Einführung in die Numerik

Guido Kanschat

26. Juli 2019

Inhaltsverzeichnis

Kapitel 1

Orthogonale Polynome

1.1 Polynomräume

1.1.1 Satz: Ein reelles Polynom vom Grad n hat höchstens n Nullstellen oder es ist das Nullpolynom.

Beweis. Für $n = 1$ handelt es sich um ein lineares Polynom und die Aussage des Satzes ist unmittelbar klar. Sei nun p ein Polynom strikt vom Grad $n > 1$ mit Nullstelle x_0. Dann gibt es nach dem euklidischen Algorithmus zur Division mit Rest ein Polynom q vom Grad $n - 1$ und eine Konstante c, so dass

$$p(x) = (x - x_0)q(x) + c. \tag{1.1}$$

Daraus folgt $p(x_0) = c$, so dass folgt $c = 0$. Wir können dieses Verfahren für alle weiteren Nullstellen $x_1, \ldots, x_m$ wiederholen und erhalten

$$p(x) = r(x) \prod_{k=0}^{m} (x - x_i), \tag{1.2}$$

wobei $r(x)$ ein Polynom vom Grad $n - m$ sein muss, da p vom Grad n ist. Insbesondere muss gelten $m \leq n$. □

1.1.2 Korollar: Zwei reelle Polynome vom Grad n sind identisch, wenn sie in mindestens $n + 1$ Punkten übereinstimmen. Insbesondere ist jede Menge von Monomen $\{x^{k_0}, x^{k_1}, \ldots, x^{k_n}\}$ für paarweise verschiedene Exponenten k_i linear unabhängig.

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

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 $\alpha, \beta \in \mathbb{R}$ gilt

$$a(\alpha u + \beta v, w) = \alpha a(u, w) + \beta a(v, w) \tag{1.3}$$
$$a(w, \alpha u + \beta v) = \alpha a(w, u) + \beta a(w, v). \tag{1.4}$$

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

$$a(u, v) = a(v, u). \tag{1.5}$$

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

$$a(u, u) = 0 \quad \Longrightarrow \quad u = 0. \tag{1.6}$$

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

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

$$\|u\| = \sqrt{\langle u, u \rangle} \tag{1.7}$$

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

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

$$\langle p, q \rangle = \int_{-1}^{1} p(x) q(x) \, \mathrm{d}x \tag{1.8}$$

ein Skalarprodukt definiert.

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

$$\langle u, v \rangle = 0. \tag{1.9}$$

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

$$\langle u, v \rangle = 0 \quad \forall v \in W. \tag{1.10}$$

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

1.2.6 Lemma (Bunjakowski-Cauchy-Schwarzsche Ungleichung): Für zwei beliebige Elemente $u, v \in V$ gilt

$$\langle u, v \rangle \leq \|u\| \, \|v\|. \tag{1.11}$$

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

Beweis. Lineare Algebra, auch Wikipedia. □

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

$$\|u + v\|^2 = \|u\|^2 + \|v\|^2 \tag{1.12}$$

Beweis. Hausaufgabe. □

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

$$\|u - u_b\| = \min_{v \in A} \|u - v\|. \tag{1.13}$$

1.3.2 Satz: 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

$$\langle u - u_b, v\rangle = 0 \quad \forall v \in W. \tag{1.14}$$

Beweis. Siehe [Rannacher, 2017, Satz 2.14] oder [Deuflhard and Hohmann, 2008, Satz 3.4]. □

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

$$W^\perp = \big\{ v \in V \big| \langle v, w\rangle = 0 \quad \forall w \in W \big\}. \tag{1.15}$$

Die Lösung $\Pi_W u = u_b \in W$ der Bestaproximationsaufgabe nennen wir die **orthogonale Projektion** von $u \in V$ auf W.

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

Beweis. Satz 1.3.2. □

1.3.5 Beispiel: Die Aufgabe der Gaußschen Ausgleichsrechnung 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

$$\int_{-1}^{1} \big(f(x) - p(x)\big)^2 \,\mathrm{d}x = \min_{q \in \mathbb{P}_n} \int_{-1}^{1} \big(f(x) - q(x)\big)^2 \,\mathrm{d}x. \tag{1.16}$$

Die Lösung erfüllt

$$\int_{-1}^{1} p(x)q(x) \,\mathrm{d}x = \int_{-1}^{1} f(x)q(x) \,\mathrm{d}x \qquad \forall q \in \mathbb{P}_n. \tag{1.17}$$

1.4 Orthogonale Basen

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

$$\mathbf{Gx} = \mathbf{b}. \tag{1.18}$$

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

$$g_{ij} = \langle \varphi_i, \varphi_j \rangle, \qquad b_i = \langle u, \varphi_i \rangle. \tag{1.19}$$

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

$$v = \sum_{i=1}^{n} x_i \varphi_i \tag{1.20}$$

die Identität

$$\|v\|^2 = \sum_{i=1}^{n} x_i^2. \tag{1.21}$$

1.4.6 Lemma: 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

$$x_i = b_i = \langle u, \varphi_i \rangle. \tag{1.22}$$

1.4.7 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:

$$\begin{aligned} &\varphi_1 = \tfrac{1}{\|v_1\|}\, v_1 \\ &w_j = v_j - \sum_{i=1}^{j-1} \langle v_j, \varphi_i \rangle\, \varphi_i \qquad \varphi_j = \tfrac{1}{\|w_j\|}\, w_j \qquad j = 2, \ldots, n \end{aligned} \tag{1.23}$$

Für alle $1 \leq k \leq n$ gilt

$$\operatorname{span}\{\varphi_1, \ldots, \varphi_k\} = \operatorname{span}\{v_1, \ldots, v_k\} \tag{1.24}$$

Beweis. Siehe [Rannacher, 2017]. □

1.4.8 Algorithmus (Gram-Schmidt):

```
def gram_schmidt(v):
  (n,m) = v.shape
  for j in range(n):
    delta = np.zeros(m)
    for i in range(j-1):
      r = sprod(v[:,j],v[:,i])
      delta += r*v[:,i]
    v[:,j] -= delta
    norm = np.sqrt(sprod(v[:,j],v[:,j]))
    v[:,j] /= norm
```

1.4.9 Beispiel: Wir wählen für Polynome das L^2-Skalarprodukt aus Lemma 1.2.3 und die Basis $\{1, x, \ldots, x^{n-1}\}$ für $\mathbb{P}_{n-1}$. Wir verwenden die Iplementation in Algorithmus 1.4.8 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

1.4.10 Algorithmus (Modifizierter Gram-Schmidt):

```
def modified_gram_schmidt(v):
  (n,m) = v.shape
  for j in range(m):
    r = np.zeros(m)
    for i in range(j-1):
      r = sprod(v[:,j],v[:,i])
      v[:,j] -= r*v[:,i]
    norm = np.sqrt(sprod(v[:,j],v[:,j]))
    v[:,j] /= norm
```

1.4.11 Beispiel: In dieser Tabelle wiederholen wir die Zahlen $\max_{i\neq j}|g_{ij}|$ aus Beispiel 1.4.9 und stellen sie den entsprechenden Ergebnissen des modifizierten Verfahrens in Algorithmus 1.4.10 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 1.4.12. 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 gibt es genau eine Folge von orthogonalen Polynomen $p_k \in \mathbb{P}_k$ mit führendem Koeffizienten eins. Sie genügen der Dreiterm-Rekursionsformel

$$p_k(x) = (x - a_k)p_{k-1}(x) - b_k p_{k-2}(x), \qquad k = 1, 2, \ldots \tag{1.25}$$

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

$$a_k = \frac{\langle x p_{k-1}, p_{k-1} \rangle}{\langle p_{k-1}, p_{k-1} \rangle} \qquad \text{und} \qquad b_k = \frac{\langle p_{k-1}, p_{k-1} \rangle}{\langle p_{k-2}, p_{k-2} \rangle}. \tag{1.26}$$

Beweis. Siehe [Deuflhard and Hohmann, 2008, Satz 6.2] □

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

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

$$L_k = \tfrac{2k-1}{k} x L_{k-1}(x) - \tfrac{k-1}{k} L_{k-2}(x). \tag{1.27}$$

Sie sind orthogonal bezüglich des L^2-Skalarprodukts in Lemma 1.2.3.

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

$$\int_{-1}^{1} (f-p)^2 \,\mathrm{d}x = \min_{q \in \mathbb{P}_n} \int_{-1}^{1} (f-q)^2 \,\mathrm{d}x. \tag{1.28}$$

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

$$p(x) = \sum_{i=0}^{n} \alpha_i L_i(x) \qquad \text{mit} \qquad \alpha_i = \frac{1}{\|L_i\|^2} \int_{-1}^{1} f L_i(x) \,\mathrm{d}x. \tag{1.29}$$

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

$$T_k = 2xT_{k-1}(x) - T_{k-2}(x). \tag{1.30}$$

Sie sind orthogonal bezüglich des Skalarprodukts

$$\langle p, q \rangle = \int_{-1}^{1} \tfrac{1}{\sqrt{1-x^2}}\, p(x)q(x) \,\mathrm{d}x. \tag{1.31}$$

Kapitel 2

Konditionierung und Stabilität

2.1 Fließkommazahlen

2.1.1 Definition: Die Darstellung einer numerischen größe als **Fließkommazahl** (auch **Gleitkommazahl**) beruht auf einer Basis $2 \leq b \in \mathbb{N}$. Sie besteht aus einer **Mantisse** $1/b \leq m < 1$ und einem Exponenten e, so dass eine Zahl $x \neq 0$ die Gestalt

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

hat. Diese Darstellung ist durch die Normierung von m eindeutig. 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 $\mathbb{M}$.

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

2.1.3 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.

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

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

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

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

2.1.6 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 gild also per definitionem

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

2.1.7 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}	$\approx 4.88 \cdot 10^{-4}$

2.1.8 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

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

2.1.9 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 2.1.10. Das vorherige Lemma gibt einen ersten Hinweis, warum die beiden Varianten des Gram-Schmidt-Verfahrens sich so verschieden verhalten.

2.1.11 Beispiel (Harmonische Reihe in Fließkommaarithmetik):

```
sum = np.float16(1.0)
i = 1.
old = np.float16(0.0)
while (sum != old):
    old = sum
    i += 1.
    diff = np.float16(1./i)
    sum += diff
print (sum)
```

Bricht das Programm ab? Warum?

2.1.12 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.

2.1.13 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 $\texttt{eps} \to 0$.

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

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

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

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**

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

2.2.12 Beispiel (Konditionierung der Multiplikation): Es gilt

$$y_1 = f(x_1, x_2) = x_1 x_2, \quad \frac{\partial f}{\partial x_1} = x_2, \quad \frac{\partial f}{\partial x_2} = x_1. \tag{2.19}$$

Damit folgt

$$\kappa_{11} = \kappa_{12} = 1, \tag{2.20}$$

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

$$y_1 = f(x_1, x_2) = x_1 + x_2, \quad \frac{\partial f}{\partial x_1} = 1, \quad \frac{\partial f}{\partial x_2} = 1. \tag{2.21}$$

Damit folgt

$$\kappa_{11} = \frac{1}{1 + \frac{x_2}{x_1}}, \qquad \kappa_{12} = \frac{1}{1 + \frac{x_1}{x_2}}. \tag{2.22}$$

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 1.4.7 ein Algorithmus, wie auch die beiden Beschreibungen in Python in Algorithmus 1.4.8 und Algorithmus 1.4.10. 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?

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

Numerische Intensität: Die Anzahl an Rechenoperationen pro Speichertransfer

Bemerkung 2.4.3. Bei der **Vektorisierung** von Algorithmen nutzt man aus, dass moderne Rechenwerke sehr effizient dieselbe Operation auf mehreren Daten ausführen können. Dazu ist es wichtig, dass logische Verzweigungen im Algorithmus auf möglichst hoher Ebene angesiedelt sind. Damit ist der Algorithmus, den man sowohl in [Rannacher, 2017] als auch in [Deuflhard and Hohmann, 2008] für die Berechnung von Nullstellen quadratischer Polynome findet, für die Vektorisierung ungeeignet.

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 Eigenschaft

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

Die Lagrange-Polynome sind orthonormal bezüglich des Skalarprodukts

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

Daher sind sie linear unabhängig und formen eine Basis von $\mathbb{P}_n$.

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

$$p(x; f; x_0, \ldots, x_n) = \sum_{i=0}^{n} f_i \ell_{i;x_0,\ldots,x_n}(x). \tag{3.8}$$

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

Bemerkung 3.1.5. Die Lagrangesche Interpolationsaufgabe kann auch als Gaußsche Ausgleichsrechnung mit dem obigen Skalarprodukt aufgefasst werden.

3.1.6 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

$$\|\delta y\| \leq \kappa^{\text{abs}} \|\delta x\| \qquad \forall \delta x \in X. \tag{3.9}$$

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

$$\|y\| \leq \kappa^{\text{abs}} \|x\| \qquad \forall x \in X. \tag{3.10}$$

Bemerkung 3.1.7. 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.

3.1.8 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**

$$\Lambda_{x_0,\ldots,x_n} = \max_{x\in[a,b]} \sum_{i=0}^{n} |\ell_{i;x_0,\ldots,x_n}(x)|. \tag{3.11}$$

Es gilt also

$$\max_{x\in[a,b]} |I_n f(x)| \leq \Lambda_{x_0,\ldots,x_n} \max_{x\in[a,b]} |f(x)|. \tag{3.12}$$

Diese Abschätzung ist scharf.

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

3.1.9 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

3.1.10 Lemma (Aitken): Für das Interpolationspolynom

$$p_{0,\dots,n}(x) = p(x; f; x_0, \dots, x_n) \tag{3.13}$$

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

$$p_{0,\dots,n}(x) = \frac{(x - x_0)p_{1,\dots,n}(x) - (x - x_n)p_{0,\dots,n-1}(x)}{x_n - x_0}. \tag{3.14}$$

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 $i = 1, \dots, n-1$

$$\begin{aligned} \varphi(x_i) &= \frac{(x_i - x_0)p_{1,\dots,n}(x_i) - (x_i - x_n)p_{0,\dots,n-1}(x_i)}{x_n - x_0} \\ &= \frac{(x_i - x_0)f_i - (x_i - x_n)f_i}{x_n - x_0} \\ &= f_i. \end{aligned} \tag{3.15}$$

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

3.1.11 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

$$p_{i0} = f_i \qquad i = 0, \dots, n. \tag{3.16}$$

2. Für $k = 1, \dots, n$ berechne

$$p_{ik} = p_{i,k-1} + \frac{x - x_i}{x_i - x_{i-k}}\big(p_{i,k-1} - p_{i-1,k-1}\big) \qquad i = k, \dots, n. \tag{3.17}$$

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

$$\omega_i(x) = \omega_{0,\dots,i}(x) = \prod_{j=0}^{i-1}(x - x_j), \qquad i = 0, \dots, n \tag{3.18}$$

wobei das leere Produkt für $i = 0$ den Wert 1 annehme.

3.1.13 Lemma: Sei $Q_k \in \mathbb{P}_n$ ein Polynom dargestellt bezüglich der Newton-Basis durch

$$Q_k(x) = \sum_{i=0}^{k} a_i \omega_i(x), \qquad k = 0, \dots, n. \tag{3.19}$$

Dann gilt

$$Q_k(x) = Q_{k-1}(x) + a_k \omega_k(x), \tag{3.20}$$

und a_k ist der Koeffizient vor x^k in der Monomdarstellung von $Q_k(x)$.

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

$$[x_i]f = f_i \tag{3.21}$$

$$[x_i, \dots, x_{i+k}]f = \frac{[x_{i+1}, \dots, x_{i+k}]f - [x_i, \dots, x_{i+k-1}]f}{x_{i+k} - x_i} \tag{3.22}$$

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

$$p_{i,\dots,i+k}(x) = \sum_{j=i}^{i+k} [x_i, \dots, x_{i+k}]f \; \frac{\omega_j(x)}{\omega_i(x)}. \tag{3.23}$$

Beweis. In der Newton-Darstellung gilt

$$p_{i,\dots,i+k}(x) = p_{i,\dots,i+k-1}(x) + \alpha \frac{\omega_{i+k}(x)}{\omega_i(x)}. \tag{3.24}$$

Zu zeigen ist also $\alpha = [x_i, \ldots, x_{i+k}]$, was nach Lemma 3.1.13 der Koeffizient vor x^k ist. Nach Induktionsannahme ist

$$\begin{aligned} p_{i,\ldots,i+k-1}(x) &= [x_i, \ldots, x_{i+k-1}] f\, x^{k-1} + \mathcal{O}(x^{k-2}) \\ p_{i+1,\ldots,i+k}(x) &= [x_{i+1}, \ldots, x_{i+k}] f\, x^{k-1} + \mathcal{O}(x^{k-2}) \end{aligned} \tag{3.25}$$

Nach dem Lemma von Aitken gilt

$$p_{i,\ldots,i+k} = \frac{(x - x_i) p_{i+1,\ldots,i+k} - (x - x_{i+k}) p_{i,\ldots,i+k-1}}{x_{i+k} - x_i}. \tag{3.26}$$

Dessen höchster Koeffizient ist aber gerade die dividierte Differenz. □

Bemerkung 3.1.16. Der Bruch im vorherigen Satz ist nicht problematisch, da

$$\frac{\omega_j(x)}{\omega_i(x)} = \prod_{\ell=i}^{j-1} (x - x_\ell). \tag{3.27}$$

3.1.17 Satz: Sei $f \in C^{n+1}[a, b]$ und $p \in \mathbb{P}_n$ die Lagrange-Interpolierende zu den Stützstellen $a = x_0 \neq \cdots \neq x_n = b$. 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

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{0,\ldots,n}(x). \tag{3.28}$$

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. Sei nun

$$F(y) = f(y) - p(y) - \alpha \omega_n(y) \tag{3.29}$$

und α soll so gewählt werden, dass $F(x) = 0$. Damit hat $F(y)$ im Intervall I insgesamt die $n+2$ Nullstellen $x, x_0, \ldots, x_n$. 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

$$0 = F^{(n+1)}(\xi) = f^{(n+1)}(\xi) - \alpha(n+1)! \tag{3.30}$$

und damit

$$\alpha = \frac{f^{(n+1)}(\xi)}{(n+1)!}. \tag{3.31}$$

□

3.1.18 Korollar: Sei $f \in C^{n+1}[a,b]$ und alle Stützstellen x_i im Intervall $[a,b]$. Dann gibt es $\xi \in [a,b]$, so dass

$$[x_0, \ldots, x_n]f = \frac{f^{(n)}}{n!}(\xi). \tag{3.32}$$

Beweis. Formel (3.29) gibt gerade an, dass α der Koeffizient vor dem nächsten Newton-Basispolynom ist, wenn man den Punkt x der Menge der Stützstellen hinzufügt. □

3.1.19 Korollar: Es gelten die Voraussetzungen von Satz 3.1.17. Dann gibt es eine Konstante C, die nur von der Wahl der Stützstellen abhängt, so dass

$$\max_{x\in[a,b]} |f(x) - p_{0,\ldots,n}(x)| \leq \frac{C|b-a|^{n+1}}{(n+1)!} \max_{x\in[a,b]} |f^{(n+1)}(x)| \tag{3.33}$$

Bemerkung 3.1.20. Die Fehlerabschätzung in Korollar 3.1.19 können wir auch kürzer schreiben als

$$\|f - p_{0,\ldots,n}\|_\infty \leq \frac{C|b-a|^{n+1}}{(n+1)!} \|f^{(n+1)}\|_\infty. \tag{3.34}$$

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.

3.1.21. 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 3.1.17 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.

3.1.22 Lemma: Die Tschebyscheff-Polynome, die der Rekursionsformel in Definition 1.5.5 genügen, haben die Darstellung

$$T_k = \cos(k \arccos x) \tag{3.35}$$

Insbesondere gilt

$$T_k(1) = 1 \tag{3.36}$$
$$T_k(-1) = (-1)^k \tag{3.37}$$
$$|T_k(x)| \leq 1, \qquad x \in [-1, 1] \tag{3.38}$$
$$T_k(x) = (-1)^j, \quad x = \cos\left(\frac{j}{k}\pi\right), \qquad j = 0, \ldots, k \tag{3.39}$$
$$T_k(x) = 0, \qquad x = \cos\left(\frac{2j-1}{2k}\pi\right), \quad j = 1, \ldots, k \tag{3.40}$$

Beweis. Hausaufgabe □

3.1.23 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

$$\frac{1}{2^{n-1}} T_n(x) = \operatorname*{argmin}_{\substack{p \in \mathbb{P}_n \\ p = x^n + \cdots}} \max_{x \in [-1,1]} |p(x)|. \tag{3.41}$$

Beweis. Siehe auch [Deuflhard and Hohmann, 2008, Satz 7.19]. Aus der Rekursionsformel 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

$$\max_{x \in [-1,1]} |p(x)| < 1. \tag{3.42}$$

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

$$T_n(\tilde{x}_j) = 1, \qquad p(\tilde{x}_j) < 1 \qquad q_n(\tilde{x}_j) > 0, \qquad j \text{ gerade} \tag{3.43}$$
$$T_n(\tilde{x}_j) = -1, \qquad p(\tilde{x}_j) > -1 \qquad q_n(\tilde{x}_j) < 0, \qquad j \text{ ungerade.} \tag{3.44}$$

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}

$$\min_{\substack{p\in\mathbb{P}_n \\ p=x^n+\cdots}} \max_{x\in[-1,1]} |p(x)| \geq 1, \tag{3.45}$$

und Gleichheit für das skalierte Tschebyscheff-Polynom. □

3.1.24 Korollar: Wählt man als Stützstellen die Werte

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{2i+1}{2n+2}\pi\right), \qquad i = 0, \ldots, n, \tag{3.46}$$

So gilt für den Fehler der Lagrange-Interpolation

$$\|f - p_{0,\ldots,n}\|_\infty \leq \frac{|b-a|^{n+1}}{2^{2n}(n+1)!} \|f^{(n+1)}\|_\infty. \tag{3.47}$$

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

$$x = \Phi(\xi) = \frac{a+b}{2} + \frac{b-a}{2}\xi. \tag{3.48}$$

Es gilt $\Phi(-1) = a$, $\Phi(1) = b$ und di Punkte x_i sind die Bilder der Tschebyscheff-Abszissen ξ_i zu T_{n+1}. Es gilt $\Phi'(\xi) = (b-a)/2$ und für die Funktion $F(\xi) = f(\Phi(\xi))$ gilt

$$\frac{d^k}{d\xi^k} F(\xi) = \left(\frac{b-a}{2}\right)^k \frac{d^k}{dx^k} f(x). \tag{3.49}$$

Auf $[-1, 1]$ folgern wir aus Satz 3.1.17 und Satz 3.1.23, dass für die Interpolation gilt

$$\max_{\xi\in[-1,1]} |F(\xi) - P(\xi)| \leq 2^{1-n} \max_{\xi\in[-1,1]} |F^{(n+1)}(\xi)|, \tag{3.50}$$

woraus folgt

$$\max_{x\in[a,b]} |f(x) - p(x)| \leq 2^{1-n} \left(\frac{b-a}{2}\right)^{n+1} \max_{x\in[a,b]} |f^{(n+1)}(x)|. \tag{3.51}$$

□

3.1.3 Hermite-Interpolation

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

$$\frac{d^j p}{dx^j}(x_i) = f_i^j, \qquad i = 0, \ldots, m, \quad j = 0, \ldots, n_i - 1, \tag{3.52}$$

und es gilt

$$\sum_i n_i = n + 1. \tag{3.53}$$

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

3.1.26 Satz: Definition 3.1.25 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

$$p(x) = \sum_{i=0}^{m} \sum_{j=0}^{n_i - 1} f_i^j H_{ij}(x). \tag{3.54}$$

Zunächst führen wir die Hilfspolynome

$$q_{ij}(x) = \frac{(x - x_i)^j}{j!} \prod_{k \neq i} \left(\frac{x - x_k}{x_i - x_k} \right)^{n_k} \tag{3.55}$$

ein. Es gilt

$$\begin{aligned} \frac{d^j q_{i,n_i-1}}{dx^j}(x_k) &= 0, \qquad k \neq i, \qquad j = 0, \ldots, n_k - 1, \\ \frac{d^j q_{i,n_i-1}}{dx^j}(x_i) &= 0, \qquad j = 0, \ldots, n_i - 2, \\ \frac{d^{n_i-1} q_{i,n_i-1}}{dx^{n_i-1}}(x_i) &= 1. \end{aligned} \tag{3.56}$$

Damit können wir rekursiv definieren

$$\begin{aligned} H_{i,n_i-1}(x) &= q_{i,n_i-1}(x) && i = 0, \dots, m \\ H_{ij}(x) &= q_{ij}(x) - \sum_{k=j+1}^{n_i-1} q_{ij}^{(k)}(x_i) H_{ik}(x), \end{aligned} \tag{3.57}$$

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

$$\frac{d^\ell}{dx^\ell} H_{ij}(x_k) = \delta_{ik}\delta_{j\ell}. \tag{3.58}$$

□

3.1.27 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

$$a = x_0 \leq x_1 \leq \cdots \leq x_n = b. \tag{3.59}$$

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.

3.1.28 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

$$p(x; f; x_0, \dots, x_n) = \sum_{k=0}^{n} \frac{(x-x_0)^k}{k!} f^{(k)}(x_0). \tag{3.60}$$

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

$$p(a), p'(a), p(b), p'(b). \tag{3.61}$$

3.1.30 Satz: Das Hermite-Interpolationspolynom genügt der Darstellung

$$p_{0,\dots,n}(x) = \sum_{j=0}^{n} [x_0, \dots, x_j] f \; \omega_j(x) \tag{3.62}$$

mit den verallgemeinerten dividierten Differenzen definiert durch die Rekursion

$$[x_i, \dots, x_{i+k}] f = \begin{cases} \frac{f^{(k)}(x_i)}{k!} & x_i = x_{i+k} \\ \frac{[x_{i+1},\dots,x_{i+k}] f - [x_i,\dots,x_{i+k-1}] f}{x_{i+k} - x_i} & x_i \neq x_{i+k}. \end{cases} \tag{3.63}$$

Beweis. Der Beweis folgt im wesentlichen dem analogen Satz 3.1.15. Wir müssen dort nur die Argumente anpassen, die auf paarweise verschiedenen Stützstellen beruhen.

Zunächst benutzen wir Korollar 3.1.18, wonach für $x_i < x_{i+1} < \cdots < x_{i+k}$ gilt: es gibt ein $\xi \in [x_i, x_{i+k}]$ mit

$$[x_i, \dots, x_{i+k}] f = \frac{f^{(k)}(\xi)}{k!}. \tag{3.64}$$

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

$$[x_i, \dots, x_{i+k}] f \to \frac{f^{(k)}(x_i)}{k!}, \tag{3.65}$$

sowie

$$\omega_{i,\dots,i+k}(x) \to (x - x_i)^k. \tag{3.66}$$

Für das zugehörige Interpolationspolynom gilt dann

$$\frac{d^j}{dx^j} p_{i,\dots,i+k}(x_i) = f^j(x_i) \qquad j = 0, \dots, k-1. \tag{3.67}$$

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

$$q(x) = \frac{(x - x_i) p_{i+1,\dots,i+k}(x) - (x - x_k) p_{i,\dots,i+k-1}(x)}{x_{i+k} - x_i} \tag{3.68}$$

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. □

3.1.31 Satz: Sei $f \in C^{n+1}[a,b]$ und $p \in \mathbb{P}_n$ die Hermite-Interpolierende zu den Stützstellen $a = x_0 \leq \cdots \leq x_n = b$. 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

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{0,\dots,n}(x). \tag{3.69}$$

Beweis. Der Beweis folgt exakt denselben Argumenten wie der von Satz 3.1.17. □

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

$$p(x) = \sum_{i=0}^{n} \frac{f^i(x_0)}{i!}(x - x_0)^i \tag{3.70}$$

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

$$f(x) - p(x) = \frac{f^{n+1}(\xi)}{(n+1)!}(x - x_0)^{n+1}. \tag{3.71}$$

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

$$a = x_0 < x_1 \cdots < x_n = b \tag{3.72}$$

stets

$$\mathcal{I}_h = \big\{ I_i = [x_{i-1}, x_i] \big| \; i = 1, \ldots, n \big\} \tag{3.73}$$

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

$$[a, b] = \bigcup_{i=1}^{n} I_h. \tag{3.74}$$

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 Notation: 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

$$\begin{aligned} \Phi_i \colon \hat{I} &\to I_i \\ \hat{x} &\mapsto \tfrac{x_i + x_{i-1}}{2} + \tfrac{h_i}{2} \hat{x}. \end{aligned} \tag{3.75}$$

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

$$x_{ij} = \Phi_i(\hat{x}_j), \qquad j = 1, \ldots, k \tag{3.76}$$

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

$$\|\hat{f} - \hat{p}\|_{\infty;\hat{I}} \leq C\|\hat{f}^{(k+1)}\|_{\infty;\hat{I}}. \tag{3.77}$$

Dann ist der Fehler der stückweisen Interpolation beschränkt ist durch

$$\|f - s\|_{\infty;[a,b]} \leq \frac{C}{2^{k+1}} h^{k+1} \|f^{(k+1)}\|_{\infty;[a,b]}. \tag{3.78}$$

3.2.6 Bemerkung: Genauere Betrachtung der Analyse ergibt die schärfere Abschätzung

$$\|f - s\|_{\infty;[a,b]} \leq \frac{C}{2^{k+1}} \max_{i=1,\ldots,n} \left(h_i^{k+1} \|f^{(k+1)}\|_{\infty;I_i} \right). \tag{3.79}$$

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**

$$S_h^{(k,m)} \left\{ s \in C^m[a,b] \middle| s_{|I_i} \in \mathbb{P}_k, i = 1, \ldots, n \right\} \tag{3.80}$$

mit $m < k$.

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

$$\dim S_h^{(k,m)} = (k - m)n + m + 1 \tag{3.81}$$

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

$$s(x_i) = f_i, \qquad i = 0, \ldots, n. \tag{3.82}$$

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

$$s''(a) = s''(b) = 0 \tag{3.83}$$

Periodisch

$$s'(a) = s'(b) \quad \wedge \quad s''(a) = s''(b) \tag{3.84}$$

Vollständig approximierend

$$s'(a) = f'(a) \quad \wedge \quad s'(b) = f'(b) \tag{3.85}$$

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

$$s \in N_h = \big\{ w \in C^2[a,b] \;\big|\; w(x_i) = 0, \quad i = 0, \ldots, n \big\}. \tag{3.86}$$

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

$$\int_{I_i} s''(x)w''(x)\,\mathrm{d}x = s''w'\Big|_{x_{i-1}}^{x_i} - \int_{I_i} s^{(3)}(x)w'(x)\,\mathrm{d}x \tag{3.87}$$

$$= s''w'\Big|_{x_{i-1}}^{x_i} - s^{(3)}w\Big|_{x_{i-1}}^{x_i} + \int_{I_i} s^{(4)}(x)w(x)\,\mathrm{d}x \tag{3.88}$$

$$= s''w'\Big|_{x_{i-1}}^{x_i}. \tag{3.89}$$

Summieren wir über alle Intervalle, so ergibt sich

$$\int_a^b s''(x)w''(x)\,\mathrm{d}x = \sum_{i=1}^{n} s''w'\Big|_{x_{i-1}}^{x_i} = s''(b)w'(b) - s''(a)w'(a). \tag{3.90}$$

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

$$\int_a^b |s''(x)|^2\,\mathrm{d}x = 0 \tag{3.91}$$

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 $f(x_i) = y_i$, $i = 0,\dots,n$ ist der natürliche Spline $s \in S_h^{(3,2)}$, der diese Punkte interpoliert, diejenige mit der kleinsten mittleren zweiten Ableitung, es gilt also

$$\int_a^b |s''(x)|^2\,\mathrm{d}x \le \int_a^b |f''(x)|^2\,\mathrm{d}x \qquad \forall f \in C^2[a,b]. \tag{3.92}$$

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

3.2.13 Lemma: Seien die Momente

$$M_i = s''(x_i), \qquad i = 0,\dots,n \tag{3.93}$$

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

$$s_{|I_i}(x) = a_{i0} + a_{i1}(x - x_i) + a_{i2}(x - x_i)^2 + a_{i3}(x - x_i)^3, \tag{3.94}$$

aus den Formeln

$$a_{i0} = f_i, \qquad a_{i1} = \tfrac{f_i - f_{i-1}}{h_i} + \tfrac{h_i(2M_i + M_{i-1})}{6}, \tag{3.95}$$

$$a_{i2} = \tfrac{M_i}{2} \qquad a_{i3} = \tfrac{M_i - M_{i-1}}{6h_i}. \tag{3.96}$$

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

$$s''(x) = M_{i-1}\frac{x_i - x}{h_i} + M_i\frac{x - x_{i-1}}{h_i}, \qquad x \in I_i. \tag{3.97}$$

Daraus erhalten wir durch Integration

$$\begin{aligned} s'(x) &= -M_{i-1}\frac{(x_i - x)^2}{2h_i} + M_i\frac{(x - x_{i-1})^2}{2h_i} + A_i \\ s(x) &= M_{i-1}\frac{(x_i - x)^3}{6h_i} + M_i\frac{(x - x_{i-1})^3}{6h_i} + A_i(x - x_{i-1}) + B_i \end{aligned} \tag{3.98}$$

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

$$B_i = y_{i-1} - M_{i-1}\frac{h_i^2}{6}, \qquad A_i = \frac{f_i - f_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1}). \tag{3.99}$$

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

$$\begin{pmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & \mu_n & 2 \end{pmatrix} \begin{pmatrix} M_0 \\ \vdots \\ M_n \end{pmatrix} = \begin{pmatrix} d_0 \\ \vdots \\ d_n \end{pmatrix} \tag{3.100}$$

wobei für $i = 1, \ldots, n-1$

$$\lambda_i = \tfrac{h_{i+1}}{h_i + h_{i+1}}, \qquad \mu_i = 1 - \lambda_i = \tfrac{h_i}{h_i + h_{i+1}}, \tag{3.101}$$

$$d_i = \tfrac{6}{h_i + h_{i+1}}\left[\tfrac{f_{i+1} - f_i}{h_{i+1}} - \tfrac{f_i - f_{i-1}}{h_i}\right] \tag{3.102}$$

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

$$d_0 = \frac{6}{h_1}\left(\frac{f_1 - f_0}{h_1} - f_0'\right), \qquad d_n = \frac{6}{h_n}\left(f_n' - \frac{f_n - f_{n-1}}{h_n}\right). \tag{3.103}$$

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 (3.98) hergeleitete Form: für $x \in I_i$ gilt

$$s'(x) = -M_{i-1}\frac{(x_i - x)^2}{2h_i} + M_i\frac{(x - x_{i-1})^2}{2h_i} + \frac{f_i - f_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1}). \tag{3.104}$$

Damit gilt am Punkt x_i

$$s'(x_i) = \frac{f_i - f_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1}) + M_i\frac{h_i}{2} \tag{3.105}$$

$$= \frac{f_i - f_{i-1}}{h_i} + \frac{h_i}{3}M_i + \frac{h_i}{6}M_{i-1} \tag{3.106}$$

$$s'(x_i) = \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{h_{i+1}}{6}(M_{i+1} - M_i) - M_i\frac{h_{i+1}}{2} \tag{3.107}$$

$$= \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{h_{i+1}}{3}M_i - \frac{h_{i+1}}{6}M_{i+1} \tag{3.108}$$

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

$$\frac{h_i}{6}M_{i-1} + \frac{h_i + h_{i+1}}{3}M_i + \frac{h_{i+1}}{6}M_{i+1} = \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i}. \tag{3.109}$$

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

$$A = \begin{pmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & \mu_n & 2 \end{pmatrix} \tag{3.110}$$

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

$$\|x\|_\infty \leq \|y\|_\infty. \tag{3.111}$$

Insbesondere ist A invertierbar.

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

$$y_k\mu_k x_{k-1} + 2x_k + \lambda_k x_{k+1}. \tag{3.112}$$

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

$$\min_i h_i \geq ch. \tag{3.113}$$

Dann gilt für den vollständig approximierenden Spline s zu den Funktionswerten $f(x_i)$ die Abschätzung

$$\|f^{(\nu)} - s^{(\nu)}\|_{\infty;[a,b]} \leq c_\nu c h^{4-\nu} \|f^{(4)}\|_{\infty;[a,b]} \tag{3.114}$$

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

$$\|M - g\|_\infty \leq \tfrac{3}{4}\|f^{(4)}\|_\infty h^2. \tag{3.115}$$

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

$$\|M - g\|_\infty \leq \|r\|_\infty. \tag{3.116}$$

Wir betrachten den Punkt x_0 und nutzen die Taylor-Interpolation

$$f(x_1) = f(x_0) + h_1 f'(x_0) + \frac{h_1^2}{2} f''(x_0) + \frac{h_1^3}{6} f^{(3)}(x_0) + \frac{h_1^4}{24} f^{(4)}(\xi_0), \tag{3.117}$$

$$f''(x_1) = f''(x_0) + h_1 f^{(3)}(x_0) + \frac{h_1^2}{2} f^{(4)}(\xi_1) \tag{3.118}$$

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

$$\begin{aligned} r_0 =& d_0 - 2f''(x_0) - f''(x_1) && (3.119)\\ =& \frac{h_1}{6}\left(\frac{f_1 - f_0}{h_1} - f_0'\right) - 2f''(x_0) - f''(x_1) && (3.120)\\ =& \frac{6}{h_1}\left[f'(x_0) + \frac{h_1}{2}f''(x_0) + \frac{h_1^2}{6}f^{(3)}(x_0) + \frac{h_1^3}{24}f^{(4)}(\xi_0) - f'(x_0)\right] && (3.121)\\ & - 2f''(x_0) - \left[f''(x_0) + h_1 f^{(3)}(x_0) + \frac{h_1^2}{2}f^{(4)}(\xi_1)\right] && (3.122)\\ =& \frac{h_1^2}{4}f^{(4)}(\xi_0) - \frac{h_1^2}{2}f^{(4)}(\xi_1). && (3.123) \end{aligned}$$

Damit gilt

$$|r_0| \leq \tfrac{3}{4}\|f^{(4)}\|_\infty h^2. \tag{3.124}$$

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

$$\begin{aligned} r_i &= d_i - \mu_i f''(x_{i-1}) - 2f''(x_i) - \lambda_i f''(x_{i+1}) && (3.125)\\ &= \frac{1}{h_i + h_{i+1}}\left[\frac{h_{i+1}^3}{4}f^{(4)}(\xi_1) + \frac{h_i^3}{4}f^{(4)}(\xi_2) - \frac{h_{1+1}^3}{2}f^{(4)}(\xi_3) - \frac{h_i^3}{2}f^{(4)}(\xi_4)\right]. \end{aligned}$$

Daher ist

$$|r_i| \leq \tfrac{3}{4}\|f^{(4)}\|_{\infty;[a,b]}h^2, \qquad i = 1, \ldots, n-1. \tag{3.126}$$

Aus (3.116) schließen wir

$$\|M - g\|_\infty \leq \|r\|_\infty \leq \frac{3}{4}h^2\|f^{(4)}\|_{\infty;[a,b]}. \tag{3.127}$$

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

$$\begin{aligned} e^{(3)}(x) = \frac{M_i - M_{i-1}}{h_i} - f^{(3)}(x)& \\ = \frac{M_i - f''(x_i)}{h_i} - \frac{M_{i-1} - f''(x_{i-1})}{h_i}& \\ + \frac{f''(x_i) - f''(x) - \big(f''(x_{i-1}) - f''(x)\big)}{h_i} - f^{(3)}(x).& \quad (3.128) \end{aligned}$$

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

$$\begin{aligned} f''(x_i) &= f''(x) + f^{(3)}(x)(x_i - x) + \frac{f^{(4)}(\xi_1)}{2}(x_i - x)^2\\ f''(x_{i-1}) &= f''(x) + f^{(3)}(x)(x_{i-1} - x) + \frac{f^{(4)}(\xi_1)}{2}(x_{i-1} - x)^2. \end{aligned} \tag{3.129}$$

Setzen wir dies und (3.127) in (3.128) ein, so erhalten wir

$$\begin{aligned}|s^{(3)}(x) - f(x)^{(3)}| &\leq \frac{3}{2}\frac{h^2}{h_i}\|f^{(4)}\|_{\infty;[a,b]} + \frac{h_i^2}{2h_i}\|f^{(4)}\|_{\infty;[a,b]} \\ &\leq 2ch\|f^{(4)}\|_{\infty;[a,b]} \end{aligned} \tag{3.130}$$

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

$$e''(x) = e''(\tilde{x}) + \int_{\tilde{x}}^{x} e^{(3)}(t)\,\mathrm{d}t, \tag{3.131}$$

so dass wir mit (3.127) und (3.130) folgern

$$\begin{aligned}|s''(x) - f''(x)| &\leq \frac{3}{4}h^2\|f^{(4)}\|_{\infty;[a,b]} + ch^2\|f^{(4)}\|_{\infty;[a,b]} \\ &\leq \frac{7}{4}ch^2\|f^{(4)}\|_{\infty;[a,b]}. \end{aligned} \tag{3.132}$$

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 $x \in I_i$

$$e'(x) = \int_{\xi_i}^{x} e''(t)\,\mathrm{d}t \tag{3.133}$$

und daher mit (3.132)

$$|s'(x) - f'(x)| \leq \frac{7}{4}ch^3\|f^{(4)}\|_{\infty;[a,b]}. \tag{3.134}$$

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

$$e(x) = \int_{\tilde{x}}^{x} e'(t)\,\mathrm{d}t \tag{3.135}$$

die Abschätzung

$$|s'(x) - f'(x)| \leq \frac{7}{8}ch^4\|f^{(4)}\|_{\infty;[a,b]}. \tag{3.136}$$

□

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

$$\|f^{(\nu)} - s^{(\nu)}\|_{\infty;[a,b]} = \mathcal{O}(h^{4-\nu}), \tag{3.137}$$

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

3.3 Interpolatorische Quadratur

3.3.1 Summierte Quadratur

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

$$Q_{[a,b]}(f) \approx \int_a^b f(x)\,\mathrm{d}x \tag{3.138}$$

in der Form

$$Q_{[a,b]}(f) = \sum_{i=0}^{n} \omega_i f(x_i). \tag{3.139}$$

Die Stützstellen x_i bezeichnen wir auch als **Quadraturpunkte**, die Zahlen ω_i als **Quadraturgewichte**.
Lässt sich die Quadraturormel bezüglich einer Zerlegung $\mathcal{I}_h$ des Intervalls $[a,b]$ in der Form

$$Q_{[a,b]}(f) = \sum_{i=1}^{n} Q_{I_i}(f) \tag{3.140}$$

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

3.3.2 Definition: Gilt bei einer summierten Quadraturformel die Abschätzung

$$\left| \int_{I_i} f(x)\,\mathrm{d}x - Q_{I_i}(f) \right| = \mathcal{O}\left(h_i^{k+1}\right) \tag{3.141}$$

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

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

$$c_q \min_{I_i \in \mathcal{I}_h} h_i \geq h. \tag{3.142}$$

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

$$\left| \int_a^b f(x)\,\mathrm{d}x - Q_{[a,b]}(f) \right| = \mathcal{O}\left(h^k\right). \tag{3.143}$$

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

$$\left| \int_{I_i} f(x)\,\mathrm{d}x - Q_{I_i}(f) \right| \leq c h_i^{k+1}. \tag{3.144}$$

Damit schätzen wir ab

$$\begin{aligned} \left| \int_a^b f(x)\,\mathrm{d}x - Q_{[a,b]}(f) \right| &= \sum_{I_i \in \mathcal{I}_h} \left| \int_{I_i} f(x)\,\mathrm{d}x - Q_{I_i}(f) \right| && (3.145)\\ &\leq \sum_{I_i \in \mathcal{I}_h} c h^{k+1} && (3.146)\\ &\leq n_{\max} c h^{k+1} = \mathcal{O}\left(h^k\right). && (3.147) \end{aligned}$$

□

3.3.2 Quadratur auf Einzelintervallen

3.3.4 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 $x_1, \ldots, x_n$. 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.

3.3.5 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

$$\int_I p(x)\,\mathrm{d}x - Q_I(p) = 0 \qquad \forall p \in \mathbb{P}_k. \tag{3.148}$$

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

$$\left| \int_I f(x)\,\mathrm{d}x - Q_I(f) \right| = \mathcal{O}\big(h^{k+2}\big) \tag{3.149}$$

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

3.3.8 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

$$\omega_i = \int_I \ell_{i;x_1,\ldots,x_n}(x)\,\mathrm{d}x, \tag{3.150}$$

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

$$\int_I \ell_i = \sum_{k=1}^{n} \omega_k \ell_i(x_k) = \omega_i. \tag{3.151}$$

□

3.3.9 Definition: Werden die Quadraturpunkte $x_1, \ldots, x_n$ 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	3	0	$1/2$	1		$1/6$	$4/6$	$1/6$	
$3/8$-Regel	4	0	$1/3$	$2/3$	1	$1/8$	$3/8$	$3/8$	$1/8$

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

$$\left| \int_I f \,\mathrm{d}x - Q_I(f) \right| \leq \begin{cases} \dfrac{h^3}{12} \max\limits_{\xi \in I} |f''(\xi)| & \text{Trapezregel} \\ \dfrac{h^5}{2880} \max\limits_{\xi \in I} |f^{(4)}(\xi)| & \text{Simpson-Regel} \\ \dfrac{h^5}{6480} \max\limits_{\xi \in I} |f^{(4)}(\xi)| & 3/8\text{-Regel} \end{cases} \tag{3.152}$$

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:

$$p(x_1) = f(x_1) \qquad p(x_2) = f(x_2) \tag{3.153}$$
$$p(x_3) = f(x_3) \qquad p'(x_2) = f'(x_2). \tag{3.154}$$

Die letzte Bedingung ist aus der Quadraturformel nicht ersichtlich. Folgen wir jedoch der Basiskonstruktion im Satz 3.1.26 über die Wohlgestelltheit der Hermite-Interpolationsaufgabe, so erhalten wir

$$H_{10}(x) = \tfrac{4(x-x_2)^2(x_3-x)}{h^3} \qquad H_{20}(x) = \tfrac{4(x-x_1)(x-x_3)}{h^2} \tag{3.155}$$
$$H_{30}(x) = \tfrac{4(x-x_2)^2(x-x_1)}{h^3} \qquad H_{21}(x) = \tfrac{4(x-x_2)(x-x_1)(x-x_3)}{h^2} \tag{3.156}$$

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

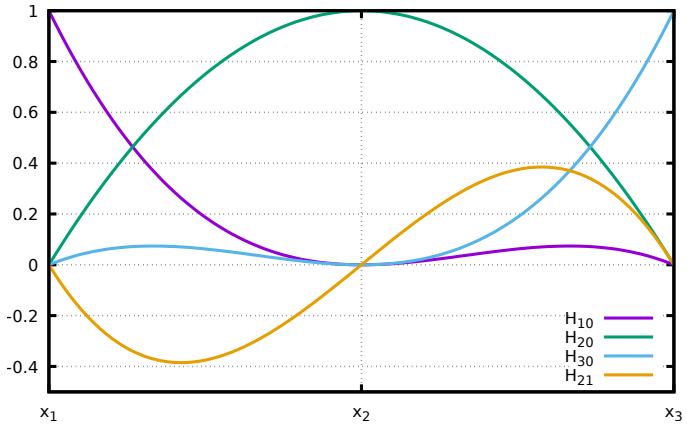

Abbildung 3.1: Basisfunktionen für die Simpsonregel. Beachte, dass H_21 in allen Quadraturpunkten verschwindet und auch das Integral null ist.

Intervallmitte. Daher verschwindet ihr Integral und das zugehörige Integrationsgewicht ist null. Die Simpson-Regel lässt sich also schreiben als

$$Q_I = \frac{h}{6} f(x_1) + \frac{4h}{6} f(x_2) + \frac{h}{6} f(x_3) + 0 f'(x_2). \tag{3.157}$$

Für die obige Interpolation gilt nach Satz 3.1.31 die Fehlerdarstellung

$$f(x) - p(x) = \frac{f^4(\xi(x)}{4!} \omega_{x_1,x_2,x_2,x_3}(x). \tag{3.158}$$

Integration ergibt

$$\left| \int_I f \,\mathrm{d}x - Q_I(f) \right| = \left| \int_I \big(f(x) - p(x)\big) \,\mathrm{d}x \right| \tag{3.159}$$

$$\leq \max_{\xi \in I} \frac{f^4(\xi}{4!} \int_I \omega_{x_1,x_2,x_2,x_3}(x) \,\mathrm{d}x. \tag{3.160}$$

Schließlich berechnen wir

$$\int_I \omega_{x_1,x_2,x_2,x_3}(x) \,\mathrm{d}x = \int_I (x - x_1)(x - x_2)^2 (x - x_3) = \frac{1}{120} \tag{3.161}$$

□

Bemerkung 3.3.11. Ab Grad ??? haben die Newton-Cotes-Formeln negative Gewichte...

3.3.3 Gauß-Quadratur

3.3.12 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 [Rannacher, 2017, Satz 3.1] □

3.3.13 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

$$\omega_i = \int_I \ell_{i;x_1,\dots,x_n}(x)\,\mathrm{d}x. \tag{3.162}$$

3.3.14 Satz: Die n-Punkt-Gauß-Legendre-Formel ist wohldefiniert und exakt für beliebige Polynome vom Grad $2n-1$.

3.3.15 Satz: Seien die Quadraturformeln Q_k mit Quadraturpunkten $x_1^{(k)},\dots,x_k^{(k)}$ für $k=1,\dots,n$ auf dem Intervall $I=[-1,1]$ exakt für beliebige $p\in\mathbb{P}_{2k-1}$. Dann sind die Polynome

$$p_k(x)=\prod_{i=1}^{k}(x-x_i^{(k)})\in\mathbb{P}_k \tag{3.163}$$

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_k und die n-Punkt-Gauss-Legendre-Formel ist die einzige Formel mit n Punkten, deren Grad der Exaktheit $2n-1$ ist.

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

$$\omega_i=\int_{-1}^{1}\prod_{j\neq i}\left(\frac{x-x_j}{x_i-x_j}\right)^2\mathrm{d}x. \tag{3.164}$$

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

$$\left| \int_I f \, \mathrm{d}x - Q_n(f) \right| \leq \max_{\xi \in I} \frac{f^{(2n)}(\xi)}{(2n)!} \int_{-1}^{1} \prod_{i=1}^{n} (x - x_i)^2. \tag{3.165}$$

Bemerkung 3.3.18. Alle Resultate dieses Abschnitts gelten für Skalarprodukte der Form

$$\langle p, q \rangle = \int_I \omega(x) p(x) q(x) \, \mathrm{d}x \tag{3.166}$$

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

3.3.19 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

$$|T(0) - p(0)| = \mathcal{O}(h^{np}). \tag{3.167}$$

Bemerkung 3.3.20. 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

$$T(h) - T(0) = \tau_1 h^p + \tau_2 h^{2p} + \ldots \tau_n h^{np} + \mathcal{O}(h^{(n+1)p}). \tag{3.168}$$

3.3.21 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.

3.3.22 Algorithmus (Romberg-Quadratur):

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

3.3.5 Praktische Aspekte

Bemerkung 3.3.23. Die Konvergenzabschätzungen der Form

$$\left| \int_I f \,\mathrm{d}x - Q_h(f) \right| \leq c h^p \|f^{p+1}\|_{\infty;I} \tag{3.169}$$

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

$$\left| \int_I f \,\mathrm{d}x - Q_h(f) \right| \leq c \|f\|_{\infty;I}. \tag{3.170}$$

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

Ist aber $f \in C^{p+1}(I)$, so gilt die obige Abschätzung für hinreichend kleine $h_1 > h_2$ in der stärkeren Form

$$\left| \int_I f \,\mathrm{d}x - Q_{h_2}(f) \right| \approx \left(\frac{h_2}{h_1} \right)^p \left| \int_I f \,\mathrm{d}x - Q_{h_1}(f) \right|. \tag{3.171}$$

Man spricht hier vom asymptotischen Bereich, für größere h vom präasymptotischen Bereich.

Kapitel 4

Lösung linearer Gleichungssysteme

4.1 Vektor- und Matrixnormen

4.1.1 Grundlagen

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

$$\begin{aligned} \|\cdot\| \colon V &\to \mathbb{R} \\ x &\mapsto \|x\| \end{aligned} \tag{4.1}$$

mit den Eigenschaften

Homogenität:	$\|\alpha x\| = \|\alpha\|\|x\|$	$\forall \alpha \in \mathbb{R}, x \in V$	(4.2)
Dreiecksungleichung:	$\|x + y\| \leq \|x\| + \|y\|$	$\forall x, y \in V$	(4.3)
Definitheit:	$\|x\| \geq 0$	$\forall x \in V$	(4.4)
	$\|x\| \neq 0$	$\forall x \neq 0$	(4.5)

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

4.1.2 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

$$c\|v\|_X \leq \|v\|_Y \leq C\|v\|_X \qquad \forall v \in V. \tag{4.6}$$

4.1.3 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

$$\forall \varepsilon > 0 \; \exists k_0 \in \mathbb{N} \; \forall k \geq k_0, i = 1, \ldots, n : |x_i^{(k)} - x_i| < \varepsilon. \tag{4.7}$$

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

$$\forall \varepsilon > 0 \; \exists k_0 \in \mathbb{N} \; \forall k \geq k_0 : \|x^{(k)} - x\| < \varepsilon. \tag{4.8}$$

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

$$f \colon x \mapsto \|x\| \tag{4.9}$$

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

$$\max_{i=1,\ldots,n} \left| \left(x_i^{(k)} - x_i \right) \|e_i\| \right| < \frac{\varepsilon}{n} \qquad \forall k \geq k_0. \tag{4.10}$$

Hier ist e_i der i-te Einheitsvektor. Dann folgt

$$\|x^{(k)} - x\| = \left\| \sum_{i=1}^{n} \left(x_i^{(k)} - x_i \right) e_i \right\| \tag{4.11}$$

$$\leq \sum_{i=1}^{n} \left\| \left(x_i^{(k)} - x_i \right) e_i \right\| \tag{4.12}$$

$$< n \frac{\varepsilon}{n} = \varepsilon. \tag{4.13}$$

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

Die „Einheitssphäre“

$$S = \left\{ x \in \mathbb{R}^n \middle| \|x\|_\infty = 1 \right\} \tag{4.14}$$

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

$$c\|x\|_\infty \leq \|x\| \leq C\|x\|_\infty. \tag{4.15}$$

□

4.1.5 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

$$\begin{aligned} c_X\|x\|_\infty \leq \quad \|x\|_X \quad &\leq C_X\|x\|_\infty \\ c_Y\|x\|_\infty \leq \quad \|x\|_Y \quad &\leq C_Y\|x\|_\infty. \end{aligned} \tag{4.16}$$

Daher gilt

$$\begin{aligned} \|x\|_Y \leq C_Y\|x\|_\infty &\leq \frac{C_Y}{c_X}\|x\|_X \\ \|x\|_X \leq C_X\|x\|_\infty &\leq \frac{C_X}{c_y}\|x\|_Y \end{aligned} \tag{4.17}$$

□

4.1.6 Definition: Auf dem Vektorraum der Matrizen $\mathbb{R}^{m\times n}$ ist durch Definition 4.1.1 eine Norm definiert. Gilt zusätzlich

$$\|Ax\| \leq \|A\|\|x\| \qquad \forall A \in \mathbb{R}^{m\times n}, x \in \mathbb{R}^n, \tag{4.18}$$

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

$$\|AB\| \leq \|A\|\|B\|. \tag{4.19}$$

Ferner definieren wir die **Operatornorm** oder **natürliche Norm**

$$\|A\| = \sup_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|. \tag{4.20}$$

4.1.7 Lemma: Die Operatornorm ist verträglich und submultiplikativ.

4.1.8 Beispiel: Die Operatornormen zu den Vektornormen $\|.\|_1$ und $\|.\|_\infty$ sind die **Spaltensummennorm** und die **Zeilensummennorm**

$$\|A\|_1 = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{ji}| \tag{4.21}$$

$$\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{ij}| \tag{4.22}$$

4.1.2 Eigenwerte und die Spektralnorm

4.1.9 Definition: Sei $A \in \mathbb{R}^{n\times n}$. Gilt für einen Vektor $0 \neq x \in \mathbb{R}^n$

$$Ax = \lambda x, \tag{4.23}$$

so nennen wir λ **Eigenwert** von A und x einen zugehörigen **Eigenvektor**. Wir notieren die Zugehörigkeit zur Matrix A auch explizit durch $\lambda(A)$.

4.1.10 Lemma: Für alle Eigenwerte $\lambda \in \mathbb{C}$ einer Matrix $A \in \mathbb{R}^{n\times n}$ gilt

$$|\lambda| \leq \|A\| \tag{4.24}$$

für jede zu einer beliebigen Vektornorm verträglichen Norm.

4.1.11 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. □

4.1.12 Satz: Die Operatornorm zur euklidischen Norm ist die **Spektralnorm**

$$\|A\|_2 = \max_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} \sqrt{\frac{x^T A^T A x}{x^T x}} = \sqrt{\lambda_{\max}(A^T A)}. \tag{4.25}$$

Insbesondere gilt für symmetrische Matrizen $\|A\|_2 = \max_i |\lambda_i(A)|$.

Beweis. Nach Satz 4.1.11 gibt es eine Basis des $\mathbb{R}^n$ von Eigenvektoren $v^{(i)}$ von $A^T A$. Jeder beliebige Vektor x besitzt damit die Darstellung

$$x = \sum_{i=1}^{n} \alpha_i v^{(i)}. \tag{4.26}$$

Es gilt nach der Parsevalschen Gleichung $\|x\|_2 = \|\alpha\|_2$. Ferner gilt mit den Eigenwerten $\lambda_i = \lambda_i(A^T A)$

$$\|Ax\|_2^2 = x^T A^T A x = \sum_{i=1}^{n} \lambda_i \alpha_i^2. \tag{4.27}$$

Daher gilt

$$\|A\|_2^2 = \max_{x\in\mathbb{R}^n} \frac{\|Ax\|^2}{\|x\|^2} = \max_{\alpha} \frac{\sum \lambda_i \alpha_i^2}{\sum \alpha_i^2} \leq \lambda_{\max}(A^T A). \tag{4.28}$$

Da für symmetrische Matrizen $A = A^T$, so ist

$$\lambda_{\max}(A^T A) = \lambda_{\max}(A^2) = \lambda_{\max}^2(A) \tag{4.29}$$

□

4.1.13 Definition: Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt **positiv definit**, wenn

$$x^T A x > 0 \qquad \forall 0 \neq x \in \mathbb{R}^n. \tag{4.30}$$

4.1.14 Satz: Eine symmetrische Matrix $A \in \mathbb{R}^{n\times n}$ ist positiv definit genau dann, wenn ihre Eigenwerte alle positiv sind.

4.1.3 Konditionierung der Lösung

4.1.15 Definition: Die Aufgabe, das lineare Gleichungssystem

$$Ax = b \tag{4.31}$$

zu lösen wandelt die Eingabedaten (A, b) in das Ausgabedatum x um. Die zugehörige gestörte Aufgabe ist

$$(A + \delta A)(x + \delta x) = b + \delta b, \tag{4.32}$$

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

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa \left[\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right]. \tag{4.33}$$

4.1.16 Lemma: Sei $B \in \mathbb{R}^{n \times n}$ mit $\|B\| < 1$. Dann ist $I - B$ invertierbar und es gilt

$$\|(I - B)^{-1}\| \leq (1 - \|B\|)^{-1} \tag{4.34}$$

Beweis. Siehe [Rannacher, 2017, Hilfssatz 4.4]. □

4.1.17 Satz: Sei die Matrix $A \in \mathbb{R}^{n \times n}$ invertierbar und

$$\|\delta A\| < \frac{1}{\|A^{-1}\|}. \tag{4.35}$$

Dann ist die gestörte Matrix $A + \delta A$ ebenfalls invertierbar und es gilt die Fehlerabschätzung

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\mathrm{cond}(A)}{1 - \mathrm{cond}(A)^{\|\delta A\|/\|A\|}} \left[\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right]. \tag{4.36}$$

Hierzu definieren wir die **Konditionszahl** der Matrix A zur Norm $\|\cdot\|$

$$\mathrm{cond}(A) = \|A\| \, \|A^{-1}\| \tag{4.37}$$

Beweis. Siehe [Rannacher, 2017, Satz 4.1]. □

Bemerkung 4.1.18. Satz 4.1.17 gilt unabhängig von der Wahl der Norm, sobald die Konditionszahl der Matrix konsistent definiert ist. Es ist dabei durchaus möglich, dass die Bedingung (4.35) 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 4.1.19. Die Bedingung (4.35) 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

$$\kappa = \frac{\operatorname{cond}(A)}{1 - \operatorname{cond}(A)^{\|\delta A\|}/_{\|A\|}} = \operatorname{cond}(A)\frac{1}{1 - \|\delta A\|\|A^{-1}\|}, \tag{4.38}$$

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 4.1.20. Ohne Einschränkung der Allgemeinheit ist das Resultat von Satz 4.1.17 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.

4.2 Die LR-Zerlegung

4.2.1 Notation: Da wir uns in diesem Abschnitt mit der Lösung quadraischer Gleichungssysteme beschäftigen, gelte für alle Matrizen, soweit nicht anders vermerkt, dass ihre Dimension $n \times n$ sei.

4.2.1 Dreiecksmatrizen und Frobeniusmatrizen

4.2.2 Definition: Für eine **untere Dreiecksmatrix** $L \in \mathbb{R}^{n\times n}$ gilt

$$\ell_{ij} = 0, \qquad j > i. \tag{4.39}$$

Für eine **obere Dreiecksmatrix** $R \in \mathbb{R}^{n\times n}$ gilt

$$r_{ij} = 0, \qquad j < i. \tag{4.40}$$

4.2.3 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 □

4.2.4 Korollar: Eine Dreiecksmatrix ist invertierbar genau dann, wenn alle ihre Diagonalelemente von null verschieden sind.

4.2.5 Algorithmus: Die Lösung der linearen Gleichungssysteme

$$Lx = b \qquad Rx = b \tag{4.41}$$

mit einer unteren Dreiecksmatrix L und einer oberen Dreiecksmatrix R lässt sich sukzessive durch Vorwärts- bzw. Rückwärtseinsetzen berechnen.

```
def forward_subst(A,b):
  (m,n) = A.shape
  x = np.zeros(n)
  for i in range(0,n):
    x[i] = b[i]
    for j in range(0,i):
      x[i] -= A[i,j]*x[j]
    x[i] /= A[i,i]
  return x
```

```
def backward_subst(A,b):
  (m,n) = A.shape
  x = np.zeros(n)
  for i in range(n-1,-1,-1):
    x[i] = b[i]
    for j in range(i+1,n):
      x[i] -= A[i,j]*x[j]
    x[i] /= A[i,i]
  return x
```

4.2.6 Definition: Eine Matrix der Gestalt

$$G_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & g_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & g_{nk} & & 1 \end{bmatrix} \tag{4.42}$$

mit von null verschiedenen Subdiagonaleinträgen nur in Spalte k heißt **Frobenius-Matrix**.

4.2.7 Lemma: Für Frobenius-Matrizen gilt

$$G_k^{-1} = 2\mathbb{I} - G_k. \tag{4.43}$$

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.
Sei $k_1 < \cdots < k_m$ eine aufsteigende Folge von Indizes. Dann ist

$$G_{k_1} \cdots G_{k_m} = \sum_{i=1}^{m} G_i - (m-1)\mathbb{I}. \tag{4.44}$$

Insbesondere gilt

$$G_1 \dots G_n = \begin{bmatrix} 1 & & & \\ g_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ g_{n1} & \cdots & g_{n,n-1} & 1 \end{bmatrix} \tag{4.45}$$

4.2.2 Konstruktion der LR-Zerlegung

4.2.8 Lemma: Bei der Gauß-Elimination lässt sich die Elimination der Subdiagonalelemente der k-ten Spalte als Matrix-Produkt

$$A^{(k+1)} = L_k^{-1} A^{(k)}, \qquad b^{(k+1)} = L_k^{-1} b^{(k)}, \qquad k = 1, \ldots, n-1 \quad (4.46)$$

mit $A^{(1)} = A$, $b^{(1)} = b$ und den Frobenius-Matrizen

$$L_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & \ell_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & \ell_{nk} & & 1 \end{bmatrix}, \qquad \ell_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \quad (4.47)$$

schreiben.

4.2.9 Satz: Nach $n-1$ Schritten der Gauß-Elminiation erhält man das transformierte lineare Gleichungssystem

$$Rx = y, \qquad R = L^{-1}A, \qquad y = L^{-1}b, \qquad L = L_1 \cdots L_{n-1}, \quad (4.48)$$

und die LR-Zerlegung

$$A = LR \quad (4.49)$$

mit einer oberen Dreiecksmatrix R und einer unteren Dreiecksmatrix L, deren Diagonale aus Einsen besteht.

4.2.10 Algorithmus (LR-Zerlegung):

```
def lu_decomposition(A):
  (m,n) = A.shape
  for k in range (0,n-1):
    piv = 1./ A[k,k]
    for i in range (k+1,n):
      A[i,k] *= piv
      for j in range (k+1,n):
        A[i,j] -= A[k,j]*A[i,k]
```

4.2.11 Algorithmus (Vorwärts-Rückwärts-Einsetzen):

```
def forward_backward(LR,b):
  (m,n) = LR.shape
  x = np.zeros(n)
  for i in range(0,n):
    x[i] = b[i]
    for j in range(0,i):
      x[i] -= LR[i,j]*x[j]
  for i in range(n-1,-1,-1):
    for j in range(i+1,n):
      x[i] -= LR[i,j]*x[j]
    x[i] /= LR[i,i]
  return x
```

4.2.12 Lemma: Der Aufwand der LR-Zerlegung einer $n \times n$-Matrix ist

$$\tfrac{1}{3}n^3 + \mathcal{O}(n^2). \tag{4.50}$$

4.2.13 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.

4.2.14 Definition: Führt man im k-ten Schritt der Gauß-Elimination eine Zeilenvertauschung durch, so dass

$$|a_{kk}^{(k)}| = \max_{j \geq k} |a_{jk}^{(k)}|, \tag{4.51}$$

so spricht man von Gauß-Elimination mit **Spalten-Pivotierung**. Vertauscht man sogar die verbleibenden Zeilen und spalten, so dass

$$|a_{kk}^{(k)}| = \max_{i,j \geq k} |a_{ij}^{(k)}|, \tag{4.52}$$

handelt es sich um **vollständige Pivotierung**.

4.2.15 Lemma: Führt man die Gauß-Elimination mit Spalten-Pivotierung durch, so gilt für die Matrix L:

$$|\ell_{ij}| \leq 1, \qquad 1 \leq i, j \leq n. \tag{4.53}$$

4.2.16 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

$$P_\pi L_k P_\pi^{-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & \ell_{\pi(k+1),k} & 1 & \\ & & \vdots & & \ddots \\ & & \ell_{\pi(n)k} & & 1 \end{bmatrix} \tag{4.54}$$

wieder eine Frobenius-Matrix gleicher Struktur wier L_k.

Beweis. Jede Permutation ist das Produkt von Transpositionen. Für solche wird die Aussage in der Hausaufgabe gezeigt. □

4.2.17 Satz: Nach $n-1$ Schritten der Gauß-Elimination mit Spalten-Pivotierung erhält man die Zerlegung

$$PA = LR \tag{4.55}$$

mit einer Permutationsmatrix P und den Dreiecksmatrizen L und R.

Beweis. Siehe [Deuflhard and Hohmann, 2008, Abschnitt 1.3]. □

4.2.3 Fehleranalyse

Ohne Beweis geben wir die folgenden Resultate zur Rundungsfehleranalyse der Lösung linearer Gleichungssysteme mit der LR-Zerlegung an.

4.2.18 Lemma: Die Realisierung des Vorwärtseinsetzens für das Gleichungssystem $LX = b$ in Fließkommaarithmetik berechnet die Lösung $\hat{x}$ des gestörten Systems $\hat{L}\hat{x} = b$ mit

$$|\ell_{ij} - \hat{\ell}_{ij}| \lessdot n|l_{ij}|\texttt{eps}. \tag{4.56}$$

4.2.19 Lemma: Die Matrix A besitze eine LR-Zerlegung. Dann berechnet das Gaußsche Eliminationsverfahren für das Gleichungssystem $AX = b$ die Lösung $\hat{x}$ des Systems $(A+\delta A)\hat{x} = b$ für eine Matrix $A+\delta A$ mit

$$|\delta a_{ij}| \lessdot 2n|\hat{\ell}_{ij}||\hat{r}_{ij}|\texttt{eps}. \tag{4.57}$$

4.2.20 Satz (Wilkinson): Das Gaußsche Eliminationsverfahren mit Spaltenpivotierung für das Gleichungssystem $Ax = b$ berechnet die Lösung $\hat{x}$ des Systems $\hat{A}\hat{x} = b$ für eine Matrix $\hat{A}$ mit

$$\frac{\|\delta A\|_\infty}{\|A\|_\infty} \lessdot 2n^3 \frac{\alpha_{\max}}{\max|a_{ij}|}\texttt{eps}, \tag{4.58}$$

wobei

$$\alpha_{\max} = \max_{1\le k,i,j\le n} |a_{ij}^{(k)}|. \tag{4.59}$$

4.2.4 Anmerkungen zur LR-Zerlegung

Bemerkung 4.2.21. Es gibt einige Aussagen über die LR-Zerlegung von Matrizen mit spezielleren Strukturen. So gilt

1. Ist die Matrix A invertierbar und schwach diagonaldominant, das heißt,

$$|a_{ii}| \ge \sum_{j\neq i} |a_{ij}|, \qquad i = 1, \ldots, n, \tag{4.60}$$

   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 4.2.22. 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.

4.3 Die QR-Zerlegung

4.3.1. Betrachten wir die Konstruktion der LR-Zerlegung als Folge von Operationen auf Matrizen, so ergibt sich das Bild

$$\begin{aligned} A^{(1)} &\mapsto L_1^{-1} A^{(2)} \\ A^{(2)} &\mapsto L_2^{-1} A^{(3)} \\ A^{(n-1)} &\mapsto L_{n-1}^{-1} R. \end{aligned} \tag{4.61}$$

Es findet also in jedem Schritt eine Umformung der Matrix des aktuellen Schritts mit einer Frobeniusmatrix statt.

In der Fehlerabschätzung steht der Faktor

$$\varrho = \frac{\alpha_{\max}}{\max|a_{ij}|} = \frac{\max|a_{ij}^{(k)}|}{\max|a_{ij}|}, \tag{4.62}$$

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

$$\|A^{(n-1)}\| = \cdots = \|A^{(k)}\| = \cdots = \|A^{(1)}\| \tag{4.63}$$

für eine geeignete Norm gilt.

4.3.1 Orthogonale Matrizen

4.3.2 Definition: Eine **orthogonale Matrix** ist eine quadratische Matrix, deren Spaltenvektoren bzw. deren Zeilenvektoren eine Orthonormalbasis des $\mathbb{R}^n$ bilden.

4.3.3 Satz: Für eine orthogonale Matrix Q gilt

$$Q^{-1} = Q^T. \tag{4.64}$$

Umgekehrt folgt aus dieser Beziehung die Orthogonalität der Zeilenvektoren und Spaltenvektoren.

Beweis. Nehmen wir an, die Spaltenvektoren $q^{(1)}, \dots, q^{(n)}$ von Q seinen eine ONB. Dann gilt für die Matrix $A = Q^T Q$:

$$a_{ij} = \sum_{k=1}^{n} q_{ki} q_{kj} = \sum_{k=1}^{n} q_k^{(i)} q_k^{(j)} = \left(q^{(i)}\right)^T q^{(j)} = \delta_{ij}. \tag{4.65}$$

Daher gilt $Q^T Q = I$. Multiplizieren wir diese Gleichung von rechts mit Q^{-1}, so erhalten wir (4.64). Setzen wir umgekehrt $Q^T Q = I$, so ergibt obige Rechnung die Orthogonalität der Spaltenvektoren.

Aus $Q^T = Q^{-1}$ folgt aber durch Transponieren

$$Q = Q^{-T}, \tag{4.66}$$

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 4.3.4. Die Rotationsmatrix

$$Q = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix} \tag{4.67}$$

ist orthogonal. Dasselbe gilt für die Reflexionsmatrix an einem Vektor $w \in \mathbb{R}^n$,

$$Q = I - 2\frac{ww^T}{w^T w} \tag{4.68}$$

4.3.5 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

$$\|Qx\|_2 = \|x\|_2, \qquad \|QA\|_2 = \|A\|_2. \tag{4.69}$$

4.3.2 Existenz und Konstruktion

4.3.6 Definition: Bei der **QR-Zerlegung** wird die Matrix $A \in \mathbb{R}^{n\times n}$ in das Produkt

$$A = QR \tag{4.70}$$

aus einer orthogonalen Matrix Q und einer oberen Dreiecksmatrix R zerlegt.

4.3.7 Lemma: Seien $q^{(1)}, \ldots, q^{(n)}$ die Spaltenvektoren von Q und $a^{(1)}, \ldots, a^{(n)}$ die Spaltenvektoren von A. Dann gilt

$$a^{(k)} = \sum_{i=1}^{k} r_{ik} q^{(i)}. \tag{4.71}$$

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

$$\operatorname{span}\{q^{(1)}, \ldots, q^{(k)}\} = \operatorname{span}\{a^{(1)}, \ldots, a^{(k)}\} \qquad k = 1, \ldots, n. \tag{4.72}$$

4.3.8 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 (4.71) mit $r_{ij} = \langle a^{(j)}, q^{(i)} \rangle$.

Zur Eindeutigkeit nehmen wir an, es gelte $A = Q_1 R_1 = Q_2 R_2$. Es gilt dann für die Hilfsmatrix $P = Q_2^T Q_1$

$$P = Q_2^{-1} Q_1 = R_2 A^{-1} A R_1^{-1} \tag{4.73}$$

und ebenso $P^T = R_1 R_2^{-1}$. 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

$$PR_1 = Q_2^T Q_1 R_1 = Q_2^T A = Q_2^T Q_2 R_2, \tag{4.74}$$

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 $R_2 R_1^{-1} = \mathbb{I}$ folgt dann $R_1 = R_2$ und

$$Q_1 = A R_1^{-1} = A R_2^{-1} = Q_2. \tag{4.75}$$

□

4.3.9 Definition: Die **Givens-Rotation** Ω_{jk} zum Winkel ϑ bildet ab

$$\Omega_{jk} \colon \mathbb{R}^n \to \mathbb{R}^n \tag{4.76}$$
$$x \mapsto y \tag{4.77}$$

mit

$$y_i = \begin{cases} c x_j + s x_k & i = j \\ -s x_j + c x_k & i = k \\ x_i & \text{sonst} \end{cases} \tag{4.78}$$

mit $c = \cos\vartheta$ und $s = \sin\vartheta$.

4.3.10 Algorithmus (Givens-Rotation):

4.3.11 Algorithmus (QR-Zerlegung mit Givens-Rotation):

4.3.12 Definition (Householder-Reflexion): Ist ein Vektor $w \in \mathbb{R}^n$ gegeben, so beschreibt die Matrix

$$Q_w = \mathbb{I} - 2\frac{w w^T}{w^T w} \tag{4.79}$$

die Abbildung, die einen Vektor y an der Hyperebene senkrecht zu w spiegelt.

4.3.13 Lemma: 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

$$Q_{w^\pm} y = \pm \alpha e_1 = \begin{pmatrix} \pm\alpha \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{4.80}$$

Beweis. Orthogonale Abbildungen sind normerhaltend. Daher muss gelten

$$|\alpha| = \|y\|_2. \tag{4.81}$$

oder $\alpha = \pm\|y\|_2$. Ferner gilt wegen (4.80) für einen geeigneten Reflexionsvektor w:

$$Q_w y = y - 2\frac{ww^T y}{w^T w} = y - 2\frac{w^T y}{w^T w}w = \alpha e_1. \tag{4.82}$$

Daher ist w ein Vielfaches von $y - \alpha e_1$. Da durch die Norm von w geteilt wird, wählen wir

$$w^{\pm} = y \mp \|y\|_2 e_1. \tag{4.83}$$

□

Bemerkung 4.3.14. Zur vermeidung von Auslöschung in der ersten Stelle von w verwendet man in der QR-Zerlegung den Faktor α so dass

$$\operatorname{sign}(\alpha) = -\operatorname{sign}(y_1) \tag{4.84}$$

und damit

$$w = y + \operatorname{sign}(y_1)\|y\|_2 e_1 = \begin{pmatrix} y_1 + \operatorname{sign}(y_1)\|y\|_2 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}. \tag{4.85}$$

4.3.15 Algorithmus (Householder-Reflexion):

4.3.16 Algorithmus (QR-Zerlegung mit Householder-Reflexion):

4.4 Lineare Ausgleichsrechnung

4.4.1. Die Methode der kleinsten Fehlerquadrate führt auf die Minimierungsaufgabe

$$\|Ax - b\|_2 = \min. \tag{4.86}$$

4.4.2 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

$$\|Ax - b\|_2 = \min, \tag{4.87}$$

wenn x Lösung der **Normalengleichungen**

$$A^T Ax = A^T b \tag{4.88}$$

ist. Insbesondere ist die Minimierungsaufgabe eindeutig lösbar, wenn A vollen Rang hat.

Bemerkung 4.4.3. 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.

4.4.4 Definition: Die Konditionszahl einer rechteckigen Matrix maximalen Rangs bezüglich der Operatornorm zur Vektornorm $\|\cdot\|$ ist

$$\mathrm{cond}(A) = \frac{\sup_{\|x\|=1}\|Ax\|}{\inf_{\|x\|=1}\|Ax\|}. \tag{4.89}$$

Die Definition ist konsistent zur Definition für invertierbare, quadratische Matrizen.

4.4.5 Lemma: Für eine Matrix $A \in \mathbb{R}^{m\times n}$ maximalen Rangs mit $m \geq n$ gilt

$$\mathrm{cond}(A^T A) = \mathrm{cond}(A)^2. \tag{4.90}$$

4.4.6 Lemma: Zu jeder Matrix $A \in \mathbb{R}^{m\times n}$ maximalen Rangs mit $m \leq n$ gibt es eine QR-Zerlegung

$$A = QR \tag{4.91}$$

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.

4.4.7 Satz:

$$Rx = Q^T b. \tag{4.92}$$

Beweis. Einsetzen der QR-Zerlegung in die Normalengleichungen ergibt

$$R^T Q^T Q R x = R^T R x = R^T Q^T b. \tag{4.93}$$

Da R^T invertierbar ist, können wir die Inverse von links anwenden und erhalten das Resultat. □

Kapitel 5

Iterationsverfahren

5.1 Grundlagen

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

$$x^{(k+1)} = G(x^{(k)}), \qquad k = 0, 1, 2, \ldots. \tag{5.1}$$

Das Verfahren heißt **konvergent**, wenn gilt $x^{(k)} \to x$.

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

$$\left\| x^{(k+1)} - x \right\| \leq c \left\| x^{(k)} - x \right\|^p \tag{5.2}$$

gilt. Es ist linear konvergent, wenn

$$\left\| x^{(k+1)} - x \right\| \leq c \left\| x^{(k)} - x \right\| \tag{5.3}$$

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

$$\left\| x^{(k+1)} - x \right\| = o\left(\left\| x^{(k)} - x \right\| \right) \tag{5.4}$$

5.1.3 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

$$\|f(x) - f(y)\| \leq \varrho \|x - y\| \qquad \forall x, y \in M. \tag{5.5}$$

5.1.4 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

$$x = f(x). \tag{5.6}$$

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

$$x^* = \operatorname*{argmin}_{x \in \mathbb{R}^n} g(x), \tag{5.7}$$

notwendig

$$\nabla g(x^*) = 0. \tag{5.8}$$

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 einer Funktion $f\colon \mathbb{R}^n \to \mathbb{R}^n$ zu finden, durch die Minimierung des Funktionals $g(x) = \|f(x)\|$ darstellen.

5.2 Nichtlineare Gleichungssysteme

5.2.1 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

$$x^{(k+1)} = x^{(k)} - \left(\nabla f(x^{(k)})\right)^{-1} f(x^{(k)}). \tag{5.9}$$

5.2.2 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

$$\|\nabla f(x) - \nabla f(y)\| \le \gamma \|x - y\| \qquad \forall x, y \in M. \tag{5.10}$$

mit einer Konstanten γ. Dann gilt für alle $x, y \in M$

$$\|f(x) - f(y) - \nabla f(y)(x - y)\| \le \frac{\gamma}{2}\|x - y\|^2. \tag{5.11}$$

Beweis. Wir folgen [Stoer, 1983, Hilfssatz 5.3.1]. Sei $\varphi\colon [0,1] \to \mathbb{R}^n$ die Hilfsfunktion definiert durch

$$\varphi(t) = f\big(y + t(x - y)\big), \tag{5.12}$$

so dass

$$f(x) - f(y) - \nabla f(y)(x - y) = \varphi(1) - \varphi(0) + \varphi'(0) = \int_0^1 (\varphi'(t) - \varphi'(0))\,\mathrm{d}t, \tag{5.13}$$

denn nach der Kettenregel gilt

$$\varphi'(t) = \nabla f\big(y + t(x - y)\big)(x - y). \tag{5.14}$$

Den Integranden schätzen wir ab durch

$$\begin{aligned} \|\varphi'(t) - \varphi'(0)\| &= \|\Big(\nabla f\big(y - t(x - y)\big) - \nabla f(y)\Big)(x - y)\| && (5.15)\\ &\le \|\nabla f\big(y - t(x - y)\big) - \nabla f(y)\|\|(x - y)\| && (5.16)\\ &\le \gamma\, t\|x - y\|^2. && (5.17) \end{aligned}$$

Einsetzen ins Integral ergibt

$$\|f(x) - f(y) - \nabla f(y)(x - y)\| \le \frac{\gamma}{2}\|x - y\|^2. \tag{5.18}$$

□

5.2.3 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

$$\|\nabla f(x) - \nabla f(y)\| \leq \gamma \|x - y\|, \qquad \|\big(\nabla f(x)\big)^{-1}\| \leq \beta. \tag{5.19}$$

Gibt es dann eine Konstante α, so dass

$$\|\big(\nabla f(x^{(0)})\big)^{-1} f(x^{(0)})\| \leq \alpha \tag{5.20}$$

$$h := \frac{\alpha\beta\gamma}{2} < 1 \tag{5.21}$$

$$\overline{B_r(x^{(0)})} \subseteq M, \qquad \text{mit } r = \frac{\alpha}{1-h}, \tag{5.22}$$

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

$$\|x^{(k)} - x^*\| \leq \alpha \frac{h^{2^k - 1}}{1 - h^{2^k}}. \tag{5.23}$$

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 (5.20). Nach der Verfahrensvorschrift können wir abschätzen:

$$\|x^{(k+1)} - x^{(k)}\| = \|\big(\nabla f(x^{(k)})\big)^{-1} f(x^{(k)})\| \tag{5.24}$$

$$\leq \beta \|f(x^{(k)})\| \tag{5.25}$$

$$= \beta \|f(x^{(k)}) - f(x^{(k-1)}) - \nabla f(x^{(k)})(x^{(k)} - x^{(k-1)})\|, \tag{5.26}$$

wobei wir die letzte Zeile aus der Multiplikation der Verfahrensvorschrift mit ∇f gewonnen haben. Hierauf wenden wir nun Lemma 5.2.2 an und bekommen die quadratische Konvergenz, wenn der Abstand zweier Folgenglieder einmal klein genug ist:

$$\|x^{(k+1)} - x^{(k)}\| \leq \frac{\beta\gamma}{2} \|x^{(k)} - x^{(k-1)}\|^2. \tag{5.27}$$

Es bleibt zu zeigen, dass die Folge in $B_r(x^{(0)})$ bleibt. Dazu zeigen wir per Induktion, dass

$$\|x^{(k+1)} - x^{(k)}\| \leq \alpha h^{2^k - 1}. \tag{5.28}$$

Für $k = 0$ folgt $\|x^{(1)} - x^{(0)}\| \leq \alpha$ direkt aus (5.20). Für den Induktionsschritt benutzen wir unsere Konvergenzabschätzung (5.27):

$$\|x^{(k+1)} - x^{(k)}\| \leq \frac{\beta\gamma}{2}\|x^{(k)} - x^{(k-1)}\|^2 \leq \frac{\beta\gamma}{2}(\alpha h^{2^{k-1}-1})^2 = \frac{\alpha\beta\gamma}{2}\alpha h^{2^k-2} = \alpha h^{2^k-1}. \tag{5.29}$$

Nun können wir mit einer Teleskopsumme abschätzen

$$\|x^{(k+1)} - x^{(0)}\| \leq \sum_{j=0}^{k} \|x^{(j+1)} - x^{(j)}\| \tag{5.30}$$

$$\leq \alpha(1 + h + h^3 + h^7 + \cdots + h^{2^k-1}) \tag{5.31}$$

$$< \frac{\alpha}{1-h} = r, \tag{5.32}$$

Aus (5.28) folgt mit dieser Abschätzung, dass $x^{(k)}$ Cauchy Folge ist und durch Grenzübergang die Abschätzung (5.23). □

5.2.4 Definition: Sei $g\colon \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar. Dann definieren wir den Kegel positiven Anstiegs zum Parameter γ als

$$S_\gamma(x) = \big\{ s \in \mathbb{R}^n \big| \|s\| = 1 \ \wedge \ \nabla g(x)s \geq \gamma \|\nabla g(x)\| \big\}. \tag{5.33}$$

5.2.5 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

$$h(x - \mu s) \leq h(x) - \frac{\mu\gamma}{4}\|\nabla g(y)\|. \tag{5.34}$$

5.2.6 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

$$x^{(k+1)} = x^{(k)} - \alpha_k s^{(k)}, \tag{5.35}$$

so dass

$$g\big(x^{(k+1)}\big) \leq g\big(x^{(k)}\big) - \frac{\gamma_k \alpha_k}{4}\|\nabla g(x^{(k)}\|. \tag{5.36}$$

Bemerkung 5.2.7. Lemma Lemma 5.2.5 stellt sicher, dass es in jedem Schritt ein positives α_k gibt, das die Bedingung erfüllt.

5.2.8 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

$$\alpha_k = \operatorname*{argmin}_{\alpha>0} g\big(x^{(k)} - \alpha s^{(k)}\big) \tag{5.37}$$

bestimmt. Danach setze

$$x^{(k+1)} = x^{(k)} - \alpha_k s^{(k)}. \tag{5.38}$$

5.2.9 Satz: Sei $g\colon \mathbb{R}^n \to \mathbb{R}$ und $x^{(0)} \in \mathbb{R}^n$ so gewählt, dass die Menge

$$K = \Big\{ x \in \mathbb{R}^n \;\Big|\; g(x) \le g\big(x^{(0)}\big) \Big\} \tag{5.39}$$

kompakt und g stetig differenzierbar auf einer Umgebung von K ist. Dann besitzt die Folge $\{x^{(k+1)}\}$ des Abstiegsverfahrens mindestens einen Häufungspunkt in K. Gilt zusätzlich in der Umgebung eines Häufungspunkts $\alpha_k \ge \alpha > 0$, so ist er ein stationärer Punkt von g.

5.2.10 Lemma: Sei $f\colon \mathbb{R}^n \to \mathbb{R}^n$ stetig differenzierbar und $g(x) = \|f(x)\|_2^2$. Dann sind die Suchrichtungen

$$s^{(k)} = \frac{d^{(k)}}{\|d^{(k)}\|_2}, \qquad d^{(k)} = \big(\nabla f(x^{(k)})\big)^{-1} f(x^{(k)}) \tag{5.40}$$

Abstiegsrichtungen für $g(x)$ und es gilt

$$s^{(k)} \in S_\gamma\big(x^{(k)}\big), \qquad \gamma = \frac{1}{\operatorname{cond}_2(\nabla f(x^{(k)}))} \tag{5.41}$$

5.2.11 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

$$\left\| f(x^{(k)} - 2^{-j} d^{(k)}) \right\|_2^2 \leq \left\| f(x^{(k)}) \right\|_2^2 - 2^{-j} \frac{1}{4 \operatorname{cond}_2(\nabla f(x^{(k)}))} \left\| f^T \left(x^{(k)} \nabla f(x^{(k)}) \right) \right\|_2 \quad (5.42)$$

3. Setze $x^{(k+1)} = x^{(k)} - 2^{-j} d^{(k)}$

Bemerkung 5.2.12. 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 (5.36) dazu dient, eine hinreichende Kontraktion in der Nähe eines Häufungspunkts sicherzustellen. Die Existenz eines solchen kann bereits aus

$$g\left(x^{k+1}\right) < g\left(x^k\right) \quad (5.43)$$

gefolgert werden. Umgekehrt wird, wenn die Funktion f die Bedingungen des Konvergenzsatzes Satz 5.2.3 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

$$\left\| f\left(x^k - 2^{-j} d^{(k)}\right) \right\|_2^2 < \left\| f\left(x^k\right) \right\|_2^2. \quad (5.44)$$

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.

5.2.13 Algorithmus (Newton-Verfahren mit Schrittweitensteuerung):

5.3 Dünnbesetzte lineare Gleichungssysteme

5.3.1. Bei vielen Aufgaben zur approximativne Simulation physikalischer Vorgänge, zum Beispiel bei der Diskretisierung partieller Differentialgleichungen,

treten sehr große ($n = 10^6$ oder gar $n = 10^9$) Gleichungssysteme auf. Die Matrizen zeichnen sich dadurch aus, dass sie **dünn besetzt** ist, das heißt, in jeder Zeile und Spalte sind nur wenige Einträge von null verschieden. Diese Einträge sind aber nicht in einem schmalen Band angeordnet. Für solche Matrizen ist die Berechnung einer LR- oder QR-Zerelgung unverhältnismäßig aufwendig, da die Zerlegungen die Struktur nicht erhalten und Nullen innerhalb des Bandes auffüllen. Die Multiplikation einer solchen Matrix mit einem Vektor hingegen kann in $\mathcal{O}(n)$ Operationen durchgeführt werden.

Für solche Matrizen leiten wir hier iterative Lösungsverfahren her, die nur auf der Multiplikation beruhen.

5.3.2 Lemma: Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch positiv definit. Ein Vektor $x \in \mathbb{R}^n$ minimiert die Funktion

$$F(x) = \tfrac{1}{2}x^T Ax - x^T b \tag{5.45}$$

genau dann, wenn

$$Ax = b. \tag{5.46}$$

5.3.3 Definition: Das **Verfahren des steilsten Abstiegs** zur Lösung linearer Gleichungssysteme lautet: gegeben ein Vektor $x^{(0)} \in \mathbb{R}^n$, berechne die Folge $\{x^{(k)}\}$ durch die Iterationsvorschrift

$$x^{(k+1)} = x^{(k)} + \alpha_k r^{(k)}, \tag{5.47}$$

wobei $r^{(k)} = b - Ax^{(k)}$ das Residuum von $x^{(k)}$ ist und

$$\alpha_k = \operatorname*{argmin}_{\alpha>0} F\big(x^{(k)} + \alpha r^{(k)}\big). \tag{5.48}$$

5.3.4 Lemma: Der Parameter α_k im Verfahren des steilsten Abstiegs kann durch den Ausdruck

$$\alpha_k = \frac{\big\langle r^{(k)}, r^{(k)}\big\rangle}{\big\langle Ar^{(k)}, r^{(k)}\big\rangle} \tag{5.49}$$

berechnet werden.

Beweis. Durch Einsetzen erhalten wir (unter Weglassen der Indizes k)

$$F(x+\alpha r) = \tfrac{1}{2}\langle A(x+\alpha r), x+\alpha r\rangle - \langle x+\alpha r, b\rangle \tag{5.50}$$

$$= \tfrac{1}{2}\langle Ax, x\rangle + \alpha\langle Ax, r\rangle + \frac{\alpha^2}{2}\langle Ar, r\rangle - \langle x, b\rangle - \alpha\langle r, b\rangle. \tag{5.51}$$

Dieser Ausdruck lässt sich nun leicht ableiten:

$$\frac{d}{d\alpha}F(x+\alpha r) = \langle Ax, r\rangle + \alpha\langle Ar, r\rangle - \langle b, r\rangle, \tag{5.52}$$

$$= \alpha\langle Ar, r\rangle - \langle r, r\rangle \tag{5.53}$$

$$\frac{d^2}{d\alpha^2}F(x+\alpha r) = \langle Ar, r\rangle. \tag{5.54}$$

Da A s.p.d sehen wir, dass eine Nullstelle der ersten Ableitung nur ein Minimum sein kann. Nullsetzen der ersten Ableitung ergibt die behauptete Formel. □

5.3.5 Lemma: Für den Fehler $e^{(k)} = x - x^{(k)}$ im Verfahren des steilsten Abstiegs gilt

$$\left\langle Ae^{(k+1)}, r^{(k)} \right\rangle = 0, \tag{5.55}$$

der Fehler nach einem Schritt ist also bezüglich des A-Skalarprodukts orthogonal zum Residuum des vorigen Schritts,

$$\left\langle e^{(k+1)}, r^{(k)} \right\rangle_A = 0 \tag{5.56}$$

und $x^{(k+1)}$ ist die orthogonale Projektion von x auf den von $r^{(k)}$ aufgespannten Raum.

Beweis. Zunächst bemerken wir:

$$x^{(k+1)} = x^{(k)} + \alpha_k r^{(k)} \qquad \Rightarrow \qquad r^{(k+1)} = r^{(k)} - \alpha_k A r^{(k)} \tag{5.57}$$

und $r^{(k+1)} = Ae^{(k+1)}$. Aus

$$\left\langle r^{(k+1)}, r^{(k)} \right\rangle = \left\langle r^{(k)}, r^{(k)} \right\rangle - \frac{\left\langle r^{(k)}, r^{(k)} \right\rangle}{\left\langle Ar^{(k)}, r^{(k)} \right\rangle} \left\langle Ar^{(k)}, r^{(k)} \right\rangle = 0 \tag{5.58}$$

folgt damit die Orthogonalität. □

5.3.6 Lemma (Kantorowitsch-Ungleichung): Für eine symmetrisch positiv definite Matrix $A \in \mathbb{R}^{n\times n}$ mit minimalen und maximalen Eigenwerten $\lambda_{\min}$ und $\lambda_{\max}$. Dann gilt

$$\frac{\langle Ax, x\rangle\langle A^{-1}x, x\rangle}{\langle x, x\rangle^2} \leq \frac{(\lambda_{\max} + \lambda_{\min})^2}{4\lambda_{\min}\lambda_{\max}}. \tag{5.59}$$

5.3.7 Satz: Für den Fehler $e^{(k)} = x - x^{(k)}$ des Verfahrens des steilsten Abstiegs gilt die Abschätzung

$$\|e^{(k+1)}\|_A \leq \varrho \|e^{(k)}\|_A \tag{5.60}$$

mit

$$\varrho = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} = \frac{\operatorname{cond}_2(A) - 1}{\operatorname{cond}_2(A) + 1} \tag{5.61}$$

$$= 1 - \frac{2}{\operatorname{cond}_2(A)} + \mathcal{O}(\operatorname{cond}_2(A)^{-2}). \tag{5.62}$$

Das Verfahren ist eine Kontraktion und konvergiert mit jedem Startwert $x^{(0)} \in \mathbb{R}^n$ gegen die Lösung $x = A^{-1}b$.

5.3.8 Definition: Das **cg-Verfahren** (Verfahren der konjugierten Gradianten, conjugate gradient method) berechnet zu einer symmetrisch positiv definiten Matrix $A \in \mathbb{R}^{n \times n}$, einer rechten Seite $b \in \mathbb{R}^n$ und einem Startvektor $x^{(0)} \in \mathbb{R}^n$ eine Folge $\{x^{(k)}\}$ nach folgender Vorschrift:

1. Initialisierung:

$$r^{(0)} = b - Ax^{(0)} \tag{5.63}$$

$$p^{(0)} = r^{(0)} \tag{cg0}$$

2. Iteration: für $k = 0, 1, \ldots$ berechne falls $p^{(k)} \neq 0$

$$\alpha_k = \frac{\left\langle r^{(k)}, r^{(k)} \right\rangle}{\left\langle Ap^{(k)}, p^{(k)} \right\rangle} \tag{cg1}$$

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)} \tag{cg2}$$

$$r^{(k+1)} = r^{(k)} - \alpha_k A p^{(k)} \tag{cg3}$$

$$\beta_k = \frac{\left\langle r^{(k+1)}, r^{(k+1)} \right\rangle}{\left\langle r^{(k)}, r^{(k)} \right\rangle} \tag{cg4}$$

$$p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)} \tag{cg5}$$

5.3.9 Lemma: Sei k gewählt so dass $p^{(j)} \neq 0$ für $j \leq k$. Dann gilt für die Vektoren des cg-Verfahrens

$$r^{(k)} = b - Ax^{(k)} \neq 0 \tag{5.64}$$

$$\left\langle r^{(k)}, p^{(j)} \right\rangle = 0 \qquad 0 \leq j < k \tag{cg-o1}$$

$$\left\langle r^{(k)}, p^{(k)} \right\rangle = \left\langle r^{(k)}, r^{(k)} \right\rangle \tag{cg-o2}$$

$$\left\langle Ap^{(j)}, p^{(k)} \right\rangle = 0 \qquad 0 \leq j < k \tag{cg-o3}$$

$$\left\langle r^{(j)}, r^{(k)} \right\rangle = 0 \qquad 0 \leq j < k \tag{cg-o4}$$

Beweis. Für $k = 0$ sind diese Beziehungen offensichtlich erfüllt, einige, weil sie leere Bedingungen sind. Wir nehmen nun also an, dass sie für k bewiesen sind und schließen auf $k+1$. Zunächst ist α_k wohldefiniert, da $p^{(k)} \neq 0$ und A positiv definit. Es gilt

$$b - Ax^{(k+1)} = b - A(x^{(k)} + \alpha_k p^{(k)}) = r^{(k)} - \alpha_k Ap^{(k)} = r^{(k+1)}, \tag{5.65}$$

und damit der erste Teil von (5.64) bewiesen. Bleibt zu zeigen, dass $r^{(k)} \neq 0$. Für $k = 0$ würde dies nach (cg0) bedeuten, dass $p^{(0)} = 0$, was der Annahme widerspricht. Für $k > 0$ ergäbe sich aus (cg5) $p^{(k)} = \beta_{k-1} p^{(k-1)}$ und damit der Widerspruch

$$0 < \left\langle Ap^{(k)}, p^{(k)} \right\rangle = \beta_{k-1} \left\langle Ap^{(k-1)}, p^{(k)} \right\rangle = 0. \tag{5.66}$$

Nun beweisen wir (cg-o1): nach (cg1), (cg3) und (cg-o2) gilt

$$\left\langle r^{(k+1)}, p^{(k)} \right\rangle = \left\langle r^{(k)} - \alpha Ap^{(k)}, p^{(k)} \right\rangle \tag{5.67}$$

$$= \left\langle r^{(k)}, p^{(k)} \right\rangle - \frac{\left\langle r^{(k)}, r^{(k)} \right\rangle}{\left\langle Ap^{(k)}, p^{(k)} \right\rangle} \left\langle Ap^{(k)}, p^{(k)} \right\rangle \tag{5.68}$$

$$= 0. \tag{5.69}$$

Direkter folgt aus (cg-o1) und (cg-o3) für $j < k$

$$\left\langle r^{(k+1)}, p^{(j)} \right\rangle = \left\langle r^{(k)}, p^{(j)} \right\rangle - \alpha \left\langle Ap^{(k)}, p^{(j)} \right\rangle = 0. \tag{5.70}$$

Zum Beweis von (cg-o2) bemerken wir zunächst, dass β_k wohldefiniert ist, da $r^{(k)} \neq 0$. Nun gilt, da wir (cg-o1) schon bewiesen haben

$$\left\langle r^{(k+1)}, p^{(k+1)} \right\rangle = \left\langle r^{(k+1)}, r^{(k+1)} \right\rangle - \beta_k \left\langle r^{(k+1)}, p^{(k)} \right\rangle = \left\langle r^{(k+1)}, r^{(k+1)} \right\rangle. \tag{5.71}$$

Nun zum Beweis von (cg-o3). Nach (cg5) ist

$$\left\langle p^{(k+1)}, Ap^{(j)} \right\rangle = \left\langle r^{(k+1)}, Ap^{(j)} \right\rangle - \beta_k \left\langle p^{(k)}, Ap^{(j)} \right\rangle. \tag{5.72}$$

Für den ersten Term nutzen wir (cg3) und lösen nach $Ap^{(j)}$ auf:

$$\alpha_k Ap^{(j)} = r^{(j)} - r^{(j+1)} = p^{(j)} - \beta_{j-1} p^{(j-1)} - p^{(j+1)} + \beta_j p^{(j)} =: y. \tag{5.73}$$

Für $j < k$ ist $\left\langle p^{(k)}, Ap^{(j)} \right\rangle = 0$ und nach bereits für $k+1$ bewiesenem (cg-o1)

$$\left\langle r^{(k+1)}, Ap^{(j)} \right\rangle = \left\langle r^{(k+1)}, y \right\rangle = 0. \tag{5.74}$$

Es bleibt der Beweis für $j = k$, der die bereits für $k+1$ bewiesenen (cg-o1) und (cg-o2) mit den Definitionen von α_k und β_k verbindet:

$$\begin{aligned}
\alpha_k \left\langle p^{(k+1)}, Ap^{(k)} \right\rangle =& \left\langle r^{(k+1)}, p^{(k)} - \beta_{j-1} p^{(k-1)} - p^{(k+1)} + \beta_j p^{(k)} \right\rangle && (5.75)\\
&+ \alpha_k \beta_k \left\langle p^{(k)}, Ap^{(k)} \right\rangle && (5.76)\\
=& - \left\langle r^{(k+1)}, p^{(k+1)} \right\rangle + \alpha_k \beta_k \left\langle p^{(k)}, Ap^{(k)} \right\rangle && (5.77)\\
=& \frac{\left\langle r^{(k)}, r^{(k)} \right\rangle}{\left\langle p^{(k)}, Ap^{(k)} \right\rangle} \frac{\left\langle r^{(k+1)}, r^{(k+1)} \right\rangle}{\left\langle r^{(k)}, r^{(k)} \right\rangle} \left\langle p^{(k)}, Ap^{(k)} \right\rangle && (5.78)\\
&- \left\langle r^{(k+1)}, r^{(k+1)} \right\rangle. && (5.79)
\end{aligned}$$

Zum Beweis von (cg-o4) lösen wir (cg5) nach $r^{(j)}$ auf und erhalten

$$\left\langle r^{(j)}, r^{(k+1)} \right\rangle = \left\langle p^{(j)} - \beta_{j-1} p^{(j-1)}, r^{(k+1)} \right\rangle = 0. \tag{5.80}$$

□

5.3.10 Korollar: Der Fall $p^{(k)} = 0$ tritt genau dann ein, wenn $x^{(k)}$ die Lösung des Gleichungssystems ist. Damit ist das Verfahren in jedem Punkt außer der Lösung durchführbar.

5.3.11 Korollar: Der Parameter α_k des cg-Verfahrens ist so gewählt, dass

$$\alpha_k = \operatorname*{argmin}_{\alpha > 0} F\big(x^{(k)} + \alpha p^{(k)}\big). \tag{5.81}$$

5.3.12 Korollar: Die Suchrichtungen $p^{(j)}$ für $j = 1, \ldots, k-1$ des cg-Verfahrens bilden eine A-orthogonale (auch konjugiert genannte) Basis des sogenannten **Krylov-Raum**s

$$\mathcal{K}_k(A, r^{(0)}) = \operatorname{span}\{r^{(0)}, Ar^{(0)}, A^2r^{(0)}, \ldots, A^{k-1}r^{(0)}\}. \tag{5.82}$$

Der Fehler $e^{(k)}$ ist A-orthogonal zu $\mathcal{K}_k$ und $x^{(k)}$ ist die Bestapproximation von x in $x^{(0)}+\mathcal{K}_k$ bzgl. der A-Norm. Das Verfahren bricht nach spätestens n Schritten mit der exakten Lösung ab.

5.3.13 Satz: Für den Fehler des cg-Verfahrens gilt bezüglich der Eigenwerte λ_i der Matrix A die Optimalitätseigenschaft

$$\|e^{(k)}\|_A = \min_{\substack{p\in\mathbb{P}_k \\ p(0)=1}} \max_{\lambda_i} |p(\lambda_i)| \|e^{(0)}\|_A \tag{5.83}$$

Beweis. Für den Krylov-Raum der Dimension k gilt

$$\begin{aligned} \mathcal{K}_k(A, r^{(0)}) &= \operatorname{span}\{r^{(0)}, Ar^{(0)}, A^2r^{(0)}, \ldots, A^{k-1}r^{(0)}\} && (5.84)\\ &= \operatorname{span}\{Ae^{(0)}, A^2e^{(0)}, A^3e^{(0)}, \ldots, A^ke^{(0)}\}. && (5.85) \end{aligned}$$

Daher kann jeder Vektor $y \in \mathcal{K}_k(A, r^{(0)})$ mit einem Polynom $q \in \mathbb{P}_{k-1}$ dargestellt werden als

$$y = q(A)r^{(0)} = Aq(A)e^{(0)}. \tag{5.86}$$

Nach den Rekursionsformeln für $r^{(k+1)}$ und $p^{(k+1)}$ folgt, dass beide in $\mathcal{K}_k(A, r^{(0)})$ liegen. Seien nun $q_0, \ldots q_k$ die Polynome, so dass $p^{(j)} = Aq_j(A)e^{(0)}$. Dann gilt

$$e^{(1)} = x - x^{(0)} - \alpha_0 p^{(0)} = e^{(0)} - \alpha_0 Ae^{(0)} \qquad = (\mathbb{I} - \alpha_0 Aq_0(A))e^{(0)} \tag{5.87}$$

$$\vdots$$

$$e^{(k+1)} = x - x^{(k)} - \alpha_k p^{(k)} = e^{(k)} - \alpha_k Aq_k(A)e^{(k)} = (\mathbb{I} - \alpha_k Aq_k(A))e^{(k)}. \tag{5.88}$$

Daher gilt dann:

$$e^{(k+1)} = Q(A)e^{(0)}, \qquad Q(x) = \prod_{j=0}^{k}(1 - \alpha_k x q_k(x)). \tag{5.89}$$

Wir sehen, dass $Q \in \mathbb{P}_{k+1}$ und $Q(0) = 1$. Nun benutzen wir die Spektraldarstellung

$$e^{(0)} = \sum_{i=1}^{n} \eta_i v^i, \qquad Ae^{(0)} = \sum_{i=1}^{n} \lambda_i \eta_i v^i, \tag{5.90}$$

wobei $\lambda_i > 0$ die Eigenwerte von A mit zugehörigen orthonormalen Eigenvektoren v^i sind. Es gilt dann

$$e^{(k+1)} = \sum_{i=1}^{n} Q(\lambda_i) \eta_i v^i. \tag{5.91}$$

Damit reduziert sich jede Komponente um den Faktor $Q(\lambda_i)$ und dich schlechteste um $\max|Q(\lambda_i)|$. Es gilt damit auch

$$\|e^{(k+1)}\|_A^2 = \sum_{i=1}^{n} \lambda_i Q(\lambda_i)^2 \eta_i^2 \tag{5.92}$$

$$\leq \max_i |Q(\lambda_i)| \sum_{i=1}^{n} \lambda_i \eta_i^2 \tag{5.93}$$

$$= \max_i |Q(\lambda_i)| \|e^{(0)}\|_A^2. \tag{5.94}$$

□

5.3.14 Lemma: Unter allen Polynomen $p \in \mathbb{P}_n$ aus der Menge

$$K = \left\{ p \in \mathbb{P}_n \;\middle|\; \max_{x \in [-1,1]} |p(x)| = 1 \right\} \tag{5.95}$$

ist das Tschebyscheff-Polynom T_n dasjenige, das außerhalb des Intervalls $[-1, 1]$ am schnellsten wächst, also

$$|T(x)| \geq p(x) \qquad \forall p \in K, \quad \forall |x| > 1. \tag{5.96}$$

Beweis. Der Beweis verläuft ähnlich zu Satz 3.1.23. Wir führen ihn für $x > 1$, wo gilt $T_n(x) > 0$. Sei $\tilde{p} \in K$ mit $|\tilde{p}(y)| \geq T_n(y)$ for some $y > 1$. Dann gilt mit $\gamma = T_n(y)/\tilde{p}(y)$ und $p(x) = \tilde{p}(x)\gamma$, dass die Differenz $q(x) = p(x) - T_n(x) \in \mathbb{P}_n$ in y eine Nullstelle hat. Ferner gilt

$$\max_{x \in [-1,1]} |p(x)| = \gamma < 1 \tag{5.97}$$

In den $n+1$ Tschebyscheff-Abszissen $\tilde{x}_j \in [-1, 1]$ hat damit $q(x)$ alternierendes Vorzeichen, und nach dem Zwischenwertsatz n Nullstellen im Intervall $(-1, 1)$. Damit hat es insgesamt $n+1$ Nullstellen und es gilt $q \equiv 0$, woraus folgt $p \equiv T_n$ und, da bereits $\|p\|_{\infty;[0,1]} = 1$ gilt, auch $\tilde{p} = p$. □

5.3.15 Korollar: Sei $[a, b]$ ein Intervall und $0 < a$. Dann löst das Polynom

$$\widehat{T}_n(x) = \frac{T_n\left(\frac{a+b-2t}{b-a}\right)}{T_n\left(\frac{a+b}{b-a}\right)} \tag{5.98}$$

die Minimierungsaufgabe

$$\widehat{T}_n(x) = \underset{\substack{p\in\mathbb{P}_n\\ p(0)=1}}{\operatorname{argmin}} \max_{x\in[a,b]} |p(x)|. \tag{5.99}$$

Es gilt

$$\max_{x\in[a,b]} |\widehat{T}_n(x)| = \left(T_n\left(\frac{a+b}{b-a}\right)\right)^{-1} \tag{5.100}$$

5.3.16 Satz: Der Fehler des cg-Verfahrens genügt der Abschätzung

$$\|e^{(k)}\|_A \leq 2\left(\frac{\sqrt{\operatorname{cond}_2(A)}-1}{\sqrt{\operatorname{cond}_2(A)}+1}\right)^k \|e^{(0)}\|_A \tag{5.101}$$

Beweis. Für die Tschebyscheff-Polynome gilt die Darstellung

$$T_k = \frac{1}{2}\left(\left(x-\sqrt{x^2-1}\right)^k + \left(x+\sqrt{x^2-1}\right)^k\right), \qquad |x| \geq 1. \tag{5.102}$$

Es gilt

$$\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}} = \frac{\operatorname{cond}_2(A)+1}{\operatorname{cond}_2(A)-1}. \tag{5.103}$$

Einsetzen in (5.100) mit $a = \lambda_{\min}$ und $b = \lambda_{\max}$ ergibt

$$T_k\left(\frac{\operatorname{cond}_2(A)+1}{\operatorname{cond}_2(A)-1}\right) = \tag{5.104}$$

□

5.4 Abbruchkriterien

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