(t) = o(1) \tag{2.12}$$

dargestellt wird. Hier insbesondere ist der Schluss aus dem Sinn schwierig, da die unabhängige Variable nicht im o-Ausdruck erscheint.

2.2.9 Beispiel: Als Definition der Ableitung der Funktion f im Punkt x kennen wir

$$\lim_{h\to 0} \frac{f(x+h) - f(x)}{h} = f'(x). \tag{2.13}$$

In unserer Schreibweise

$$\begin{aligned} f(x+h) - f(x) - hf'(x) &= o(h) \\ \frac{f(x+h) - f(x)}{h} - f'(x) &= o(1) \qquad \text{für } h \to 0 \\ \frac{f(x+h) - f(x)}{h} &\doteq f'(x) \qquad \text{für } h \to 0. \end{aligned} \tag{2.14}$$

2.2.10 Beispiel: Nach dem Satz von Taylor gilt für eine zweimal stetig differenzierbare Funktion f mit $\xi \in (x, x+h)$

$$f(x+h) = f(x) + hf'(x) + \tfrac{h^2}{2} f''(\xi) \tag{2.15}$$

Damit können wir schreiben

$$\begin{aligned} f(x+h) - f(x) &= \mathcal{O}(h) \\ f(x+h) - f(x) &= hf'(x) + \mathcal{O}(h^2). \end{aligned} \tag{2.16}$$

Oder

$$\frac{f(x+h) - f(x)}{h} \doteq f'(x), \tag{2.17}$$

wobei wir im letzten Beispiel Information veschenkt haben.

2.2.11 Lemma: Sei die Funktion f in Definition 2.2.2 stetig differenzierbar um das Datum x. Dann gilt für die relativen Fehler

$$\frac{\delta y_i}{y_i} \doteq \sum_{j=1}^{m} \kappa_{ij} \frac{\delta x_j}{x_j}$$

mit den **Konditionszahlen**

$$\kappa_{ij} = \frac{\partial f_i}{\partial x_j}(x) \frac{x_j}{y_i} \tag{2.18}$$

Beweis. Wir betrachten die Funktion

$$g(h) = f_i(x + h\delta x), \tag{2.19}$$

wobei i im Bereich $[1, m]$ liegt und x beziehungsweise δx die vektorwertige Eingabe und ihr Fehler sind. Für stetig differenzierbares g gilt

$$g(h) = g(0) + hg'(0) + o(h). \tag{2.20}$$

Die Ableitung von g lässt sich aber nach der Kettenregel berechnen:

$$g'(0) = \sum_{j=1}^{n} \frac{\partial f_i(x)}{\partial x_j} \delta x_j. \tag{2.21}$$

Diese ist nach Voraussetzung stetig und somit darf g stetig differenzierbar angenommen werden. Schließlich gilt

$$\begin{aligned} y_i + \delta y_i &= g(1) \\ &\doteq g(0) + g'(0) \\ &= f_i(x) + \sum_{j=1}^{n} \frac{\partial f_i(x)}{\partial x_j} \delta x_j. \end{aligned} \tag{2.22}$$

Damit haben wir den absoluten Fehler des Resultats durch den absoluten Fehler der Eingabedaten abgeschätzt. Der Rest ergibt sich durch Division.

Beachten Sie, dass wir auch in diesem Beweis die Komplikation eines mehrdimensionalen Arguments durch die Einführung der Hilfsfunktion g umschifft haben. □

2.2.12 Beispiel (Konditionierung der Multiplikation): Es gilt

$$y_1 = f(x_1, x_2) = x_1 x_2, \quad \frac{\partial f}{\partial x_1} = x_2, \quad \frac{\partial f}{\partial x_2} = x_1. \tag{2.23}$$

Damit folgt

$$\kappa_{11} = \kappa_{12} = 1, \tag{2.24}$$

die Multiplikation ist also gut konditioniert, da die relativen Fehler der Ausgabedaten gleich denen der Eingabedaten sind.

2.2.13 Beispiel (Konditionierung der Addition): Es gilt

$$y_1 = f(x_1, x_2) = x_1 + x_2, \quad \frac{\partial f}{\partial x_1} = 1, \quad \frac{\partial f}{\partial x_2} = 1. \tag{2.25}$$

Damit folgt

$$\kappa_{11} = \frac{1}{1 + \frac{x_2}{x_1}}, \qquad \kappa_{12} = \frac{1}{1 + \frac{x_1}{x_2}}. \tag{2.26}$$

Für den Fall $x_1 \approx -x_2$ ist die Addition also schlecht konditioniert.

2.2.14 Bemerkung: Man nennt die schlechte Konditionierung der Subtraktion fast gleicher Zahlen auch anschaulich **Auslöschung**, was wir an folgendem Beispiel erklären:

$$\begin{array}{r} 0.1234569 \\ \text{-}0.1234567 \\ \hline 0.0000002 = 0.2 \cdot 10^{-6}. \end{array}$$

Bei der Subtraktion zweier Zahlen mit 7-stelliger Mantisse haben sich 6 Stellen ausgelöscht und es bleibt nur eine einzige signifikante Stelle.

2.3 Stabilität eines Algorithmus

2.3.1. Im letzten Abschnitt haben wir untersucht, wie sich die Fehler von Eingabedaten bei der Anwendung eines mathematisch exakten Algorithmus auf die Augabedaten auswirken. Hier nun beschäftigen wir uns mit den Auswirkungen inexakter Rechnungen auf das Ergebnis.

2.3.2 Definition: Ein **Algorithmus** ist eine eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen. Algorithmen bestehen aus endlich vielen, wohldefinierten Einzelschritten. Damit können sie zur Ausführung in ein Computerprogramm implementiert, aber auch in menschlicher Sprache formuliert werden. Bei der Problemlösung wird eine bestimmte Eingabe in eine bestimmte Ausgabe überführt.

nach Wikipedia (22.4.2019)

2.3.3 Bemerkung (Eigenschaften von Algorithmen):

Determiniertheit: Gleiche Eingabe, gleiche Ausgabe

Statische Finitheit: Beschreibung endlicher Länge

Dynamische Finitheit: Endlicher Speicherplatz

Terminiertheit: Endet nach endlich vielen Schritten

Effektivität: Der Effekt jedes Schrittes ist eindeutig festgelegt

Bemerkung 2.3.4. Wir betrachten Algorithmen auf verschiedenen Abstraktionsebenen. So ist sowohl das Gram-Schmidt-Verfahren in Theorem 1.4.8 ein Algorithmus, wenn die Gültigkeit des Assoziativgesetzes vorausgesetzt ist. Auch die beiden Beschreibungen in Python in Algorithmus 1.4.9 und Algorithmus 1.4.12 sind Algorithmen. Zur genaueren Spezifikation bezeichnen wir ersteren auch als **mathematisches Verfahren**, letztere auch als **Implementation**.

Bemerkung 2.3.5. Mathematische Verfahren benutzen eine Formelsprache, in die die Assoziativität der Grundoperationen bereits eingebaut ist. Während wir eine Summe sequenziell denken mögen, so gibt die Formel keine Reihenfolge strikt vor.

2.3.6 Definition: Wir nennen einen Algorithmus, bzw. seine Implementation auf einem Computer **stabil**, wenn die Akkumulation von Rundungsfehlern bei der Durchführung den Fehler durch die Konditionierung nicht wesentlich verschlechtert.

Bemerkung 2.3.7. Früher, als Fließkommazahlen mit 32 Bit der Standard waren, war die Analyse der Fehler von Rechenverfahren ein zentrales Thema der Numerik. Heute, im Zeitalter von 64 Bit hat sich das relativiert und wir legen mehr Gewicht auf mathematische Eigenschaften der Algorithmen.

Andererseits bieten die modernsten Graphikkarten (GPU) sehr schnelle Berechnung mit 16 Bit an, was sich in einer Implementation mit verschiedenen Genauigkeiten nutzen lässt.

Der Sinn dieses Abschnitts liegt damit statt des rigorosen Studiums der Fehleranalyse mehr darin, Bewusstsein für das Konzept der Stabilität zu wecken. Bereits beim Gram-Schmidt-Verfahren haben wir gesehen, dass es in der Tat Algorithmen gibt, bei denen Stabilität ein Problem ist. Glücklicherweise sind das wenige und es genügt zumeist, aus der Literatur die stabile Variante zu wählen.

Die Rundungsfehleranalyse ist aufwändig und nach Ansicht des Autors nur dann anzuraten, wenn der Verdacht auf Instabilität besteht. Daher werden wir hier nur exemplarisch vorgehen und sie an einigen Beispielen erläutern.

2.3.8 Definition: Bei der **Vorwärtsanalyse** von Rundungsfehlern folgt man den Elementaroperationen des Algorithmus und berücksichtigt in jedem Schritt den Rundungsfehler.
Ein Algorithmus ist stabil im Sinne der Vorwärtsanalyse, wenn die akkumulierten Rundungsfehler den Fehler durch die Konditionierung nicht wesentlich überschreiten.

Bemerkung 2.3.9. Graphisch ist die Vorwärtsanalyse in [Deuflhard and Hohmann, 2008, Abschnitt 2.3.1] veranschaulicht

Beispiel 2.3.10. Exemplarisch führen wir die Rundungsfehleranalyse am Beispiel in [Rannacher, 2017, Abschnitt 1.3.2] durch.

2.3.11 Definition: Bei der **Rückwartsanalyse** des Rundungsfehlers bestimmt man zur genäherten Lösung $\tilde{y} = \tilde{f}(x)$ einen Eingabewert $\tilde{x}$, so dass $\tilde{y} = f(\tilde{x})$ das Ergebnis des exakten Algorithmus angewandt auf die fehlerhaften Daten $\tilde{x}$ ist.
Ein Algorithmus ist stabil im Sinne der Rückwärtsanalyse, wenn $\|\tilde{x} - x\|$ in derselben Größenordnung wie der erwartete Eingabefehler ist.

Beispiel 2.3.12. Exemplarisch führen wir die Rundungsfehleranalyse am Beispiel in [Deuflhard and Hohmann, 2008, Lemma 2.30] durch.

2.4 Effizienz eines Algorithmus

Bemerkung 2.4.1. Der aktuelle Entwicklungshorizont ist **Exascale computing**, das heißt, berechnungen mit 10^{18} Fließkommaoperationen (**FLOP**) pro Sekunde. Das führt zu etwa 10^9 gleichzeitigen Operationen, deren zeitliche Abfolge nicht mehr festgelegt ist. Dies steht wegen der fehlenden Assoziativität im Widerspruch zu Determiniertheit und Effektivität.

Die Idee des Algorithmus als eine „Abfolge von Elementaroperationen“ stößt hier an ihre Grenzen, wie auch die Idee der Turing-Maschine als Muster aller Computer.

Bemerkung 2.4.2. Eine wichtige Eigenschaft von Algorithmen fehlte in der Auflistung Bemerkung 2.3.3, die durch die theoretische Informatik geprägt ist, nämlich die **Effizienz**, also die möglichst schnelle Abarbeitung auf einer gegebenen Rechenmaschine.

Maße für Effizienz sind vielfältig und reichen von mathematischen Eigenschaften zum Zusammenspiel von Mathematik und Maschine. Wir listen hier die am häufigsten benutzen auf.

Aufwand: Die Anzahl an Fließkommaoperationen, die insgesamt zur Berechnung des Ergebnisses nötig sind

Auslastung: Anzahl der (sinnvollen) Fließkommaoperationen pro Zeiteinheit verglichen mit dem maximal Möglichen des Computers in „%-peak performance“.

Starke Skalierbarkeit: Wie verringert sich die Bearbeitungszeit, wenn mehr parallele Prozesse für dasselbe Problem eingesetzt werden (strong scaling)?

Schwache Skalierbarkeit: Wie verhält sich die Bearbeitungszeit beim Einsatz von mehr parallelen Prozessen, wenn die Problemgröße proportional zur Zahl der Prozesse wächst (weak scaling)?

Numerische Intensität: Die Anzahl an Rechenoperationen pro Speichertransfer (numerical intensity)

Bemerkung 2.4.3. Die Quantifizierung der Effizienz eines Algorithmus is deswegen so schwierig, weil es heute nicht mehr möglich ist, einfach die Instruktionen, die der Rechner ausführt, nacheinander zu betrachten und die benötigten Zeiten aufzuaddieren.

Superskalare Prozessorarchitekturen, und das sind alle modernen CPUs, verfügen über eine sogenannte **instruction pipeline** die mehrere Anweisungen gleichzeitig laden und möglichst parallel ausführen soll, ohne dass dies der Benutzer merkt. Dabei eliminiert die Logik in der CPU Seiteneffekte, die dadurch entstehen können, wenn mehrfach dasselbe Datum verändert wird. Auf einem solchen Rechner werden bei kurzen Verzweigungen mit **if** beide Zweige ausgeführt und erst anschließend ein Ergebnis verworfen. Es lohnt also in diesem Fall nicht, einen Algorithmus für spezielle Werte zu optimieren, da alle Programmzweige ausgeführt werden.

Bei der **Vektorisierung** von Algorithmen nutzt man aus, dass moderne Rechenwerke sehr effizient dieselbe Operation auf mehreren Daten ausführen können.

Das funktioniert auch nur, wenn keine Verzweigungen auftreten, bzw. wenn logische Verzweigungen im Algorithmus auf möglichst hoher Ebene angesiedelt sind.

Das Ausführen arithmentischer Operationen ist heutzutage wesentlich schneller und auch billiger als das Laden der Daten aus dem Speicher. Deshalb ist die Grundregel aus den ersten Jahrzehnten des numerischen Rechnens „speichern statt zweimal zu rechnen“ heute in der Regel falsch und könnte eher heißen “lieber zehnmal rechnen als einmal speichern.“ Als folge sind einzelne Rechenoperationen oft in der Ausführungszeit nicht sichtbar.

Komplizierter wird dies alles durch eine Hierarchie von Cache-Speichern, bei denen das Laden und Speichern schneller geht, die aber beschränkte Größe haben.

Kapitel 3

Interpolation und Quadratur

3.0.1. Ziel dieses Kapitels ist die Herleitung von Methoden zur Approximation des Integrals einer Funktion über ein Intervall $[a, b]$. Diese Aufgabe wird in zwei Teile geteilt:

1. Wir unterteilen das Intervall in Subintervalle und summieren die Teilintegrale

$$\int_a^b f \,\mathrm{d}x = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f \,\mathrm{d}x, \qquad a = x_0 < x_1 < \cdots < x_n = b. \tag{3.1}$$

2. Auf jedem Teilintervall finden wir Approximationen für das lokale Integral.

Da wir Polynome exakt integrieren können, nutzen wir wieder die Approximation von Funktionen durch Polynome, um uns diesem Problem zu nähern.

3.1 Polynominterpolation

3.1.1 Definition und Konditionsabschätzung

3.1.1 Definition: Die **Interpolation**saufgabe nach Lagrange lautet: seien $n+1$ paarweise verschiedene **Stützstellen** $x_0, \ldots, x_n$ mit zugehörigen Funktionswerten f_i gegeben. Finde ein Polynom $p \in \mathbb{P}_n$ mit der Eigenschaft

$$p(x_i) = f_i. \tag{3.2}$$

Alternativ ist die Interpolationsaufgabe aufzufassen als eine Abbildung

$$\begin{aligned} I_n \colon C[a,b] &\to \mathbb{P}_n \\ p(x_i) &= f(x_i), \end{aligned} \tag{3.3}$$

wobei das Interval $[a,b]$ alle Stützpunkte enthält. Wir nennen diese Abbildung den **Lagrange-Interpolationsoperator** oder kurz **Lagrange-Interpolation**.

3.1.2 Satz: Die Interpolationsaufgabe nach Lagrange hat eine eindeutige Lösung, bezeichnet als (Lagrange-)**Interpolierende** der Funktion f

$$p(x; f; x_0, \ldots, x_n) \tag{3.4}$$

Beweis. Der Beweis ist eine direkte Konsequenz des folgenden Lemmas. □

3.1.3 Lemma: Seien die Punkte $x_0, \ldots, x_n$ paarweise verschieden. Dann gilt für die **Lagrange-Polynome**

$$\ell_i(x) = \ell_{i;n}(x) = \ell_{i;x_0,\ldots,x_n}(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{3.5}$$

die **Interpolationseigenschaft**

$$\ell_i(x_j) = \delta_{ij}, \qquad 0 \leq i, j \leq n. \tag{3.6}$$

Sie sind linear unabhängig und formen eine Basis von $\mathbb{P}_n$.
Die Lagrange-Polynome sind orthonormal bezüglich des Skalarprodukts

$$\langle p, q \rangle = \sum_{i=0}^{n} p(x_i) q(x_i). \tag{3.7}$$

Beweis. Der Beweis der Basiseigenschaft ist Übungsaufgabe. Es bleibt zu zeigen, dass $\langle \cdot, \cdot \rangle$ ein Skalarprodukt ist und die Polynome orthogonal. Dabei sind die Symmetrie und Homogenität und Orthogonalität elementar nachzurechnen. Was zu zeigen bleibt, ist die Definitheit, also für $p \in \mathbb{P}_n$:

$$\langle p, p \rangle = \sum_{i=0}^{n} p(x_i)^2 = 0 \qquad \Rightarrow \qquad p = 0. \tag{3.8}$$

Dazu nutzen wir die Basiseigenschaft der Polynome ℓ_j und schreiben mit geeigneten Koeffizienten α_j:

$$p(x) = \sum_{j=0}^{n} \alpha_j \ell_j(x). \tag{3.9}$$

Es gilt dann wegen der Interpolationseigenschaft

$$\langle p, p \rangle = \sum_{i=0}^{n} \left(\sum_{j=0}^{n} \ell_j(x_i) \right)^2 = \sum_{i=0}^{n} \alpha_j^2. \tag{3.10}$$

Offensichtlich ist diese Summe genau dann null, wenn dies für jedes α_j gilt, also $p = 0$. □

3.1.4 Korollar: Die Lösung der Interpolationsaufgabe nach Lagrange erlauben die Darstellung

$$p(x; f; x_0, \dots, x_n) = \sum_{i=0}^{n} f_i \ell_{i;x_0,\dots,x_n}(x). \tag{3.11}$$

Die Lagrange-Interpolation eingeschränkt auf den Raum $\mathbb{P}_n$ ist die Identität

3.1.5. Die Lagrangesche Interpolationsaufgabe kann auch als Gaußsche Ausgleichsrechnung mit dem obigen Skalarprodukt aufgefasst werden. Allerdings erreichen wir es hier, dass der Abstand in der zugehörigen Norm zu null wird.

Ein alternativer Beweis zur Eindeutigkeit der Lösung der Interpolationsaufgsbe benutzt nicht die explizite Darstellung in der Lagrange-Basis, sondern folgendes Resultat über die Nullstellen eines Polynoms.

3.1.6 Satz: Hat ein Polynom $p \in \mathbb{P}_n$ auf der reellen Achse $n+1$ verschiedene Nullstellen, so ist es notwendig das Nullpolynom.

Beweis. Seien die Nullstellen $x_0 < x_1 < \cdots < x_n$ sortiert nach Größe. Dann gilt nach dem Satz von Rolle, dass in jedem Intervall (x_i, x_{i+1}) mit $i = 0, \dots, n-1$ eine Nullstelle der Ableitung $p' \in \mathbb{P}_{n-1}$ liegt, insgesamt n Stück.

Diesen Prozess setzen wir rekursiv fort und schließen, dass die n-te Ableitung $p^{(n)} \in \mathbb{P}_0$ eine Nullstelle hat. Damit ist muss $p^{(n)} \equiv 0$ gelten und daher ist $p^{(n-1)} \in \mathbb{P}_0$. Somit schließen wir dass $p^{(n-2)} = p^{(n-3)} = \cdots = p \equiv 0$ □

3.1.7 Lemma: Sei $f\colon X \to Y$ eine lineare Abbildung zwischen Vektorräumen X und Y. Dann sind folgende Aussagen äquivalent:

1. In einem beliebigen Punkt $x \in X$ gilt für das gestörte Problem $y + \delta y = f(x + \delta x)$ die Abschätzung

$$\|\delta y\| \leq \kappa^{\text{abs}} \|\delta x\| \qquad \forall \delta x \in X. \tag{3.12}$$

2. Für $y = f(x)$ gilt die Abschätzung

$$\|y\| \leq \kappa^{\text{abs}} \|x\| \qquad \forall x \in X. \tag{3.13}$$

Insbesondere ist eine lineare Abbildung genau dann stetig, wenn es eine solche Schranke gibt.

Beweis. Der Linearität und $y = f(x)$ wegen gilt

$$y + \delta y = f(x + \delta x) = f(x) + f(\delta x), \tag{3.14}$$

beziehungsweise

$$\delta y = f(x) - y + f(\delta x) = f(\delta x). \tag{3.15}$$

Gilt nun die zweite Aussage, so folgt sofort die erste. Betrachte ich die erste im Punkt $x = 0$, so dass wegen der Linearität $y = f(x) = 0$ gilt, so ergibt sich aus der Bedingung „für alle δx“ die zweite. □

Bemerkung 3.1.8. Es genügt also, die Konditionierung um die null zu untersuchen, was die Analyse vereinfacht.

Nun gilt für eine lineare Abbildung $f(0) = 0$. In diesem Falle ist also die Konditionszahl für den relativen Fehler aus Definition 2.2.3 bzw. Lemma 2.2.11 nicht sinnvoll definiert. Wir benutzen daher die Konditionszahlen für den absoluten Fehler.

3.1.9 Satz (Konditionszahl der Lagrange-Interpolation): Die Konditionszahl des absoluten Fehlers in der Supremumsorm der Lagrange-Interpolation zu den Punkten $a = x_0 < \cdots < x_n = b$ ist die **Lebesgue-Konstante**

$$\Lambda_{x_0,\ldots,x_n} = \max_{x\in[a,b]} \sum_{i=0}^{n} |\ell_{i;x_0,\ldots,x_n}(x)|. \tag{3.16}$$

Es gilt also

$$\max_{x\in[a,b]} |I_n f(x)| \leq \Lambda_{x_0,\ldots,x_n} \max_{x\in[a,b]} |f(x)|. \tag{3.17}$$

Diese Abschätzung ist scharf.

Beweis. Siehe auch [Deuflhard and Hohmann, 2008, Satz 7.3].

Zunächst zeigen wir die Beschränktheit der Interpolation. Es gilt:

$$|I_n(f)(x)| = \left| \sum_{i=0}^{n} f(x_i)\ell_{i;n}(x) \right| \tag{3.18}$$

$$\leq \sum_{i=0}^{n} |f(x_i)\ell_{i;n}(x)| \tag{3.19}$$

$$\leq \max_{i=0,\ldots,n} |f_i| \sum_{i=0}^{n} |\ell_{i;n}(x)| \tag{3.20}$$

$$\leq \max_{x\in[a,b]} |f(x)| \max_{x\in[a,b]} \sum_{i=0}^{n} |\ell_{i;n}(x)| \,. \tag{3.21}$$

Damit gilt also die behauptete Abschätzung

$$\|I_n(f)\|_\infty \leq \Lambda_{x_0,\ldots,x_n} \|f\|_\infty, \tag{3.22}$$

wobei die Supremumsnorm auf dem Intervall $[a, b]$ genommen wurde.

Wir zeigen, dass die Abschätzung scharf ist, indem wir für eine Funktion die Gleichheit zeigen, also: es existiert eine auf $[a, b]$ stetige Funktion g, so dass es einen Wert $x \in [a, b]$ gibt an dem gilt:

$$|I(g)(x)| = \|g\|_\infty \max_{x\in[a,b]} \sum_{i=0}^{n} |\ell_{i;n}(x)| \,. \tag{3.23}$$

Sei zunächst ξ der Wert, an dem die Summe ihr Maximum annimt, also

$$\sum_{i=0}^{n} |\ell_{i;n}(\xi)| = \max_{x\in[a,b]} \sum_{i=0}^{n} |\ell_{i;n}(x)| \,. \tag{3.24}$$

Nun wählen wir g als die stückweise lineare Funktion, die in den Werten x_i die Werte $g(x_i) = \operatorname{sgn} \ell_i(\xi)$ annimmt und links und rechts des Intervalls, das von den Punkten $\{x_i\}$ aufgespannt wird, konstant ist. Es gilt damit $\|g\|_\infty = 1$ und

$$I_n(g)(\xi) = \sum_{i=0}^{n} \operatorname{sgn} \ell_i(\xi)\ell_i(\xi) \tag{3.25}$$

$$= \sum_{i=0}^{n} |\ell_i(\xi)| \tag{3.26}$$

$$= \Lambda_{x_0,\ldots,x_n} \|g\|_\infty \tag{3.27}$$

□

3.1.10 Beispiel: Für äquidistante Stützstellen erhält man exemplarisch die Konditionszahlen in der zweiten Spalte. Später entwickeln wir einen optimalen Satz von Stützstellen. Die Konditionszahlen dazu sind in der rechten Spalte.

	$\Lambda_{0,\ldots,n}$	
n	äquidistant	optimal
5	3.1	2.1
10	30	2.5
15	512	2.7
20	10986	2.9

Quelle: [Deuflhard and Hohmann, 2008]

3.1.2 Rekursive Interpolation

3.1.11 Lemma (Aitken): Für das Interpolationspolynom

$$p_{0,\dots,n}(x) = p(x; f; x_0, \dots, x_n) \tag{3.28}$$

zu paarweise verschiedenen Stützstellen $x_0, \dots, x_n$ gilt die Rekursionsformel

$$p_{0,\dots,n}(x) = \frac{(x - x_0)p_{1,\dots,n}(x) - (x - x_n)p_{0,\dots,n-1}(x)}{x_n - x_0}. \tag{3.29}$$

Beweis. Der Beweis benutzt wieder Induktion. Für eine einzige Stützstelle ist das Interpolationspolynom konstant, $p_i(x) = f_i$ und daher $p_i \in P_0$. Sei nun $\varphi(x)$ der Bruch auf der rechten Seite. Durch Induktion sehen wir sofort, dass $\varphi \in \mathbb{P}_n$. Ferner gilt für $i = 1, \dots, n-1$

$$\begin{aligned} \varphi(x_i) &= \frac{(x_i - x_0)p_{1,\dots,n}(x_i) - (x_i - x_n)p_{0,\dots,n-1}(x_i)}{x_n - x_0} \\ &= \frac{(x_i - x_0)f_i - (x_i - x_n)f_i}{x_n - x_0} \\ &= f_i. \end{aligned} \tag{3.30}$$

Ebenso verschwindet für x_0 und x_n je ein Term und es gilt dieselbe Aussage. □

3.1.12 Algorithmus (Neville): Sei für eine Stelle x an der das Interpolationspolynom berechnet werden soll $p_{ik} = p_{i-k,\dots,i}(x)$ für $i \geq k$. Dann lässt sich $p_{0,\dots,n}(x) = p_{nn}$ rekursiv berechnen durch

1. Für $k = 0$ setze

$$p_{i0} = f_i \qquad i = 0, \dots, n. \tag{3.31}$$

2. Für $k = 1, \dots, n$ berechne

$$p_{ik} = p_{i,k-1} + \frac{x - x_i}{x_i - x_{i-k}}\big(p_{i,k-1} - p_{i-1,k-1}\big) \qquad i = k, \dots, n. \tag{3.32}$$

3.1.13 Definition: Als **Newton-Basis** der Lagrange-Interpolation bezeichnen wir die Polynome

$$\omega_k(x) = \omega_{0,\dots,k-1}(x) = \prod_{j=0}^{k-1}(x - x_j), \qquad k = 0, \dots, n \tag{3.33}$$

wobei das leere Produkt für $k = 0$ den Wert 1 annehme. Ebenso können wir Newton-Polynome mit Startindex verschieden von null definieren und erhalten für $k \geq 0$

$$\omega_{i,\dots,i+k} = \prod_{j=i}^{i+k}(x - x_j) = \frac{\omega_{i+k+1}(x)}{\omega_i(x)}. \tag{3.34}$$

3.1.14 Lemma: Sei $p \in \mathbb{P}_n$ ein Polynom dargestellt bezüglich der Monombasis und der Newton-Basis durch

$$p(x) = \sum_{i=0}^{n} a_i x^i = \sum_{i=0}^{n} b_i \omega_i(x), \tag{3.35}$$

Dann gilt $b_n = a_n$.

Beweis. Es gilt

$$p(x) = q_n(x) = q_{n-1}(x) + b_n \omega_n(x). \tag{3.36}$$

Da $q_{n-1} \in \mathbb{P}_{n-1}$ und der führende Koeffizient von ω_n eins ist, folgt sofort $b_n = a_n$. □

3.1.15 Definition: Als **dividierte Differenzen** zur Lagrange-Interpolationsaufgabe bezeichnen wir die rekursiv definierten Werte

$$[x_i]f = f_i \tag{3.37}$$

$$[x_i, \dots, x_{i+k}]f = \frac{[x_{i+1}, \dots, x_{i+k}]f - [x_i, \dots, x_{i+k-1}]f}{x_{i+k} - x_i} \tag{3.38}$$

3.1.16 Satz: Für das Lagrange-Interpolationspolynom $p_{i,\dots,i+k}(x)$ zu den paarweise verschiedenen Stützpunkten $x_i, \dots, x_{i+k}$ gilt

$$p_{i,\dots,i+k}(x) = \sum_{j=i}^{i+k}[x_i, \dots, x_j]f \; \frac{\omega_j(x)}{\omega_i(x)}. \tag{3.39}$$

Beweis. In der Newton-Darstellung gilt

$$p_{i,\dots,i+k}(x) = p_{i,\dots,i+k-1}(x) + \alpha \frac{\omega_{i+k}(x)}{\omega_i(x)}. \tag{3.40}$$

Zu zeigen ist also $\alpha = [x_i, \dots, x_{i+k}]f$, was nach Lemma 3.1.14 der Koeffizient vor x^k ist. Nach Induktionsannahme ist

$$\begin{aligned} p_{i,\dots,i+k-1}(x) &= [x_i, \dots, x_{i+k-1}]f\, x^{k-1} + \mathcal{O}(x^{k-2}) \\ p_{i+1,\dots,i+k}(x) &= [x_{i+1}, \dots, x_{i+k}]f\, x^{k-1} + \mathcal{O}(x^{k-2}) \end{aligned} \tag{3.41}$$

Nach dem Lemma von Aitken gilt

$$p_{i,\dots,i+k} = \frac{(x - x_i)p_{i+1,\dots,i+k} - (x - x_{i+k})p_{i,\dots,i+k-1}}{x_{i+k} - x_i}. \tag{3.42}$$

Dessen höchster Koeffizient ist aber gerade die dividierte Differenz. □

Bemerkung 3.1.17. Der Bruch im vorherigen Satz ist nicht problematisch, da

$$\frac{\omega_j(x)}{\omega_i(x)} = \prod_{\ell=i}^{j-1}(x - x_\ell). \tag{3.43}$$

3.1.18 Satz: Sei $f \in C^{n+1}[a, b]$ und $p \in \mathbb{P}_n$ die Lagrange-Interpolierende zu den Stützstellen $x_0, \dots, x_n \in [a, b]$. Dann gibt es zu jedem $x \in [a, b]$ einen Punkt $\xi \in (a, b)$, so dass

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{0,\dots,n}(x). \tag{3.44}$$

Wir bezeichnen diese Aussage als **Fehlerdarstellung**, die rechte Seite auch als **Restglied**.

Beweis. Der Beweis folgt [Stoer, 1983, Satz 2.1.4.1]. Zunächst bemerken wir, dass für alle Stützstellen x_i gilt, dass $f(x_i) - p(x_i) = 0$. Dort ist also nichts zu beweisen. Für die Fehlerfunktion $F(y)$ machen wir nun den Ansatz

$$F(y) = f(y) - p(y) - \alpha\omega_{0,\dots,n}(y) \tag{3.45}$$

und α soll so gewählt werden, dass $F(x) = 0$. Damit hat $F(y)$ im Intervall I insgesamt die $n+2$ Nullstellen $x, x_0, \dots, x_n$. Wiederholte Anwendung des Satzes von Rolle ergibt, dass $F'(y)$ insgesamt $n + 1$ Nullstellen hat und das $F^{(n+1)}(y)$ eine Nullstelle ξ besitzt. Da $p \in \mathbb{P}_n$ gilt

$$0 = F^{(n+1)}(\xi) = f^{(n+1)}(\xi) - \alpha(n + 1)! \tag{3.46}$$

und damit

$$\alpha = \frac{f^{(n+1)}(\xi)}{(n+1)!}. \tag{3.47}$$

□

3.1.19 Korollar: Sei $f \in C^{n+1}[a,b]$ und alle Stützstellen x_i im Intervall $[a,b]$. Dann gibt es ξ aus dem kleinsten Intervall, das die Punkte a, b und x enthält, so dass

$$[x_0, \ldots, x_n]f = \frac{f^{(n)}}{n!}(\xi). \tag{3.48}$$

Beweis. Wenden wir Formel (3.45) auf $p_{0,\ldots,n-1}$ und $x = x_n$ an, so sehen wir, dass α der Koeffizient vor dem nächsten Newton-Basispolynom $\omega_{0,\ldots,n-1}$ ist. Dieser ist aber gerade die angegebene dividierte Differenz. □

3.1.20 Korollar: Es gelten die Voraussetzungen von Satz 3.1.18. Dann gibt es eine Konstante C, die nur von der Wahl der Stützstellen abhängt, so dass

$$\max_{x\in[a,b]} |f(x) - p_{0,\ldots,n}(x)| \leq \frac{C|b-a|^{n+1}}{(n+1)!} \max_{x\in[a,b]} |f^{(n+1)}(x)| \tag{3.49}$$

Bemerkung 3.1.21. Die Fehlerabschätzung in Korollar 3.1.20 können wir auch kürzer schreiben als

$$\|f - p_{0,\ldots,n}\|_\infty \leq \frac{C|b-a|^{n+1}}{(n+1)!} \|f^{(n+1)}\|_\infty. \tag{3.50}$$

Die rechte Seite besteht dabei aus dem Produkt aus einem Teil, der nur von den Daten abhängt, $\|f^{(n+1)}\|_\infty$ und einem Anteil, der durch das Verfahren bestimmt ist.

Wir sehen, dass Interpolation auf einem Intervall um so genauer ist, je kürzer das Intervall ist.

3.1.22. Der Rest dieses Abschnitts befasst sich mit der Frage, wie die Stützstellen $x_0, \ldots, x_n$ gewählt werden können, damit die Konstante C in der Fehlerabschätzung optimal ist. Aus der Fehlerdarstellung in Satz 3.1.18 folgt, dass wir dazu ein Polynom finden müssen, dessen führender Koeffizient 1 ist, und das minimalen Betrag auf dem Intervall $[a,b]$ hat. Tatsächlich erlauben uns die Tschebyscheff-Polynome, diese Optimalität zu erreichen.

Insbesondere erlaubt uns die Fehlerdarstellung, die Minimierung über die Stützpunkte durch eine Minimierung im Polynomraum zu ersetzen. Letztere hat Vektorraumstruktur und ist damit deutlich strukturierter.

3.1.23 Lemma: Die Tschebyscheff-Polynome, die der Rekursionsformel in Definition 1.5.6 genügen, haben die Darstellung

$$T_k = \cos(k \arccos x) \tag{3.51}$$

Insbesondere gilt

$$T_k(1) = 1 \tag{3.52}$$
$$T_k(-1) = (-1)^k \tag{3.53}$$
$$|T_k(x)| \leq 1, \qquad x \in [-1, 1] \tag{3.54}$$
$$T_k(x) = (-1)^j, \quad x = \cos\left(\frac{j}{k}\pi\right), \qquad j = 0, \ldots, k \tag{3.55}$$
$$T_k(x) = 0, \qquad x = \cos\left(\frac{2j-1}{2k}\pi\right), \quad j = 1, \ldots, k \tag{3.56}$$

Beweis. Hausaufgabe □

3.1.24 Satz: Jedes Polynom $p \in \mathbb{P}_n$ mit führendem Koeffizienten 1 nimmt im Intervall $[-1, 1]$ einen Wert $|p(x)| \geq 1/2^{n-1}$ an und es gilt

$$\frac{1}{2^{n-1}} T_n(x) = \operatorname*{argmin}_{\substack{p \in \mathbb{P}_n \\ p = x^n + \cdots}} \max_{x \in [-1,1]} |p(x)|. \tag{3.57}$$

Beweis. Siehe auch [Deuflhard and Hohmann, 2008, Satz 7.19]. Aus der Rekursionsformel für Tschebyscheff-Polynome folgt sofort, dass der höchste Koeffizient von T_n den Wert 2^{n-1} annimmt. Sei nun als Widerspruchsannahme $p \in \mathbb{P}_n$ ein weiteres Polynom mit höchstem Koeffizienten 2^{n-1}, so dass

$$\max_{x \in [-1,1]} |p(x)| < 1. \tag{3.58}$$

Dann ist $q_n = T_n - p \in \mathbb{P}_{n-1}$ und für die **Tschebyscheff-Abszissen** $\tilde{x}_j = \cos(j\pi/n)$ mit $j = 0, \ldots, n$ gilt

$$T_n(\tilde{x}_j) = 1, \qquad p(\tilde{x}_j) < 1 \qquad q_n(\tilde{x}_j) > 0, \qquad j \text{ gerade} \tag{3.59}$$
$$T_n(\tilde{x}_j) = -1, \qquad p(\tilde{x}_j) > -1 \qquad q_n(\tilde{x}_j) < 0, \qquad j \text{ ungerade.} \tag{3.60}$$

q_n wechselt also an mindestens n Stellen das Vorzeichen und hat damit als stetige Funktion mindestens ebensoviele Nullstellen. Aus $q_n \in \mathbb{P}_{n-1}$ folgt damit im Widerspruch $q_n = 0$ und $p = T_n$. Damit gilt nach Skalierung um den Faktor 2^{n-1}

$$\min_{\substack{p\in\mathbb{P}_n \\ p=x^n+\cdots}} \max_{x\in[-1,1]} |p(x)| \geq 1, \tag{3.61}$$

und Gleichheit für das skalierte Tschebyscheff-Polynom. □

3.1.25 Korollar: Wählt man als Stützstellen die Werte

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{2i+1}{2n+2}\pi\right), \qquad i = 0, \ldots, n, \tag{3.62}$$

So gilt für den Fehler der Lagrange-Interpolation

$$\|f - p_{0,\ldots,n}\|_\infty \leq \frac{|b-a|^{n+1}}{2^{2n+1}(n+1)!} \|f^{(n+1)}\|_\infty. \tag{3.63}$$

Beweis. Zunächst transformieren wir die Aufgabe vom Intervall $[a, b]$ auf das Intervall $[-1, 1]$ durch die Abbildung

$$x = \Phi(\xi) = \frac{a+b}{2} + \frac{b-a}{2}\xi. \tag{3.64}$$

Es gilt $\Phi(-1) = a$, $\Phi(1) = b$ und die Punkte x_i sind die Bilder der Tschebyscheff-Knoten ξ_i zu T_{n+1}. Insbesondere ist nun für diese Knoten

$$\omega_{0,\ldots,n}(\xi) = \frac{1}{2^n} T_{n+1}(\xi). \tag{3.65}$$

Es gilt $\Phi'(\xi) = (b-a)/2$ und für die Funktion $F(\xi) = f(\Phi(\xi))$ gilt

$$\frac{d^k}{d\xi^k} F(\xi) = \left(\frac{b-a}{2}\right)^k \frac{d^k}{dx^k} f(x). \tag{3.66}$$

Auf $[-1, 1]$ folgern wir aus Satz 3.1.18 und Satz 3.1.24, dass für die Interpolation gilt

$$\max_{\xi\in[-1,1]} |F(\xi) - P(\xi)| \leq \frac{2^{-n}}{(n+1)!} \max_{\xi\in[-1,1]} |F^{(n+1)}(\xi)|, \tag{3.67}$$

woraus folgt

$$\max_{x\in[a,b]} |f(x) - p(x)| \leq \frac{2^{-n}}{(n+1)!} \left(\frac{b-a}{2}\right)^{n+1} \max_{x\in[a,b]} |f^{(n+1)}(x)|. \tag{3.68}$$

□

3.1.26 Bemerkung: Wird das Interpolationspolynom an einem Punkt x ausgewertet, der nicht zwischen den Interpolationspunkten liegt, so sprechen wir von Extrapolation. Aus der Fehlerdarstellung mit den Newton-Polynomen können wir ablesen, dass die Abschätzungen sehr schnell schlechter werden, wenn sich x von den Stützstellen entfernt.

3.1.3 Hermite-Interpolation

3.1.27 Definition: Die **Hermite-Interpolation** benutzt neben Funktionswerten auch Ableitungswerte zur Interpolation. Das Interpolationspolynom $p \in \mathbb{P}_n$ genügt in $m+1$ paarweise verschiedenen Punkten den Bedingungen

$$\frac{d^j p}{dx^j}(x_i) = f_i^j, \qquad i = 0, \ldots, m, \quad j = 0, \ldots, n_i - 1, \tag{3.69}$$

und es gilt

$$\sum_i n_i = n + 1. \tag{3.70}$$

Die definierenden Funktionale[a] der Gestalt ${}^{d^j}/_{dx^j} p(x_i)$ werden auch als **Knotenwerte** oder **Knotenfunktionale** bezeichnet.

[a] Als Funktional bezeichnet man eine Abbildung aus einem Vektorraum in den zugehörigen Körper

3.1.28 Satz: Definition 3.1.27 bestimmt das Interpolationspolynom eindeutig.

Beweis. Analog zur Lagrange-Interpolation identifizieren wir wieder eine Basis $\{H_{ij}(x)\}$, diesmal doppelt indiziert, die bezüglich der Interpolationsbedingungen orthogonal ist. Damit stellen wir das Interpolationspolynom dar als

$$p(x) = \sum_{i=0}^{m} \sum_{j=0}^{n_i - 1} f_i^j H_{ij}(x). \tag{3.71}$$

Zunächst führen wir die Hilfspolynome

$$q_{ij}(x) = \frac{(x - x_i)^j}{j!} \prod_{k \neq i} \left(\frac{x - x_k}{x_i - x_k} \right)^{n_k} \tag{3.72}$$

ein. Es gilt, dass q_{ij} in jedem Punkt $x_k \neq x_i$ eine n_k-fache Nullstelle hat, also

$$\frac{d^m q_{ij}}{dx^m}(x_k) = 0, \qquad k \neq i, \qquad m = 0, \dots, n_k - 1.$$

Ferner hat q_{ij} in x_i eine j-fache Nullstelle, also

$$\frac{d^m q_{ij}}{dx^m}(x_i) = 0, \qquad m = 0, \dots, j - 1.$$

Schließlich gilt

$$\frac{d^j q_{ij}}{dx^j}(x_i) = 1. \tag{3.73}$$

Damit können wir im Punkt x_i rekursiv definieren

$$\begin{aligned} H_{i,n_i-1}(x) &= q_{i,n_i-1}(x) \\ H_{ij}(x) &= q_{ij}(x) - \sum_{m=j+1}^{n_i-1} q_{ij}^{(m)}(x_i) H_{im}(x), \end{aligned} \tag{3.74}$$

wobei die letzte Zeile die Anwendung des Gram-Schmidt-Verfahrens ist. Per constructionem gilt für diese Polynome

$$\frac{d^\ell}{dx^\ell} H_{ij}(x_k) = \delta_{ik}\delta_{j\ell}, \tag{3.75}$$

was die lineare Unabhängigkeit impliziert. Da ihre Anzahl gleich der Raumdimension ist, bilden sie eine Basis. Die Aussage des Satzes ergibt sich nun aus der Basisdarstellung (3.71). Insbesondere impliziert $\equiv 0$ für alle Knotenwerte $f_i^j = 0$. □

3.1.29 Notation: Bei der Polynominterpolation ist die Anordnung der Interpolationspunkte beliebig. Das ist auch weiterhin der Fall. Für die Darstellung der Resultate und Beweise ist es aber oft hilfreich anzunehmen, dass sie in aufsteigender Folge angeordnet sind. Wir nehmen daher ab jetzt an, dass

$$a = x_0 \leq x_1 \leq \cdots \leq x_n = b. \tag{3.76}$$

Dabei sollen k-fach wiederholte Stützstellen bedeuten, dass dort nicht nur der der Funktionswert, sondern auch die ersten $k - 1$ Ableitungen interpoliert werden. Damit haben wir für die Interpolation in $\mathbb{P}_n$ immer eine Folge von $n + 1$ Stützstellen.

3.1.30 Beispiel: Sind alle Stützstellen $x_0 = \cdots = x_n$ identisch, so erhalten wir duch Interpolation einer Funktion $f \in C^n[a,b]$ das Taylor-Polynom vom Grad n

$$p(x; f; x_0, \ldots, x_n) = \sum_{k=0}^{n} \frac{(x - x_0)^k}{k!} f^{(k)}(x_0). \tag{3.77}$$

3.1.31 Beispiel: Die kubische Hermite-Interpolation auf dem Intervall $[a,b]$ ist definiert durch die Knotenwerte

$$p(a), p'(a), p(b), p'(b). \tag{3.78}$$

3.1.32 Satz: Das Hermite-Interpolationspolynom genügt der Darstellung

$$p_{0,\ldots,n}(x) = \sum_{j=0}^{n} [x_0, \ldots, x_j] f \; \omega_j(x) \tag{3.79}$$

mit den verallgemeinerten dividierten Differenzen definiert durch die Rekursion

$$[x_i, \ldots, x_{i+k}] f = \begin{cases} \frac{f^{(k)}(x_i)}{k!} & x_i = x_{i+k} \\ \frac{[x_{i+1}, \ldots, x_{i+k}] f - [x_i, \ldots, x_{i+k-1}] f}{x_{i+k} - x_i} & x_i \neq x_{i+k}. \end{cases} \tag{3.80}$$

Hier ist das Newton-Polynom mit wiederholten Stützstellen so definiert, dass die zugehörigen Linearfaktoren wiederholt werden.

Beweis. Der Beweis folgt im wesentlichen dem analogen Satz 3.1.16. Wir müssen dort nur die Argumente anpassen, die auf paarweise verschiedenen Stützstellen beruhen.

Dazu betrachten wir die Lagrange-Interpolation mit $x_i < x_{i+1} < \cdots < x_{i+k}$ und zugehörigem Interpolationspolynom $p_{i,\ldots,i+k} \in \mathbb{P}_k$. Wir benutzen Korollar 3.1.19, wonach gilt: es gibt ein $\xi \in [x_{i+1}, x_{i+k}]$ mit

$$[x_i, \ldots, x_{i+k}] f = \frac{f^{(k)}(\xi)}{k!}. \tag{3.81}$$

Da diese Eigenschaft unabhängig vom Abstand der Stützstellen gilt, können wir den Limes $x_j \to x_i$ für $j = 1, \ldots, k$ bilden, und es gilt im Limes $x_i = \cdots = x_{i+k}$

$$[x_i, \ldots, x_{i+k}] f \to \frac{f^{(k)}(x_i)}{k!}, \tag{3.82}$$

sowie

$$\omega_{i,\dots,i+k-1}(x) \to (x - x_i)^k. \tag{3.83}$$

Für das zugehörige Interpolationspolynom gilt dann

$$\frac{d^k}{dx^k} p_{i,\dots,i+k}(x_i) = f^k(x_i). \tag{3.84}$$

Ist x_i ein mehrfacher Interpolationspunkt, so gilt dies Argument für alle $x = 1, \dots, n_i - 1$ (Achtung, wir wechseln hier in der Diskussion die Numerierung zwischen Definition 3.1.27 und Notation 3.1.29).

Damit haben wir im Neville-Schema den Induktionsanfang geschafft. Es bleibt zu zeigen, dass die Rekursionsformel von Aitken auch weiterhin für $x_i \neq x_{i+k}$ gilt. Das ist unmittelbar einsichtig, wenn $x_i \neq x_{i+1}$ und $x_{i+k-1} \neq x_{i+k}$, da dann beide Polynome in der Rekursion alle Zwischenpunkte x_j interpolieren.

Sei nun zunächst $x_i = x_{i+1} = x_{i+r} < x_{r+1} \leq \dots < x_{i+k}$. Es ist zu zeigen, dass das Polynom

$$q(x) = \frac{(x - x_i)p_{i+1,\dots,i+k}(x) - (x - x_k)p_{i,\dots,i+k-1}(x)}{x_{i+k} - x_i} \tag{3.85}$$

alle Knotenfunktionale interpoliert. Für $x_j \neq x_i$ folgt dies wie bei der Lagrange-Interpolation aus der Induktionsannahme. Doch auch für x_i gilt dies, da der erste Term in der Summe verschwindet und $p_{i,\dots,i+k-1}$ bereits alle geforderten Ableitungen interpoliert. □

3.1.33 Satz: Sei $f \in C^{n+1}[a, b]$ und $p \in \mathbb{P}_n$ die Hermite-Interpolierende zu den Stützstellen $a = x_0 \leq \dots \leq x_n = b$. Dann gibt es zu jedem $x \in \mathbb{R}$ einen Punkt ξ im kleinsten Intervall I, das die Punkte x, a und b enthält, so dass

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{0,\dots,n}(x). \tag{3.86}$$

Beweis. Der Beweis folgt exakt denselben Argumenten wie der von Satz 3.1.18. □

3.1.34 Korollar: Für das **Taylor-Polynom** zu $f \in C^{n+1}(a,b)$ in einem Punkt $x_0 \in (a,b)$,

$$p(x) = \sum_{i=0}^{n} \frac{f^i(x_0)}{i!}(x - x_0)^i \tag{3.87}$$

gilt die folgende Fehlerdarstellung: es gibt ein $\xi \in [x_0, x]$ so dass gilt

$$f(x) - p(x) = \frac{f^{n+1}(\xi)}{(n+1)!}(x - x_0)^{n+1}. \tag{3.88}$$

3.2 Interpolation mit Splines

Dieser Abschnitt folgt recht eng der Darstellung in [Rannacher, 2017, Abschnitt 2.3].

3.2.1 Interpolation auf Teilintervallen

3.2.1 Notation: In diesem Abschnitt bezeichne für die monotone Folge

$$a = x_0 < x_1 \cdots < x_n = b \tag{3.89}$$

stets

$$\mathcal{I}_h = \big\{ I_i = [x_{i-1}, x_i] \big|\ i = 1, \ldots, n \big\} \tag{3.90}$$

eine **Zerlegung** des Intervalls $I = [a,b]$, also

$$[a,b] = \bigcup_{i=1}^{n} I_h. \tag{3.91}$$

Die Länge der Teilintervalle bezeichnen wir mit $h_i = |I_i| = x_i - x_{i-1}$, mit $h = \max h_i$ die **Feinheit** der Unterteilung.

3.2.2 Definition: Wir bezeichnen $\hat{I} = [-1, 1]$ als **Referenzintervall**. Jedes Intervall I_i einer Zerlegung $\mathcal{I}_h$ ergibt sich als Bild von $\hat{I}$ unter der affinen Abbildung (**Referenzabbildung**)

$$\begin{aligned} \Phi_i \colon \hat{I} &\to I_i \\ \hat{x} &\mapsto \tfrac{x_i + x_{i-1}}{2} + \tfrac{h_i}{2}\hat{x}. \end{aligned} \tag{3.92}$$

3.2.3 Definition (Stückweise Interpolation): Sei $\mathcal{I}_h$ eine Zerlegung von $[a, b]$. Auf dem Referenzintervall $\hat{I}$ sei eine Interpolationsaufgabe durch die Stützstellen $\hat{x}_0, \ldots, \hat{x}_k$ definiert. Dann lautet die Aufgabe der stückweisen Interpolation auf $\mathcal{I}_h$: finde eine Funktion s auf $[a, b]$, so dass für jedes $i = 1, \ldots, n$ die Einschränkung $s_{|I_i} \in \mathbb{P}_k$ der Interpolationsaufgabe mit den Stützstellen

$$x_{ij} = \Phi_i(\hat{x}_j), \qquad j = 1, \ldots, k \tag{3.93}$$

genügt.

3.2.4 Lemma: Die stückweise Interpolationsaufgabe hat eine eindeutige Lösung, wenn die Interpolationsaufgabe auf dem Referenzintervall eine solche besitzt.

3.2.5 Lemma (Skalierungsargument): Für die Lösung $\hat{p} \in \mathbb{P}_k$ der Interpolationsaufgabe auf dem Referenzintervall gelte mit einer Konstanten C unabhängig von $\hat{f} \in C^{k+1}(\hat{I})$ die Fehlerabschätzung

$$\|\hat{f} - \hat{p}\|_{\infty;\hat{I}} \leq C \|\hat{f}^{(k+1)}\|_{\infty;\hat{I}}. \tag{3.94}$$

Dann ist der Fehler der stückweisen Interpolation beschränkt ist durch

$$\|f - s\|_{\infty;[a,b]} \leq \frac{C}{2^{k+1}} h^{k+1} \|f^{(k+1)}\|_{\infty;[a,b]}. \tag{3.95}$$

Beweis. Für jedes Teilintervall I_j sei $\Phi_j \colon \hat{I} \to I_j$ die affine Referenzabbildung gemäß Definition 3.2.2. Es gelte dann $\hat{f}_j\big(\Phi_j^{-1}(x)\big) = f(x)$ und analog für $\hat{p}_j$. Offensichtlich gilt

$$\|f - p\|_{\infty;I_j} = \|\hat{f}_j - \hat{p}_j\|_{\infty;\hat{I}}. \tag{3.96}$$

Nun nutzen wir nach Voraussetzung

$$\|\hat{f}_j - \hat{p}_j\|_{\infty;\hat{I}} \leq C \|\hat{f}_j^{(k+1)}\|_{\infty;\hat{I}}. \tag{3.97}$$

Es gilt aber mit $x = \Phi_j(\hat{x})$ und $f(x) = \hat{f}(\hat{x})$

$$\hat{f}'(\hat{x}) = f'(x)\Phi_j'(\hat{x}) = \frac{h_j}{2} f'(x). \tag{3.98}$$

Damit gilt auch

$$\hat{f}_j^{(k+1)}(\hat{x}) = \left(\frac{h_j}{2}\right)^n f^{(n)}(x). \tag{3.99}$$

Einsetzen in (3.97) ergibt das Resultat. □

3.2.6 Bemerkung: Genauere Betrachtung der Analyse ergibt die schärfere Abschätzung

$$\|f - s\|_{\infty;[a,b]} \leq \frac{C}{2^{k+1}} \max_{i=1,\ldots,n} \left(h_i^{k+1} \|f^{(k+1)}\|_{\infty;I_i} \right). \tag{3.100}$$

Diese setzt die Ableitungen und Intervallängen lokal in Beziehung, was eine adaptierung der Intervallänge an die „Glattheit“ der Funktion erlaubt.

3.2.2 Splines

3.2.7 Definition: Für stückweise Polynome auf dem Intervall $[a, b]$ mit einer Zerlegung $\mathcal{I}_h$ definieren wir die **Spline-Räume**

$$S_{\mathcal{I}_h}^{(k,m)} = S_h^{(k,m)} = \left\{ s \in C^m[a,b] \middle| s_{|I_i} \in \mathbb{P}_k, i = 1, \ldots, n \right\} \tag{3.101}$$

mit $m < k$.

3.2.8 Lemma: Die Dimension von $S_h^{(k,m)}$ ist

$$\dim S_h^{(k,m)} = (k - m)n + m + 1 \tag{3.102}$$

Beweis. Betrachten wir die n Wiederholungen des Raums $\mathbb{P}_k$, eine für jedes Intervall I_i, so ergibt sich $(k + 1)n$. Die Bedingung $s \in C^m[a, b]$ bedeutet, dass die Werte und die ersten m Ableitungen der Funktionen in $S^{(k,m)}$ in jedem inneren Punkt x_i für die beiden Intervalle I_i und I_{i+1} übereinstimmen. Daraus ergeben sich $(n - 1)(m + 1)$ lineare Beschränkungen, so dass die Dimension $(k + 1)n - (n - 1)(m + 1)$ ist. □

3.2.9 Definition: Die Interpolationsaufgabe mit kubischen **Splines** lautet: finde eine Funktion $s \in S_h^{(3,2)}$, so dass

$$s(x_i) = f_i, \qquad i = 0, \ldots, n. \tag{3.103}$$

Hierbei definieren die Stützstellen x_i die Zerlegung $\mathcal{I}_h$.

3.2.10 Definition: Da die Anzahl der Interpolationsbedingungen um 2 geringer ist als die Dimension des Raumes $S_h^{(3,2)}$ definieren wir folgende, alternative Randbedingungen:

Natürlich

$$s''(a) = s''(b) = 0 \tag{3.104}$$

Periodisch

$$s'(a) = s'(b) \quad \wedge \quad s''(a) = s''(b) \tag{3.105}$$

Vollständig approximierend

$$s'(a) = f'(a) \quad \wedge \quad s'(b) = f'(b) \tag{3.106}$$

3.2.11 Satz: Die stückweise kubische Spline-Interpolierende $s \in S_h^{(3,2)}$ mit natürlicher Randbedingung existiert und ist eindeutig bestimmt.

Beweis. Wie meistens beginnen wir mit der Eindeutigkeit. Seinen s_1 und s_2 zwei Interpolierende der Werte f_i in den Punkten x_i, $i = 0, \ldots, n$ und $s = s_2 - s_1$. Dann gilt

$$s \in N_h = \big\{ w \in C^2[a,b] \;\big|\; w(x_i) = 0, \quad i = 0, \ldots, n \big\}. \tag{3.107}$$

Zusätzlich gilt $s_{|I_i} \in \mathbb{P}_3$ für alle Intervalle. Wir beobachten, dass für beliebiges $w \in N_h$ gilt

$$\int_{I_i} s''(x)w''(x)\,\mathrm{d}x = s''w'\Big|_{x_{i-1}}^{x_i} - \int_{I_i} s^{(3)}(x)w'(x)\,\mathrm{d}x \tag{3.108}$$

$$= s''w'\Big|_{x_{i-1}}^{x_i} - s^{(3)}w\Big|_{x_{i-1}}^{x_i} + \int_{I_i} s^{(4)}(x)w(x)\,\mathrm{d}x \tag{3.109}$$

$$= s''w'\Big|_{x_{i-1}}^{x_i}. \tag{3.110}$$

Summieren wir über alle Intervalle, so ergibt sich

$$\int_a^b s''(x)w''(x)\,\mathrm{d}x = \sum_{i=1}^{n} s''w'\Big|_{x_{i-1}}^{x_i} = s''(b)w'(b) - s''(a)w'(a). \tag{3.111}$$

Wegen der natürlichen Randbedingung ist dies aber null. Insbesondere können wir $w = s$ einsetzen und erhalten

$$\int_a^b |s''(x)|^2\,\mathrm{d}x = 0 \tag{3.112}$$

und s muss ein lineares Polynom sein. Aus $s(a) = s(b) = 0$ folgt damit $s \equiv 0$ im Widerspruch zur Annahme, dass es zwei Lösungen gebe.

Nach Lemma 3.2.8 hat $S_h^{(3,2)}$ die Dimension $n+3$. Andererseits haben wir $n+1$ Interpolationsbedingungen und 2 Randbedingungen, so dass aus der Eindeutigkeit die Existenz folgt. □

3.2.12 Lemma: Unter allen Funktionen $f \in C^2[a,b]$ mit vorgegebenen Funktionswerten $f(x_i) = y_i$, $i = 0,\ldots,n$ ist der natürliche Spline $s \in S_h^{(3,2)}$, der diese Punkte interpoliert, diejenige mit der kleinsten mittleren zweiten Ableitung, es gilt also

$$\int_a^b |s''(x)|^2\,\mathrm{d}x \le \int_a^b |f''(x)|^2\,\mathrm{d}x \qquad \forall f \in C^2[a,b]. \tag{3.113}$$

Beweis. Siehe [Rannacher, 2017, Satz 2.9] □

3.2.13 Lemma: Seien die Momente

$$M_i = s''(x_i), \qquad i = 0,\ldots,n \tag{3.114}$$

bekannt. Dann berechnen sich die Koeffizienten der Polynome auf den Teilintervallen I_i , $i = 1,\ldots,n$, dargestellt durch

$$s_{|I_i}(x) = a_{i0} + a_{i1}(x - x_i) + a_{i2}(x - x_i)^2 + a_{i3}(x - x_i)^3, \tag{3.115}$$

aus den Formeln

$$a_{i0} = f_i, \qquad a_{i1} = \tfrac{f_i - f_{i-1}}{h_i} + \tfrac{h_i(2M_i + M_{i-1})}{6}, \tag{3.116}$$

$$a_{i2} = \tfrac{M_i}{2} \qquad a_{i3} = \tfrac{M_i - M_{i-1}}{6h_i}. \tag{3.117}$$

Beweis. Siehe [Stoer, 1983, Abschnitt 2.4.2]. Wir bemerken: s'' ist eine stückweise lineare Funktion, die die Werte M_i interpoliert. Daher gilt

$$s''(x) = M_{i-1}\frac{x_i - x}{h_i} + M_i\frac{x - x_{i-1}}{h_i}, \qquad x \in I_i. \tag{3.118}$$

Daraus erhalten wir durch Integration

$$\begin{aligned} s'(x) &= -M_{i-1}\frac{(x_i - x)^2}{2h_i} + M_i\frac{(x - x_{i-1})^2}{2h_i} + A_i \\ s(x) &= M_{i-1}\frac{(x_i - x)^3}{6h_i} + M_i\frac{(x - x_{i-1})^3}{6h_i} + A_i(x - x_{i-1}) + B_i \end{aligned} \tag{3.119}$$

mit Integrationskonstanten A_i und B_i. Wegen der Interpolationsbedingungen in x_{i-1} und x_i muss gelten

$$B_i = y_{i-1} - M_{i-1}\frac{h_i^2}{6}, \qquad A_i = \frac{f_i - f_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1}). \tag{3.120}$$

Aus dieser Darstellung und der Beziehung $s^{(j)}(x_i) = j!a_{ij}$ erhalten wir die gewünschten Koeffizienten. □

3.2.14 Lemma: Die Momente M_i genügen dem linearen Gleichungssystem

$$\begin{pmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & \mu_n & 2 \end{pmatrix} \begin{pmatrix} M_0 \\ \vdots \\ M_n \end{pmatrix} = \begin{pmatrix} d_0 \\ \vdots \\ d_n \end{pmatrix} \tag{3.121}$$

wobei für $i = 1, \ldots, n-1$

$$\lambda_i = \tfrac{h_{i+1}}{h_i + h_{i+1}}, \qquad \mu_i = 1 - \lambda_i = \tfrac{h_i}{h_i + h_{i+1}}, \tag{3.122}$$

$$d_i = \tfrac{6}{h_i + h_{i+1}}\left[\tfrac{f_{i+1} - f_i}{h_{i+1}} - \tfrac{f_i - f_{i-1}}{h_i}\right] \tag{3.123}$$

Für natürliche Splines sind $\lambda_0 = \mu_n = 0$ und $d_0 = d_n = 0$. Für vollständig approximierende Splines ist $\lambda_0 = \mu_n = 1$ und

$$d_0 = \frac{6}{h_1}\left(\frac{f_1 - f_0}{h_1} - f_0'\right), \qquad d_n = \frac{6}{h_n}\left(f_n' - \frac{f_n - f_{n-1}}{h_n}\right). \tag{3.124}$$

Beweis. Siehe [Stoer, 1983, Abschnitt 2.4.2]. Die Stetigkeit von $s(x)$ und $s''(x)$ in den inneren Punkten x_i ergibt sich im vorhergehenden Beweis aus der Interpolation der f_i und M_i. Zusätzlich müssen wir die Stetigkeit von $s'(x)$ fordern.

Dazu benutzen wir die in Gleichung (3.119) hergeleitete Form: für $x \in I_i$ gilt

$$s'(x) = -M_{i-1}\frac{(x_i - x)^2}{2h_i} + M_i\frac{(x - x_{i-1})^2}{2h_i} + \frac{f_i - f_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1}). \tag{3.125}$$

Damit gilt am Punkt x_i

$$s'(x_i) = \frac{f_i - f_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1}) + M_i\frac{h_i}{2} \tag{3.126}$$

$$= \frac{f_i - f_{i-1}}{h_i} + \frac{h_i}{3}M_i + \frac{h_i}{6}M_{i-1} \tag{3.127}$$

$$s'(x_i) = \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{h_{i+1}}{6}(M_{i+1} - M_i) - M_i\frac{h_{i+1}}{2} \tag{3.128}$$

$$= \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{h_{i+1}}{3}M_i - \frac{h_{i+1}}{6}M_{i+1} \tag{3.129}$$

Aus der Gleichheit ergibt sich damit für $i = 1, \ldots, n-1$

$$\frac{h_i}{6}M_{i-1} + \frac{h_i + h_{i+1}}{3}M_i + \frac{h_{i+1}}{6}M_{i+1} = \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i}. \tag{3.130}$$

Multiplizieren dieser Gleichungen mit $6/(h_i + h_{i+1})$ ergibt die Gestalt der Matrix. Die natürliche Randbedingung ergibt $M_0 = 0$ und $M_n = 0$, was die Einträge in der ersten und letzten Zeile ergibt. □

3.2.15 Lemma: Die Matrix

$$A = \begin{pmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & \mu_n & 2 \end{pmatrix} \tag{3.131}$$

aus Lemma 3.2.14 hat die folgende Eigenschaft: für jeden Vektor $x \in \mathbb{R}^{n+1}$ und $y = Ax$ gilt

$$\|x\|_\infty \leq \|y\|_\infty. \tag{3.132}$$

Insbesondere ist A invertierbar.

Beweis. Sei k ein Index, so dass $|x_k| = \|x\|_\infty$. Dann gilt

$$y_k = \mu_k x_{k-1} + 2x_k + \lambda_k x_{k+1}. \tag{3.133}$$

Aus der Definition folgt $|\lambda_k| < 1$ und $|\mu_k| < 1$. damit gilt

$$\begin{aligned}\|y\|_\infty \geq |y_k| &\geq 2|x_k| - \mu_k|x_{k-1}| - \lambda_k|x_{k+1}| \\ &\geq |x_k|(2 - \mu_k - \lambda_k) \\ &\geq |x_k| = \|x\|_\infty.\end{aligned}$$

Wäre nun A singulär. Dann gäbe es $x \neq 0$ mit $Ax = 0$. Nach der Normabschätzung gilt dann aber $\|x\|_\infty = 0$ im Widerspruch zur Annahme. □

3.2.16 Satz: Sei $f \in C^4[a, b]$ und sei $\mathcal{I}_h$ eine Zerlegung der Feinheit h, für die es zusätzlich eine Konstante $c > 0$ gibt mit

$$\min_i h_i \geq ch. \tag{3.134}$$

Dann gilt für die Ableitungen der Ordnung $\nu = 0, \ldots, 3$ des vollständig approximierenden Spline s zu den Funktionswerten $f(x_i)$ die Abschätzung

$$\|f^{(\nu)} - s^{(\nu)}\|_{\infty;[a,b]} \leq c_\nu c h^{4-\nu} \|f^{(4)}\|_{\infty;[a,b]} \tag{3.135}$$

mit Konstanten c_ν unabhängig von $\mathcal{I}_h$ und f.

Beweis. Sei $g = (f''(x_0), \ldots, f''(x_n))^T$ der Vektor der zweiten Ableitungen von f in den Punkten x_i. Der Schlüssel ist die Abschätzung

$$\|M - g\|_\infty \leq \tfrac{3}{4} \|f^{(4)}\|_\infty h^2. \tag{3.136}$$

Dazu untersuchen wir den Vektor $r = A(M - g) = d - Ag$. Nach Lemma 3.2.15 gilt

$$\|M - g\|_\infty \leq \|r\|_\infty. \tag{3.137}$$

Wir betrachten den Punkt x_0 und nutzen die Taylor-Interpolation

$$f(x_1) = f(x_0) + h_1 f'(x_0) + \frac{h_1^2}{2} f''(x_0) + \frac{h_1^3}{6} f^{(3)}(x_0) + \frac{h_1^4}{24} f^{(4)}(\xi_0), \tag{3.138}$$

$$f''(x_1) = f''(x_0) + h_1 f^{(3)}(x_0) + \frac{h_1^2}{2} f^{(4)}(\xi_1) \tag{3.139}$$

wobei $\xi_0, \xi_1 \in I_0$ gilt. Daraus folgt

$$\begin{aligned} r_0 =& d_0 - 2f''(x_0) - f''(x_1) && (3.140)\\ =& \frac{6}{h_1}\left(\frac{f_1 - f_0}{h_1} - f_0'\right) - 2f''(x_0) - f''(x_1) && (3.141)\\ =& \frac{6}{h_1}\left[f'(x_0) + \frac{h_1}{2}f''(x_0) + \frac{h_1^2}{6}f^{(3)}(x_0) + \frac{h_1^3}{24}f^{(4)}(\xi_0) - f'(x_0)\right] && (3.142)\\ & - 2f''(x_0) - \left[f''(x_0) + h_1 f^{(3)}(x_0) + \frac{h_1^2}{2}f^{(4)}(\xi_1)\right] && (3.143)\\ =& \frac{h_1^2}{4}f^{(4)}(\xi_0) - \frac{h_1^2}{2}f^{(4)}(\xi_1). && (3.144) \end{aligned}$$

Damit gilt

$$|r_0| \le \tfrac{3}{4}\|f^{(4)}\|_\infty h^2. \tag{3.145}$$

Dasselbe gilt für $r_n = d_n - f''(x_{n-1}) - 2f''(x_n)$. Für die anderen Punkte ist mit demselben Argument und mehr Rechenaufwand

$$\begin{aligned} r_i &= d_i - \mu_i f''(x_{i-1}) - 2f''(x_i) - \lambda_i f''(x_{i+1}) && (3.146)\\ &= \frac{1}{h_i + h_{i+1}}\left[\frac{h_{i+1}^3}{4}f^{(4)}(\xi_1) + \frac{h_i^3}{4}f^{(4)}(\xi_2) - \frac{h_{1+1}^3}{2}f^{(4)}(\xi_3) - \frac{h_i^3}{2}f^{(4)}(\xi_4)\right]. \end{aligned}$$

Daher ist

$$|r_i| \le \tfrac{3}{4}\|f^{(4)}\|_{\infty;[a,b]}h^2, \qquad i = 1, \dots, n-1. \tag{3.147}$$

Aus (3.137) schließen wir

$$\|M - g\|_\infty \le \|r\|_\infty \le \frac{3}{4}h^2\|f^{(4)}\|_{\infty;[a,b]}. \tag{3.148}$$

Nun zeigen wir die Behauptung des Satzes für $\nu = 3$. Sei $e(x) = s(x) - f(x)$. Für $x \in I_i$ ist

$$\begin{aligned} e^{(3)}(x) = \frac{M_i - M_{i-1}}{h_i} - f^{(3)}(x)& \\ = \frac{M_i - f''(x_i)}{h_i} - \frac{M_{i-1} - f''(x_{i-1})}{h_i}& \\ + \frac{f''(x_i) - f''(x) - \big(f''(x_{i-1}) - f''(x)\big)}{h_i} - f^{(3)}(x).& \quad (3.149) \end{aligned}$$

Die Werte in den Stützpunkten schätzen wir nun durch Taylor-Entwicklung um x ab:

$$\begin{aligned} f''(x_i) &= f''(x) + f^{(3)}(x)(x_i - x) + \frac{f^{(4)}(\xi_1)}{2}(x_i - x)^2\\ f''(x_{i-1}) &= f''(x) + f^{(3)}(x)(x_{i-1} - x) + \frac{f^{(4)}(\xi_1)}{2}(x_{i-1} - x)^2. \end{aligned} \tag{3.150}$$

Setzen wir dies und (3.148) in (3.149) ein, so erhalten wir

$$\begin{aligned}|s^{(3)}(x) - f^{(3)}(x)| &\leq \frac{3}{2}\frac{h^2}{h_i}\|f^{(4)}\|_{\infty;[a,b]} + \frac{h_i^2}{2h_i}\|f^{(4)}\|_{\infty;[a,b]} \\ &\leq 2ch\|f^{(4)}\|_{\infty;[a,b]} \end{aligned} \tag{3.151}$$

Nun $\nu = 2$. Sei $\tilde{x} \in \{x_{i-1}, x_i\}$ der nächste Stützpunkt zu x, so dass $|x - \tilde{x}| \leq h/2$. Es gilt für die zweiten Ableitungen

$$e''(x) = e''(\tilde{x}) + \int_{\tilde{x}}^{x} e^{(3)}(t)\,\mathrm{d}t, \tag{3.152}$$

so dass wir mit (3.148) und (3.151) folgern

$$\begin{aligned}|s''(x) - f''(x)| &\leq \frac{3}{4}h^2\|f^{(4)}\|_{\infty;[a,b]} + ch^2\|f^{(4)}\|_{\infty;[a,b]} \\ &\leq \frac{7}{4}ch^2\|f^{(4)}\|_{\infty;[a,b]}. \end{aligned} \tag{3.153}$$

Aus den Interpolationsbedingungen folgt $e(x_i) = 0$ für $i = 0, \ldots, n$. Damit gibt es nach dem Satz von Rolle in jedem Intervall I_i ein ξ_i mit $e'(\xi_i) = 0$. Somit gilt für $x \in I_i$

$$e'(x) = \int_{\xi_i}^{x} e''(t)\,\mathrm{d}t \tag{3.154}$$

und daher mit (3.153)

$$|s'(x) - f'(x)| \leq \frac{7}{4}ch^3\|f^{(4)}\|_{\infty;[a,b]}. \tag{3.155}$$

Für $\nu = 0$ können wir wieder $\tilde{x}$ wie oben wählen und erhalten aus

$$e(x) = \int_{\tilde{x}}^{x} e'(t)\,\mathrm{d}t \tag{3.156}$$

die Abschätzung

$$|s'(x) - f'(x)| \leq \frac{7}{8}ch^4\|f^{(4)}\|_{\infty;[a,b]}. \tag{3.157}$$

□

Bemerkung 3.2.17. Werte wie $7/4$ oder $7/8$ in der obigen Abschätzung suggerieren, dass sie sehr scharf ist. In der Tat ist das aber nur so, wenn weder $f^4(x)$, noch h_i stark variieren. Wir haben mehrfach $\|f^{k+1}\|_{\sup;I_i}$ durch $\|f^{k+1}\|_{\sup;[a,b]}$ sowie h_i durch h/c ersetzt. Jedesmal hat sih der Fehler erhöht.

Daraus ergibt sich, dass die wesentliche Aussage der Abschätzung ist: es gilt

$$\|f^{(\nu)} - s^{(\nu)}\|_{\infty;[a,b]} = \mathcal{O}(h^{4-\nu}), \tag{3.158}$$

wobei die Konstante von der 4. Ableitung der Funktion und der Gleichmäßigkeit des Gitters abhängt.

Diese Abschätzung gilt in der Tat nur für den voll approximierenden Spline. Für andere Randbedingungen gilt sie nur dann, wenn die Randbedingungen von f zufällig übereinstimmen.

3.3 Interpolatorische Quadratur

3.3.1 Summierte Quadratur

3.3.1 Definition: Eine **Quadraturformel** $Q_{[a,b]}(f)$ ist eine Approximation des Integrals

$$Q_{[a,b]}(f) \approx \int_a^b f(x)\,\mathrm{d}x \tag{3.159}$$

in der Form

$$Q_{[a,b]}(f) = \sum_{i=0}^{n} \omega_i f(x_i). \tag{3.160}$$

Die Stützstellen x_i bezeichnen wir auch als **Quadraturpunkte**, die Zahlen ω_i als **Quadraturgewichte**.

3.3.2 Definition: Ist eine Quadraturormel bezüglich einer Zerlegung $\mathcal{I}_h$ des Intervalls $[a, b]$ in der Form

$$Q_{[a,b]}(f) = \sum_{i=1}^{n} Q_{I_i}(f) \tag{3.161}$$

definiert, so sprechen wir von **summierter**, **iterierter** oder **stückweiser Quadratur**.

3.3.3 Satz: Die Integration einer Funktion $f \in C[a,b]$ über das Intervall $[a,b]$ genügt der Konditionsabschätzung

$$\left| \int_a^b f(x)\,\mathrm{d}x \right| \leq \kappa_{\mathrm{abs}} \max_{x\in[a,b]} |f(x)|, \qquad \kappa_{\mathrm{abs}} = b-a. \tag{3.162}$$

Für die Quadraturaufgabe gilt

$$\left|Q_{[a,b]}(f)\right| \leq \kappa_{\mathrm{abs}} \max_{x\in[a,b]} |f(x)|, \qquad \kappa_{\mathrm{abs}} = \sum_i |\omega_i|. \tag{3.163}$$

Beweis. Für die Integration ist aus der Analysis die Abschätzung

$$\left| \int_a^b f(x)\,\mathrm{d}x \right| \leq (b-a) \max_{x\in[a,b]} |f(x)| \tag{3.164}$$

bekannt. Für die Quadratur gilt genauso

$$\left| \sum_{i=1}^n \omega_i f(x_i) \right| \leq \left(\sum_{i=1}^n |\omega_i| \right) \max_{x\in[a,b]} |f(x)|. \tag{3.165}$$

Wählen wir eine Funktion mit $f(x_i) = \operatorname{sign}\omega_i$, so sehen wir, dass die zweite Abschätzung scharf ist. Für die erste ist das offensichtlich, wenn f konstant ist. □

3.3.4 Definition: Gilt bei einer summierten Quadraturformel die Abschätzung

$$\left| \int_{I_i} f(x)\,\mathrm{d}x - Q_{I_i}(f) \right| = \mathcal{O}\left(h_i^{k+1}\right) \tag{3.166}$$

für jedes Teilintervall I_i und Funktionen $f \in C^{k+1}[a,b]$, so sprechen wir von der **lokalen Fehlerordnung** $k+1$.

3.3.5 Satz: Sei $\mathcal{I}_h$ eine Zerlegung von $[a,b]$ der Feinheit h und c_q sei so gewählt, dass

$$c_q \min_{I_i \in \mathcal{I}_h} h_i \geq h. \tag{3.167}$$

Sind dann die Formeln Q_{I_i} von lokaler Fehlerordnung $k+1$ für $f \in C^{k+1}[a,b]$, so gilt für die summierte Quadratur $Q_{[a,b]}$ die Abschätzung

$$\left| \int_a^b f(x)\,\mathrm{d}x - Q_{[a,b]}(f) \right| = \mathcal{O}\left(h^k\right). \tag{3.168}$$

Beweis. Das kleinste Intervall hat die Länge h/c_q. Damit ist die Anzahl der Intervalle beschränkt durch $n_{\max} = c_q(b-a)/h$. Aus der lokalen Fehlerordnung ergibt sich die Existenz einer Konstanten c, so dass

$$\left| \int_{I_i} f(x)\,\mathrm{d}x - Q_{I_i}(f) \right| \leq c h_i^{k+1}. \tag{3.169}$$

Damit schätzen wir ab

$$\begin{aligned} \left| \int_a^b f(x)\,\mathrm{d}x - Q_{[a,b]}(f) \right| &= \sum_{I_i \in \mathcal{I}_h} \left| \int_{I_i} f(x)\,\mathrm{d}x - Q_{I_i}(f) \right| && (3.170)\\ &\leq \sum_{I_i \in \mathcal{I}_h} c h^{k+1} && (3.171)\\ &\leq n_{\max} c h^{k+1} = \mathcal{O}\left(h^k\right). && (3.172) \end{aligned}$$

□

3.3.2 Quadratur auf Einzelintervallen

3.3.6 Notation: In diesem Abschnitt integrieren wir wieder über das Intervall $I = [a, b]$, aber mit dem Gedanken, dass es sich eigentlich um die Teilintervalle I_i einer summierten Quadratur handelt.
Wir betrachten in der Regel Quadraturformeln mit n Punkten $x_1, \dots, x_n$.
Oft benutzen wir Ergebnisse aus den Abschnitten über Interpolation. Dabei ist jeweis darauf zu achten, dass die Indizes dort bei null loslaufen. Der Grund für diesen Wechsel ist, dass wir bei der Interpolation den Grad der Polynome als führende Größe angesehen haben, während hier die Anzahl der Quadraturpunkte im Vordergrund steht.
Bei der summierten Quadratur steht die Anzahl der Intervalle im Vordergrund. Deswegen werden dort die Punkte weiterhin mit null beginnend numeriert.

3.3.7 Definition: Eine Quadraturformel Q_I heißt **exakt vom Grad** k und k heißt der **Grad der Exaktheit** von Q_I, wenn sie exakt für alle Polynome vom Grad bis zu k ist, also

$$\int_I p(x)\,\mathrm{d}x - Q_I(p) = 0 \qquad \forall p \in \mathbb{P}_k. \tag{3.173}$$

Bemerkung 3.3.8. Als unmittelbare Folgerung erhalten wir für jede Quadraturformel, die mindestens exakt vom Grad null ist, dass

$$\sum \omega_i = b - a. \tag{3.174}$$

Insbesondere gilt dann, dass die Konditionierung der Quadratur gleich der der Integration ist, wenn alle Gewichte positiv sind. Umgekehrt führen negative Gewichte automatisch zur Verschlechterung der Konditionierung, weswegen solche Formeln vermieden werden.

3.3.9 Lemma: Sei die Quadraturformel Q_I exakt vom Grad k und $|I| \leq h$. Dann gilt für $f \in C^{k+1}(I)$

$$\left| \int_I f(x)\,\mathrm{d}x - Q_I(f) \right| = \mathcal{O}\big(h^{k+2}\big) \tag{3.175}$$

Beweis. Ersetzen wir f durch sein Tailorpolynom $p \in \mathbb{P}_k$ um einen Punkt $x_0 \in I$, so erhalten wir

$$f(x) = p(x) + r(x), \qquad r(x) = (x - x_0)^{k+1} \frac{f^{(k+1)}(\xi)}{(k+1)!} \tag{3.176}$$

für einen Punkt ξ zwischen x_0 und x. Aufgrund der Exaktheit gilt

$$\int_I f(x)\,\mathrm{d}x - Q_I(f) = \underbrace{\int_I p(x)\,\mathrm{d}x - Q_I(p)}_{=0} + \int_I r(x)\,\mathrm{d}x - Q_I(r). \tag{3.177}$$

Für den Rest schätzen wir ab:

$$(x - x_0) \le h, \quad \int_I (x - x_0)^{k+1} = \frac{1}{k+2}(x - x_0)^{k+2}, \qquad \sum_{\omega_i} = h, \tag{3.178}$$

und die Ableitung von f durch ihr Maximum auf I. □

Bemerkung 3.3.10. Die Überlegungen des vorhergehenden Lemmas sind insbesondere dann nützlich, wenn man die Konvergenzordnung einer Methode schnell überschlagen möchte. Man spart sich auf diese Art genauere Untersuchungen der Fehlerdarstellung. Umgekehrt sind die Konstanten in den Abschätzungen auch scharf und die Analysis erhebt auch nicht den Anspruch. Im Detail werden wir daher auch noch schärfere Abschätzungen machen.

3.3.11 Definition: Eine **interpolatorische Quadraturformel** mit n Quadraturpunkten $x_1, \ldots, x_n$ approximiert das Integral einer Funktion f durch das exakte Integral ihres Interpolationspolynoms $p \in \mathbb{P}_{n-1}$

3.3.12 Lemma: Seien $x_1, \ldots, x_n$ die Quadraturpunkte einer interpolatorischen Quadraturformel Q_I, die exakt für Polynome vom Grad $n-1$ ist. Dann sind die Gewichte gegeben durch

$$\omega_i = \int_I \ell_{i;x_1,\ldots,x_n}(x)\,\mathrm{d}x, \tag{3.179}$$

wobei $\ell_{i;x_1,\ldots,x_n}$ das Lagrange-Interpolationspolynom zum Punkt x_i ist.

Beweis. Die Lagrange-Polynome ℓ_i sind Polynome vom Grad $n-1$. Es gilt daher

$$\int_I \ell_i = \sum_{k=1}^{n} \omega_k \ell_i(x_k) = \omega_i. \tag{3.180}$$

□

3.3.13 Definition: Werden die Quadraturpunkte $a = x_1, \ldots, x_n = b$ gleichmäßig im Intervall $[a, b]$ verteit, so spricht man von einer **Newton-Cotes-Formel**. Die ersten drei klassischen Formeln sind auf dem Einheitsintervall $[0, 1]$ gegeben durch

	n	x_i				ω_i			
Trapezregel	2	0	1			$1/2$	$1/2$		
Simpson-Regel[a]	3	0	$1/2$	1		$1/6$	$4/6$	$1/6$	
$3/8$-Regel[b]	4	0	$1/3$	$2/3$	1	$1/8$	$3/8$	$3/8$	$1/8$

[a] Auch Keplersche Fassregel
[b] Von Newton auch mit dem Adjektiv „pulcherrima“ belegt.

Bemerkung 3.3.14. Genauer gesagt handelt es sich bei den Formeln in Definition 3.3.13 um **geschlossene** Newton-Cotes-Formeln, die die Intervallenden als Quadraturpunkte enthalten. Bei offenen Formeln sind die Intervallenden keine Quadraturpunkte. Während offene Newton-Cotes-Formeln von geringem Interesse sind, werden wir später bei der Gauß-Quadratur offene Formeln kennenlernen.

Summierte geschlossene Formeln werden of direkt als Summe über die Zerlegung geschrieben. Dazu numerieren wir nun alle Stützpunkte der Reihe nach, egal ob es sich um Intervallenden oder innere Stützpunkte handelt von 0 bis n. Sei h nun der Abstand zweier Quadraturpunkte der summierten Formel. dann gilt für die summierte Trapezregel

$$Q(f) = h\left(\tfrac{1}{2}f_0 + f_1 + \cdots + f_{n-1} + \tfrac{1}{2}f_n\right). \tag{3.181}$$

Bei der summierten Simpson-Regel (man beachte die Umskalierung der Intervall-Länge) ergibt sich

$$Q(f) = \tfrac{h}{3}\left(f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \cdots + 2f_{n-2} + 4f_{n-1} + f_n\right). \tag{3.182}$$

Die Anzahl der Teilintervalle ist bei dieser Darstellung n für die Trapezregel und $2n$ für die Simpson-regel.

Der Aufwand für Funktionsauswertungen in den Intervallenden halbiert sich bei dieser Darstellung der summierten Regeln. Natürlich lassen sich die summierten Formeln für Intervalle wechselnder Länge umschreiben. Bei der Simpson-Regel muss man aber dabei Bedenken, dass immer 2 Subintervalle ein Intervall der Zerlegung ergeben.

3.3.15 Satz: Die Fehler der Newton-Cotes-Formeln auf dem Intervall I der Länge h lassen sich wie folgt abschätzen

$$\left| \int_I f \,\mathrm{d}x - Q_I(f) \right| \leq \begin{cases} \frac{h^3}{12} \max_{\xi \in I} |f''(\xi)| & \text{Trapezregel} \\ \frac{h^5}{2880} \max_{\xi \in I} |f^{(4)}(\xi)| & \text{Simpson-Regel} \\ \frac{h^5}{6480} \max_{\xi \in I} |f^{(4)}(\xi)| & 3/8\text{-Regel} \end{cases} \tag{3.183}$$

Beweis. Der Beweis für die Trapezregel und die $3/8$-Regel benutzt Interpolation in den Quadraturpunkten und die Fehlerdarstellung des Interpolationsfehlers. Für die Trapezregel ist er als Hausaufgabe gestellt.

Hier führen wir nur den Beweis für die Simpson-Regel. Nachdem man experimentell beobachtet, dass die Formel exakt vom Grad 3 ist, nicht vom erwarteten Grad 2, konstruieren wir eine Interpolation auf $I = [x_1, x_3]$ mit Mittelpunkt x_2 wie folgt:

$$p(x_1) = f(x_1) \qquad p(x_2) = f(x_2) \tag{3.184}$$
$$p(x_3) = f(x_3) \qquad p'(x_2) = f'(x_2). \tag{3.185}$$

Die letzte Bedingung ist aus der Quadraturformel nicht ersichtlich. Folgen wir jedoch der Basiskonstruktion im Satz 3.1.28 über die Wohlgestelltheit der Hermite-Interpolationsaufgabe, so erhalten wir

$$H_{10}(x) = \tfrac{4(x-x_2)^2(x_3-x)}{h^3} \qquad H_{20}(x) = \tfrac{4(x-x_1)(x-x_3)}{h^2} \tag{3.186}$$
$$H_{30}(x) = \tfrac{4(x-x_2)^2(x-x_1)}{h^3} \qquad H_{21}(x) = \tfrac{4(x-x_2)(x-x_1)(x-x_3)}{h^2} \tag{3.187}$$

Die Funktion $H_{21}(x)$ ist das Produkt der Parabel $(x - x_1)(x - x_3)$, die symmetrisch zur Intervallmitte ist mit einer linearen Funktion mit Nullstelle in der Intervallmitte. Daher verschwindet ihr Integral und das zugehörige Integrationsgewicht ist null. Die Simpson-Regel lässt sich also schreiben als

$$Q_I = \frac{h}{6} f(x_1) + \frac{4h}{6} f(x_2) + \frac{h}{6} f(x_3) + 0 f'(x_2). \tag{3.188}$$

Für die obige Interpolation gilt nach Satz 3.1.33 die Fehlerdarstellung

$$f(x) - p(x) = \frac{f^4(\xi(x)}{4!} \omega_{x_1,x_2,x_2,x_3}(x). \tag{3.189}$$

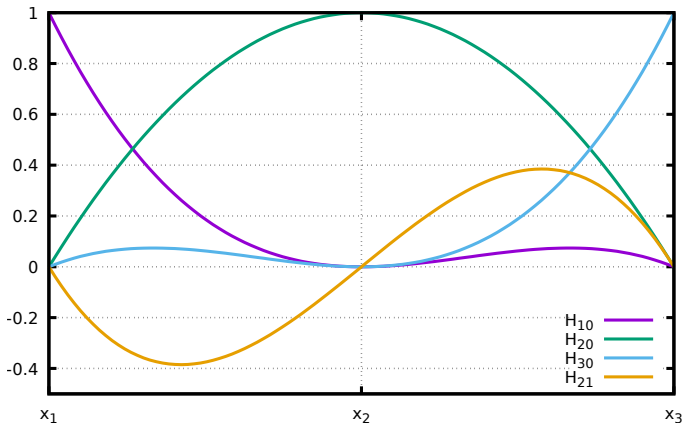

Abbildung 3.1: Basisfunktionen für die Simpsonregel. Beachte, dass H_21 in allen Quadraturpunkten verschwindet und auch das Integral null ist.

Integration ergibt

$$\left| \int_I f \,\mathrm{d}x - Q_I(f) \right| = \left| \int_I \big(f(x) - p(x)\big) \,\mathrm{d}x \right| \tag{3.190}$$

$$\leq \max_{\xi \in I} \frac{f^4(\xi}{4!} \int_I \omega_{x_1,x_2,x_2,x_3}(x) \,\mathrm{d}x. \tag{3.191}$$

Schließlich berechnen wir

$$\int_I \omega_{x_1,x_2,x_2,x_3}(x) \,\mathrm{d}x = \int_I (x - x_1)(x - x_2)^2(x - x_3) = \frac{1}{120} \tag{3.192}$$

□

Bemerkung 3.3.16. Die Newton-Cotes-Formel mit 9 Punkten hat negative Gewichte, was auch bei höheren Formeln wieder auftritt. Bei solchen Formeln wird die Konditionierung der Quadraturaufgabe schlechter als die der Integration. Auch kann es sein, dass eine nirgendwo negative Funktion durch eine solche Quadratur ein negatives Integral erhält. Deswegen werden solche Formeln nicht verwendet.

3.3.3 Gauß-Quadratur

3.3.17 Lemma: Sei Q_n eine Quadraturformel auf einem Intervall I mit n Quadraturpunkten. Dann ist Q_n maximal exakt vom Grad $2n - 1$.

Beweis. Siehe auch [Rannacher, 2017, Satz 3.1]. Für die QuadraturPunkte x_i definieren wir das quadrierte Newton-Polynom

$$p(x) = \omega_{1,\dots,n}^2 = \prod_{i=1}^{n} (x - x_i)^2 \in \mathbb{P}_{2n}. \tag{3.193}$$

Da es in allen Stützstellen verschwindet, gilt $Qp = 0$. Da es aber nichtnegativ und nicht das Nullpolynom ist, so ist sein Integral größer als null. Damit gibt es für jede n-Punkt-Formel ein Polynom, für das sie nicht exakt ist. □

3.3.18 Satz: Sei in der Folge von Quadraturformeln $\{Q_n\}$ mit Quadraturpunkten $x_1^{(n)}, \dots, x_n^{(n)}$ für $n = 1, \dots$ auf dem Intervall $I = [-1, 1]$ jede Formel Q_n exakt für beliebige $p \in \mathbb{P}_{2n-1}$. Dann sind die Polynome

$$p_n(x) = \prod_{i=1}^{n} \left(x - x_i^{(n)} \right) \in \mathbb{P}_n \tag{3.194}$$

und $p_0(x) = 1 \in \mathbb{P}_0$ paarweise orthogonal bezüglich des L^2-Skalarprodukts. Insbesondere sind sie damit Vielfache der Legendre-Polynome L_n und die Formeln sind eindeutig bestimmt.

Beweis. Siehe [Deuflhard and Hohmann, 2008, Lemma 9.9 und 9.10]. Sei $j < n$. Dann ist $p_j p_n \in \mathbb{P}_{2n-1}$. Es gilt also

$$\int_{-1}^{1} p_j p_n \,\mathrm{d}x = Q_n(p_j p_n) = \sum_{i=0}^{n} \omega_i p_j(x_i^{(n)}) p_n(x_i^{(n)}). \tag{3.195}$$

Da die Punkte $x_i^{(n)}$ aber gerade die Nullstellen von p_n sind, muss dieser Term null sein. Wir haben damit eine Folge orthogonaler Polynome steigenden Grades. Dazu hatten wir in Satz 1.5.1 nachgewiesen, dass eine solche Folge bis auf Skalierung eindeutig bestimmt ist. Die Polynome p_n sind also Vielfache der Legendre-Polynome L_n und insbesondere die Nullstellen durch die Bedingung eindeutig festgelegt. □

3.3.19 Definition: Die n-Punkt-**Gauß-Legendre-Formel** auf dem Intervall $I = [-1, 1]$ benutzt als Stützstellen $x_1, \dots, x_n$ die Nullstellen des Legendre-Polynoms $L(x)$ vom Grad n. Ihre Quadraturgewichte sind die Integrale der Lagrange-Polynome

$$\omega_i = \int_I \ell_{i;x_1,\dots,x_n}(x) \,\mathrm{d}x. \tag{3.196}$$

3.3.20 Satz: Die n-Punkt-Gauß-Legendre-Formel wohldefiniert. Sie ist exakt für beliebige Polynome vom Grad $2n-1$ genau dann, wenn sie exakt vom Grad $n-1$ ist.

Beweis. Zunächst müssen wir zeigen, dass das Legendre-Polynom L_n genau n paarweise verschiedene, reelle Nullstellen hat. Seien dazu $\lambda_1, \ldots, \lambda_m$ die Nullstellen ungerade Vielfachheit mit $m \leq n$. Wir führen das Hilfspolynom q ein, für das gelte $q \equiv 1$, falls es keine solche Nullstelle gibt und sonst

$$q(x) = \prod_{i=1}^{m} (x - \lambda_i). \tag{3.197}$$

Dieses Polynom hat m reelle Nullstellen. Das Polynom $L_n q \in \mathbb{P}_{n+m}$ hat dann keinen Vorzeichenwechsel und es gilt daher

$$\int_{-1}^{1} L_n q \, \mathrm{d}x \neq 0. \tag{3.198}$$

Aufgrund der Orthogonalität folgt damit $m = n$ und $q = L_n$.

Nun wenden wir uns dem Grad der Exhaktheit zu. Ein Polynom $p \in \mathbb{P}_{2n-1}$ können wir durch Division mit Rest als Summe

$$p(x) = q(x)L_n(x) + r(x) \tag{3.199}$$

darstellen, wobei $q, r \in \mathbb{P}_{n-1}$. Es gilt dann wegen der Orthogonalität

$$\int_{-1}^{1} p \, \mathrm{d}x = \int_{-1}^{1} qL_n \, \mathrm{d}x + \int_{-1}^{1} r \, \mathrm{d}x = \int_{-1}^{1} r \, \mathrm{d}x. \tag{3.200}$$

Für die Quadratur gilt, da die Quadraturpunkte die Nullstellen von L_n sind,

$$Q_n(p) = Q_n(qL_n) + Q(r) = Q(r). \tag{3.201}$$

Die Quadratur ist also genau dann exakt, wenn für beliebiges $r \in \mathbb{P}_{n-1}$

$$Q_n(r) = \int_{-1}^{1} r \, \mathrm{d}x. \tag{3.202}$$

□

3.3.21 Lemma: Die Gewichte der Gauss-Legendre-Formeln sind positiv und genügen der Darstellung

$$\omega_i = \int_{-1}^{1} \prod_{j \neq i} \left(\frac{x - x_j}{x_i - x_j} \right)^2 \mathrm{d}x. \tag{3.203}$$

Beweis. Da die x_i die Nullstellen von L_n sind, gilt $L_n(x) = \prod(x - x_i)$. Deshalb gilt für

$$q(x) = \left(\frac{L_n(x)}{x - x_i} \right)^2 = \prod_{j \neq i} (x - x_j)^2, \tag{3.204}$$

dass $q \in \mathbb{P}_{2n-2}$. Ferner ist $q(x_j) = 0$ für $j \neq i$ und $q(x_i) \neq 0$.. Daher gilt auf Grund der Exaktheit

$$\omega_i = \frac{1}{q(x_i)} \int q \, \mathrm{d}x. \tag{3.205}$$

Das ist aber genau die Darstellung (3.203). Da $q(x) \geq 0$ ist die rechte Seite positiv. □

3.3.22 Lemma: Für die Gauss-Legendre-Formel mit n Quadraturpunkten auf $I = [-1, 1]$ gilt die Fehlerabschätzung

$$\left| \int_I f \, \mathrm{d}x - Q_n(f) \right| \leq \max_{\xi \in I} \frac{f^{(2n)}(\xi)}{(2n)!} \int_{-1}^{1} \prod_{i=1}^{n} (x - x_i)^2. \tag{3.206}$$

Beweis. Dies ist eine direkte Anwendung der Fehlerdarstellung für die Lagrange-Interpol □

Bemerkung 3.3.23. Alle Resultate dieses Abschnitts gelten für Skalarprodukte der Form

$$\langle p, q \rangle = \int_I \omega(x) p(x) q(x) \, \mathrm{d}x \tag{3.207}$$

mit einer positiven Gewichtsfunktion $\omega(x)$, wenn man die Legendre-Polynome durch die entsprechenden orthogonalen Polynome ersetzt.

3.3.4 Richardson-Extrapolation und Romberg-Quadratur

3.3.24 Definition: Sei $T(h)$ eine numerische Methode zur Approximation des tatsächlichen Wertes $T(0)$ mit Diskretisierungsparameter h und Fehlerabschätzung $|T(h) - T(0)| = \mathcal{O}(h^p)$. Zur **Richardson-Extrapolation** wertet man diese Methode mit einer Schrittfolge $h_1, h_2, \ldots, h_n$ aus, so dass die Schrittweite (theoretisch) gegen null geht. Wertet man dann das Interpolationspolynom $p(h^p)$ an der Stelle $h = 0$ aus, so bekommt man unter stärkeren Voraussetzungen die verbesserte Approximation

$$|T(0) - p(0)| = \mathcal{O}(h^{np}). \tag{3.208}$$

Bemerkung 3.3.25. Tatsächlich genügt die einfache Fehlerabschätzung $|T(h) - T(0)| = \mathcal{O}(h^p)$ nicht, um die behauptete Konvergenzordnung zu beweisen. Man benötigt eine asymptotische Fehlerentwicklung der Form

$$T(h) - T(0) = \tau_1 h^p + \tau_2 h^{2p} + \ldots \tau_n h^{np} + \mathcal{O}(h^{(n+1)p}). \tag{3.209}$$

3.3.26 Definition: Die **Romberg-Quadratur** beruht auf einer summierten Quadraturformel Q_h der Ordnung h^p, die für eine Folge von Schrittweiten $h_1, \ldots, h_n$ angewandt wird. Aus diesen berechnet man mit dem Neville-Algorithmus Approximationen für Q_0.

3.3.27 Algorithmus (Romberg-Quadratur):

```
1  def romberg(n0,steps,order,quadrature,f):
2      res = np.zeros((steps,steps))
3      for i in range(0,steps):
4          n = n0 * 1<<i
5          res[i,0] = quadrature(n,f)
6          for j in range(1,i+1):
7              res[i,j] = res[i,j-1] + \
8              (res[i,j-1]-res[i-1,j-1])/(2**(order*j)-1.)
9      return res
```

3.3.28 Aufgabe (Romberg-Quadratur): Schreiben Sie eine Funktion, die die Funktion $f(x) = \sin(\pi x)$ über das Intervall $[0, 1]$ mit der ierierten Trapezregel integriert. Wenden Sie die Romberg-Quadratur mit der Schrittfolge

$$h = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots \tag{3.210}$$

an und beobachten Sie die Konvergenz gegen den exakten Integralwert $2/\pi$ für die verschiedenen Spalten im Tableau des Neville-Algorithmus.

3.3.29 Beispiel (Romberg-Quadratur mit Trapezregel): Mit der summierten Trapezregel Q_h approximieren wir das Integral $\int_0^1 \sin \pi x \, \mathrm{d}x$ mit den Schrittweiten h (erste beiden Spalten in der folgenden Tabelle).

h_i	$Q_{i0} = Q_{h_i}$	Q_{i1}	Q_{i2}	Q_{i3}
$1/2$	0.5			
$1/4$	0.60355339	0.63807119		
$1/8$	0.62841744	0.63670545	0.6366144	
$1/16$	0.63457315	0.63662505	0.63661969	0.63661978

Die Einträge der hinteren Spalten berechnen sich nach dem Neville-Algorithmus durch die Formel

$$Q_{ik} = Q_{i,k-1} + \frac{h_i^2}{h_{i-k}^2 - h_i^2} \left(Q_{i,k-1} - Q_{i-1,k-1} \right). \tag{3.211}$$

was der Auswertung der Fehlerentwicklung als Polynom in h^2 im Punkt null entspricht.

3.3.5 Praktische Aspekte

Bemerkung 3.3.30. Die Konvergenzabschätzungen der Form

$$\left| \int_I f \, \mathrm{d}x - Q_h(f) \right| \leq c h^p \| f^{p+1} \|_{\infty; I} \tag{3.212}$$

verlieren ihren Nutzen für große h, wenn die Ableitungen von f wachsen. Schlimmstenfalls bekommt man dann aus der Interpolationseigenschaft noch immer

$$\left| \int_I f \, \mathrm{d}x - Q_h(f) \right| \leq c \| f \|_{\infty; I}. \tag{3.213}$$

Es gibt aber keine Garantie, dass der Fehler bei feinerer Unterteilung schrumpft.

Dies gilt auch für große h, falls $f \in C^{p+1}(I)$ aber die Ableitungen mit steigender Ordnung schnell wachsen. Dann kann zunächst der Gewinn durch Wahl einer feineren Unterteilung durch das Wachsen der Ableitung annulliert werden. Man nennt das Verhalten dann **präasymptotisch**.

Für hinreichend feine Unterteilungen h_1 und h_2 gilt die obige Abschätzung aber in der stärkeren Form

$$\left| \int_I f \,\mathrm{d}x - Q_{h_2}(f) \right| \approx \left(\frac{h_2}{h_1} \right)^p \left| \int_I f \,\mathrm{d}x - Q_{h_1}(f) \right|. \tag{3.214}$$

Man beobachtet also die Konvergenzordnung als direkte Verbesserung mit jeder Wahl eines feineren Parameters. Dieses Verhalten nennt man **asymptotisch**.

Bemerkung 3.3.31. Die Konvergenzordnung eines Verfahrens lässt sich auch experimentell bestimmen. Sei dazu $T(h)$ eine numerische Methode mit Diskretisierungsparameter h und der Fehler verhalte sich wie

$$|T(h) - T(0)| = ch^p + o(h^p). \tag{3.215}$$

Wenn die exakte Lösung $T(0)$ bekannt ist, so lässt sich die linke Seite für verschiedene Parameter h berechnen. Benutzt man zwei verschiedene Schrittweiten, so lassen sich mit der Abkürzung $e(h) = |T(h) - T(0)|$ die Werte p und c aus dem linearen Gleichungssystem

$$\begin{aligned} \log c + p \log h_1 &= \log\big(e(h_1)\big), \\ \log c + p \log h_2 &= \log\big(e(h_2)\big). \end{aligned} \tag{3.216}$$

bestimmen. Da hier die unbekannten Terme $o(h)$ weggelassen wurden, ist diese Bestimmung nicht exakt, konvergiert aber für $h \to 0$. Führt man die Bestimmung für jeweils aufeinanderfolgende Paare von Parametern h und einer Folge durch, so konvergieren die Werte von p und c, so dass man dadurch die Zuverlässigkeit der Schätzung einschätzen kann.

Findet man keine relevante Aufgabe, deren exakte Lösung bekannt ist, so kann man den Wert $T(0)$ durch Richardson-Extrapolation nähern.

Der Nutzen dieser Technik liegt nicht nur in der experimentellen Bestätigung der theoretischen Beweise. Sie erlaubt es, die Konstante c zu Schätzen, für die der Beweis oft nur Existenz liefert. Auch kann dadurch die Optimalität des theoretischen Ergebnisses überprüft werden. Schließlich erlaubt diese Technik auch, die Konvergenzordnung zu schätzen um dann erst zu sehen, wie man diese Ordnung auch beweist.

Bemerkung 3.3.32. Wenn eine ekakte Lösung nicht verfügbar ist und Extrapolation zur null nicht sinnvoll, so gibt es noch die Option, die **intrinsische**

Konvergenzordnung zu betrachten. Dazu betrachten wir zum Beispiel die Folge

$$d_k = |T(2^{-k}h) - T(2^{-(k+1)}h)|, \qquad k = 1, \dots \tag{3.217}$$

Gilt $d_k \approx 2^{-pk}$ mit $p > 0$, so gilt aufgrund der Konvergenz der geometrischen Reihe

$$|T(h) - T(0)| \leq \sum_{k=1}^{\infty} \left| T\big(2^{-k}h\big) - T\big(2^{-(k+1)}h\big) \right| \leq \frac{1}{1-2^p} |T(h) - T(h/2)|. \tag{3.218}$$

Daraus schließt man, dass auch die Konvergenzordnung etwa gleich p ist. Auch dies funktioniert nur, wenn h bereits im asymptotischen Bereich liegt.

3.3.33 Aufgabe: Bestimmen sie aus dem Neville-Tableau aus Aufgabe 3.3.28 die experimentellen und intrinsischen Konvergenzraten der Spalten.

Kapitel 4

Iterationsverfahren

4.0.1. Die Bestimmung der Nullstellen einer Funktion ist eine oft wiederkehrende Aufgabe der Numerik, sei es zum Beispiel die Bestimmung der Stützstellen der Gauß-Quadratur. Eine geschlossene Darstellung ist dabei nur in den seltensten Fällen möglich, bei Polynomen zum Beispiel nur bis zum Grad 3. Stattdessen benutzt man sukzessive Annäherungen an die Nullstelle, sogenannte Iterationsverfahren.

Das folgende Heron-Verfahren war bereits zur Zeit des Königs Hammurabi in Mesopotamien, also vor fast 4000 Jahren bekannt. Es ist benannt nach Heron von Alexandria, in dessen Metrika es beschrieben ist.

4.0.2 Definition (Heron-Verfahren): Zur Bestimmung der Quadratwurzel der Zahl a kann folgende Iteration mit beliebigem, positiven Startwert $x^{(0)}$ angewandt werden:

$$x^{(k+1)} = \frac{1}{2}\left(x^{(k)} + \frac{a}{x^{(k)}}\right) \tag{4.1}$$

4.0.3 Algorithmus (Heron-Verfahren):

```
1 def heron(x,steps):
2   for k in range(1,steps):
3     x = .5*(x+a/x)
4   return x
```

4.0.4 Beispiel: Das Heron-Verfahren zur Bestimmung der Quadratwurzel von 3 liefert folgende Iterationsfolge

k	$x^{(k)}$	$x^{(k)} - \sqrt{3}$
0	1	-0.7320508075688772
1	2.0	0.2679491924311228
2	1.75	0.017949192431122807
3	1.7321428571428572	9.204957398001312e-05
4	1.7320508100147274	2.44585018904786e-09
5	1.7320508075688772	0.0

4.1 Grundlagen

4.1.1 Fixpunktiterationen

4.1.1 Definition: Ein **Iterationsverfahren** berechnet schrittweise Approximationen an die Lösung x einer Aufgabe aus einem Startwert $x^{(0)}$ mit der Verfahrensvorschrift der Form

$$x^{(k+1)} = f(x^{(k)}), \qquad k = 0, 1, 2, \ldots. \tag{4.2}$$

Wir nennen f die **Iterationsfunktion** und $\{x^{(k)}\}$ die **Iterationsfolge** zu f mit Startwert $x^{(0)}$.
Das Verfahren heißt **konvergent**, wenn gilt $x^{(k)} \to x$ für $k \to \infty$.

4.1.2 Definition: Ein Iterationsverfahren ist konvergent mindestens von Ordnung $p > 1$ zum Grenzwert x, wenn es eine Konstante $c > 0$ gibt, so dass

$$\left\| x^{(k+1)} - x \right\| \leq c \left\| x^{(k)} - x \right\|^p \tag{4.3}$$

gilt. Es ist linear konvergent, wenn

$$\left\| x^{(k+1)} - x \right\| \leq c \left\| x^{(k)} - x \right\| \tag{4.4}$$

mit einer Konstanten $c < 1$. Wir sprechen von superlinearer Konvergenz, wenn

$$\left\| x^{(k+1)} - x \right\| = o\left(\left\| x^{(k)} - x \right\| \right) \tag{4.5}$$

4.1.3 Definition: Sei $M \subset \mathbb{R}^n$. Eine Abbildung $f\colon M \to M$ ist eine **Kontraktion** auf M, wenn es eine Konstante $\varrho < 1$ gibt, so dass

$$\|f(x) - f(y)\| \leq \varrho \|x - y\| \qquad \forall x, y \in M. \tag{4.6}$$

4.1.4 Satz (Banachscher Fixpunktsatz): Sei f eine Kontraktion auf der abgeschlossenen Menge $M \subset \mathbb{R}^n$. Dann gibt es genau einen **Fixpunkt** $x^* \in M$, also

$$x^* = f(x^*). \tag{4.7}$$

Für jeden Startwert $x^{(0)} \in M$ konvergiert die Folge definiert durch

$$x^{(k+1)} = f\left(x^{(k)}\right) \tag{4.8}$$

gegen diesen Fixpunkt. Es gelten die Fehlerabschätzungen

$$\|x^{(k)} - x^*\| \leq \frac{\varrho}{1-\varrho} \|x^{(k)} - x^{(k-1)}\| \leq \frac{\varrho^k}{1-\varrho} \|x^{(1)} - x^{(0)}\|. \tag{4.9}$$

Beweis. Zunächst zeigen wir die Eindeutigkeit nach der üblichen Methode: seien $x, y \in M$ Fixpunkte der Kontraktion $f(\cdot)$. Dann gilt

$$\|x - y\| = \|f(x) - f(y)\| \leq \varrho \|x - y\|. \tag{4.10}$$

Da $\varrho < 1$ kann dies nur gelten, wenn die Differenz verschwindet.

Als nächstes zeigen wir, dass die Iterationsvorschrift eine Cauchy-Folge erzeugt, woraus die Konvergenz gegen einen Grenzwert folgt. Dazu beobachten wir zunächst, dass mit $x^{(0)} \in M$ auch alle weiteren Folgenglieder in M liegen. Ferner gilt:

$$\|x^{(k+1)} - x^{(k)}\| \leq \varrho \|x^{(k)} - x^{(k-1)}\| \leq \varrho^k \|x^{(1)} - x^{(0)}\|. \tag{4.11}$$

Daher gilt für $m \geq 1$

$$\|x^{(k+m)} - x^{(k)}\| \leq \sum_{i=1}^{m} \|x^{(k+i)} - x^{(k+i-1)}\| \tag{4.12}$$

$$\leq \sum_{i=1}^{m} \varrho^i \|x^{(k)} - x^{(k-1)}\| \tag{4.13}$$

$$\leq \sum_{i=1}^{m} \varrho^{k+i-1} \|x^{(1)} - x^{(0)}\| \tag{4.14}$$

$$\leq \varrho^k \sum_{i=0}^{\infty} \varrho^i \|x^{(1)} - x^{(0)}\| \tag{4.15}$$

$$\leq \frac{\varrho^k}{1-\varrho} \|x^{(1)} - x^{(0)}\|. \tag{4.16}$$

Daher gilt das Cauchy-Kriterium: für alle $\varepsilon > 0$ gibt es ein $k_0 \geq 0$, so dass für alle $k \geq k_0$ und alle $m \geq 1$ gilt, dass $\|x^{(k+m)} - x^{(k)}\| < \varepsilon$. Damit existiert also der Grenzwert x^* und liegt wegen der Abgeschlossenheit in M. aus der Konvergenz der Folge folgt auch

$$\|f\left(x^{(k)}\right) - x^{(k)}\| = \|x^{(k+1)} - x^{(k)}\| \to 0, \tag{4.17}$$

und damit im Limes $x^* = f(x^*)$. Für die Fehlerabschätzung beobachten wir, dass die Norm links in (4.12) für $m \to \infty$ gegen $\|x^* - x^{(k)}\|$ konvergiert, so dass mit demselben Argument wie dort die Abschätzungen gelten. □

4.1.5 Satz: Sei $f\colon \mathbb{R}^n \to \mathbb{R}^n$ eine Selbstabbildung mit einem Fixpunkt x^*, so dass für eine Kugel vom Radius R um x^* mit $p > 1$ gilt,

$$\left\|f\left(x^{(k)}\right) - x^*\right\| \leq c\left\|x^{(k)} - x^*\right\|^p. \tag{4.18}$$

Dann gibt es einen Radius $r \leq R$, so dass die zugehörige Iterationsfolge $\{x^{(0)}\}_{k=0,\ldots,\infty}$ für alle Startwerte $x^{(0)} \in B_r(x^*)$ konvergiert.

Beweis. Sei r so gewählt, dass

$$\varrho = c\|x - x^*\|^{p-1} < 1 \qquad \forall x \in B_r(x^*). \tag{4.19}$$

Dann gilt $f\colon B_r(x^*) \to B_r(x^*)$ und

$$\|f(x) - x^*\| \leq \varrho \|x - x^*\| \qquad \forall x \in B_r(x^*). \tag{4.20}$$

Für die Iterationsfolge gilt also

$$\|x^{(k)} - x^*\| \leq \varrho^k \|x^{(0)} - x^*\| \qquad \forall x \in B_r(x^*). \tag{4.21}$$

□

4.1.6 Satz: Sei $g\colon \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar. Dann gilt für eine Minimalstelle x^* von g, also

$$x^* = \operatorname*{argmin}_{x\in\mathbb{R}^n} g(x), \tag{4.22}$$

notwendig

$$\nabla g(x^*) = 0. \tag{4.23}$$

Das Minimierungsproblem lässt sich also auf das Finden einer Nullstelle von $f\colon \mathbb{R}^n \to \mathbb{R}^n$ mit $f(x) = \nabla g(x)$ reduzieren. Umgekehrt lässt sich die Aufgabe, eine Nullstelle der Funktion $f\colon \mathbb{R}^n \to \mathbb{R}^n$ zu finden, durch die Minimierung der Norm $g(x) = \|f(x)\|$ darstellen.

Beweis. Die erste Tatsache ist aus der Analysis bekannt. Die zweite Aussage ergibt sich daraus, dass $g(x)$ nirgendwo negativ ist, eine Nullstelle also ein Minimum sein muss. □

Bemerkung 4.1.7. Beide Aussagen des vorigen Satzes lassen keine Umkehr zu: weder ist jede Nullstelle des Gradienten ein Minimum, noch ist jedes lokale Minimum von g eine Nullstelle.

4.1.2 Vektor- und Matrixnormen

4.1.8 Definition: Eine **Norm** $\|\cdot\|$ auf dem Vektorraum V ist eine Abbildung

$$\begin{aligned} \|\cdot\|\colon V &\to \mathbb{R} \\ x &\mapsto \|x\| \end{aligned} \tag{4.24}$$

mit den Eigenschaften

Homogenität:	$\|\alpha x\| = \|\alpha\|\|x\|$	$\forall \alpha \in \mathbb{R}, x \in V$	(4.25)
Dreiecksungleichung:	$\|x + y\| \leq \|x\| + \|y\|$	$\forall x, y \in V$	(4.26)
Definitheit:	$\|x\| \geq 0$	$\forall x \in V$	(4.27)
	$\|x\| \neq 0$	$\forall x \neq 0$	(4.28)

Verzichtet man auf die zweite Definitheitsbedingung, so erhält man eine **Seminorm**.

4.1.9 Definition: Sei V ein reeller oder komplexer Vektorraum. Zwei Normen $\|\cdot\|_X$ und $\|\cdot\|_Y$ auf V heißen äquivalent, wenn es Konstanten $c > 0$ und $C > 0$ gibt, so dass

$$c\|v\|_X \leq \|v\|_Y \leq C\|v\|_X \qquad \forall v \in V. \tag{4.29}$$

4.1.10 Definition: Eine Folge $\{x^{(k)}\} \subset \mathbb{R}^n$ für $k = 1, 2, \ldots$ heißt **komponentenweise konvergent** gegen $x \in \mathbb{R}^n$, wenn gilt

$$\forall \varepsilon > 0 \; \exists k_0 \in \mathbb{N} \; \forall k \geq k_0, i = 1, \ldots, n : |x_i^{(k)} - x_i| < \varepsilon. \tag{4.30}$$

Die Folge heißt konvergent unter der Norm $\|\cdot\|$ oder **normkonvergent** wenn gilt

$$\forall \varepsilon > 0 \; \exists k_0 \in \mathbb{N} \; \forall k \geq k_0 : \|x^{(k)} - x\| < \varepsilon. \tag{4.31}$$

4.1.11 Lemma: Sei $\|\cdot\|$ eine beliebige Norm auf $\mathbb{R}^n$. Dann ist die Abbildung

$$f \colon x \mapsto \|x\| \tag{4.32}$$

stetig bezüglich der komponentenweisen Konvergenz. Ferner ist die Norm $\|\cdot\|$ äquivalent zur Maximumsnorm.

Beweis. Für den ersten Teil ist zu zeigen, dass zu einer komponentenweise konvergenten Folge von Vektoren auch deren Norm konvergiert. Sei $\{x^{(k)}\}$ eine solche Folge und dazu k_0 so gewählt, dass

$$\max_{i=1,\ldots,n} \left| \left(x_i^{(k)} - x_i \right) \|e_i\| \right| < \frac{\varepsilon}{n} \qquad \forall k \geq k_0. \tag{4.33}$$

Hier ist e_i der i-te Einheitsvektor. Dann folgt

$$\|x^{(k)} - x\| = \left\| \sum_{i=1}^{n} \left(x_i^{(k)} - x_i \right) e_i \right\| \tag{4.34}$$

$$\leq \sum_{i=1}^{n} \left\| \left(x_i^{(k)} - x_i \right) e_i \right\| \tag{4.35}$$

$$< n \frac{\varepsilon}{n} = \varepsilon. \tag{4.36}$$

Hiermit haben wir bereits bewiesen, dass komponentenweise Konvergenz auch Normkonvergenz impliziert.

Die „Einheitssphäre“

$$S = \left\{ x \in \mathbb{R}^n \middle| \|x\|_\infty = 1 \right\} \tag{4.37}$$

ist beschränkt und bezüglich der komponentenweisen Konvergenz abgeschlossen. Die Norm $\|.\|$ nimmt dort als stetige Funktion ihr Minimum c und ihr Maximum C an. Insbesondere gilt aber wegen der Definitheit $c > 0$. Für einen beliebigen Vektor $x \in \mathbb{R}^n$ ist $x/\|x\|_\infty \in S$, so dass gilt

$$c\|x\|_\infty \leq \|x\| \leq C\|x\|_\infty. \tag{4.38}$$

□

4.1.12 Satz: Auf $\mathbb{R}^n$ sind zwei beliebige Normen $\|.\|_X$ und $\|.\|_Y$ äquivalent.

Beweis. Nach dem vorherigen Lemma sind beide Normen äquivalent zur Maximumsnorm. Es gibt also Konstanten $c_X, c_Y, C_X, C_y > 0$ mit

$$\begin{aligned} c_X\|x\|_\infty \leq \quad \|x\|_X \quad &\leq C_X\|x\|_\infty \\ c_Y\|x\|_\infty \leq \quad \|x\|_Y \quad &\leq C_Y\|x\|_\infty. \end{aligned} \tag{4.39}$$

Daher gilt

$$\begin{aligned} \|x\|_Y \leq C_Y\|x\|_\infty &\leq \frac{C_Y}{c_X}\|x\|_X \\ \|x\|_X \leq C_X\|x\|_\infty &\leq \frac{C_X}{c_y}\|x\|_Y \end{aligned} \tag{4.40}$$

□

4.1.13 Lemma: Sind zwei Normen $\|\cdot\|_X$ und $\|\cdot\|_Y$ äquivalent, so konvergiert die Folge $x^{(k)}$ in $\mathbb{R}^n$ bezüglich der Norm $\|\cdot\|_X$ genau dann, wenn sie bezüglich der Norm $\|\cdot\|_Y$ konvergiert.

Bemerkung 4.1.14. Da die Definition der komponentenweisen Konvergenz der Konvergenz bezüglich der Maximumsnorm $\|\cdot\|_\infty$ entspricht, konvergiert damit eine Folge $x^{(k)}$ in $\mathbb{R}^n$ genau dann komponentensweise, wenn sie in einer beliebigen Norm konvergiert.

Diese Aussage überträgt sich *nicht* auf die Kontraktionseigenschaft! Eine lineare Abbildung $x \mapsto Ax$ kann sehr wohl eine Kontraktion bezüglich einer Norm sein, aber nicht einer anderen.

4.1.15 Aufgabe: Zeigen Sie Lemma 4.1.13. Zeigen Sie die Aussage über Kontraktionen in der vorigen Bemerkung durch ein Gegenbeispiel.

4.1.16 Definition: Auf dem Vektorraum der Matrizen $\mathbb{R}^{m\times n}$ ist durch Definition 4.1.8 eine Norm definiert. Gilt zusätzlich

$$\|Ax\| \leq \|A\|\|x\| \qquad \forall A \in \mathbb{R}^{m\times n}, x \in \mathbb{R}^n, \tag{4.41}$$

so heißt die Norm $\|.\|$ der Matrix **verträglich** mit der Vektornorm $\|.\|$. Wir sprechen von einer **Matrixnorm**, wenn sie zusätzlich **submultiplikativ** ist, dass heißt, für alle Matrizen A, B passender Dimensionen gilt

$$\|AB\| \leq \|A\|\|B\|. \tag{4.42}$$

Ferner definieren wir die **Operatornorm** oder **natürliche Norm**

$$\|A\| = \sup_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|. \tag{4.43}$$

Bemerkung 4.1.17. In der Definition der Operatornorm wird das Supremum über alle von null verschiedenen Vektoren gebildet. Andererseits ist der Quotient aber für $x = 0$ offensichtlich sinnlos. Um die Notation einfach zu halten sei daher im folgenden vereinbart, dass im Ausdruck

$$\sup_{x\in\mathbb{R}^n} \frac{\|Ax\|}{\|x\|} \tag{4.44}$$

der Wert $x = 0$ implizit ausgenommen sei.

4.1.18 Lemma: Die Operatornorm ist verträglich zu ihrer Vektornorm und submultiplikativ.

Beweis. Sei $\|x\|$ die gewählte Norm in $\mathbb{R}^n$ und $\|A\|$ die zugehörige Operatornorm. Die Verträglichkeit folgt aus

$$\|Ax\| = \frac{\|Ax\|}{\|x\|}\|x\| \leq \sup_{x\in\mathbb{R}^n} \frac{\|Ax\|}{\|x\|}\|x\| = \|A\|\|x\|. \tag{4.45}$$

Für die Submultiplikativität folgt aus der Verträglichkeit

$$\|AB\| = \sup_{x\in\mathbb{R}^n} \frac{\|ABx\|}{\|x\|} \leq \sup_{x\in\mathbb{R}^n} \frac{\|A\|\|Bx\|}{\|x\|} = \|A\| \sup_{x\in\mathbb{R}^n} \frac{\|Bx\|}{\|x\|} = \|A\|\|B\| \tag{4.46}$$

□

4.1.19 Beispiel: Die Operatornormen zu den Vektornormen $\|.\|_1$ und $\|.\|_\infty$ sind die **Spaltensummennorm** und die **Zeilensummennorm**

$$\|A\|_1 = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{ji}| \tag{4.47}$$

$$\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{ij}| \tag{4.48}$$

4.1.20 Notation: Werden im folgenden Normen ohne Index notiert, so handelt es sich bei Vektoren um eine beliebige Vektornorm und bei Matrizen um die zugehörige Operatornorm.

4.1.3 Eigenwerte und die Spektralnorm

4.1.21 Definition: Sei $A \in \mathbb{R}^{n\times n}$. Gilt für einen Vektor $0 \neq v \in \mathbb{R}^n$

$$Av = \lambda v, \tag{4.49}$$

so nennen wir λ **Eigenwert** von A und v einen zugehörigen **Eigenvektor**. Wir notieren die Zugehörigkeit zur Matrix A auch explizit durch $\lambda(A)$.

4.1.22 Lemma: Für alle Eigenwerte $\lambda \in \mathbb{C}$ einer Matrix $A \in \mathbb{R}^{n\times n}$ gilt

$$|\lambda| \leq \|A\| \tag{4.50}$$

für jede zu einer beliebigen Vektornorm verträglichen Norm.

Beweis. Es gilt für den Eigenwert λ mit zugehörigem Eigenvektor v:

$$|\lambda|\|v\| = \|\lambda v\| = \|Av\| \leq \|A\|\|v\|. \tag{4.51}$$

□

4.1.23 Satz: Sei $A \in \mathbb{R}^n \times n$ eine symmetrische Matrix. Dann gibt es eine Orthonormalbasis des $\mathbb{R}^n$ von Eigenvektoren $v^{(i)}$ mit zugehörigen reellen Eigenwerten λ_i.

Beweis. Resultat der linearen Algebra. □

4.1.24 Satz: Die Operatornorm zur euklidischen Norm ist die **Spektralnorm**

$$\|A\|_2 = \max_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} \sqrt{\frac{x^T A^T A x}{x^T x}} = \sqrt{\lambda_{\max}(A^T A)}. \tag{4.52}$$

Insbesondere gilt für symmetrische Matrizen $\|A\|_2 = \max_i |\lambda_i(A)|$.

Beweis. Nach Satz 4.1.23 gibt es eine Basis des $\mathbb{R}^n$ von Eigenvektoren $v^{(i)}$ von $A^T A$. Jeder beliebige Vektor x besitzt damit die Darstellung

$$x = \sum_{i=1}^{n} \alpha_i v^{(i)}. \tag{4.53}$$

Es gilt nach der Parsevalschen Gleichung $\|x\|_2 = \|\alpha\|_2$. Ferner gilt mit den Eigenwerten $\lambda_i = \lambda_i(A^T A)$

$$\|Ax\|_2^2 = x^T A^T A x = \sum_{i=1}^{n} \lambda_i \alpha_i^2. \tag{4.54}$$

Daher gilt

$$\|A\|_2^2 = \max_{x\in\mathbb{R}^n} \frac{\|Ax\|^2}{\|x\|^2} = \max_{\alpha} \frac{\sum \lambda_i \alpha_i^2}{\sum \alpha_i^2} \leq \lambda_{\max}(A^T A). \tag{4.55}$$

Da für symmetrische Matrizen $A = A^T$, so ist

$$\lambda_{\max}(A^T A) = \lambda_{\max}(A^2) = \lambda_{\max}^2(A) \tag{4.56}$$

□

4.1.25 Definition: Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt **positiv definit**, wenn

$$x^T A x > 0 \qquad \forall 0 \neq x \in \mathbb{R}^n. \tag{4.57}$$

4.1.26 Satz: Eine symmetrische Matrix $A \in \mathbb{R}^{n\times n}$ ist positiv definit genau dann, wenn ihre Eigenwerte alle positiv sind.

4.1.27 Definition: Die **Konditionszahl** der Matrix A zur Norm $\|\cdot\|$ ist das Produkt

$$\operatorname{cond}(A) = \|A\| \, \|A^{-1}\| \tag{4.58}$$

Insbesondere definieren wir die **Spektralkondition** einer s.p.d. Matrix als

$$\operatorname{cond}_2(A) = \|A\|_2 \, \|A^{-1}\|_2 = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}. \tag{4.59}$$

4.2 Das Newton-Verfahren

4.2.1 Definition: Das **Newton-Verfahren** ist ein Iterationsverfahren zum Auffinden einer Nullstelle einer Funktion $f\colon \mathbb{R}^n \to \mathbb{R}^n$. Zu einem Startwert $x^{(0)} \in \mathbb{R}^n$ berechnen sich die weiteren Iterierten durch

$$x^{(k+1)} = x^{(k)} - \big(\nabla f(x^{(k)})\big)^{-1} f(x^{(k)}). \tag{4.60}$$

4.2.2 Algorithmus (Newton-Verfahren):

```
1 def newton(x, f, Dfinv, tol):
2   r = f(x)
3   while (abs(r)>tol):
4     d = Dfinv(x,r)
5     x -= d
6     r = f(x)
7   return x
```

Die Parameter zu dieser Funktion sind der Startwert x, die Funktion $f(x)$, die Anwendung der inversen Ableitung

$$\operatorname{Dfinv}(x, r) = \big(\nabla f(x)\big)^{-1} r, \tag{4.61}$$

sowie eine Toleranz als Abbruchkriterium.

4.2.3 Lemma: Sei $M \subset \mathbb{R}^n$ konvex. Sei $f\colon M \to \mathbb{R}^n$ stetig differenzierbar auf M und die Ableitung genüge der Lipschitz-Abschätzung

$$\|\nabla f(x) - \nabla f(y)\| \le \gamma \|x - y\| \qquad \forall x, y \in M. \tag{4.62}$$

mit einer Konstanten γ. Dann gilt für alle $x, y \in M$

$$\|f(x) - f(y) - \nabla f(y)(x - y)\| \le \frac{\gamma}{2}\|x - y\|^2. \tag{4.63}$$

Beweis. Wir folgen [Stoer, 1983, Hilfssatz 5.3.1]. Sei $\varphi\colon [0,1] \to \mathbb{R}^n$ die Hilfsfunktion definiert durch

$$\varphi(t) = f\big(y + t(x - y)\big), \tag{4.64}$$

so dass

$$f(x) - f(y) - \nabla f(y)(x - y) = \varphi(1) - \varphi(0) - \varphi'(0) = \int_0^1 (\varphi'(t) - \varphi'(0))\,\mathrm{d}t, \tag{4.65}$$

denn nach der Kettenregel gilt

$$\varphi'(t) = \nabla f\big(y + t(x - y)\big)(x - y). \tag{4.66}$$

Den Integranden schätzen wir ab durch

$$\begin{aligned} \|\varphi'(t) - \varphi'(0)\| &= \|\Big(\nabla f\big(y - t(x - y)\big) - \nabla f(y)\Big)(x - y)\| && (4.67)\\ &\le \|\nabla f\big(y - t(x - y)\big) - \nabla f(y)\|\|(x - y)\| && (4.68)\\ &\le \gamma\, t\|x - y\|^2. && (4.69) \end{aligned}$$

Einsetzen ins Integral ergibt

$$\|f(x) - f(y) - \nabla f(y)(x - y)\| \le \frac{\gamma}{2}\|x - y\|^2. \tag{4.70}$$

□

4.2.4 Satz: Sei $M \subset \mathbb{R}^n$ eine offene, konvexe Menge und $f\colon \overline{M} \to \mathbb{R}^n$ stetig differenzierbar in M und stetig auf $\overline{M}$. Die **Jacobi-Matrix** $\nabla f(x)$ sei auf ganz M invertierbar und es gebe Konstanten β und γ, so dass für $x, y \in M$ gilt

$$\|\nabla f(x) - \nabla f(y)\| \leq \gamma \|x - y\|, \qquad \|\big(\nabla f(x)\big)^{-1}\| \leq \beta. \tag{4.71}$$

Gibt es dann eine Konstante α, so dass

$$\|\big(\nabla f(x^{(0)})\big)^{-1} f(x^{(0)})\| \leq \alpha \tag{4.72}$$

$$h := \frac{\alpha\beta\gamma}{2} < 1 \tag{4.73}$$

$$\overline{B_r(x^{(0)})} \subseteq M, \qquad \text{mit } r = \frac{\alpha}{1-h}, \tag{4.74}$$

So ist die Folge $x^{(k)}$ des Newton-Verfahrens für alle $k = 1, \ldots$ wohldefiniert und liegt in $B_r(x^{(0)})$. Ferner konvergiert sie quadratisch gegen einen Wert $x^* \in \overline{B_r(x^{(0)})}$ und es gilt

$$\|x^{(k)} - x^*\| \leq \alpha \frac{h^{2^k - 1}}{1 - h^{2^k}}. \tag{4.75}$$

Beweis. Wir folgen [Stoer, 1983, Satz 5.3.2]. Wir zeigen zunächst induktiv für alle $k = 1, \ldots$, dass das Folgenglied $x^{(k)}$ in $B_r(x^{(0)}) \subseteq M$ liegt. Damit existiert dann nach Voraussetzung $\big(\nabla f(x^{(k)})\big)^{-1}$ und $x^{(k+1)}$ ist wohldefiniert. Zur Verankerung bemerken wir, dass offensichtlich $x^{(0)} \in B_r(x^{(0)})$ und $x^{(1)}$ nach Voraussetzung (4.72). Nach der Verfahrensvorschrift können wir abschätzen:

$$\|x^{(k+1)} - x^{(k)}\| = \|\big(\nabla f(x^{(k)})\big)^{-1} f(x^{(k)})\| \tag{4.76}$$

$$\leq \beta \|f(x^{(k)})\| \tag{4.77}$$

$$= \beta \|f(x^{(k)}) - f(x^{(k-1)}) - \nabla f(x^{(k)})(x^{(k)} - x^{(k-1)})\|, \tag{4.78}$$

wobei wir die letzte Zeile aus der Multiplikation der Verfahrensvorschrift mit ∇f gewonnen haben. Hierauf wenden wir nun Lemma 4.2.3 an und bekommen die quadratische Konvergenz, wenn der Abstand zweier Folgenglieder einmal klein genug ist:

$$\|x^{(k+1)} - x^{(k)}\| \leq \frac{\beta\gamma}{2} \|x^{(k)} - x^{(k-1)}\|^2. \tag{4.79}$$

Es bleibt zu zeigen, dass die Folge in $B_r(x^{(0)})$ bleibt. Dazu zeigen wir per Induktion, dass

$$\|x^{(k+1)} - x^{(k)}\| \leq \alpha h^{2^k - 1}. \tag{4.80}$$

Für $k = 0$ folgt $\|x^{(1)} - x^{(0)}\| \leq \alpha$ direkt aus (4.72). Für den Induktionsschritt benutzen wir unsere Konvergenzabschätzung (4.79):

$$\|x^{(k+1)} - x^{(k)}\| \leq \frac{\beta\gamma}{2}\|x^{(k)} - x^{(k-1)}\|^2 \leq \frac{\beta\gamma}{2}(\alpha h^{2^{k-1}-1})^2 = \frac{\alpha\beta\gamma}{2}\alpha h^{2^k-2} = \alpha h^{2^k-1}. \tag{4.81}$$

Nun können wir mit einer Teleskopsumme abschätzen

$$\begin{aligned} \|x^{(k+1)} - x^{(0)}\| &\leq \sum_{j=0}^{k} \|x^{(j+1)} - x^{(j)}\| && (4.82)\\ &\leq \alpha(1 + h + h^3 + h^7 + \cdots + h^{2^k-1}) && (4.83)\\ &< \frac{\alpha}{1-h} = r, && (4.84) \end{aligned}$$

Aus (4.80) folgt mit dieser Abschätzung, dass $x^{(k)}$ Cauchy Folge ist und durch Grenzübergang die Abschätzung (4.75). □

4.3 Abstiegsverfahren und Globalisierung

4.3.1. Die lokale Konvergenz ist beim Newtonverfahren ein großes Hindernis für die Anwendung. Wählt man den Startwert nicht im Einzugsbereich einer Nullstelle, so divergiert das Verfahren. Der Einzugsbereich, wie er sich aus dem Konvergenzsatz ergibt, ist dabei oft sehr klein und daher schwer zu finden.

Ziel dieses Abschnitts ist daher, eine Modifikation des Newton-Verfahrens zu finden, die den Konvergenzbereich aufweitet, idealerweise sogar globale Konvergenz erzeugt. Solche Modifikationen findet man unter der Bezeichnung **Globalisierung**.

4.3.2 Definition: Sei $g\colon \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar. Dann definieren wir den Kegel positiven Anstiegs zum Parameter γ als

$$S_\gamma(x) = \big\{ s \in \mathbb{R}^n \big| \|s\| = 1 \ \wedge \ \nabla g(x) \cdot s \geq \gamma \|\nabla g(x)\| \big\}. \tag{4.85}$$

Die Richtung des steilsten Anstiegs im Punkt x ist $\nabla g(x)$.

Da es sich um normierte Vektoren handelt, ist die Menge $S_\gamma(x)$ eigentlich kein Kegel, sondern eine Kugel in der Einheitssphäre mit Zentrum im normierten Gradienten und Radius $\arccos\gamma$. Zusammen mit den Skalierungsfaktoren, die unten eingeführt werden, repräsentiert sie aber einen Kegel.

4.3.3 Lemma: Sei $g\colon \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar und in einem Punkt $y \in \mathbb{R}^n$ gelte $\nabla g(y) \neq 0$. Dann gibt es eine Umgebung $U(y)$ und $\lambda > 0$, so dass für alle $x \in U(y)$, $s \in S_\gamma(x)$ und $\mu \in [0, \lambda]$ gilt

$$g(x - \mu s) \leq g(x) - \frac{\mu\gamma}{4}\|\nabla g(y)\|. \tag{4.86}$$

Beweis. Wir definieren zunächst eine Umgebung um y auf der sich die Gradienten nicht zu sehr unterscheiden:

$$U_1(y) = \left\{ x \in \mathbb{R}^n \,\middle|\, \|\nabla g(x) - \nabla g(y)\| \leq \frac{\gamma}{4}\|\nabla g(y)\| \right\}. \tag{4.87}$$

Eine zweite Umgebung ist so gewählt, dass dort der Abstiegskegel in einem größeren Abstiegskegel im Punkt y enthalten ist:

$$U_2(y) = \left\{ x \in \mathbb{R}^n \middle| S_\gamma(x) \subseteq S_{\gamma/2}(y) \right\}. \tag{4.88}$$

Wähle nun $\lambda > 0$, so dass

$$\overline{B_{2\lambda}(y)} \subseteq U_1(y) \cap U_2(y). \tag{4.89}$$

und $U(y) = B_\lambda(y)$. Dann zeigen wir nun die Aussage für alle $x \in U(y)$, $s \in S_\gamma(x)$ und $\mu \in [0, \lambda]$. Nach dem Mittelwertsatz existiert $\vartheta \in (0, 1)$ so dass

$$g(x) - g(x - \mu s) = \mu \nabla g(x - \vartheta\mu s)s. \tag{4.90}$$

Wir formen weiter um:

$$\begin{aligned}
\nabla g(x - \vartheta\mu s)s &= \big(\nabla g(x - \vartheta\mu s) - \nabla g(y)\big)s + \nabla g(y)s && (4.91)\\
&\geq -\frac{\gamma}{4}\|\nabla g(y)\|\|s\| + \nabla g(y)s && (4.92)\\
&\geq -\frac{\gamma}{4}\|\nabla g(y)\| + \frac{\gamma}{2}\|\nabla g(y)\| && (4.93)\\
&\geq \frac{\gamma}{4}\|\nabla g(y)\|. && (4.94)
\end{aligned}$$

□

4.3.4 Definition: Ein **Abstiegsverfahren** für eine stetig differenzierbare Funktion $g\colon \mathbb{R}^n \to \mathbb{R}$ ist eine Iterationsvorschrift aus den folgenden Schritten: gegeben $x^{(k)}$,

1. wähle $\gamma_k > \gamma > 0$ und eine Abstiegsrichtung $s^{(k)} \in S_{\gamma_k}(x^{(k)})$.
2. Wähle eine Schrittweite $\alpha_k > 0$ und setze

$$x^{(k+1)} = x^{(k)} - \alpha_k s^{(k)}, \tag{4.95}$$

so dass die **Reduktionsbedingung**

$$g\big(x^{(k+1)}\big) \leq g\big(x^{(k)}\big) - \frac{\gamma_k \alpha_k}{4} \|\nabla g(x^{(k)})\| \tag{4.96}$$

gilt.

Bemerkung 4.3.5. Lemma Lemma 4.3.3 stellt sicher, dass es in jedem Schritt ein positives α_k gibt, das die Bedingung erfüllt.

4.3.6 Beispiel (Verfahren des steilsten Abstiegs): Sei der Vektor $x^{(k)} \in \mathbb{R}^n$ gegeben, dann wähle $s^{(k)} = \nabla g(x^{(k)})$. Die Schrittweite α_k wird aus der eindimensionalen Minimierungsaufgabe (auch **line search** genannt)

$$\alpha_k = \operatorname*{argmin}_{\alpha > 0} g\big(x^{(k)} - \alpha s^{(k)}\big) \tag{4.97}$$

bestimmt. Danach setze

$$x^{(k+1)} = x^{(k)} - \alpha_k s^{(k)}. \tag{4.98}$$

4.3.7 Satz: Sei $g\colon \mathbb{R}^n \to \mathbb{R}$ und $x^{(0)} \in \mathbb{R}^n$ so gewählt, dass die Menge

$$K = \Big\{ x \in \mathbb{R}^n \;\Big|\; g(x) \leq g\big(x^{(0)}\big) \Big\} \tag{4.99}$$

kompakt und g stetig differenzierbar auf einer Umgebung von K ist. Dann besitzt die Folge $\{x^{(k)}\}$ des Abstiegsverfahrens mindestens einen Häufungspunkt in K. Gilt zusätzlich in der Umgebung eines Häufungspunkts $\gamma_k \geq \gamma > 0$, so existiert α, so dass $\alpha_k \geq \alpha > 0$ gewählt werden kann. In diesem Fall ist der Häufungspunkt ein stationärer Punkt von g.

Beweis. Da die Folge monoton fällt, bleibt sie in K und hat der Kompaktheit wegen mindestens einen Häufungspunkt x^*. Wir benennen nun ebenfalls mit $\{x^{(k)}\}$ ebenfalls eine Teilfolge, die gegen diesen Häufungspunkt konvergiert.

Wir machen die Widerspruchsannahme, dass x^* kein stationäre Punkt von g ist, also

$$\nabla g(x^*) \neq 0. \tag{4.100}$$

Wir bemerken, dass nach Voraussetzung $S_{\gamma_k}(x^*) \subseteq S_\gamma(x^*)$ gilt. Nun gibt es nach Lemma 4.3.3 eine Umgebung $U(x^*)$ und eine Zahl $\lambda > 0$, so dass für alle $\mu \in [0, \lambda]$ gilt:

$$g(x - \mu s) \leq g(x) - \mu \frac{\gamma}{4} \|\nabla g(x^*)\|. \tag{4.101}$$

Daraus folgt, dass zu jedem $\alpha_k \leq \lambda$ die Reduktionsbedingung (4.96) erfüllt ist und dementsprechend die Bedingung $\alpha_k \geq \alpha$ für $\alpha \leq \lambda$ erfüllt werden kann.

Sei nun k_0 gewählt, so dass $x^{(k)} \in U(x^*)$ für alle $k \geq k_0$. Dann gilt nach der Konstruktion von $U(x^*)$ in (4.87), dass

$$\|\nabla g(x^{(k)})\| \geq \|\nabla g(x^*)\| - \|\nabla g(x^{(k)}) - \nabla g(x^*)\| \geq \left(1 - \frac{\gamma}{4}\right) \|\nabla g(x^*)\|. \tag{4.102}$$

Es gilt also

$$g(x^{k+1}) \leq g(x^k) - \frac{3}{4}\alpha \|\nabla g(x^*)\|. \tag{4.103}$$

Daraus folgt im Widerspruch zur Stetigkeit die Konvergenz $g(x^{(k)}) \to -\infty$, Es muss also $\nabla g(x^*) = 0$ gelten, x^* ist also ein stationärer Punkt von g. □

Bemerkung 4.3.8. Der vorherige besteht aus zwei Teilen. Die Existenz eines Häufungspunktes wird unter einer der allgemeinen Bedingung getroffen, dass die Menge K kompakt ist, also eine Abstiegsfolge nicht ins unendliche konvergieren kann. Diese Bedingung ist oft recht leicht nachzuprüfen. Insbesondere steht die Existenz mehrerer Häufungspunkte nicht im Widerspruch zu den Annahmen, so dass das Verfahren auch dann wenigstens einen davon findet.

Die weiteren Bedingungen, die sicherstellen, dass es sich bei einem Häufungspunkt um einen stationären Punkt handelt, sind lokal in einer Umgebung eines solchen gestellt. An diesem Punkt sind die Folgen γ_k und α_k nur sehr abstrakt fixiert. Wir zeigen nun, dass die Folge der α_k im Verfahren des steilsten Abstiegs die Bedingung erfüllt. Als zweite Anwendung von allgemeinen Abstiegsverfahren stellen wir dann das Newton-Verfahren mit Schrittweitensteuerung vor.

4.3.9 Korollar: Beim Verfahren des steilsten Abstiegs sind die Folgen γ_k und α_k so gewählt, das Satz 4.3.7 gilt.

Beweis. Da die Abstiegsrichtungen immer gleich dem (negativen) Gradienten sind, gilt $\gamma_k \equiv 1$. Für die Folge α_k zeigen wir nich die Beschränktheit durch α. Stattdessen bemerken wir mit λ aus Lemma 4.3.3:

$$g(x^{(k+1)}) = \min_{\alpha>0} g\left(x^{(k+1)} - \alpha s^{(k)}\right) \tag{4.104}$$

$$\leq g(x^{(k)} - \lambda s^{(k)}) \tag{4.105}$$

$$\leq g(x^{(k)}) - \frac{\lambda}{4}\|\nabla g(x^*)\|. \tag{4.106}$$

Auch hier schließen wir, dass die Folge divergiert wenn die Punkte konvergieren. □

4.3.10 Lemma: Sei $f\colon \mathbb{R}^n \to \mathbb{R}^n$ stetig differenzierbar und $g(x) = \|f(x)\|_2^2$. Dann sind die Suchrichtungen des Newton-Verfahrens

$$s^{(k)} = \frac{d^{(k)}}{\|d^{(k)}\|_2}, \qquad d^{(k)} = \big(\nabla f(x^{(k)})\big)^{-1} f(x^{(k)}) \tag{4.107}$$

Abstiegsrichtungen für $g(x)$ und es gilt

$$s^{(k)} \in S_\gamma\big(x^{(k)}\big), \qquad \gamma = \frac{1}{\operatorname{cond}_2(\nabla f(x^{(k)}))} \tag{4.108}$$

Beweis. Es gilt (Nachrechnen!)

$$\nabla g(x) = 2f(x)^T \nabla f(x). \tag{4.109}$$

Daher ist

$$\frac{\nabla g(x) s}{\|\nabla g(x)\|_2} = \frac{f(x)^T \nabla f(x) \big(\nabla f(c)\big)^{-1} f(x)}{\|f(x)^T \nabla f(x)\|_2 \big\|\big(\nabla f(c)\big)^{-1} f(x)\big\|_2} \tag{4.110}$$

$$\geq \frac{\|f(x)\|_2^2}{\|f(x)\|_2 \|\nabla f(x)\|_2 \big\|\big(\nabla f(c)\big)^{-1}\|_2 \|f(x)\|_2} \tag{4.111}$$

$$= \frac{1}{\operatorname{cond}_2(\nabla f)}. \tag{4.112}$$

□

4.3.11 Korollar: Das modifizierte Newton-Verfahren

$$x^{(k+1)} = x^{(k)} - \alpha_k \big(\nabla f(x^{(k)})\big)^{-1} f(x^{(k)}) \tag{4.113}$$

ist ein Abstiegsverfahren, wenn α_k so gewählt ist, dass die Reduktionsbedingung gilt.

Bemerkung 4.3.12. Man kann nun zum Beispiel auch das Newton-Verfahren mit line search ausführen, um globale Konvergenzeigenschaften zu erzielen. Es gilt dann zunächst die Existenz von Häufungspunkten. In der Nähe eines solchen gilt aber natürlich, dass line search nicht schlechter konvergiert, als das normale Newton-Verfahren, woraus dann dort wieder die quadratische Konvergenz gefolgert werden kann.

Wir betrachten stattdessen die folgende, einfachere Variante, die mit minimaler Modifikation ein global konvergierendes Verfahren ergibt.

4.3.13 Definition: Das Newton-Verfahren mit **Schrittweitensteuerung** berechnet iterativ $x^{(k+1)} \in \mathbb{R}^n$ aus $x^{(k)} \in \mathbb{R}^n$ in folgenden Schritten

1. Berechne $d^{(k)} = \left(\nabla f(x^{(k)})\right)^{-1} f(x^{(k)})$
2. Berechne die kleinste ganze Zahl j, so dass

$$\left\| f(x^{(k)} - 2^{-j} d^{(k)}) \right\|_2^2 \leq \left\| f(x^{(k)}) \right\|_2^2 - 2^{-j} \frac{1}{4 \operatorname{cond}_2(\nabla f(x^{(k)}))} \left\| f^T\left(x^{(k)}\right) \nabla f\left(x^{(k)}\right) \right\|_2 \quad (4.114)$$

3. Setze $x^{(k+1)} = x^{(k)} - 2^{-j} d^{(k)}$

Bemerkung 4.3.14. Der Algorithmus benötigt viele zusätzliche Berechnungen, wie die von γ_k oder ∇g. Für die praktische Anwendung lässt er sich vereinfachen. Dazu beobachten wir zunächst, dass Bedingung (4.96) dazu dient, eine hinreichende Kontraktion in der Nähe eines Häufungspunkts sicherzustellen. Die Existenz eines solchen kann bereits aus

$$g\left(x^{k+1}\right) < g\left(x^k\right) \quad (4.115)$$

gefolgert werden. Umgekehrt wird, wenn die Funktion f die Bedingungen des Konvergenzsatzes Satz 4.2.4 erfüllt, in der Nähe eines Fixpunktes ohnehin $j = 0$ gelten. Wir ersetzen daher die komplizierte Bedingung durch die wesentlich einfachere: sei j die kleinste nichtnegative ganze Zahl, so dass

$$\left\| f\left(x^k - 2^{-j} d^{(k)}\right) \right\|_2^2 < \left\| f\left(x^k\right) \right\|_2^2. \quad (4.116)$$

Es gibt in der Literatur weitere Heuristiken zur Wahl der Schrittweite im Newton-Verfahren, die man unter dem Stichwort „Globalisierung“ findet. Hier wollen wir uns mit dieser besonders einfachen und gleichzeitig effektiven Variante begnügen.

4.3.15 Algorithmus (Newton-Verfahren mit Schrittweitensteuerung):

```
1  def newton(x, f, Dfinv, tol):
2    r = f(x)
3    r_current = abs(r)
4    while (r_current>tol):
5      d = Dfinv(x,r)
6      x -= d
7      r_old = r_current
8      r = f(x)
9      r_current = abs(r)
10     while (r_current>=r_old):
11       d *= 0.5
12       x += d
13       r = f(x)
14       r_current = abs(r)
15   return x
```

Bemerkung 4.3.16. Dieser Abschnitt gibt Hinweise darauf, wie das Newton-Verfahren modifiziert werden kann und trotzdem Konvergenz erhalten wird. Neben der Schrittweitensteuerung kommen hier insbesondere approximative Berechnungen der Ableitung in Frage. Diese werden dann oft als **Quasi-Newton-Verfahren** bezeichnet, konvergieren in der Regel nur von erster Ordnung, sind aber of viel effizienter als das Newton-Verfahren selbst.

Kapitel 5

Lösung linearer Gleichungssysteme

5.1 Grundlagen

5.1.1 Satz: Für eine Matrix $A \in \mathbb{R}^{n \times n}$ sind folgende Aussagen äquivalent:

1. Das lineare Gleichungssystem $Ax = b$ hat für jede rechte Seite $b \in \mathbb{R}^n$ eine eindeutige Lösung $x \in \mathbb{R}^n$.
2. Alle Eigenwerte von A sind von null verschieden.
3. Der Rang von A ist n.
4. $\det A \neq 0$.

5.1.1 Konditionierung der Lösung

5.1.2 Definition: Die Aufgabe, das lineare Gleichungssystem

$$Ax = b \tag{5.1}$$

zu lösen wandelt die Eingabedaten (A, b) in das Ausgabedatum x um. Die zugehörige gestörte Aufgabe ist

$$(A + \delta A)(x + \delta x) = b + \delta b, \tag{5.2}$$

wobei δA und δb eine Matrix und ein Vektor sind, um die die Eingabedaten gestört sind. δx ist die resultierende Störung der Lösung.
Die Untersuchung der Konditionierung dieser Aufgabe besteht in der Bestimmung einer relativen Konditionszahl κ, so dass

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa \left[\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right]. \tag{5.3}$$

5.1.3 Lemma: Sei $B \in \mathbb{R}^{n \times n}$ mit $\|B\| < 1$. Dann ist $\mathbb{I} - B$ invertierbar und es gilt

$$\|(\mathbb{I} - B)^{-1}\| \leq (1 - \|B\|)^{-1} \tag{5.4}$$

Beweis. Siehe [Rannacher, 2017, Hilfssatz 4.4]. □

5.1.4 Satz: Sei die Matrix $A \in \mathbb{R}^{n \times n}$ invertierbar und

$$\|\delta A\| < \frac{1}{\|A^{-1}\|}. \tag{5.5}$$

Dann ist die gestörte Matrix $A + \delta A$ ebenfalls invertierbar und es gilt die Fehlerabschätzung

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\operatorname{cond}(A)}{1 - \operatorname{cond}(A)^{\|\delta A\|/\|A\|}} \left[\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right]. \tag{5.6}$$

Beweis. Siehe [Rannacher, 2017, Satz 4.1]. □

Bemerkung 5.1.5. Satz 5.1.4 gilt unabhängig von der Wahl der Norm, sobald die Konditionszahl der Matrix konsistent definiert ist. Es ist dabei durchaus möglich, dass die Bedingung (5.5) bezüglich einer Norm verletzt, bezüglich einer

anderen erfüllt ist. Die Invertierbarkeit der gestörten Matrix hängt dabei nicht von der Wahl einer Norm ab. Es genügt also, die Bedingung bezüglich einer geeigneten Norm zu überprüfen.

Bemerkung 5.1.6. Die Bedingung (5.5) wurde benutzt, um die Invertierbarkeit der gestörten Matrix zu sichern. Daraus lässt sich ableiten, dass Nichtsingularität einer Matrix A in der Regel nicht hinreicht, um auch numerisch ein Gleichungssystem lösen zu können. Man benötigt vielmehr, dass eine Matrix nicht nur invertierbar ist, sondern dass die Inverse auch hinreichend beschränkt werden kann. Insbesondere sehen wir am Nenner der Abschätzung, dass bei sehr großer Norm der Inversen schon sehr kleine Störungen der Matrix zu einer erheblichen Vergrößerung des Fehlers führen.

Damit tritt neben die rein qualitative Aussage eine Matrix sei singulär oder invertierbar die quantitative Aussage, dass eine Matrix schlecht invertierbar sei, weil sich Datenfehler sehr stark verstärken.

Zerlegen wir die Konditionszahl in

$$\kappa = \frac{\operatorname{cond}(A)}{1 - \operatorname{cond}(A)^{\|\delta A\|}/_{\|A\|}} = \operatorname{cond}(A)\frac{1}{1 - \|\delta A\|\|A^{-1}\|}, \tag{5.7}$$

so sehen wir, dass auch bei exakter Repräsentation der Matrix die Verstärkung von Fehlern der rechten Seite schon durch die Konditionszahl bestimmt ist. Da die Konditionszahl nie besser als eins ist, gibt es grundsätzlich keine Dämpfung der relativen Fehler.

Bemerkung 5.1.7. Ohne Einschränkung der Allgemeinheit ist das Resultat von Satz 5.1.4 scharf. Dennoch ist es in der Praxis oft zu pessimistisch. Für eine Verbesserung benötigt man jedoch mehr Struktureigenschaften der Matrix, zum Beispiel Symmetrie oder die Untersuchung invarianter Unterräume.

5.2 Die LR-Zerlegung

Notation 5.2.1. Da wir uns in diesem Abschnitt mit der Lösung quadratischer Gleichungssysteme beschäftigen, gelte für alle Matrizen, soweit nicht anders vermerkt, dass ihre Dimension $n \times n$ sei.

5.2.1 Dreiecksmatrizen und Frobeniusmatrizen

5.2.2 Definition: Für eine **untere Dreiecksmatrix** $L \in \mathbb{R}^{n\times n}$ gilt

$$\ell_{ij} = 0, \qquad j > i. \tag{5.8}$$

Für eine **obere Dreiecksmatrix** $R \in \mathbb{R}^{n\times n}$ gilt

$$r_{ij} = 0, \qquad j < i. \tag{5.9}$$

5.2.3 Satz: Die Mengen der invertierbaren oberen und unteren Dreiecksmatrizen bilden jeweils eine multiplikative Gruppe. Die Determinante einer Dreiecksmatrix ist das Produkt ihrer Diagonalelemente.

Beweis. Hausaufgabe □

5.2.4 Korollar: Eine Dreiecksmatrix ist invertierbar genau dann, wenn alle ihre Diagonalelemente von null verschieden sind.

5.2.5 Algorithmus: Die Lösung der linearen Gleichungssysteme

$$Lx = b \qquad Rx = b \tag{5.10}$$

mit einer unteren Dreiecksmatrix L und einer oberen Dreiecksmatrix R lässt sich sukzessive durch Vorwärts- bzw. Rückwärtseinsetzen berechnen.

```
1  def forward_subst(A,b):
2    (m,n) = A.shape
3    x = np.zeros(n)
4    for i in range(0,n):
5      x[i] = b[i]
6      for j in range(0,i):
7        x[i] -= A[i,j]*x[j]
8      x[i] /= A[i,i]
9    return x
```

```
1  def backward_subst(A,b):
2    (m,n) = A.shape
3    x = np.zeros(n)
4    for i in range(n-1,-1,-1):
5      x[i] = b[i]
6      for j in range(i+1,n):
7        x[i] -= A[i,j]*x[j]
8      x[i] /= A[i,i]
9    return x
```

5.2.6 Definition: Eine Matrix der Gestalt

$$G_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & g_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & g_{nk} & & 1 \end{bmatrix} \tag{5.11}$$

mit von null verschiedenen Subdiagonaleinträgen nur in Spalte k heißt **Frobenius-Matrix**.

5.2.7 Lemma: Das Ergebnis des Produktes $G_k A$ einer Frobeniusmatrix mit einer beliebigen Matrix ergibt sich aus A dadurch, dass auf die j-te Zeile das g_{jk}-fache der k-ten Zeile addiert wird.
Für Frobenius-Matrizen gilt

$$G_k^{-1} = 2\mathbb{I} - G_k. \tag{5.12}$$

Sei $k_1 < \cdots < k_m$ eine aufsteigende Folge von Indizes. Dann gilt für Produkte die Darstellung

$$G_{k_1} \cdots G_{k_m} = \sum_{i=1}^{m} G_i - (m-1)\mathbb{I}. \tag{5.13}$$

Insbesondere gilt $$G_1 \cdots G_{n-1} = \begin{bmatrix} 1 & & & \\ g_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ g_{n1} & \cdots & g_{n,n-1} & 1 \end{bmatrix} \tag{5.14}$$

5.2.2 Konstruktion der LR-Zerlegung

5.2.8 Lemma: Bei der Gauß-Elimination lässt sich die Elimination der Subdiagonalelemente der k-ten Spalte als Matrix-Produkt

$$A^{(k+1)} = L_k^{-1} A^{(k)}, \qquad b^{(k+1)} = L_k^{-1} b^{(k)}, \qquad k = 1, \ldots, n-1 \quad (5.15)$$

mit $A^{(1)} = A$, $b^{(1)} = b$ und den Frobenius-Matrizen

$$L_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & \ell_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & \ell_{nk} & & 1 \end{bmatrix}, \qquad \ell_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \quad (5.16)$$

schreiben.

5.2.9 Satz: Nach $n-1$ Schritten der Gauß-Elminiation erhält man das transformierte lineare Gleichungssystem

$$Rx = y, \qquad R = L^{-1}A, \qquad y = L^{-1}b, \qquad L = L_1 \cdots L_{n-1}, \quad (5.17)$$

und die LR-Zerlegung

$$A = LR \quad (5.18)$$

mit einer oberen Dreiecksmatrix R und einer unteren Dreiecksmatrix L, deren Diagonale aus Einsen besteht.

5.2.10 Algorithmus (LR-Zerlegung):

```
1 def lu_decomposition(A):
2   (m,n) = A.shape
3   for k in range (0,n-1):
4     piv = 1./ A[k,k]
5     for i in range (k+1,n):
6       A[i,k] *= piv
7       for j in range (k+1,n):
8         A[i,j] -= A[k,j]*A[i,k]
```

Bemerkung 5.2.11. Im vorigen Algorithmus wird die LR-Zerlegung (engl. LU decomposition) so durchgeführt, dass sie die Matrix A ersetzt. Dadurch wird

kein zusätzlicher Speicher benötigt. Nach ausführen der Funktion hat dann A nicht mehr die Bedeutung einer Matrix, sondern ist ein quadratisches Zahlenfeld, für dessen Einträge a_{ij} gilt

$$a_{ij} = \begin{cases} r_{ij} & i \leq j \\ \ell_{ij} & i > j. \end{cases} \tag{5.19}$$

Von den Diagonaleinträgen von L wissen wir, dass sie den Wert 1 haben, deswegen werden sie nicht gespeichert.

Für dieses spezielle Datenformat gibt es dann auch eine spezialisierte Version der Auflösung der gestaffelten Gleichungssysteme:

5.2.12 Algorithmus (Vorwärts-Rückwärts-Einsetzen):

```
1  def forward_backward(LR,b):
2    (m,n) = LR.shape
3    x = np.zeros(n)
4    for i in range(0,n):
5      x[i] = b[i]
6      for j in range(0,i):
7        x[i] -= LR[i,j]*x[j]
8    for i in range(n-1,-1,-1):
9      for j in range(i+1,n):
10       x[i] -= LR[i,j]*x[j]
11     x[i] /= LR[i,i]
12   return x
```

5.2.13 Lemma: Der Aufwand der LR-Zerlegung einer $n \times n$-Matrix ist

$$\tfrac{1}{3}n^3 + \mathcal{O}(n^2). \tag{5.20}$$

5.2.14 Satz: Ist die Matrix A invertierbar, dann ist im k-ten Schritt der Gauß-Elimination wenigstens eins der Elemente $a_{jk}^{(k)}$ mit $j \geq k$ von null verschieden. für den Fall, dass $a_{kk}^{(k)} = 0$, kann damit die Elimination nach Vertauschen der Zeilen j und k fortgesetzt werden.

5.2.15 Definition: Führt man im k-ten Schritt der Gauß-Elimination eine Zeilenvertauschung durch, so dass

$$|a_{kk}^{(k)}| = \max_{j \geq k} |a_{jk}^{(k)}|, \tag{5.21}$$

so spricht man von Gauß-Elimination mit **Spalten-Pivotierung**. Vertauscht man sogar die verbleibenden Zeilen und spalten, so dass

$$|a_{kk}^{(k)}| = \max_{i,j \geq k} |a_{ij}^{(k)}|, \tag{5.22}$$

handelt es sich um **vollständige Pivotierung**.

5.2.16 Lemma: Führt man die Gauß-Elimination mit Spalten-Pivotierung durch, so gilt für die Matrix L:

$$|\ell_{ij}| \leq 1, \qquad 1 \leq i, j \leq n. \tag{5.23}$$

5.2.17 Lemma: Sei π eine Permutation der Zahlen $1, \ldots, n$, so dass die Zahlen $1, \ldots, k$ unverändert bleiben, und P_π die Matrix der entsprechenden Zeilenvertauschungen. Dann ist

$$P_\pi L_k P_\pi^{-1} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \ell_{\pi(k+1),k} & 1 & & \\ & & \vdots & & \ddots & \\ & & \ell_{\pi(n)k} & & & 1 \end{bmatrix} \tag{5.24}$$

wieder eine Frobenius-Matrix gleicher Struktur wie L_k.

Beweis. Jede Permutation ist das Produkt von Transpositionen. Für solche wird die Aussage in der Hausaufgabe gezeigt. □

5.2.18 Satz: Nach $n-1$ Schritten der Gauß-Elimination mit Spalten-Pivotierung erhält man die Zerlegung

$$PA = LR \tag{5.25}$$

mit einer Permutationsmatrix P und den Dreiecksmatrizen L und R.

Beweis. Siehe [Deuflhard and Hohmann, 2008, Abschnitt 1.3]. □

5.2.3 Fehleranalyse

Ohne Beweis geben wir die folgenden Resultate zur Rundungsfehleranalyse der Lösung linearer Gleichungssysteme mit der LR-Zerlegung an. Teile der Beweise finden sich in [Stoer, 1983].

5.2.19 Notation: Zu einer Matrix A sei $|A|$ die Matrix der Absolutbeträge, also

$$|A| = \begin{pmatrix} |a_{11}| & \cdots & |a_{1n}| \\ \vdots & & \vdots \\ |a_{nn}| & \cdots & |a_{nn}| \end{pmatrix}. \tag{5.26}$$

5.2.20 Lemma: Die Berechnung der LR-Zerlegung einer Matrix A in Fließkommaarithmetik resultiert in einer Zerlegung $\widehat{L}\widehat{R} = A + \delta A$ und es gilt die Abschätzung

$$|\delta A| \leq 2\alpha_{\max} \frac{\texttt{eps}}{1 - \texttt{eps}} \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 2 & 2 & \cdots & 2 & 2 \\ 1 & 2 & 3 & \cdots & 3 & 3 \\ \vdots & \vdots & \vdots & \cdots & n-1 & n-1 \end{pmatrix} \tag{5.27}$$

Hierbei ist $\alpha_{\max}$ der betragsmäßig größte Eintrag aller Matrizen $A^{(k)}$, die im Verfahren auftreten,

$$\alpha_{\max} = \max_{1 \leq k,i,j \leq n} |a_{ij}^{(k)}|. \tag{5.28}$$

5.2.21 Lemma: Die Realisierung des Vorwärtseinsetzens für das Gleichungssystem $Lx = b$ in Fließkommaarithmetik berechnet die Lösung $\widehat{x}$ des gestörten Systems $\widehat{L}\widehat{x} = b$ mit

$$|L - \widehat{L}| \leq \frac{\texttt{eps}}{1 - n\texttt{eps}} \left(|L| \begin{pmatrix} 1 & & & \\ & 2 & & \\ & & \ddots & \\ & & & n \end{pmatrix} . - \mathbb{I} \right) \tag{5.29}$$

5.2.22 Satz (Wilkinson): Das Gaußsche Eliminationsverfahren mit Spaltenpivotierung für das Gleichungssystem $Ax = b$ berechnet die Lösung $\widehat{x}$ des Systems $\widehat{A}\widehat{x} = b$ für eine Matrix $\widehat{A}$ mit

$$\frac{\|\delta A\|_\infty}{\|A\|_\infty} \lessapprox 2n^3 \frac{\alpha_{\max}}{\max|a_{ij}|}\texttt{eps}, \tag{5.30}$$

wobei

$$\alpha_{\max} = \max_{1\le k,i,j\le n} |a_{ij}^{(k)}|. \tag{5.31}$$

Bemerkung 5.2.23. Im Allgemeinen kann die Konstante $\alpha_{\max}$ durch den Wert $2^{n-1}\max a_{ij}$ abgeschätzt werden. Dies führt bei größeren Matrizen sehr schnell zu inakzeptablen Fehlern. Es gibt aber einige Aussagen über die LR-Zerlegung von Matrizen mit spezielleren Strukturen. So gilt

1. Ist die Matrix A invertierbar und schwach diagonaldominant, das heißt,
$$a_{ii} \ge \sum_{j\ne i} |a_{ij}|, \qquad i = 1, \dots, n, \tag{5.32}$$
so kann die LR-Zerlegung ohne Pivotierung durchgeführt werden.
2. Ist die Matrix A positiv definit, so kann die LR-Zerlegung ohne Pivotierung durchgeführrt werden und alle Diagonalelemente $a_{kk}^{(k)}$ sind positiv (siehe [Rannacher, 2017, Satz 4.7]).
3. Für symmetrisch positiv definite Matrizen führt man die Gauß-Elimination in der Variante des Choleski-Verfahrens durch, das eine LL^T-Zerlegung mit dem halben Aufwand der LR-Zerlegung produziert.
4. Hat die Matrix eine Struktur, bei der sich alle von null verschiedenen Einträge um die Diagonale konzentrieren, man spricht von Band- und Skyline-Matrizen, so kann man bei der LR-Zerlegung diese Struktur ausnutzen und erheblich an Operationen sparen.

Bemerkung 5.2.24. Hat man über die LR-Zerlegung eine Näherungslösung $\widehat{x} = x + \delta x$ von $Ax = b$ berechnet, so kann das Residuum

$$r(\widehat{x}) = b - A\widehat{x} \tag{5.33}$$

berechnet werden und gibt Aufschluss über den Fehler. Insbesondere gilt

$$\delta x = A^{-1} r(\widehat{x}), \tag{5.34}$$

was aber nicht berechenbar ist, da wir die Inverse von A nicht exakt kennen. Wir können aber δx mit derselben relativen Genauigkeit wie x durch den Vektor $\widehat{\delta x}$ approximieren, indem wir mit der bereits berechneten LR-Zerlegung

$$\widehat{L}\widehat{R}\widehat{\delta x} = r(\widehat{x}) \tag{5.35}$$

lösen. Dann ist aber $\widehat{x} + \widehat{\delta x}$ eine Approximation an x, deren absoluter Fehler dem von $\widehat{\delta x}$ entspricht, ist also genauer als $\widehat{x}$. Offenbar lässt sich dieser Prozess wiederholen

5.2.25 Definition: Sei $\widehat{L}\widehat{R}$ die fehlerbehaftete LR-Zerlegung der Matrix A und $x^{(0)} = \widehat{R}^{-1}\widehat{L}^{-1}b$. Dann besteht die **Nachiteration** (engl. **iterative refinement**) aus der Iterationsvorschrift

$$x^{(k+1)} = x^{(k)} + \widehat{R}^{-1}\widehat{L}^{-1}\big(b - Ax^{(k)}\big). \tag{5.36}$$

5.2.26 Aufgabe: Nutzen Sie Satz 5.2.22 und die Konditionierung der Lösung eines linearen Gleichungssystems um eine Bedingung an die Genauigkeit der Zerlegung zu stellen, unter der Sie die Konvergenz der Nachiteration durch den Banachschen Fixpunktsatz beweisen können.

Bemerkung 5.2.27. Die Funktionsbibliothek LAPACK zur linearen Algebra wird heute als Standardimplementation für viele der hier diskutierten Algorithmen benutzt, zum Beispiel die LR-Zerlegung. Sie enthält viele Optimierungen und benutzt auch automatisch Spaltenpivotierung.

5.3 Die QR-Zerlegung

5.3.1. Betrachten wir die Konstruktion der LR-Zerlegung als Folge von Operationen auf Matrizen, so ergibt sich das Bild

$$\begin{aligned} A^{(1)} &\mapsto L_1 A^{(2)} \\ A^{(2)} &\mapsto L_2 A^{(3)} \\ A^{(n-1)} &\mapsto L_{n-1} R. \end{aligned} \tag{5.37}$$

Es findet also in jedem Schritt eine Umformung der Matrix des aktuellen Schritts mit einer Frobeniusmatrix statt.

In der Fehlerabschätzung steht der Faktor

$$\varrho = \frac{\alpha_{\max}}{\max|a_{ij}|} = \frac{\max|a_{ij}^{(k)}|}{\max|a_{ij}|}, \tag{5.38}$$

wobei das Wachstum der Zähler von den Eigenschaften der Frobeniusmatrizen abhängt. Aus [Deuflhard and Hohmann, 2008] zitieren wir dazu folgende Werte abhängig von der Struktur der Matrix

Matrix	ϱ
invertierbar	2^{n-1}
diagonaldominant	2
s.p.d.	1

Hierbei wird für allgemein invertierbare Matrizen Spaltenpivotierung angewandt, für die anderen nicht. Für allgemeine Matrizen kann dieser Faktor also sehr schnell anwachsen und der Algorithmus instabil werden.

Dieser Abschnitt beschäftigt sich nun mit alternativen Transformationen, die immer zu einer stabilen Zerlegung führen. Gesucht werden dazu Matrizen so dass

$$\|A^{(n-1)}\| = \cdots = \|A^{(k)}\| = \cdots = \|A^{(1)}\| \tag{5.39}$$

für eine geeignete Norm gilt.

5.3.1 Orthogonale Matrizen

5.3.2 Definition: Eine **orthogonale Matrix** ist eine quadratische Matrix, deren Spaltenvektoren bzw. deren Zeilenvektoren eine Orthonormalbasis des $\mathbb{R}^n$ bilden.

5.3.3 Satz: Für eine orthogonale Matrix Q gilt

$$Q^{-1} = Q^T. \tag{5.40}$$

Umgekehrt folgt aus dieser Beziehung die Orthogonalität der Zeilenvektoren und Spaltenvektoren.

Beweis. Nehmen wir an, die Spaltenvektoren $q^{(1)}, \ldots, q^{(n)}$ von Q seinen eine ONB. Dann gilt für die Matrix $A = Q^T Q$:

$$a_{ij} = \sum_{k=1}^{n} q_{ki} q_{kj} = \sum_{k=1}^{n} q_k^{(i)} q_k^{(j)} = \left(q^{(i)}\right)^T q^{(j)} = \delta_{ij}. \tag{5.41}$$

Daher gilt $Q^T Q = I$. Multiplizieren wir diese Gleichung von rechts mit Q^{-1}, so erhalten wir (5.40). Setzen wir umgekehrt $Q^T Q = I$, so ergibt obige Rechnung die Orthogonalität der Spaltenvektoren.

Aus $Q^T = Q^{-1}$ folgt aber durch Transponieren

$$Q = Q^{-T}, \tag{5.42}$$

wobei Q^{-T} die Inverse von Q^T ist. Multiplizieren wir die letzte Gleichung von rechts mit Q^T, so erhalten wir $QQ^T = I$, was äquivalent zur Orthonormalität der Zeilenvektoren ist.

Wir hätten diesen Beweis auch mit den Zeilenvektoren beginnen können und $QQ^T = I$ folgern. Der Rest verläuft dann analog. □

Beispiel 5.3.4. Die Rotationsmatrix

$$Q = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix} \tag{5.43}$$

ist orthogonal. Dasselbe gilt für die Reflexionsmatrix an einem Vektor $w \in \mathbb{R}^n$,

$$Q = I - 2\frac{ww^T}{w^T w} \tag{5.44}$$

5.3.5 Lemma: Für jede orthogonale Matrix $Q \in \mathbb{R}^{n\times n}$, jeden Vektor $x \in \mathbb{R}^n$ und jede beliebige Matrix $A \in \mathbb{R}^{n\times n}$ gilt

$$\|Qx\|_2 = \|x\|_2, \qquad \|QA\|_2 = \|A\|_2. \tag{5.45}$$

5.3.2 Existenz und Konstruktion

5.3.6 Definition: Bei der **QR-Zerlegung** wird die Matrix $A \in \mathbb{R}^{n\times n}$ in das Produkt

$$A = QR \tag{5.46}$$

aus einer orthogonalen Matrix Q und einer oberen Dreiecksmatrix R zerlegt.

5.3.7 Lemma: Seien $q^{(1)}, \ldots, q^{(n)}$ die Spaltenvektoren von Q und $a^{(1)}, \ldots, a^{(n)}$ die Spaltenvektoren von A. Dann gilt

$$a^{(k)} = \sum_{i=1}^{k} r_{ik} q^{(i)}. \tag{5.47}$$

Gilt $r_{ii} \neq 0$ für $i = 1, \ldots, k$ so ist die Beziehung eindeutig umkehrbar. Insbesondere besteht dann die Folge der Spaltenvektoren von Q aus den orthogonalisierten Spaltenvektoren von A mit

$$\operatorname{span}\{q^{(1)}, \ldots, q^{(k)}\} = \operatorname{span}\{a^{(1)}, \ldots, a^{(k)}\} \qquad k = 1, \ldots, n. \tag{5.48}$$

5.3.8 Satz: Zu jeder invertierbaren quadratischen Matrix $A \in \mathbb{R}^{n\times n}$ gibt es eine QR-Zerlegung. Unter der Zusatzbedingung $r_{ii} > 0$ ist diese eindeutig.

Beweis. Nach dem vorigen Lemma können die Spaltenvektoren $q^{(i)}$ der Matrix Q mit dem Gram-Schmidt-Verfahren aus den Spaltenvektoren $a^{(i)}$ von A gewonnen werden. Da die Spalten von A linear unabhängig sind, bricht das Verfahren nicht ab.

Daraus folgt insbesondere Gleichung (5.47) mit $r_{ij} = \left\langle a^{(j)}, q^{(i)} \right\rangle$.

Zur Eindeutigkeit nehmen wir an, es gelte $A = Q_1R_1 = Q_2R_2$. Es gilt dann für die Hilfsmatrix $P = Q_2^TQ_1$

$$P = Q_2^{-1}Q_1 = R_2A^{-1}AR_1^{-1} \tag{5.49}$$

und ebenso $P^T = R_1R_2^{-1}$. Beide Produkte auf der rechten Seite sind opere Dreiecksmatrizen, woraus folgt, dass P diagonal sein muss. Wegen der Orthogonalität gilt $|p_{ii}| = 1$ für $i = 1, \ldots, n$. Schließlich benutzen wir

$$PR_1 = Q_2^TQ_1R_1 = Q_2^TA = Q_2^TQ_2R_2, \tag{5.50}$$

woraus folgt $p_{ii}r_{1;ii} = r_{2;ii}$. Wegen der Positivität der Diagonalelemente von R_1 und R_2 ist damit $p_{ii} = 1$ und $P = \mathbb{I}$. Aus $R_2R_1^{-1} = \mathbb{I}$ folgt dann $R_1 = R_2$ und

$$Q_1 = AR_1^{-1} = AR_2^{-1} = Q_2. \tag{5.51}$$

□

5.3.9 Notation: Zu einem Vektor $w \in \mathbb{R}^n$ beschreibt ww^T das **dyadische Produkt**, eine symmetrische $n \times n$-Matrix mit den Einträgen

$$\begin{pmatrix} w_1w_1 & \cdots & w_1w_n \\ \vdots & & \vdots \\ w_nw_1 & \cdots & w_nw_n \end{pmatrix}. \tag{5.52}$$

Es ist damit in gewisser Weise das „Gegenstück“ zum euklidischen Skalarprodukt w^Tw, das ein Skalar und damit eine 1×1-Matrix ist.

5.3.10 Lemma (Reflexionsmatrix): Ist ein Vektor $w \in \mathbb{R}^n$ gegeben, so beschreibt die Matrix

$$Q_w = \mathbb{I} - 2\frac{ww^T}{w^T w} \tag{5.53}$$

die Abbildung, die einen Vektor y durch $Q_w y$ an der Hyperebene senkrecht zu w spiegelt. Diese **Reflexionsmatrix** ist symmetrisch und orthogonal.

Beweis. Zu einem Vektor x ist

$$x_w = \frac{w}{\|w\|_2^2}\langle w, x\rangle \tag{5.54}$$

die orthogonale Projektion des Vektors auf den von w aufgespannten Unterraum. Subtrahiert man x_w von x, so erhält man wie im Gram-Schmidt-Verfahren einen Vektor orthogonal zu w. Subtrahiert man ein weiteres Mal, so erhält man die Spiegelung an diesme Vektor. Ändern wir nun die Notation, so ist

$$\|w\|_2^2 = w^T w \qquad \text{und} \qquad \langle w, x\rangle = w^T w. \tag{5.55}$$

Die Symmetrie der Reflexionsmatrix folgt aus der Symmetrie des dyadischen Produkts. Für die Orthogonalität setzen wir $u = {}^{w}/_{\|w\|_2}$ und berechnen

$$\begin{aligned} P^T P &= P^2 && (5.56)\\ &= (\mathbb{I} - 2uu^T)(\mathbb{I} - 2uu^T) && (5.57)\\ &= \mathbb{I} - 2uu^T - 2uu^T + 4uu^T uu^T && (5.58)\\ &= \mathbb{I} - 4uu^T + 4uu^T = \mathbb{I}. && (5.59) \end{aligned}$$

□

5.3.11 Lemma (Householder-Reflexion): Sei $y \in \mathbb{R}^n$ gegeben. Dann gibt es zwei Vektoren $w^+, w^- \in \mathbb{R}^n$ und eine Zahl $\alpha \in \mathbb{R}$, so dass

$$Q_{w^\pm} y = \pm\alpha e_1 = \begin{pmatrix} \pm\alpha \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad \text{nämlich} \qquad w^\pm = \begin{pmatrix} y_1 \pm \|y\| \\ y_2 \\ \vdots \\ y_n \end{pmatrix}. \tag{5.60}$$

Wir nennen diese Reflexionsmatrizen **Householder-Reflexion**.

Beweis. Orthogonale Abbildungen sind normerhaltend. Daher muss gelten

$$|\alpha| = \|y\|_2 \tag{5.61}$$

oder $\alpha = \pm\|y\|_2$. Ferner gilt wegen (5.60) für einen geeigneten Reflexionsvektor w:

$$\alpha e_1 = Q_w y = y - 2\frac{ww^T y}{w^T w} = y - 2\left(\frac{w^T y}{w^T w}\right) y. \tag{5.62}$$

Daher ist w ein Vielfaches von $y - \alpha e_1$. Da durch die Norm von w geteilt wird, isd die Länge beliebig und wir wählen

$$w^{\pm} = y \pm \|y\|_2 e_1. \tag{5.63}$$

□

Bemerkung 5.3.12. Zur Vermeidung von Auslöschung in der ersten Stelle von w verwendet man in der QR-Zerlegung den Faktor α so dass

$$\operatorname{sign}(\alpha) = -\operatorname{sign}(y_1) \tag{5.64}$$

und damit

$$w = y + \operatorname{sign}(y_1)\|y\|_2 e_1 = \begin{pmatrix} y_1 + \operatorname{sign}(y_1)\|y\|_2 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}. \tag{5.65}$$

5.3.13 Algorithmus (Householder-Reflexion):

```
1  def householder_compute (x):
2    s = np.dot(x,x)
3    alpha = -np.sign(x[0])*np.sqrt(s)
4    x[0] -= alpha
5    x /= np.sqrt(np.dot(x,x))
6    return alpha
7
8  def householder_apply (u,x):
9    n = x.shape
10   s = np.dot(u,x)
11   for i in range(0,n[0]):
12     x[i] -= 2*s*u[i]
```

5.3.14 Definition (QR-Zerlegung mit Householder-Reflexion): Das Verfahren berechnet eine Folge von Matrizen $A^{(k)}$ aus der Matrix $A^0 = A$ nach der Vorschrift

$$A^{(k+1)} = Q_k A^{(k)}, \qquad Q_k = \begin{bmatrix} \mathbb{I}_k & \\ & \mathbb{I}_{n-k} - 2uu^T \end{bmatrix}, \tag{5.66}$$

wobei u der normierte Vektor der Householder-Reflexion zu den letzten $n-k$ Komponenten der k-ten Spalte von A ist.

5.3.15 Lemma: Es gilt, dass der obere linke Block der Dimension k der Matrix $A^{(k)}$ obere Dreicksgestalt hat. Insbesondere ist

$$A^{(n-2)} = R. \tag{5.67}$$

Für die Matrix Q der QR-Zerlegung gilt

$$Q = Q_0 \cdots Q_{n-2}. \tag{5.68}$$

5.3.16 Lemma: Der Aufwand der QR-Zerlegung mit Householder-Reflexionen liegt mit

$$\frac{2}{3}n^3 + \mathcal{O}(n^2) \tag{5.69}$$

Operationen doppelt so hoch wie bei der LR-Zerlegung.
Das Ergebnis der QR-Zerlegung benötigt, wenn die Vektoren u der Reflexionsmatrix im unteren Dreieck der Matrix A gespeichert werden, einen weiteren Vektor zur Speicherung der der Diagonale von R.

5.3.17 Algorithmus (QR-Zerlegung mit Householder-Reflexion):

```
1  def householder_qr (A):
2    (m,n) = A.shape
3    D = np.zeros(n)
4    for k in range(0,n-1):
5      D[k] = householder_compute(A[k:m,k])
6      for i in range (k+1,n):
7        householder_apply(A[k:m,k],A[k:m,i])
8    return D
```

Achtung! Ungetestet!

5.3.18 Lemma: Liegt die Matrix $A = QR$ als QR-Zerlegung vor, so berechnet sich die Lösung des linearen Gleichungssystems $Ax = b$ in den Schritten

$$y = Q_{n-2} \cdots Q_0 b \tag{5.70}$$
$$Rx = y. \tag{5.71}$$

5.3.19 Algorithmus (Lösung mit Householder-QR-Zerlegung):

```
def householder_solve (QR,D,b)
  (m,n) = QR.shape
  for k in range (0,n-1):
    householder_apply(QR[k:m,k],b[k:m])
  x = np.zeros(n)
  for k in range(n-1,-1,-1):
    x[k] = b[k]
    for j in range(k+1,n):
      x[k] -= A[k,j]*x[j]
    x[k] /= D[k]
  return x
```

Achtung! Ungetestet!

5.3.20 Definition: Die **Givens-Rotation** Ω_{jk} mit $j < k$ zum Winkel ϑ bildet ab

$$\Omega_{jk} \colon \mathbb{R}^n \to \mathbb{R}^n, \quad x \mapsto y \qquad y = \begin{pmatrix} \mathbb{I} & & & & \\ & c & \cdots & s & \\ & \vdots & \mathbb{I} & \vdots & \\ & -s & \cdots & c & \\ & & & & \mathbb{I} \end{pmatrix} x \tag{5.72}$$

mit

$$y_i = \begin{cases} cx_j + sx_k & i = j \\ -sx_j + cx_k & i = k \\ x_i & \text{sonst} \end{cases} \tag{5.73}$$

mit $c = \cos\vartheta$ und $s = \sin\vartheta$.

Bemerkung 5.3.21. Die Einträge c und s in der Matrixdarstellung (5.72) befinden sich in den Zeilen j und k. Einheitsmatrizen $\mathbb{I}$ sollen dann jeweils die passende Dimension haben.

Multipliziert man Ω_{jk} von links an die Matrix A, so ändern sich nur die Zeilen j und k, es ist also ebenfalls eine Zeilenoperation wie bei der Gauß-Elimination.

Die Wirkung von Ω_{jk} auf einen Vektor ist eine Rotation des Vektors in der Ebene, die von den Einheitsvektoren e_j und e_k aufgespannt wird. Wenn wir das im Kopf behalten, genügt es, sie als 2×2-Matrix aufzufassen.

Die Bezeichnung Ω_{jk} scheint unzulänglich, da der Winkel ϑ fehlt. Dies liegt daran, dass wir im folgenden die Rotation nur dazu verwenden, $y_k = 0$ zu erzielen.

5.3.22 Lemma: Mit Hilfe der Givens-Rotation kann aus dem Vektor $(x_j, x_k)^T$ die zweite Komponente eliminiert werden, indem man wählt

$$r = \sqrt{x_j^2 + x_k^2}, \qquad c = \frac{x_j}{r}, \quad s = \frac{x_k}{r}. \tag{5.74}$$

Man erhält dann

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} x_j \\ x_k \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}. \tag{5.75}$$

Bemerkung 5.3.23. Die Berechnung von r im vorherigen Lemma kann bei der Implementation zu numerischem Überlauf führen, wenn eins der Argumente sehr groß ist. Es gibt in der Literatur einige Veröffentlichungen zu diesem Thema. Wir können bei der Implementation die Funktion hypot benutzen, die für die Längen der beiden Katheten die Länge der Hypothenuse eines rechtwinkligen Dreiecks zurückgibt.

Bemerkung 5.3.24. Setzt man die Givens-Rotation zur Reduktion der Spalte k einer Matrix A ein, so beginnt man am unteren Ende und benutzt $\Omega_{n-1,n}$ zur Elimination von a_{nk} und arbeitet sich dann nach oben.

5.3.25 Lemma: Die QR-Zerlegung mit Givens-Rotation berechnet mit $A^{(1)} = A$ eine Folge von Matrizen

$$A^{(k+1)} = Q_k^T A^{(k)}, \qquad k = 1, \ldots, n-1, \tag{5.76}$$

so dass $A^{(n)} = R$ obere Dreiecksgestalt hat. Es gilt

$$Q_k^T = \Omega_{k,k+1}\Omega_{k+1,k+2} \cdots \Omega_{n-1,n-2}\Omega_{n-1,n} \tag{5.77}$$

mit den Givens-Rotationen $\Omega_{j-1,j}$, die jeweils das aktuelle Element a_{jk} zu null setzen.

5.3.26 Algorithmus (QR-Zerlegung mit Givens-Rotation):

5.3.27 Aufgabe: Eine obere **Hessenbergmatrix** ist eine Matrix mit nur einer unteren Nebendiagonalen, also

$$a_{ij} = 0 \qquad \forall i > j + 1. \tag{5.78}$$

1. Zeigen Sie, dass die QR-Zerlegung einer solchen Matrix mit nur $n - 1$ mit Givens-Rotationen möglich ist.
2. Überlegen Sie, wie Sie die gesamte Information für Q und R auf dem Speicherplatz der Matrix A unterbringen.
3. Schätzen Sie den Aufwand.

5.3.28 Lemma: Der Aufwand der QR-Zerlegung mit Givens-Rotation beträgt

$$\frac{4}{3}n^3 + \mathcal{O}(n^2) \tag{5.79}$$

Multiplikationen und Additionen plus $n^2/2$ Quadratwurzeln.
Für jede Rotation sind zwei Zahlen zu speichern, so dass die Givens-Rotation zum Speicher der Originalmatrix ein weiteres unteres Dreieck benötigt.
Da bei einer Matrix in Hessenbergform nur eine Rotation pro Spalte nötig ist, speichert man die Hypotenuse als Diagonalelement von R und den Kosinus in der Subdiagonale. Der Sinus lässt sich dann berechnen.

5.4 Lineare Ausgleichsrechnung

5.4.1. Die Methode der kleinsten Fehlerquadrate führt auf die Minimierungsaufgabe

$$\|Ax - b\|_2 = \min. \tag{5.80}$$

5.4.2 Satz: Sei $A \in \mathbb{R}^{m\times n}$ mit $m \geq n$ und $b \in \mathbb{R}^m$. Dann ist $x \in \mathbb{R}^n$ genau dann eine Lösung des linearen Ausgleichsproblems

$$\|Ax - b\|_2 = \min, \tag{5.81}$$

wenn x Lösung der **Normalengleichungen**

$$A^T Ax = A^T b \tag{5.82}$$

ist. Insbesondere ist die Minimierungsaufgabe eindeutig lösbar, wenn A vollen Rang hat.

Bemerkung 5.4.3. Wir können die Normalengleichungen lösen, indem wir die symmetrische Matrix $C = A^T A \in \mathbb{R}^{n\times n}$ berechnen und dann eines der Verfahren der vorigen Abschnitte auf diese Matrix anwenden.

Das Lemma nach der nächsten Definition legt nahe, dass das keine gute Idee ist, da sich die Konditionszahl durch das Matrixprodukt quadriert und damit die Lösungsgenauigkeit leidet.

5.4.4 Definition: Die Konditionszahl einer rechteckigen Matrix maximalen Rangs bezüglich der Operatornorm zur Vektornorm $\|\cdot\|$ ist

$$\mathrm{cond}(A) = \frac{\sup\limits_{\|x\|=1} \|Ax\|}{\inf\limits_{\|x\|=1} \|Ax\|}. \tag{5.83}$$

Die Definition ist konsistent zur Definition für invertierbare, quadratische Matrizen.

5.4.5 Lemma: Für eine Matrix $A \in \mathbb{R}^{m\times n}$ maximalen Rangs mit $m \geq n$ gilt

$$\mathrm{cond}_2(A^T A) = \mathrm{cond}_2(A)^2. \tag{5.84}$$

Beweis.

$$\begin{aligned}\operatorname{cond}_2(A)^2 &= \frac{\sup\|Ax\|_2^2}{\inf\|Ax\|_2^2} && (5.85)\\ &= \frac{\sup(x^TA^TAx)}{\inf(x^TA^TAx)} && (5.86)\\ &= \frac{\lambda_{\max}(A^TA)}{\lambda_{\in}(A^TA)} && (5.87)\\ &= \operatorname{cond}_2(A^TA). && (5.88)\end{aligned}$$

□

5.4.6 Lemma: Zu jeder Matrix $A \in \mathbb{R}^{m\times n}$ maximalen Rangs mit $m \geq n$ gibt es eine QR-Zerlegung

$$A = QR \tag{5.89}$$

mit einer oberen Dreiecksmatrix $R \in \mathbb{R}^{n\times n}$ und einer Matrix $Q \in \mathbb{R}^{m\times n}$, deren Spalten ein Orthonormalsystem bilden. Unter der Zusatzbedingung $r_{ii} > 0$ ist diese Zerlegung eindeutig.

5.4.7 Satz: Sei $QR = A$ eine QR-Zerlegung. Dann kann die Lösung der Normalengleichungen berechnet werden durch die Lösung des Systems

$$Rx = Q^Tb. \tag{5.90}$$

Beweis. Einsetzen der QR-Zerlegung in die Normalengleichungen ergibt

$$R^TQ^TQRx = R^TRx = R^TQ^Tb. \tag{5.91}$$

Da R^T invertierbar ist, können wir die Inverse von links anwenden und erhalten das Resultat. □

Bemerkung 5.4.8. Bei der linearen Ausgleichsrechnung (engl.: **least-squares**) erweitern wir den Lösungsbegriff für lineare Gleichungssysteme auf überbestimmte Systeme, indem wir die Bedingung „Residuum gleich null“ durch die Bedingung „Residuum minimal“ ersetzen.

Zur definition von „minimal“ haben wir dabei ganz selbstverständlich die euklidische Norm benutzt. Das ist aber willkürlich. Zunächst können wir die Definition und alle weiteren Argumente auf allgemeine Skalarprodukte und ihre zugehörigen Normen ersetzen. Dazu müssen wir die transponierte Matrix A^T durch die im Skalarprodukt adjungierte Matrix A^* definiert durch

$$\langle A^*x, y\rangle = \langle x, Ay\rangle \qquad \forall x, y \in \mathbb{R}^n \tag{5.92}$$

ersetzen. Ebenso müssen dann natürlich die Spalten von Q orthonormal bezüglich dieses Skalarprodukts sein.

Normen, die nicht durch Skalarprodukte definiert sind, führen auf nichtlineare Bestimmungsgleichungen. Solche Aufgaben behandelt die Optimierung.

Ein wichtiger Schluss aus dieser Bemerkung ist, dass wir zwar den Begriff der Lösung erweitert haben, dabei aber nicht nur die Matrix und die rechte Seite die Lösung bestimmen, sondern zusätzlich die gewählte Norm die Lösung beeinflusst.

5.5 Die Singulärwertzerlegung

5.5.1. Nach den Überlegungen zur Lösung von überbestimmten Gleichungssystemen stellt sich die Frage, ob wir das Konzept auch auf Systeme ohne Eindeutige Lösung erweitern können. Zu diesem Zweck müssen wir im Lösungsraum Elemente auszeichnen, zum Beispiel durch Minimierung ihrer Norm. Als Vorbereitung für das kommende erinnern wir uns an zwei Tatsachen aus der linearen Algebra:

1. Betrachten wir die Matrix $A \in \mathbb{R}^{m\times n}$ als lineare Abbildung $A\colon \mathbb{R}^n \to \mathbb{R}^m$, so ist die eingeschränkte Abbildung

$$\hat{A}\colon (\ker(A))^\perp \to \operatorname{im}(A) \tag{5.93}$$

   ein Isomorphismus, kann also nach Wahl einer Basis in beiden Räumen als invertierbare Matrix geschrieben werden.

2. Es gelten die Beziehungen

$$\begin{aligned} (\ker(A))^\perp &= \operatorname{im}\left(A^T\right), \\ (\operatorname{im}(A))^\perp &= \ker\left(A^T\right). \end{aligned} \tag{5.94}$$

Unser Ziel wird es also sein, Projektionen auf die Bildräume von A und A^T zu bestimmen und auf diesen dann A^{-1} zu berechnen. Ein mächtiges Werkzeug dafür ist die Singulärwertzerlegung.

5.5.2 Notation: Mit $\mathrm{diag}(a_1, a_2, \ldots, a_r) \in \mathbb{R}^{m\times n}$ sei allgemein die $m \times n$-Matrix A bezeichnet, deren erste r Diagonalelemente die Werte a_i annehmen. Alle anderen Einträge sind null. Sie hat die Darstellung

$$A = \mathrm{diag}(a_1, a_2, \ldots, a_r) = \left(\begin{array}{ccc|c} a_1 & & \mathbf{0} & \\ & \ddots & & \mathbf{0} \\ \mathbf{0} & & a_r & \\ \hline & \mathbf{0} & & \mathbf{0} \end{array}\right), \tag{5.95}$$

wobei untere und rechte Nullblöcke leer sein dürfen.

5.5.3 Definition: Die **Singulärwertzerlegung** (engl.: **singular value decomposition**, **SVD**) einer Matrix $A \in \mathbb{R}^{m\times n}$ hat die Form

$$A = U\Sigma V^T \tag{5.96}$$

mit orthogonalen Matrizen $U \in \mathbb{R}^{m\times m}$ und $V \in \mathbb{R}^{n\times n}$. Die Matrix $\Sigma \in \mathbb{R}^{m\times n}$ ist

$$\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r) \tag{5.97}$$

mit positiven, reellen Einträgen $\sigma_1, \ldots, \sigma_r$ und $r \leq \min\{m, n\}$, den **Singulärwerten**. Die Singulärwerte seien der Größe nach fallend sortiert.

5.5.4 Satz: Jede Matrix $A \in \mathbb{R}^{m\times n}$ besitzt eine Singulärwertzerlegung.

Beweis. Siehe auch [Rannacher, 2017, Satz 4.11]. Der Beweis läuft induktiv über die Spalten von U und V. Wir bemerken zunächst, dass es wegen der Stetigkeit der Norm einen Vektor $x \in \mathbb{R}^n$ mit $\|x\|_2 = 1$ gibt, so dass

$$\|Ax\|_2 = \|A\|_2\|x\|_2. \tag{5.98}$$

Wir definieren $\sigma_1 = \|A\|$ und es sei $y \in \mathbb{R}^m$ so dass $\sigma_1 y = Ax$. Wir ergänzen x und y jeweils zu Orthogonalbasen und nennen die Matrizen der Basisvektoren $U^{(1)}$ und $V^{(1)}$. Es gilt dann

$$\left(U^{(1)}\right)^T A^{(1)} V^{(1)} = \begin{pmatrix} \sigma & w^T \\ 0 & B \end{pmatrix} \tag{5.99}$$

mit einem Vektor $w \in \mathbb{R}^{n-1}$ und einer Matrix $b \in \mathbb{R}^{m-1\times n-1}$. Da U und V orthogonal sind, gilt

$$\|A^{(1)}\| = \|A\| = \sigma. \tag{5.100}$$

Multipliziert man die Matrix $A^{(1)}$ mit dem Vektor $z = (\sigma, w)^T$, so erhält man

$$A^{(1)}z = \begin{pmatrix} \sigma & w^T \\ 0 & B \end{pmatrix} \begin{pmatrix} \sigma \\ w \end{pmatrix} = \begin{pmatrix} \sigma^2 + \|w\|^2 \\ Bw \end{pmatrix} \tag{5.101}$$

Daher gilt

$$\|A^{(1)}z\|_2^2 = \left(\sigma^2 + \|w\|^2\right)^2 + \|Bw\|_2^2 \geq (\sigma^2 + \|w\|^2)\|z\|_2^2. \tag{5.102}$$

Daher muss $\|w\| = 0$ und damit $w = 0$ gelten. Die Matrix hat also die Gestalt

$$\left(U^{(1)}\right)^T A^{(1)} V^{(1)} = \begin{pmatrix} \sigma & 0 \\ 0 & B \end{pmatrix}. \tag{5.103}$$

Nun wenden wir induktiv dasselbe Verfahren auf B an.

An einem Punkt kann es vorkommen, dass $\|B\| = 0$. In diesem Fall können wir die Konstruktion bereits abbrechen, da nun alle weiteren Vektoren im Kern der Matrix liegen. Zu diesen können wir beliebige Bildvektoren im orthogonalen Komplement des bereits konstruierten Raums wählen. □

5.5.5 Lemma: Für die Singulärwertzerlegung $A = U\Sigma V^T$ gilt für die Spalten von U und V

$$Av^{(i)} = \sigma_i u^{(i)}, \qquad A^T u^{(i)} = \sigma_i v^{(i)}, \qquad i \leq \min\{m, n\}. \tag{5.104}$$

Sei $\sigma_r \neq 0$ der letzte von null verschiedene Singulärwert. Dann gilt für den Rang der Matrix $\operatorname{rank} A = r$ und

$$\begin{aligned} \operatorname{im}(A) &= \operatorname{span}\left\{u^{(1)}, \ldots, u^{(r)}\right\} & \ker(A) &= \operatorname{span}\left\{v^{(r+1)}, \ldots, v^{(n)}\right\} \\ \operatorname{im}\left(A^T\right) &= \operatorname{span}\left\{v^{(1)}, \ldots, v^{(r)}\right\} & \ker\left(A^T\right) &= \operatorname{span}\left\{u^{(r+1)}, \ldots, u^{(m)}\right\} \end{aligned} \tag{5.105}$$

Beweis. Die Eigenschaften (5.104) liest man direkt der Darstellung ab, denn aufgrund der Orthogonalität gilt $V^T v^{(i)} = e_i \in \mathbb{R}^n$. Damit gilt $\Sigma V^T v^{(i)} = \sigma_i e_i \in \mathbb{R}^m$. Schliesslich selektiert Ue_i den i-ten Spaltenvektor von U. Für die transponierte Matrix gilt dies wegen

$$A^T = V\Sigma^T U^T, \tag{5.106}$$

wobei bei der Transposition von Σ nur die Dimensionen getauscht werden, die Diagonalelemente bleiben natürlich gleich.

Die Aussage (5.105) ist dann eine direkte Folge. Insbesondere ist der Rang der Matrix durch den Index des letzten positiven Singulärwerts charakterisiert. □

Bemerkung 5.5.6. Aus Gleichung (5.105) lesen wir direkt die bekannten Beziehungen aus der linearen Algebra ab. Oder umgekehrt, die Singulärwertzerlegung erzeugt Orthonormalbasen der Vektorräume, die in Kern und orthogonales Komplement zerlegen. Zusätzlich wird der (eingeschränkte) Isomorphismus noch diagonalisiert.

5.5.7 Satz: Sei $A \in \mathbb{R}^{m\times n}$ mit Singulärwertzerlegung $A = U\Sigma V^T$ und Rang r. Sei

$$\Sigma^+ = \operatorname{diag}\left(\tfrac{1}{\sigma_1}, \tfrac{1}{\sigma_2}, \ldots, \tfrac{1}{\sigma_r}\right) \in \mathbb{R}^{m\times n}. \tag{5.107}$$

Dann ist der Vektor $x^* \in \mathbb{R}^n$ mit

$$x^* = V\Sigma^+ U^T b \tag{5.108}$$

die eindeutig bestimmte Lösung der Normalengleichungen mit minimaler Norm. Für das Residuum gilt

$$\|Ax^* - b\|_2^2 = \sum_{i=r+1}^{m} \left(\left(u^{(i)}\right)^T b\right)^2. \tag{5.109}$$

Beweis. Sei $x \in \mathbb{R}^n$ und $z = V^T x$ sei seine Koordinatendarstellung in der Basis V. Dann gilt

$$\begin{aligned}
\|Ax - b\|_2^2 &= \|AVV^T x - b\|_2^2 && (5.110)\\
&= \|U^T AVz - U^T b\|_2^2 && (5.111)\\
&= \|\Sigma z - U^T b\|_2^2 && (5.112)\\
&= \sum_{i=1}^{r}\left(\sigma_i z_i - \left(u^{(i)}\right)^T b\right)^2 + \sum_{i=r+1}^{m}\left(\left(u^{(i)}\right)^T b\right)^2. && (5.113)
\end{aligned}$$

Da alle Summanden nichtnegativ sind, wird das Minimum für

$$z_i = \frac{1}{\sigma_i}\left(u^{(i)}\right)^T b, \qquad i = 1, \ldots, r \tag{5.114}$$

angenommen. Damit verschwindet die erste Summe und (5.109) ist für den so bestimmten Vektor z bewiesen. Offensichtlich ist die Norm des Vektors z minimal, wenn alle weiteren Komponenten verschwinden, also

$$z_i = 0, \qquad i = r+1, \ldots, n. \tag{5.115}$$

Damit können wir zusammenfassend schreiben

$$z = \Sigma^+ U^T b. \tag{5.116}$$

Da V orthogonal ist, überträgt sich die Minimalitätseigenschaft auf $x^* = Vz$ □

Bemerkung 5.5.8. Für den Fall eine invertierbaren Matrix $A \in \mathbb{R}^{n\times n}$ entspricht (5.108) gerade der Inversen des Produkts. Im Falle das A vollen Rank hat mit $m \geq n$ bekommen wir die eindeutige Lösung der Normalengleichungen und die Bedingung „mit minimaler Norm“ entfällt.

5.5.9 Definition: Die Matrix $A^+ = V\Sigma^+U^T \in \mathbb{R}^{n\times m}$ ist eine Verallgemeinerung der Inversen, die als **Pseudoinverse**, auch als **Moore-Penrose-Inverse** bezeichnet wird. Sie ist für jede Matrix $A \in \mathbb{R}^{m\times n}$ definiert.

5.5.10 Satz (Penrose-Axiome): Für die Pseudoinverse $A^+ \in \mathbb{R}^{n\times m}$ einer Matrix $A \in \mathbb{R}^{m\times n}$ gelten folgende Gleichungen:

$$\left(A^+A\right)^T = A^+A, \tag{5.117}$$
$$\left(AA^+\right)^T = AA^+, \tag{5.118}$$
$$A^+AA^+ = A^+, \tag{5.119}$$
$$AA^+A = A. \tag{5.120}$$

Insbesondere ist A^+A die orthogonale Projektion auf im $\left(A^T\right)$ und AA^+ die orthogonale Projektion auf im (A).

Beweis. Es gilt

$$A^+A = V\Sigma^+U^TU\Sigma V^T = V\Sigma^+\Sigma V^T = VE_rV^T, \tag{5.121}$$

wobei $E_r \in \mathbb{R}^{n\times n}$ mit

$$E_r = \operatorname{diag}(\underbrace{1,\ldots,1}_{r \text{ mal}}). \tag{5.122}$$

Daraus folgen sofort Symmetrie und Projektionseigenschaft, sowie aus letzterer (5.119). Dieseben Argumente bleiben korrekt, wenn man A^+ und A vertauscht, wobei dann $E_r \in \mathbb{R}^{m\times m}$ ist. □

Bemerkung 5.5.11. Die Eigenschaft eines Vektors, im Nullraum der Matrix A zu liegen ist natürlich nicht invariant unter Störungen von A. Im Gegenteil wird im Allgemeinen die kleinste Störung dazu führen, dass alle Eigenwerte von null verschieden sind. Daher benötigen wir für die stabile Zerlegung eines Vektorraums in den Kern und sein orthogonales Komplement ein neues Konzept des Nullraums bzw. des Rangs einer Matrix.

5.5.12 Definition: Zu einer positiven Zahl ε ist der ε-Rang einer Matrix A ist definiert als

$$\operatorname{rank}_\varepsilon A = \min_{\|A-B\|_2 \leq \varepsilon} \operatorname{rank} B. \tag{5.123}$$

5.5.13 Satz: Sei $A \in \mathbb{R}^{m\times n}$ eine Matrix vom Rang r mit Singulärwertzerlegung $A = U\Sigma V^T$. Sei dazu $A_k = U\Sigma_k V^T$ die abgeschnittene SVD mit

$$\Sigma_k = \operatorname{diag}(\sigma_1, \ldots, \sigma_k). \tag{5.124}$$

Dann ist A_k die Bestapproximation zu A von maximalem Rang k in der Spektralnorm, also

$$\|A - A_k\|_2 = \min_{\operatorname{rank} B \leq k} \|A - B\|_2. \tag{5.125}$$

Es gilt ferner, dass

$$\|A - A_k\|_2 = \sigma_{k+1}. \tag{5.126}$$

Beweis. Es gilt

$$U^T(A - A_k)V = \operatorname{diag}(0, \ldots, 0, \sigma_{k+1}, \ldots, \sigma_r). \tag{5.127}$$

Da U und V orthogonal sind, gilt damit

$$\|A - A_k\|_2 = \sigma_{k+1}. \tag{5.128}$$

Nun müssen wir zeigen, dass für jede Matrix B mit $\operatorname{rank} B \leq k$ gilt

$$\|A - B\|_2 \geq \sigma_{k+1}. \tag{5.129}$$

Es gilt $\dim \ker(B) \geq n-k$. Daher hat dieser Nullraum einen nichtleeren Schnitt mit dem Erzeugnis der ersten $k+1$ Spaltenvektoren von V. Sei w ein Vektor in dieser Schnittmenge mit $\|w\|_2 = 1$. Es gilt dann $Bw = 0$. Schreiben wir

$$w = \sum_{i=1}^{k+1} z_i v^{(i)}, \tag{5.130}$$

so gilt nach der Parsevalschen Gleichung $\|z\|_2 = \|w\|_2 = 1$ und wir erhalten

$$Aw = \sum_{i=1}^{k+1} \sigma_i z_i u^{(i)}. \tag{5.131}$$

Es folgt

$$\|A-B\|_2^2 \geq \|(A-B)w\|_2^2 = \|Aw\|_2^2 \\ = \sum_{i=1}^{k+1} \sigma_i^2 z_i^2 \geq \sigma_{k+1}^2 \|z\|_2^2 = \sigma_{k+1}^2. \quad (5.132)$$

Für die zweite Ungleichung haben wir ausgenutzt, dass die Singulärwerte der Größe nach sortiert sind. □

5.5.14 Korollar: Der ε-Rang einer Matrix A mit Singulärwertzerlegung $A = U\Sigma V^T$ ist die Anzahl r der Singulärwerte $\sigma_k > \varepsilon$. Es gilt also

$$\sigma_1 \geq \cdots \geq \sigma_r > \varepsilon \geq \sigma_{r+1} \geq \cdots \quad (5.133)$$

Beweis. Im vorherigen Satz haben wir bewiesen, dass für die Matrix $A_r = U\Sigma_r V^T$ gilt

$$\|A - A_r\|_2 = \sigma_{r+1} \leq \varepsilon. \quad (5.134)$$

Damit gilt also bereits

$$\operatorname{rank}_\varepsilon A \leq \operatorname{rank} A_r = r. \quad (5.135)$$

Gäbe es nun eine Matrix B vom Rang $k < r$ mit $\|A - B\|_2 \leq \varepsilon$, so gälte nach dem Bestapproximationsresultat

$$\sigma_{k+1} = \|A - A_k\| \leq \|A - B\| \leq \varepsilon < \sigma_r \quad (5.136)$$

im Widerspruch zur Annahme, dass die Singulärwerte der Größe nach sortiert sind. □

Bemerkung 5.5.15. Das Korollar bietet eine einfachere Definition des ε-Rangs einer Matrix, nämlich als Anzahl aller Singulärwerte größer als ε. Das ist konsistent mit der Aussage, dass der Rang einer Matrix die Anzahl der von null verschiedenen Singulärwerte ist.

Bemerkung 5.5.16. Die Bedeutung des ε-Rangs besteht darin, dass bei der Berechnung der Pseudoinversen die Invertierung der Diagonalelemente nicht erst bei $\sigma_k = 0$, sondern bereits bei $\sigma_k \leq \varepsilon$ stoppen sollte, also

$$\Sigma_\varepsilon^+ = \operatorname{diag}\left(\tfrac{1}{\sigma_1}, \tfrac{1}{\sigma_2}, \ldots, \tfrac{1}{\sigma_{r_\varepsilon}}\right). \quad (5.137)$$

Die folgenden von null verschiedenen Singulärwerte $\sigma_k \leq \varepsilon$ dürfen nicht invertiert werden, da sonst das Ergebnis der Multiplikation mit der Matrix $A^+ =$

$V\Sigma_\varepsilon^+ U^T$ von diesen Werten dominiert würde. Die Elemente von Σ^+ an diesen Stellen setzen wir zu null, da wir die Singulärwerte ja numerisch als null ansehen.

Die Singulärwertzerelgung bietet also neben der Möglichkeit, die Pseudoinverse zu berechnen auch Kontrolle darüber, wie groß die Norm der inversen einer fast singulären Matrix werden darf.

Bemerkung 5.5.17. Die Aufgabe der Singulärwertzerlegung ist gut konditioniert und es gibt stabile Algorithmen zu ihrer Berechnung. Dazu benötigen wir aber mehr Informationen zur Eigenwertberechnung. Dies wird in der Vorlesung „Numerical Linear Algebra“ untersucht.

Literaturverzeichnis

[Deuflhard and Hohmann, 2008] Deuflhard, P. and Hohmann, A. (2008). *Numerische Mathematik 1.* de Gruyter, Berlin.

[Rannacher, 2017] Rannacher, R. (2017). *Numerik 0. Einführung in die Numerische Mathematik.* Lecture Notes Mathematik. Heidelberg University Publishing, Heidelberg.

[Stoer, 1983] Stoer, J. (1983). *Einführung in die Numerische Mathematik I.* Springer, 4 edition.

Index