Numerische Mathematik

Paul Breiding

Vorwort

Dieses Vorlesungsskript entstand während der Vorlesung *Numerische Mathematik* im Sommersemester 2025 an der Universität Osnabrück. Als Grundlage für dieses Skript dienten die Unterlagen von Michael Gnewuch. Zusätzliche Quellen sind [1, 3, 4, 5, 8, 9].

Osnabrück, 16. Juli 2025

Paul Breiding

Inhaltsverzeichnis

1 Grundbegriffe

Ziel der numerischen Mathematik ist die Entwicklung und Analyse von Algorithmen zur computergestützten Lösung mathematischer Probleme. In der Numerik werden „kontinuierliche“ Probleme aus der Linearen Algebra (in $\mathbb{R}$- oder $\mathbb{C}$-Vektorräumen) oder der Analysis studiert. In vielen Situationen lässt sich ein Problem als eine Funktion

$$f : E \to R,$$

wobei $E \subseteq V$ und $R \subseteq W$ offene Teilmengen von Vektorräumen V, W sind. Das “Problem” ist dann die Berechnung der *Ausgabe* $f(x) \in R$ für die *Eingabe* $x \in E$. Beispiele sind die Nullstellenbestimmung von Funktionen, das Lösen linearer Gleichungssysteme, das Auswerten von Integralen oder das Lösen von Differenzialgleichungen.

Kriterien für gute Algorithmen sind:

a) *Genauigkeit*: Die vom Algorithmus ausgegebene Lösung einer Probleminstanz soll die wirkliche Lösung möglichst genau approximieren, also einen möglichst kleinen *Fehler* aufweisen.

b) *Numerische Stabilität*: Kleine Fehler in den Ausgangsdaten sollen nur kleine Fehler im Ergebnis erzeugen. Die Abhängigkeit der Lösung von der Störung der Eingangsdaten wir häufig durch eine Konditionszahl erfasst. (Große Konditionszahl heißt *schlecht konditioniert*, kleine Konditionszahl heißt *gut konditioniert*).

c) *Laufzeit*: Der Algorithmus soll in möglichst kurzer, aber in jedem Fall in praktikabler Laufzeit eine Lösung liefern. Die Laufzeit wird bei der theoretischen Analyse von Algorithmen meist durch die Anzahl der benötigten Operationen beschrieben (um eine Kenngröße zu erhalten, die nur vom Algorithmus und nicht von der genutzten Rechnerarchitektur abhängt).

d) *Einfache Beschreibung*: Einfache Verfahren werden auf Grund ihrer einfachen Implementierung bevorzugt.

In dieser Vorlesung werden wir den Fokus auf diese Kriterien legen. Weitere Kritieren sind:

e) *Speicherbedarf*: Bei großen Datenmengen sind Algorithmen wünschenswert, welche nur wenig Speicher benötigen. Dies erlaubt die Verwendung von schnelleren Speichermedien (z.B. Cache statt Hauptspeicher).

f) *Parallelisierung*: Ist es möglich, durch moderne Architekturen (mehrere Kerne, Vektorrechner, Grafikprozessoren) den Algorithmus zu beschleunigen (vektorwertige statt skalarwertige Operationen; Aufteilen auf mehrere Prozessoren)?

Bemerkung 1.1. „Diskrete“ Probleme, wie sie typischerweise in der Informatik oder der Algebra auftreten, werden normalerweise nicht dazugerechnet. Die Implementierung von Algorithmen fällt ebenfalls nicht in den Gegenstandsbereich der Numerik.

1.1 Gleitkommazahlen

In diesem Abschnitt schauen wir uns an, wie Computer intern Zahlen darstellen und damit rechnen. Die Arithmetik üblicher Computer arbeitet mit zwei verschiedenen Zahlensystemen, und zwar mit ganzen Zahlen und mit *Gleitkommazahlen* (Engl.: „Floating Point Numbers“).

Definition 1.2. Eine *Menge von Gleitkommazahlen* ist charakterisiert durch die folgenden vier Größen:

a) Die Basis $b \in \mathbb{N}$,

b) die Genauigkeit (= Anzahl der Mantissenstellen) $\ell \in \mathbb{N}$,

c) der unteren und der oberen Schranke $\underline{e}$, $\overline{e} \in \mathbb{Z}$, $\underline{e} \le \overline{e}$, für den Exponenten.

Eine *Gleitkommazahl* hat dann die Form

$$g = \pm \,.\, \underbrace{d_1 d_2 \dots d_\ell}_{\text{Mantisse}} \cdot b^p := \pm \left(d_1 b^{-1} + d_2 b^{-2} + \dots + d_\ell b^{-\ell}\right) \cdot b^p, \tag{1.1}$$

mit $d_i \in \{0, 1, \dots, b-1\}$ und Exponent $p \in \{\underline{e}, \underline{e}+1, \dots, \overline{e}\}$.

Eine Gleitkommazahl heißt *normalisiert*, wenn $d_1 > 0$. Zu gegebenem $(b, \ell, \underline{e}, \overline{e})$ definieren wir die *Menge der normalisierten Gleitkommazahlen* durch

$$\mathbb{G} := \{0\} \cup \{g \in \mathbb{R} \,:\, g \text{ hat die Form (1.1) mit } d_1 > 0\}.$$

Wir bezeichnen die größte und die (betragsmäßig) kleinste Gleitkommazahl durch

$$g_{\max} := \max \mathbb{G} \quad \text{und} \quad g_{\min} := \min\{g \in \mathbb{G} \,:\, g > 0\}. \tag{1.2}$$

Die *Maschinengenauigkeit* ist

$$\text{eps} := \frac{1}{2} b^{1-\ell}. \tag{1.3}$$

Üblicherweise verwendet man als Basis $b = 2$ (*Binärsystem*), manchmal auch $b = 16$ (*Hexadezimalsystem*). Im Binärsystem heißt jede verwendete Zahl $d_i \in \{0, 1\}$ ein *Bit*.

Beispiel 1.3. Die Zahl $1/3$ kann im Binärsystem nicht exakt dargestellt werden kann, denn es gilt

$$\frac{1}{3} = \frac{1}{1 - \frac{1}{4}} - 1 = \sum_{k=1}^{\infty} \frac{1}{2^{2k}} = (0.01010101\dots)_2.$$

1.2 Der IEEE Standard

Die Darstellung einer Gleitkommazahl im Rechner ist in der realen Welt architekturabhängig und kann auch bei gleichem $(b, \ell, \underline{e}, \overline{e})$ unterschiedlich sein. Wir nennen die am Computer darstellbaren Zahlen auch *Maschinenzahlen* und bezeichnen die Menge der Maschinenzahlen als $\mathbb{M}$.

Auf realen Computern wird für die Zahldarstellung $\mathbb{M}$ nicht genau unsere idealisierte Darstellung der Gleitkommazahlen $\mathbb{G}$ genutzt, sondern fast ausschließlich der IEEE Standard for Binary Floating-Point Arithmetic for Microprocessor Systems. Dabei steht IEEE für „Institute of Electrical and Electronics Engineers“ und ANSI für „American National Standards Institute“.

Dazu nutzt man die Basis $b = 2$. Es folgt für eine normalisierte Zahl g der Form (1.1), dass $d_1 = 1$. Daher bietet es sich an, eine Darstellung

$$g = \pm\,(1.\,m_1 m_2 \ldots m_\ell)_2 \cdot 2^p$$

bei normalisierten Zahlen zu verwenden und die führende 1 nicht abzuspeichern. Um den Exponenten p festzulegen, kann man eine nichtnegative Binärzahl P verwenden, von der man einen festen Biaswert B abzieht:

$$p = P - B.$$

Besteht P aus r Bits, so setzt man

$$B = 2^{r-1} - 1.$$

Die nicht-normierte Zahl 0 wird mit Hilfe von $P = 0$ dargestellt. Die Konvention ist hier, dass, wenn $P = 0$, dann $g = \pm(0.m_1 m_2 \ldots, m_\ell)_2$ und $p = 2^{r-1} - 2$. Diese Zahlen nennt man *unnormalisiert*. Weitere unnnormalisierte Zahlen erhalten wir für $P = (1.11\ldots1)_2$. In diesem Fall ist $g = \infty$, falls $m_1 = \cdots = m_\ell = 0$, und $g =$NaN (“Not a Number”) sonst. Die kleinste positive normalisierte Zahl ist somit

$$g_{\min} = (1.0\cdots0)_2 \cdot 2^{1-(2^{r-1}-1)} = 2^{-(2^{r-1}-2)}.$$

Die größte unnnormalisierte Zahl kleiner als ∞ ist

$$\begin{aligned} h_{\max} &= (0.1\cdots10)_2 \cdot 2^{0-(2^{r-1}-2)} \\ &= \Big(\sum_{i=1}^{\ell} 2^{-i}\Big) \cdot 2^{-(2^{r-1}-2)} \\ &= (1 - 2^{-\ell}) \cdot 2^{-(2^{r-1}-2)}. \end{aligned}$$

Hier sehen wir den Grund für die spezielle Wahl des Exponenten $p = 2^{r-1} - 2$ für unnormalisierte Zahlen. Für diese Wahl ist $h_{\max}$ die in Maschinenpräzision $\text{eps} = \frac{1}{2}2^{1-\ell} = 2^{-\ell}$ nächste Zahl kleiner als $g_{\min}$.

Das nächste Beispiel illustriert diese Architektur anhand von 6-Bit Zahlen.

Beispiel 1.4. Nehmen wir das konkrete Beispiel der 6-Bit Zahlen zur Basis $b = 2$. Wir benutzen

- 1 Bit für das Vorzeichen,
- 2 Bits für die Mantisse $m = (1.m_1m_2)_2$ und
- 3 Bits für den Exponenten P.

Der Biaswert ist hier $B = 2^2 - 1 = 3$ und der echte Exponent p kann mittels $P - 3$ berechnet werden. Die möglichen Exponenten sind $P \in \{1 = (001)_2, \ldots, (110)_2 = 6\}$. Diese entsprechen also $2^p \in \{2^{-2}, \ldots, 2^3\}$. Zum Beispiel erhalten wir für

$$m = (1.01)_2 \text{ und } P = (010)_2 : \quad (1 + \tfrac{1}{4}) \cdot 2^{2-3} = \tfrac{5}{8} = 0.625.$$

Der Exponent $P = (000)_2$ steht für Zahlen der Form $m = (0.m_1m_2)$. Z.B. gilt für

$$m = (1.01)_2 \text{ und } P = (000)_2 : \quad (0 + \tfrac{1}{4}) \cdot 2^{-2} = \tfrac{1}{16} = 0.0625.$$

Der Exponent $(111)_2$ steht für unendlich, falls $m_1 = m_2 = 0$, und sonst für NaN.

Die Tabelle zeigt alle darstellbaren 6-Bit-Maschinenzahlen mit positivem Vorzeichen.

P / m	$(000)_2$ special	$(001)_2$ 2^{-2}	$(010)_2$ 2^{-1}	$(011)_2$ 2^0	$(100)_2$ 2^1	$(101)_2$ 2^2	$(110)_2$ 2^3	$(111)_2$ special
$(1.00)_2$	0	0.25	0.5	1	2	4	8	∞
$(1.01)_2$	0.0625	0.3125	0.625	1.25	2.5	5	10	NaN
$(1.10)_2$	0.125	0.375	0.75	1.5	3	6	12	NaN
$(1.11)_2$	0.1875	0.4375	0.875	1.75	3.5	7	14	NaN

Betrachten wir nur die normalisierten Gleitkommazahlen, so ist $g_{\min} = 0.25$ die betragsmäßig kleinste darstellbare Zahl und $g_{\max} = 14$ die größte darstellbare Zahl.

Der IEEE754 Standard legt folgende Zahlentypen fest:

a) Einfache Genauigkeit (32 Bit): 1 Vorzeichenbit, 8 Exponentenbits, 23 Mantissenbits. (Zahlenraum etwa 10^{-45} bis 10^{38}, Biaswert $2^7 - 1$, Werte des Exponenten $-126 \leq p \leq 127$.)

b) Doppelte Genauigkeit (64 Bit): 1 Vorzeichenbit, 11 Exponentenbits, 52 Mantissenbits. (Zahlenraum etwa 10^{-323} bis 10^{308}, Biaswert $2^{10} - 1$, Werte des Exponenten $-1022 \leq p \leq 1023$.)

Der IEEE754 Standard liefert auch genaue Vorschriften für Operationen mit Gleitkommazahlen, wie Rundungen, arithmetische Operationen oder Wurzelberechnungen, siehe [7].

1.3 Gleitkommazahlarithmetik

Im Folgenden arbeiten wir mit $\mathbb{G}$ (und nicht mit $\mathbb{M}$).

Für $x, y \in \mathbb{G}$ gilt i.A. $x+y, x-y, x\cdot y, x/y \notin \mathbb{G}$. Um sinvolle arithmetische Operationen für Gleitkommazahlen zu definieren, müssen wir daher eine *Rundungsfunktion* einführen. Dazu erinnern wir uns an die Definition von $g_{\max}$ und $g_{\min}$ aus (1.2).

Definition 1.5. Setze $\mathbb{G}' := \{0\} \cup \{x \in \mathbb{R} : g_{\min} \leq |x| \leq g_{\max}\}$. Die Rundungsfunktion $\mathrm{rd} : \mathbb{G}' \to \mathbb{G}$ ist definiert durch

$$|x - \mathrm{rd}(x)| = \min_{g \in \mathbb{G}} |x - g|;$$

ist das Minimum nicht eindeutig bestimmt, so wählen wir stets die betragsgrößere Zahl.

Wir definieren nun (gerundete) Ersatzoperationen $\oplus$, $\ominus$, $\odot$ und $\oslash$:

$$x \oplus y = \mathrm{rd}(x + y) \quad \text{für alle } x, y \in \mathbb{G} \text{ mit } x + y \in \mathbb{G}'$$

und entsprechend auch für die anderen Grundrechenarten. Die gerundeten Operationen führen Rechenfehler ein. Dabei unterscheiden wir *absolute* und *relative* Fehler.

Definition 1.6. Für alle $x \in \mathbb{R}$ und jede Approximation $\tilde{x} \in \mathbb{R}$ von x heißen

$$|x - \tilde{x}| \quad \text{und, falls } x \neq 0, \quad \left|\frac{x - \tilde{x}}{x}\right|$$

der absolute und der relative Fehler von $\tilde{x}$ bzgl. x.

Der relative Fehler ist normalerweise die interessantere Größe, da er dimensionslos ist und nicht wie der absolute Fehler von der gewählten Maßeinheit abhängt.

Lemma 1.7. *Für alle $x \in \mathbb{G}' \setminus \{0\}$ gilt*

$$\left|\frac{x - \mathrm{rd}(x)}{x}\right| \leq \mathrm{eps};$$

insbesondere existiert ein $\Delta \in \mathbb{R}$ mit $|\Delta| \leq \mathrm{eps}$ und

$$\mathrm{rd}(x) = x(1 + \Delta).$$

Beweis. Gilt $b^{p-1} \leq |x| < b^p$ für ein $p \in \mathbb{Z}$, so haben die beiden Gleitkommazahlen, die x einschließen, den Abstand $b^{p-\ell}$. Somit ergibt sich

$$\left|\frac{x - \mathrm{rd}(x)}{x}\right| \leq \frac{1}{2}\frac{b^{p-\ell}}{|x|} \leq \frac{1}{2}b^{1-\ell} = \mathrm{eps}.$$

□

Aus den vorherigen Überlegungen folgt, dass für alle Gleitkommazahlen $x, y \in \mathbb{G}$ geeignete $\Delta_\oplus, \Delta_\ominus, \Delta_\odot, \Delta_\oslash \in \mathbb{R}$ existieren mit

$$|\Delta_\oplus|, |\Delta_\ominus|, |\Delta_\odot|, |\Delta_\oslash| \leq \text{eps}$$

und

$$\begin{aligned} x \oplus y &= (x+y)(1+\Delta_\oplus), \\ x \ominus y &= (x-y)(1+\Delta_\ominus), \\ x \odot y &= (x \cdot y)(1+\Delta_\odot), \\ x \oslash y &= (x/y)(1+\Delta_\oslash). \end{aligned} \tag{1.4}$$

Bemerkung 1.8.

- Die Ersatzoperationen sind nicht exakt.
- Die Maschinenaddition und -multiplikation sind nicht assoziativ. Zum Beispiel gilt für $b = 10$ und $\ell = 2$ (also $\text{eps} = 10^{1-2}/2 = 0.05$)

  $$(100 \oplus 4) \oplus 4 = 100 \neq 110 = 100 \oplus (4 \oplus 4).$$

  Dieser Effekt kann natürlich noch zu größeren Kuriositäten führen. Sei wie oben $b = 10$ und $\ell = 2$. Möchte man für $N \geq 2$ die endliche Summe

  $$\sum_{n=0}^{N} a_n \quad \text{für } a_0 = 1 \text{ und } 0 \leq a_n < 0.05 \text{ für } n \geq 1$$

  berechnen, so ergibt die aufsteigende Summation der Summanden das Ergebnis 1, egal wie groß die Summe wirklich ist. Eine Abhilfe wäre hier, sukzessive jeweils die beiden kleinsten Zahlen zu einem Zwischenergebnis zu summieren.
- Falls $|x+y| > g_{\max}$, so liefert die Addition einen sogenannten *Überlauf* (Overflow), falls $0 < |x+y| < g_{\min}$, so spricht man von einem *Unterlauf* (Underflow). In beiden Fällen braucht (1.4) nicht mehr zu gelten.

Der problematischte Effekt der Gleitkommaarithmetik ist die sogenannte *Auslöschung*. Das folgende Beispiel illustriert dieses Phänomen.

Beispiel 1.9. a) Im Zehnersystem seien $x = 1.234567$ und $y = -1.234560$. Dies wird nach Transfer in die Gleitkommazahlen ($b = 10$, $\ell = 6$) durch

$$\text{rd}(x) = 1.23457 \quad \text{und} \quad \text{rd}(y) = -1.23456 = y$$

dargestellt. Die Zahlen haben eine relative Genauigkeit von

$$\frac{|\text{rd}(x) - x|}{x} \approx 2 \cdot 10^{-6} \quad \text{und} \quad \frac{|\text{rd}(y) - y|}{y} = 0.$$

Es gilt $x+y = 0.000007 = 7 \cdot 10^{-6}$. Das Ergebnis der Addition $x+y$ am Computer ist aber

$$\mathrm{rd}(x) \oplus \mathrm{rd}(y) = \mathrm{rd}(\mathrm{rd}(x) + \mathrm{rd}(y)) = \mathrm{rd}(1 \cdot 10^{-5}) = 1 \cdot 10^{-5}.$$

Der relative Fehler ist

$$\left| \frac{(\mathrm{rd}(x) \oplus \mathrm{rd}(y)) - (x+y)}{x+y} \right| = \left| \frac{1 \cdot 10^{-5} - 7 \cdot 10^{-6}}{7 \cdot 10^{-6}} \right| = \frac{3}{7} \approx 0.43.$$

Es ist keine Dezimalstelle des Ergebnisses exakt. Kleine Fehler können sich so sehr verstärken.

b) Wir betrachten im Zehnersystem die exakte Rechnung

$$0.1236 + 1.234 - 1.356 = 0.0016 = 0.1600 \cdot 10^{-2}.$$

Gleitkommarechnung mit $b = 10$, $\ell = 4$ liefert

$$(0.1236 \oplus 1.234) \ominus 1.356 = 1.358 \ominus 1.356 = 0.2000 \cdot 10^{-2},$$

d.h. auch hier ist die führende Stelle des berechneten Ergebnisses falsch. Der Rundungsfehler der ersten Addition wird durch die nachfolgende Subtraktion extrem verstärkt.

2 Fehleranalyse

Der Fehler (= Differenz von gesuchter und ausgegebener Lösung) bei numerischen Rechnungen setzt sich meist aus mehreren Fehlerarten zusammen. Beim numerischen Lösen mathematischer Probleme treten i.A. vier verschiedene Fehlerarten auf:

a) *Modellfehler*: Das tatsächliche Problem wird in der Regel mit Hilfe eines vereinfachten mathematischen Modells beschrieben.

b) *Datenfehler*: Das mathematische Modell hängt von Parametern ab, die für das tatsächlich vorliegende Problem zu bestimmen sind. Diese werden häufig durch Messungen ermittelt und (meist unvermeidbare) Messungenauigkeiten führen dementsprechend zu Fehlern.

c) *Verfahrensfehler*: Die Auswirkungen der bei der Konstruktion eines Algorithmus gemachten Vereinfachungen auf das gesuchte Resultat werden als Verfahrensfehler bezeichnet. Üblicherweise unterscheidet man zwei Arten von Verfahrensfehlern:

   - *Diskretisierungsfehler*: Statt kontinuierlicher (unendlicher) Information wird diskrete (endliche) Information verwendet.
   - *Abbruchfehler* (engl.: Truncation Error): Fehler, den man durch den Abbruch eines unendlichen Verfahrens (wie z.B. der Berechnung einer unendlichen Reihe oder einem Iterationsverfahren) begeht.

   Der Verfahrensfehler ist also die Abweichung der durch die gemachten Vereinfachungen bedingten und durch rundungsfehlerfreie Berechnungen gewonnenen Näherungslösung von der exakten Lösung des (i.A. bereits mit Modellfehler und Datenfehlern behafteten) mathematischen Problems.

d) *Rundungsfehler*: Da ein Computer nur endlich viele Maschinenzahlen zur Verfügung hat, können nicht alle Zahlen beliebig genau dargestellt und nicht alle Berechnungen exakt ausgeführt werden.

Die numerische Mathematik beschäftigt sich hauptsächlich mit Diskretisierungs-, Abbruch- und Rundungsfehlern.

Im Folgenden wollen wir verstehen, welchen Effekt Fehler auf die numerische Lösung von Problemen haben. Dazu unterscheiden wir problemspezifische Effekte und algorithmusspezifische Effekte. Erstere verstehen wir mit Hilfe von *Konditionszahlen*. Für Letztere führen wir das Konzept der *Stabilität* eines Algorithmus ein.

Bemerkung 2.1. Bei der Modelierung können numerische Überlegungen aber auch eine wichtige Rolle spielen. Von Interesse ist ebenfalls, wie sich Datenfehler, bedingt durch die Problemstellung oder das gewählte numerische Verfahren, auf das Endergebnis auswirken.

Beispiel 2.2 (Exakter Algorithmus)**.** Das folgende Beispiel ist aus [2, Chapter 9].

Gegeben ist eine Matrix $A \in \mathbb{R}^{2\times 2}$ mit $\det(A) \neq 0$, wir wollen die Inverse Matrix A^{-1} berechnen. Wir betrachten zwei Fälle dieses Problems. Die Fehler werden durch die euklidische Norm gemessen.

(a) Zunächst sei $A = \left(\begin{smallmatrix} 1 & 1 \\ -1 & 1 \end{smallmatrix}\right)$. Wir betrachten diese Matrix als die echten Eingabedaten. Ein kleiner Messfehler ergibt die neuen Eingabedaten $\tilde{A} = \left(\begin{smallmatrix} 1 & 1 \\ -1+\varepsilon & 1 \end{smallmatrix}\right)$, $0 < \varepsilon \ll 1$. Die *exakten* Lösungen A^{-1} und $\tilde{A}^{-1}$ sind dann

$$A^{-1} = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \quad \text{und} \quad \tilde{A}^{-1} = \frac{1}{2-\varepsilon}\begin{pmatrix} 1 & -1 \\ 1-\varepsilon & 1 \end{pmatrix}$$
$$= A^{-1} + \frac{\varepsilon}{2(2-\varepsilon)}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

Vergleicht man die Fehler, so sehen wir, dass $\|A^{-1} - \tilde{A}^{-1}\| \approx \|A - \tilde{A}\|$. Mit anderen Worten, die Fehler in der Eingabe $\|A - \tilde{A}\|$ und in der Ausgabe $\|A^{-1} - \tilde{A}^{-1}\|$ sind für kleine Werte von ε grob gleich und beide sind in der Größenordnung $O(\varepsilon)$.

(b) Die echten Eingabedaten in unserem zweiten Beispiel sind die Matrix $B = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 1+\delta \end{smallmatrix}\right)$, wobei $|\delta| \neq 0$ klein ist. Wir perturbieren die Eingabe, indem wir ε zum unteren linken Eintrag hinzufügen. Die perturbierten Eingabedaten sind $\tilde{B} = \left(\begin{smallmatrix} 1 & 1 \\ 1+\varepsilon & 1+\delta \end{smallmatrix}\right)$. Die Matrizeninversen sind

$$B^{-1} = \frac{1}{\delta}\begin{pmatrix} 1+\delta & -1 \\ -1 & 1 \end{pmatrix} \quad \text{und} \quad \tilde{B}^{-1} = \frac{1}{\delta-\varepsilon}\begin{pmatrix} 1+\delta & -1 \\ -1-\varepsilon & 1 \end{pmatrix}$$
$$= B^{-1} + \frac{\varepsilon}{\delta(\varepsilon-\delta)}\begin{pmatrix} -(1+\delta) & 1 \\ 1+\delta & -1 \end{pmatrix}.$$

Das impliziert $\|B^{-1} - \tilde{B}^{-1}\| \approx \frac{1}{\delta(\varepsilon-\delta)} \cdot \|B - \tilde{B}\|$. Wenn $\varepsilon < \delta$, dann wird der Fehler um einen Faktor von ungefähr δ^{-2} verstärkt, was groß ist. Das Verhalten hier ist anders als zuvor.

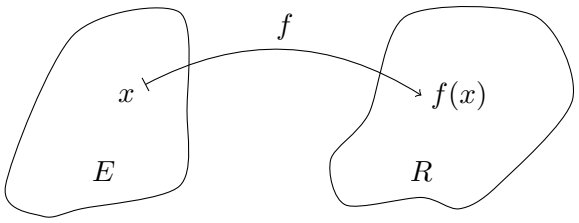
Wir haben einen exakten Algorithmus auf das Problem der Matrixinversion angewendet und dabei erhebliche Unterschiede in der Ausgabe beobachtet. Im ersten Beispiel war die Ausgabe $\tilde{A}^{-1}$ für die perturbierten Daten $\tilde{A}$ nah an der wahren Ausgabe A^{-1}. Andererseits war im zweiten Beispiel die Ausgabe $\tilde{B}^{-1}$ für die perturbierten Daten $\tilde{B}$ weit von der wahren Ausgabe B^{-1} entfernt. Die Schwierigkeit im Lösen des Problems liegt also in den Eingabedaten! Im nächsten Kapitel definieren wir Konditionszahlen, um dieses Phänomen zu quantifizieren.

2.1 Kondition eines Problems

Ein Problem ist modelliert durch eine Abbildung

$$f : E \to R, ,$$

wobei $E \subseteq V$ und $R \subseteq W$ offene Teilmengen von Vektorräumen V und W sind. Das Problem ist es zu gegebener Eingabe $x \in E$ den Wert $f(x) \in R$ zu berechnen. Die Abbildung f nennt man auch *Lösungsoperator* des Problems.



Damit die Ausgabe $f(x)$ interpretierbar ist, sollten kleine Änderungen in x nur kleine Änderungen in $f(x)$ bewirken. Schlecht ist es, wenn kleine Änderungen in x große Änderungen in $f(x)$ bewirken können. Die Konditionszahl von f in x beschreibt die maximale Änderung von $f(x)$ durch Störungen in x.

Um ein sinnvolles Konzept von "Änderung" zu haben, definieren wir auf den Vektorräumen V und W jeweils eine Norm $\|\cdot\|_V$ und $\|\cdot\|_W$. In anderen Worten, V und W sind *normierte Vektorräume*. Die formale Definition der Konditionszahl ist wie folgt.

Definition 2.3. Die *absolute Konditionszahl* κ_{abs} der Probleminstanz (f, x) ist

$$\kappa_{\text{abs}}(f,x) := \lim_{\epsilon \to 0} \sup_{\|\tilde{x}-x\|_V \leq \epsilon} \frac{\|f(\tilde{x}) - f(x)\|_W}{\|\tilde{x} - x\|_V}.$$

Die *relative Konditionszahl* κ_{rel} ist gegeben durch

$$\kappa_{\text{rel}}(f,x) := \lim_{\epsilon \to 0} \sup_{\|\tilde{x}-x\|_V \leq \epsilon} \frac{\|f(\tilde{x}) - f(x)\|_W \cdot \|x\|_V}{\|f(x)\|_W \cdot \|\tilde{x} - x\|_V} = \kappa_{\text{abs}}(f,x) \cdot \frac{\|x\|_V}{\|f(x)\|_W}.$$

Das Problem (f, x) heißt *schlecht gestellt* (Engl.: ill-posed.), falls $\kappa_{\text{abs}}(f, x) = \infty$ (bzw. $\kappa_{\text{rel}}(f, x) = \infty$). Das Problem heißt *schlecht konditioniert*, wenn $\kappa_{\text{abs}}(f, x)$ sehr groß ist und *gut konditioniert*, wenn $\kappa_{\text{abs}}(f, x)$ vernünftig beschränkt ist (bzw. entsprechend für $\kappa_{\text{rel}}(f, x)$).

Die relative Konditionszahl ist im Kontext von Gleitkommazahlen wichtiger, denn die Verwendung von Gleitkommazahlen führt relative Fehler ein.

Im Folgenden lassen wir den Index der Norm weg. Der folgende Satz liefert eine hilfreiche Formulierung der Konditionszahl, falls f stetig differenzierbar ist.

Satz 2.4. *Seien V und W endlich-dimensionale reelle Vektorräume und $f : V \to W$ stetig differenzierbar. Sei $Df(x) : V \to W$ die Ableitung von f an der Stelle x. Dann gilt*

$$\kappa_{\text{abs}}(f, x) = \max_{v \in V:\ \|v\|=1} \|Df(x)\, v\|$$

und

$$\kappa_{\text{rel}}(f, x) = \max_{v \in V:\ \|v\|=1} \frac{\|Df(x)\, v\| \cdot \|x\|}{\|f(x)\|}.$$

Beweis. Sei $\Delta x := \hat{x} - x$ und $\Delta y := f(\hat{x}) - f(x)$. Mit dem Mittelwertsatz der Differenzialrechnung erhalten wir für ein geeignetes ξ mit $\|x - \xi\| \leq \|\Delta x\|$:

$$\Delta y = f(x + \Delta x) - f(x) = Df(\xi)\, \Delta x.$$

Hieraus ergibt sich die absolute Konditionszahl

$$\begin{aligned}
\kappa_{\text{abs}}(f, x) = \lim_{\epsilon \to 0} \sup_{\|\Delta x\| \leq \epsilon} \frac{\|\Delta y\|}{\|\Delta x\|} &= \lim_{\epsilon \to 0} \sup_{\|\Delta x\| \leq \epsilon} \frac{\|Df(\xi)\Delta x\|}{\|\Delta x\|} \\
&= \lim_{\epsilon \to 0} \max_{v \in V:\ \|v\|=1} \|Df(\xi)\, v\| \\
&= \max_{v \in V:\ \|v\|=1} \|Df(x)\, v\|,
\end{aligned}$$

wobei wir in der vorletzten Gleichung benutzt haben, dass die Einheitssphäre in einem endlich-dimensionalen Vektorraum kompakt ist und somit das Supremum durch ein Maximum ersetzt werden kann. In der letzten Gleichung wurde verwendet, dass $\xi \to x$ für $\varepsilon \to 0$ und dass f stetig differenzierbar ist. Die relative Konditionszahl ist dann entsprechend

$$\kappa_{\text{rel}}(f, x) = \kappa_{\text{abs}}(f, x) \cdot \frac{\|x\|}{\|f(x)\|} = \max_{v \in V:\ \|v\|=1} \frac{\|Df(x)\, v\| \cdot \|x\|}{\|f(x)\|}.$$

□

In der Praxis verwendet man in Theorem 2.4 die Jacobi-Matrix $Jf(x)$ anstelle der Ableitung $Df(x)$. (Erinnerung: Die Jacobi Matrix $Jf(x)$ ist die Matrixdarstellung von $Df(x)$ bezüglich einer Basis).

Beispiel 2.5. Sei $f : \mathbb{R} \to \mathbb{R}$ gegeben durch $f(x) = \frac{1-\cos(x)}{x}$. Dann ist

$$f'(x) = \frac{x\,\sin(x) - 1 + \cos(x)}{x^2}, \quad \kappa_{\text{rel}}(f, x) = \frac{|f'(x)| \cdot |x|}{|f(x)|} = \left| \frac{x\,\sin(x) - 1 + \cos(x)}{1 - \cos(x)} \right|.$$

Es ist $\kappa_{\text{rel}}(f, 0) = 1$, denn mit zweifacher Anwendung der Regel von de L'Hospital ergibt sich

$$\lim_{x \to 0} \kappa_{\text{rel}}(f, x) = \lim_{x \to 0} \left| \frac{x\,\sin(x) - 1 + \cos(x)}{1 - \cos(x)} \right| = \lim_{x \to 0} \left| \frac{x \cos x}{\sin x} \right| = \lim_{x \to 0} \left| \frac{\cos x - x \sin x}{\cos x} \right| = 1.$$

Also ist das Problem für $x = 0$ gut konditioniert.

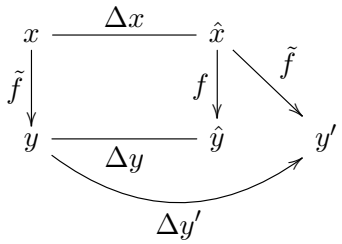
Beispiel 2.6 (Addition)**.** Seien $a \neq 0$ und $f_a : \mathbb{R} \to \mathbb{R}, x \mapsto a + x$. Dann ist $f_a'(x) = 1$. Wir haben also

$$\kappa_{\text{rel}}(f, x) = \frac{|f_a'(x)| \cdot |x|}{|f_a(x)|} = \frac{|x|}{|a + x|}.$$

Insbesondere ist das Problem für $x \approx -a$ schlecht konditioniert. Dies haben wir schon in Beispiel 1.9 beobachten können.

2.2 Stabilität eines Algorithmus

In der Praxis wird ein mathematisches Problem f durch einen Algorithmus gelöst. Wir können den Algorithmus auch als eine Abbildung $\tilde{f}$ verstehen, welche f approximiert. Im diesem Abschnitt verwenden wir folgende Notation.



(2.1)

Definition 2.7. Sei $\tilde{f}$ ein Algorithmus für das Problem f und $x, \hat{x} \in E$ zulässige Eingaben. Sei $y = \tilde{f}(x)$ und $y' = \tilde{f}(\hat{x})$. Der Algorithmus $\tilde{f}$ heißt *numerisch stabil*, falls kleine Störungen $\Delta x = \hat{x} - x$ der Eingabe x nur kleine Störungen $\|\Delta y'\|/\|y\|$ im Resultat bewirken, wobei $\Delta y' = y' - y$.

Stabilität ist eine Eigenschaft von Algorithmen, wohingegen Kondition eine Eigenschaft von Problemen ist (und somit unabhängig vom gewählten Algorithmus).

Siehe Beispiel 2.2, wo eine Störung in den Eingabedaten große Änderungen in den Ausgabedaten verursacht hat, unabhängig vom gewählten Algorithmus. Das nächste Beispiel zeigt ein Problem und zwei Algorithmen mit verschiedenen Stabilitätseigenschaften.

Beispiel 2.8 (Quadratische Gleichungen)**.** Die quadratische Gleichung $x^2 - 2px + q = 0$ mit $p, q > 0$ und $p^2 > q$ hat zwei reelle Lösungen $z = p + \sqrt{p^2 - q}$ und $y = p - \sqrt{p^2 - q}$. Wir betrachten das Problem y auszurechnen. Die Lösungsabbildung ist

$$f : U := \{(p, q) \in \mathbb{R}^2 \mid p^2 > q\} \to \mathbb{R}, \quad (p, q) \mapsto p - \sqrt{p^2 - q}.$$

Ist $p \gg q$, so führt ein naives Aussrechnen der Lösung $y = p - \sqrt{p^2 - q}$ zu Auslöschung. Besser ist es nach den Satz von Vieta zu benutzen:

$$z = p + \sqrt{p^2 - q}, \quad y = \frac{q}{z}.$$

Dies führt zu zwei Algorithmen $\tilde{f}_{\text{naiv}}$ und $\tilde{f}_{\text{Vieta}}$ um f zu berechnen.

Für unser Beispiel betrachten wir die Eingabe $(p, q) = (2 \cdot 10^6, 1)$ zusammen mit der Störung $(\hat{p}, \hat{q}) = (p, q) + (10^{-4}, 10^{-4})$. Der relative Fehler in der Euklidischen Norm ist

$$\frac{\|(p, q) - (\hat{p}, \hat{q})\|}{\|(p, q)\|} \approx 7.0710^{-11}.$$

Wir berechnen y in IEEE754 64-bit Gleitkommazahlarithmetik mit Hilfe der zwei Algorithmen. Wir erhalten:

$$\text{für } \tilde{f}_{\text{naiv}}\text{: } \frac{\|y - \hat{y}\|}{\|y\|} \approx 0.000931; \qquad \text{für } \tilde{f}_{\text{Vieta}}\text{: } \frac{\|y - \hat{y}\|}{\|y\|} \approx 0.000099.$$

Der Vorwärtsfehler für $\tilde{f}_{\text{Vieta}}$ ist ungefähr um den Faktor 10 kleiner.

Der Begriff "klein" in der Definition von numerischer Stabilität ist nicht formal definiert. In der Literatur wird oft gefordert, dass Konstanten c, r existiert, so dass

$$\frac{\|\hat{y} - y\|}{\|y\|} \leq \kappa(f, x) \cdot c \cdot r$$

für alle $\tilde{x}$ mit $\|\hat{x} - x\|/\|x\| \leq r$. Hierbei ist $\kappa_{\text{rel}}(f, x)$ die Konditionszahl des Problems f an der Stelle x. Eine übliche Wahl ist $r = \text{eps}$, die Maschinenpräzision aus (1.3).

Die Eigenschaft in Definition 2.7 heißt auch *vorwärts stabil* und der Fehler $\|\hat{y} - y\|/\|y\|$ heißt *Vorwärtsfehler*. Im Gegensatz dazu definiert man auch *rückwärts stabil*.

Definition 2.9. Der Algorithmus $\tilde{f}$ heißt *rückwärts stabil*, falls für jede Eingabe $x \in E$ eine Eingabe $\hat{x}$ mit $\tilde{f}(x) = f(\hat{x})$ existiert, so dass der *Rückwärtsfehler* $\|\Delta x\|/\|x\|$ klein ist. (In der Notation von (2.1) gilt also $y = \hat{y}$.)

Die Motivation für diese Definition ist, dass ein rückwärts stabiler Algorithmus wie ein exakter Algorithmus auf gestörten Daten funktioniert. Unter dieser Perspektive fokussiert sich die Fehleranalyse auf die Eingabedaten.

Beispiel 2.10. Wir betrachten das Problem $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, (a, b) \mapsto y = a^T b$. Um f zu berechnen, verwenden wir folgenden Algorithmus. Zunächst berechnen wir die n Produkte $a_i b_i$ in Gleitkommazahlarithmetik und erhalten $a_i b_i(1+\delta_{i,0})$, wobei $|\delta_{i,0}| \leq \text{eps}$. Dann addieren wir rekursiv in Zweierpaaren. Dies erfordert $m := \lceil \log_2(n) \rceil$ Additionen in Gleitkommazahlarithmetik. Insgesamt erhalten wir als Ausgabe unseres Algorithmus

$$\tilde{y} = a_1 b_1 \prod_{j=0}^{m} (1 + \delta_{1,j}) + a_2 b_2 \prod_{j=0}^{m} (1 + \delta_{2,j}) + \cdots + a_n b_n \prod_{j=0}^{m} (1 + \delta_{n,j}),$$

wobei $|\delta_{i,j}| \leq \text{eps}$ für alle i, j. Sei nun $\tilde{a}_i := a_i \prod_{j=0}^{m} (1 + \delta_{i,j})$. Dann gilt

$$\tilde{y} = \tilde{a}^T b = f(\tilde{a}, b).$$

D.h., $\tilde{y}$ ist die korrekte Ausgabe zur Eingabe $(\tilde{a}, b)$. Der Rückwärtsfehler ist

$$\frac{\|\tilde{a} - a\|_\infty}{\|a\|_\infty} = \frac{\max_i |a_i \cdot (-1 + \prod_{j=0}^{m}(1 + \delta_{i,j}))|}{\max_i |a_i|} \leq \big| -1 + (1 + \text{eps})^m \big|.$$

Weiterhin gilt

$$0 < -1 + (1 + \text{eps})^m = \sum_{k=1}^{m} \binom{m}{k} \text{eps}^k \leq \sum_{k=1}^{m} (m \cdot \text{eps})^k = \frac{m \cdot \text{eps} - (m \cdot \text{eps})^{k+1}}{1 - m \cdot \text{eps}}.$$

Falls $m \cdot \text{eps} \ll 1$, erhalten wir somit die Abschätzung

$$\frac{\|\tilde{a} - a\|_\infty}{\|a\|_\infty} < \frac{m \cdot \text{eps}}{1 - m \cdot \text{eps}}.$$

In diesem Fall ist unser Algorithmus zum Berechnen von f also rückwärts-stabil.

Für 64-bit Gleitkommaoperationen im IEEE754 Standard ist $\text{eps} = 2^{-53}$. Da $n \leq 2^m$, ist die Annahme $m \cdot \text{eps} \ll 1$ also keine wesentliche Einschränkung.

Die Konzepte von Vorwärts– und Rückwärtsstabilität werden in der nächsten Definition kombiniert.

Definition 2.11. Der Algorithmus $\tilde{f}$ heißt *gemischt vorwärts-rückwärts stabil*, falls für alle Eingaben $x \in E$ eine Eingabe $\hat{x}$ existiert, so dass sowohl der Rückwärtsfehler $\|\Delta x\|/\|x\|$ als auch der Vorwärtsfehler $\|\Delta y\|/\|y\|$ klein sind, wobei

$$\Delta y := f(\hat{x}) - \tilde{f}(x) = \hat{y} - y.$$

(Für mehr Details verweisen wir auf [5, Kapitel 1.5].)

Algorithmen, die gemischt vorwärts-rückwärts stabil, nennt man meist einfach stabil.

Sei nun $\tilde{f}$ ein Algorithmus für f Dann gilt für die Vorwärtsfehler $\Delta y = \hat{y} - y$ und $\Delta y' = y' - y$ nach der Dreiecksungleichung:

$$\|\Delta y'\| \leq \|\Delta y\| + \|y' - \hat{y}\|.$$

Angenommen $\tilde{f}$ ist ein stabiler Algorithmus, dann ist $\|\Delta y\|/\|y\|$ klein. Wenn dann auch

$$\frac{\|y' - \hat{y}\|}{\|y\|} = \frac{\|\tilde{f}(\hat{x}) - f(\hat{x})\|}{\|\tilde{f}(x)\|}$$

klein ist, ist $\tilde{f}$ vorwärts stabil im Sinne von Definition 2.7. Falls die Konditionszahl von $\tilde{f}$ klein ist, gilt $\|\tilde{f}(x)\| \approx \|\tilde{f}(\hat{x})\|$ und in diesem Fall ist der obige Ausdruck approximativ $\|\tilde{f}(\hat{x}) - f(\hat{x})\|/\|\tilde{f}(\hat{x})\|$. Einen Algorithmus $\tilde{f}$, für den Letzteres klein ist, nennt man *konsistent* oder *genau*. Das folgende Beispiel zeigt einen Algorithmus, der nicht genau ist.

Beispiel 2.12. In Beispiel 2.5 haben wir gezeigt, dass das Problem

$$y = f(x) = \frac{1 - \cos x}{x}$$

für $x = 0$ gut konditioniert ist. Die naive Berechnung von y, wobei erst $1 - \cos x$ ausgerechnet und dann das Ergebnis durch x geteilt wird, ist jedoch für kleine x nicht genau: Wir betrachten ein Beispiel mit 6 Stellen Genauigkeit beim Rechnen. Sei $x = 1.234 \cdot 10^{-3}$. Dann ist $\cos(x) \approx 0.999999$ und $1 - \cos(x) \approx 0.000001$. Dies gibt $\tilde{f}(x) \approx 0.810373 \cdot 10^{-3}$. Das exakte Ergebnis ist jedoch $f(x) = 0.616999921\ldots \cdot 10^{-3}$. Bereits in der ersten signifikanten Dezimalstelle ist ein Fehler!

Besser ist die Reihenentwicklung

$$\frac{1 - \cos x}{x} = \frac{1}{x}\left(\sum_{k=1}^{\infty} (-1)^{k-1} \frac{x^{2k}}{(2k)!}\right) = \frac{x}{2}\left(1 - \frac{x^2}{12} \pm \cdots\right).$$

Sei $|x| \leq 1$. Da die Reihe alternierend und die Nullfolge $(\frac{x^{2k}}{(2k)!})_{k \in \mathbb{N}_0}$ monoton fallend ist, kann der Fehler der Approximation der Reihe durch eine endliche Partialsumme immer durch den Betrag des ersten weggelassenen Terms abgeschätzt werden. Approximieren wir also $(1 - \frac{x^2}{12} \pm \cdots)$ durch 1, so erhalten wir für $|x| < 10^{-4}$ daher einen Fehler von höchstens $\frac{1}{12}(10^{-4})^2 < 10^{-9}$. Das ergibt einen relativen Fehler von höchstens

$$\frac{\frac{1}{12}(10^{-4})^2}{1 - \frac{1}{12}(10^{-4})^2} < 10^{-9},$$

d.h. unser Ergebnis ist auf mindestens 8 Stellen exakt. In unserem konkreten Beispiel mit $x = 1.234 \cdot 10^{-3}$ ergibt die Approximation von $f(x)$ durch $\tilde{f}(x) = \frac{\hat{x}}{2}$ das Ergebnis $0.617 \cdot 10^{-3}$.

2.3 Matrixnormen

In diesem Abschnitt studieren wir Normen definiert auf Vektorräume linearer Operatoren. Zunächst aber fokussieren wir uns auf Normen in allgemeinen n-dimensionalen Vektorräumen. Nach Wahl einer Basies ist ein solcher Vektorraum kanonisch isomorph zu $\mathbb{K}^n$. Für uns sind die folgenden Normen auf $\mathbb{K}^n$ interessant:

a) *Summennorm*: $\|x\|_1 := \sum_{j=1}^n |x_j|$.

b) *Euklidische Norm*: $\|x\|_2 := \sqrt{\langle x, x\rangle} = \sqrt{\sum_{j=1}^n |x_j|^2}$.

c) *Maximumsnorm*: $\|x\|_\infty := \max_{j=1,\ldots,n} |x_j|$.

d) *p-Norm* für $1 \leq p < \infty$: $\|x\|_p := \left(\sum_{j=1}^n |x_j|^p\right)^{\frac{1}{p}}$.

Das standard innere Produkt auf $\mathbb{K}^n$ ist

$$\langle x, y\rangle = \sum_{i=1}^{n} \overline{x_i} \cdot y_i. \tag{2.2}$$

Die Cauchy-Schwarz-Ungleichung ist

$$\big|\langle x, y\rangle\big| \leq \|x\|_2 \cdot \|y\|_2.$$

Lemma 2.13. *Alle Normen auf $\mathbb{K}^n$ sind äquivalent, d.h. sind $\|\cdot\|$ und $|||\cdot|||$ Normen auf $\mathbb{K}^n$, so gibt es Konstanten $c_1, c_2 > 0$ mit $c_1|||x||| \leq \|x\| \leq c_2|||x|||$ für alle $x \in \mathbb{K}^n$.*

Beweis. Es genügt den Fall $|||\cdot||| = \|\cdot\|_1$ zu betrachten, denn äquivalent sein ist eine transitive Eigenschaft von Normen. Bezeichnen $e_1, \ldots, e_n$ die Standardeinheitsvektoren auf $\mathbb{K}^n$, so gilt mit der Dreiecksungleichung für $x = (x_1, \ldots, x_n) = x_1e_1 + \ldots + x_ne_n \in \mathbb{K}^n$

$$\|x\| \leq \sum_{j=1}^{n} |x_j| \cdot \|e_j\| \leq \big(\max_{j=1,\ldots,n} \|e_j\|\big)\, \|x\|_1 =: c_2\|x\|_1.$$

Es bleibt noch zu zeigen, dass es ein $c_1 > 0$ gibt mit $c_1\|x\|_1 \leq \|x\|$. Dazu beobachten wir zunächst, dass $\|\cdot\|_1 : \mathbb{K}^n \to \mathbb{R}$ als Summe stetiger Funktionen stetig ist. Wir zeigen, dass auch $\|\cdot\|$ stetig ist (bzgl. der von $\|\cdot\|_1$ induzierten Topologie). Nach Dreiecksungleichung ist $\big|\|x\| - \|y\|\big| \leq \|x - y\|$. Dies impliziert mit Hilfe des ersten Teils des Beweises, dass $\big|\|x\| - \|y\|\big| \leq c_1\|x - y\|_1$. Also $0 \leq \lim_{y\to x} \big|\|x\| - \|y\|\big| \leq \lim_{y\to x} c_1\|x - y\|_1 = 0$. Es folgt, dass $\|\cdot\|$ stetig ist.

Die Einheitssphäre $S := \{x \in \mathbb{K}^n : \|x\|_1 = 1\}$ ist beschränkt und abgeschlossen, also kompakt. Daher nimmt die stetige Funktion $\|\cdot\|$ auf S ein Minimum $c_1 \geq 0$ an. Es gilt $c_1 \neq 0$, da $0 \notin S$. Folglich gilt $\|x\| \geq c_1 > 0$ für alle x mit $\|x\|_1 = 1$. Für beliebiges $x \in \mathbb{K}^n \backslash \{0\}$ gilt somit $\|\frac{x}{\|x\|_1}\| \geq c_1$, d.h. $\|x\| \geq c_1\|x\|_1$. □

Bemerkung 2.14. Lemma 2.13 zeigt, dass alle Normen auf dem endlich-dimensionalen Raum $\mathbb{K}^n$ äquivalent sind, d.h. sich gegenseitig mit Hilfe von geeigneten Konstanten abschätzen lassen. Dies impliziert, dass alle von Normen induzierten Topologien in $\mathbb{K}^n$ gleich sind.

Allerdings sind die konkreten Werte der Konstanten für uns von großer Bedeutung, so gilt z.B. für alle $x \in \mathbb{K}^n$, dass

$$\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \leq n\,\|x\|_\infty$$

Bemerkung 2.15. Seien $1 \leq p \leq q \leq \infty$.

a) Dann gilt $\|x\|_q \leq \|x\|_p$ für alle $x \in \mathbb{R}^n$.

b) Da nach Lemma 2.13 alle Normen auf dem $\mathbb{R}^n$ äquivalent sind, findet man auch eine Konstante $c_{p,q} \geq 1$ derart, dass $\|x\|_p \leq c_{p,q}\|x\|_q$ für alle $x \in \mathbb{R}^n$.

Nun gehen wir einen Schritt weiter und betrachten Normen auf einem Vektorraum V mit der zusätzlichen Struktur, dass die Elemente aus V lineare Operatoren sind. Für zwei normierte $\mathbb{K}$-Vektorräume X und Y sei dazu

$$V = L(X, Y) := \{f : X \to Y \,:\, f \text{ ist stetig und linear}\}.$$

Definition 2.16. Für $T \in V = L(X, Y)$ definiert man die *Operatornorm*

$$\|T\|_{\text{op}} := \sup_{\|x\|_X \leq 1} \|Tx\|_Y = \sup_{x \in X, x \neq 0} \frac{\|Tx\|_Y}{\|x\|_X}.$$

(Die zweite Formel gilt nur, wenn X nicht der triviale Raum $\{0\}$ ist.)

Lemma 2.17. $\|\cdot\|_{\text{op}}$ *ist eine Norm auf* V.

Beweis. Da $T \in V$ stetig ist, ist das Bild des Einheitsballs unter T beschränkt. Somit ist $\|T\|_{\text{op}} \in \mathbb{R}$ wohldefiniert. Nach Definition gilt für alle $T \in V$, dass $\|T\|_{\text{op}} \geq 0$. Außerdem ist $\|T\|_{\text{op}} = 0$ genau dann, wenn $Tx = 0$ für alle $x \in X$, also $T = 0$. Weiterhin gilt für $\lambda \in \mathbb{K}$, dass $\|\lambda T\|_{\text{op}} = \sup_{\|x\|_X \leq 1} \|\lambda Tx\|_Y = |\lambda| \sup_{\|x\|_X \leq 1} \|Tx\|_Y = |\lambda| \, \|T\|_{\text{op}}$. Zuletzt beweisen wir die Dreiecksungleichung:

$$\|T + S\|_{\text{op}} = \sup_{\|x\|_X \leq 1} \|Tx + Sx\|_Y \leq \sup_{\|x\|_X \leq 1} \|Tx\| + \|Sx\|_Y \leq \|T\|_{\text{op}} + \|S\|_{\text{op}}.$$

Dies zeigt, dass die Operatornorm tatsächlich eine Norm ist. □

Aus der Definition der Operatornorm folgt direkt, dass

$$\|Tx\|_Y \leq \|T\|_{\text{op}} \cdot \|x\|_X \quad \text{für alle } x \in X. \tag{2.3}$$

Jede lineare stetige Abbildung $T : X \to Y$ ist also bereits *Lipschitz-stetig*, d.h. sie erfüllt

$$\|Tx - Ty\|_Y \leq C\|x - y\|_X \quad \text{für alle } x, y \in X \tag{2.4}$$

mit einer *Lipschitz-Konstanten* $C > 0$, und man kann zeigen, dass $C := \|T\|_{\text{op}}$ die kleinste (also optimale!) Lipschitz-Konstante ist, für die (2.4) gilt.

Da man jede Matrix $A \in \mathbb{K}^{m \times n}$ via $f(x) := Ax$ auch als Element aus $L(\mathbb{K}^n, \mathbb{K}^m)$ auffassen kann, können wir mit Hilfe von Operatornormen Normen für Matrizen definieren. Je nachdem mit welchen Normen wir $\mathbb{K}^n$ und $\mathbb{K}^m$ ausstatten, erhalten wir verschiedene Normen für die Matrizen. Wir nennen die so erhaltenen Normen auch *induzierte Normen*. Sind $\mathbb{K}^n$ und $\mathbb{K}^m$ beide mit der demselben Typ Norm $\|\cdot\|$-Norm versehen (also z.B. beide mit der Euklidischen Norm), so bezeichnen wir die induzierter Operatornorm ebenfalls mit $\|\cdot\|$. In dem Fall folgt aus (2.3) die wichtige Abschätzung

$$\|Ax\| \leq \|A\|\|x\| \quad \text{für alle } x \in \mathbb{K}^n.$$

Im Fall $n = m$ gilt $\|\mathrm{I}\| = 1$ (I=Identität), d.h. die induzierte Norm der Einheitsmatrix ist dann immer gleich 1.

Definition 2.18. Für eine quadratische Matrix $B \in \mathbb{K}^{n\times n}$ definieren wir

$$\rho(B) = \max\{|\lambda| : \lambda \in \mathbb{C} \text{ ist Eigenwert von } B\}$$

und nennen $\rho(B)$ den *Spektralradius* von B.

Definition 2.19. Wichtige Beispiele für induzierte Matrixnormen sind wie folgt:

a) *Zeilensummennorm*: Sind $\mathbb{K}^n$ und $\mathbb{K}^m$ mit der Maximumsnorm $\|\cdot\|_\infty$ versehen, so ist die induzierte Matrixnorm

$$\|A\|_\infty = \max_{j=1,\dots,m} \sum_{k=1}^{n} |A_{jk}|.$$

Dies ist die maximale Zeilensumme.

Um zu beweisen, dass $\|A\|_\infty$ die maximale Zeilensumme ist, betrachten wir

$$\|Ax\|_\infty = \max_j |(Ax)_j| = \max_j \left|\sum_k A_{jk}x_k\right| \leq \max_j \Big(\sum_k |A_{jk}|\Big)\|x\|_\infty.$$

Hieraus folgt, dass $\|Ax\|_\infty/\|x\|_\infty \leq \max_j \sum_k |A_{jk}|$, also $\|A\|_\infty \leq \max_j \sum_k |A_{jk}|$. Dies zeigt die Abschätzung nach oben. Für die Abschätzung nach unten sei j_0 die Zeile mit maximaler Zeilensumme. Dann gilt für $x = (\overline{A_{j_0,k}} \,/\, |A_{j_0,k}|))_k$, dass $\|x\|_\infty = 1$. Wir erhalten $\|A\|_\infty \geq \|Ax\|_\infty \geq |(Ax)_{j_0}| = \max_j \sum_k |A_{jk}|$.

b) *Spaltensummennorm*: Sind $\mathbb{K}^n$ und $\mathbb{K}^m$ mit der Summennorm $\|\cdot\|_1$ versehen, so ist die induzierte Matrixnorm

$$\|A\|_1 = \max_{k=1,\dots,n} \sum_{j=1}^{m} |A_{jk}|.$$

Dies ist die maximale Spaltensumme.

Die folgende Ungleichung zeigt die Abschätzung nach oben:

$$\begin{aligned}\|Ax\|_1 &= \sum_j |(Ax)_j| = \sum_j \left|\sum_k A_{jk}x_k\right| \leq \sum_j \sum_k |A_{jk}||x_k| \\ &\leq \sum_k \Big(\sum_j |A_{jk}|\Big)|x_k| \leq \Big(\max_k \sum_j |A_{jk}|\Big)\|x\|_1.\end{aligned}$$

Ist k_0 die Spalte mit maximaler Spaltensumme, so ergibt der Vektor $x = e_{k_0}$ (Standardvektor mit 1 bei Position k_0) die Abschätzung nach unten.

c) *Spektralnorm*: Sind $\mathbb{K}^n$ und $\mathbb{K}^m$ mit der Euklidischen Norm $\|\cdot\|_2$ versehen, so ist die induzierte Matrixnorm

$$\|A\|_2 = \sqrt{\rho(A^*A)}, \tag{2.5}$$

wobei $\rho(A)$ den *Spektralradius* von A bezeichnet (siehe Definition 2.18). Der Beweis von (2.5) folgt weiter unten in Satz 2.24.

d) *Frobenius-Norm*: Man kann auf $\mathbb{K}^{m\times n}$ auch folgende Norm definieren:

$$\|A\|_{\mathrm{F}} := \sqrt{\sum_{j,k} |A_{jk}|^2}.$$

Dies ist jedoch keine von einer Norm induzierte Operatornorm; im Fall $n = m > 2$ erkennt man das daran, dass die Norm der Einheitsmatrix ungleich 1 ist.

Beachte, dass die Matrix $B := A^*A$ in (2.5) nur nichtnegative reelle Eigenwerte hat. Da B hermitesch ist (d.h. $B^* = B$), hat B nur reelle Eigenwerte. Sei nun $\lambda \in \mathbb{R}$ ein Eigenwert von B mit Eigenvektor V, dann gilt $\lambda \cdot \langle v, v\rangle = \langle v, Bv\rangle = \langle Av, Av\rangle \geq 0$, also $\lambda \geq 0$. hermitesche Matrizen, deren Eigenwerte alle nichtnegativ sind, nennen wir *positiv semidefinit*. Sind alle Eigenwerte sogar echt positiv, nennen wir B *positiv definit*.

Satz 2.20 (Submultiplikativität der induzierten Matrixnorm)**.** *Sei $\|\cdot\|$ eine induzierte Matrixnorm. Dann gilt*

$$\|AB\| \leq \|A\|\|B\|.$$

Beweis. Dies folgt aus $\|ABx\| \leq \|A\|\|Bx\| \leq \|A\|\|B\|\|x\|$. □

Lemma 2.21. *Sei $\|\cdot\|$ eine Norm auf $\mathbb{K}^n$. Dann gilt für die induzierte Matrixnorm immer $\rho(A) \leq \|A\|$ für alle $A \in \mathbb{K}^{n\times n}$.*

Beweis. Sei λ ein Eigenwert von A zum Eigenvektor x, so ist $|\lambda|\cdot\|x\| = \|Ax\| \leq \|A\|\|x\|$. Also folgt $|\lambda| \leq \|A\|$. Da dies für alle Eigenwerte gilt, folgt $\rho(A) \leq \|A\|$. □

Wir wollen im Folgenden (2.5) und weitere Eigenschaften der oben genannten Normen beweisen. Dazu brauchen wir die Definition der orthogonalen/unitären Matrix.

Definition 2.22. Eine Matrix $Q \in \mathbb{K}^{n\times n}$ heißt unitäre Matrix (bzw. für $\mathbb{K} = \mathbb{R}$ orthogonale matrix), falls $Q^*Q = \mathrm{I}$.

Es gilt dann $\|Qx\|_2 = \|x\|_2$ für alle $x \in \mathbb{K}^n$, d.h. unitäre Matrizen sind *längenerhaltend* für die Euklidische Norm. Dies folgt aus

$$\|Qx\|_2^2 = \langle Qx, Qx\rangle = \langle Q^*Qx, x\rangle = \langle x, x\rangle = \|x\|_2^2,$$

wobei $\langle\cdot,\cdot\rangle$ das innere Produkt aus (2.2) ist. Hieraus folgt unmittelbar $\|Q\|_2 = 1$.

Eine zentrale Zerlegung von Matrizen durch unitäre/orthogonale Matrizen ist die *Singulärwertzerlegung*.

Satz 2.23 (Die Singulärwertzerlegung). *Es sei $A \in \mathbb{K}^{m\times n}$ eine Matrix mit $m \geq n$. Dann existieren unitäre Matrizen $U \in \mathbb{K}^{m\times m}$ und $V \in \mathbb{K}^{n\times n}$ (bzw. im Fall $\mathbb{K} = \mathbb{R}$ orthogonale Matrizen) und eindeutig bestimmte $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$, genannt Singularwärte von A, mit*

$$A = U \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \\ 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{pmatrix} V^*.$$

Insbesondere ist, falls $m = n$, die Matrix A invertierbar genau dann, wenn $\sigma_n > 0$.

Beweis. Sei $B := A^*A \in \mathbb{K}^{n\times n}$. Dann ist B positiv semidefinit. Insbesondere ist B diagonalisierbar und hat Eigenwerte $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$. Wir setzen

$$\sigma_i := \sqrt{\lambda_i} \geq 0.$$

Sei weiterhin v_i ein Eigenvektor zu λ_i, so dass die Matrix $V \in \mathbb{K}^{n\times n}$ mit Spalten v_i eine unitäre/orthogonale Matrix bildet. Dann setzen wir $u_i := \sigma_i^{-1} Av_i$. Es gilt dann:

$$\langle u_i, u_j\rangle = \frac{1}{\sigma_i\,\sigma_j}\langle Av_i, Av_j\rangle = \frac{1}{\sigma_i\,\sigma_j}\langle v_i, Bv_j\rangle = \frac{\sigma_j}{\sigma_i}\langle v_i, v_j\rangle = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$$

D.h. die u_i bilden ein orthonormales System von Vektoren. Wir können daher die u_i zu einer orthonormalen Basis $\{u_1, \ldots, u_n\}$ von $\mathbb{K}^n$ vervollständigen. Wir definieren dann U als die Matrix mit Spalten $u_1, \ldots, u_n$. Es gilt nach Konstruktion, dass

$$U^T AV = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \\ 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{pmatrix}.$$

Wenn wir diese Gleichung von links mit U und von rechts mit V^* multiplizieren, erhalten wir behauptete Zerlugung von A. Dies zeigt die Existenz der Singulärwertzerlegung. Die Eindeutigkeit der Singulärwerte folgt aus der Eindeutigkeit der Eigenwerte von B. □

Die Spaltenvektoren der Matrizen U und V aus Satz 2.23 heißen *Singulärvektoren.*

Eine alternative Formulierungen der SVD (Abkürzung für "*singular value decomposition*") ist

$$A = U\mathrm{diag}(\sigma_1, \ldots, \sigma_n)V^*,$$

wobei $U \in \mathbb{K}^{m\times n}$ mit $U^*U = \mathrm{I}$. Solche Matrizen heißen *Stiefel-Matrizen.* Manchmal werden in der Diagonalmatrix nur die positiven Singulärwerte gelistet und sowohl U als auch V als Stiefelmatrizen gewählt.

Satz 2.24. *Für $A \in \mathbb{K}^{m\times n}$ gilt*

a) $\|A\|_2 = \sqrt{\rho(A^*A)}$.

b) Sind $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{m,n\}} \geq 0$ *die Singulärwerte von* A*, so gilt* $\|A\|_2 = \sigma_1$.

c) $\|A\|_2 \leq \|A\|_{\mathrm{F}}$.

d) $\|A^*\|_1 = \|A\|_\infty$.

e) $\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}$.

f) Falls $m = n$ *gilt:* $\rho(A) = \rho(A^*) \leq \|A\|_2 = \|A^*\|_2$.

g) Falls $m = n$ *und* $A = A^*$ *gilt:* $\|A\|_2 = \rho(A)$.

Beweis. Zu a): Es gilt

$$\|A\|_2^2 = \sup_{\|x\|_2\leq 1} \|Ax\|_2^2 = \sup_{\|x\|_2\leq 1} \langle Ax, Ax\rangle = \sup_{\|x\|_2\leq 1} \langle A^*Ax, x\rangle.$$

Da $B := A^*A \in \mathbb{K}^{n\times n}$ hermitesch ist, können wir eine unitäre Matrix $Q \in \mathbb{K}^{n\times n}$ und eine Diagonalmatrix $D \in \mathbb{R}^{n\times n}$ finden mit $A^*A = Q^*DQ$. Die Diagonalelemente in D sind dabei gerade die n Eigenwerte von A^*A und die Spaltenvektoren von Q^* dazugehörige normierte (orthogonale) Eigenvektoren. Wir können das obige innere Produkt wie folgt umschreiben: $\langle A^*Ax, x\rangle = \langle Q^*DQx, x\rangle = \langle D(Qx), Qx\rangle$. Da $\|Qx\|_2 = \|x\|_2$, folgt hieraus

$$\|A\|_2^2 = \sup_{\|y\|_2\leq 1} \langle Dy, y\rangle = \sup_{\|y\|_2\leq 1} \sum_{i=1}^{n} D_{ii} \cdot |y_i|^2 = \max_{j=1,\ldots,n} |D_{jj}| = \rho(D).$$

Für die vorletzte Gleichung können wir Definition 2.19 b) verwenden und den Ausdruck als induzierte Norm der $1\times n$ Matrix $(D_{11}, \ldots, D_{nn})$ bezüglich der 1-Norm interpretieren. Da $\rho(D) = \rho(A^*A)$, folgt schließlich$\|A\|_2^2 = \rho(A^*A)$.

Zu b): Es sei $A = U\Sigma V^*$ die Singulärwertzerlegung von A, wobei U und V unitäre/orthogonale Matrizen sind und $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$. Dann gilt

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \max_{\|x\|_2=1} \|U\Sigma Vx\|_2 = \max_{\|y\|_2=1} \|\Sigma y\|_2 = \|\Sigma\|_2,$$

wobei wir im vorletzten Schritt benutzt haben, dass U und V normerhaltend sind. Für die Diagonalmatrix Σ gilt dann $\|\Sigma\|_2 = \max \sigma_i = \sigma_1$.

Zu c): Es gilt mit Cauchy-Schwarz

$$\|Ax\|_2^2 = \sum_{j=1}^{m} \left|\sum_{k=1}^{n} A_{jk}x_k\right|^2 \leq \sum_{j=1}^{m} \left(\sum_{k=1}^{n} |A_{jk}|^2 \sum_{k=1}^{n} |x_k|^2\right) = \|A\|_{\mathrm{F}}^2 \|x\|_2^2,$$

also ist $\|Ax\|_2^2/\|x\|_2^2 \leq \|A\|_{\mathrm{F}}$. Daraus folgt $\|A\|_2 \leq \|A\|_{\mathrm{F}}$.

Zu d): Nach Definition 2.19 b) ist $\|A^*\|_1$ die Spaltensummennorm von A^*. Andererseits ist nach Definition 2.19 a) $\|A\|_\infty$ die Zeilensummennorm von A. Also $\|A^*\|_1 = \|A\|_\infty$.

Zu e): Sei v, $\|v\|_1 = 1$, Eigenvektor von A^*A zu Eigenwert λ mit $|\lambda| = \rho(A^*A)$. Dann:

$$\|A\|_2^2 = \rho(A^*A) = |\lambda| \cdot \|v\|_1 = \|A^*Av\|_1 \leq \|A^*\|_1 \|A\|_1 \|v\|_1 = \|A\|_\infty \|A\|_1.$$

Zu f): Da A und A^* bis auf komplexe Konjugation das gleiche charakteristische Polynom haben, gilt $\rho(A) = \rho(A^*)$. Ferner sei $\lambda \neq 0$ Eigenwert von A^*A und somit auch von AA^* (multipliziere $A^*Av = \lambda v$ von links mit A). Mit Vertauschung der Rollen von A und A^* folgt daher mit a)

$$\|A\|_2^2 = \rho(A^*A) = \rho(AA^*) = \|A^*\|_2^2.$$

Schließlich folgt das Ungleichheitszeichen aus Lemma 2.21.

Zu g): Da $A = A^*$, ist A unitär diagonalisierbar, d.h. $A = Q^*DQ$ mit einer unitären Matrix $Q \in \mathbb{K}^{n\times n}$ und einer Diagonalmatrix D in $\mathbb{R}^{n\times n}$. Damit ist

$$\|A\|_2^2 = \rho(A^*A) = \rho(A^2) = \rho(D^2) = \rho(D)^2 = \rho(A)^2.$$

□

Korollar 2.25. *Für alle Diagonalmatrizen $D \in \mathbb{K}^{n\times n}$ und alle $k \in \mathbb{N}$ gilt*

$$\|D^k\|_2 = \rho(D^k) = \rho(D)^k = \|D\|_2^k.$$

2.4 Kondition linearer Gleichungssysteme

Wir können nun die Kondition des Problems $Ax = b$ untersuchen. Sei hierzu $A \in \mathbb{K}^{n\times n}$ invertierbar. Wir betrachten die Konditionszahl der Lösungsabbildung

$$f : \mathbb{K}^n \to \mathbb{K}^n, \, b \mapsto x = A^{-1}b.$$

Im Folgenden sei $\|\cdot\|$ eine Norm auf $\mathbb{K}^n$ mit induzierter Norm $\|\cdot\|$ auf $\mathbb{K}^{n\times n}$.

Sei $Ax = b$. Zur Berechnung der Konditionszahl verwenden wir Satz 2.4. Die Jacobimatrix der Abbildung f ist $Jf(b) = A^{-1}$. Mit Satz 2.4 folgt, dass

$$\kappa_{\text{rel}}(f, b) = \frac{\|A^{-1}\| \cdot \|b\|}{\|f(b)\|} = \frac{\|A^{-1}\| \cdot \|Ax\|}{\|x\|} \leq \|A^{-1}\| \cdot \|A\|.$$

Aufgrund dieser Abschätzung definiert man für eine induzierte Matrixnorm die *Konditionszahl* als

$$\text{cond}_{\|\cdot\|}(A) := \|A^{-1}\| \, \|A\|. \tag{2.6}$$

Wenn $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ die Singularwärte von A sind, gilt nach Satz 2.24 b) $\|A\|_2 = \sigma_1$ und $\|A^{-1}\|_2 = \sigma_n^{-1}$. Dies impliziert die Formel

$$\operatorname{cond}_2(A) = \frac{\sigma_1}{\sigma_n}. \tag{2.7}$$

Beispiel 2.26. Sei $A := \left(\begin{smallmatrix} 2 & 0 \\ 0 & \frac{1}{3} \end{smallmatrix}\right)$. Dann sind Eigenwerte von A gerade $\frac{1}{3}$ und 2. Also ist $\|A\|_2 = 2$, $\|A^{-1}\|_2 = 3$ und $\operatorname{cond}_2(A) = 6$.

Da das Problem f linear ist, können wir eine globale Schranke für den Vorwärtsfehler $\|\Delta x\|/\|x\|$ mit Hilfe der Konditionszahl angeben (beachte, dass die Konditionszahl nur für infinitesimal kleine Fehler Δb definiert ist).

Satz 2.27. *Sei $A \in \mathbb{K}^{n\times n}$ regulär und $Ax = b$ mit $x, b \in \mathbb{K}^n$. Sei zudem*

$$A(x + \Delta x) = b + \Delta b.$$

Dann gilt

$$\frac{\|\Delta x\|}{\|x\|} \leq \operatorname{cond}_{\|\cdot\|}(A) \frac{\|\Delta b\|}{\|b\|}.$$

Satz 2.27 ist ein Korollar des folgenden allgemeineren Satzes.

Satz 2.28. *Sei $A \in \mathbb{K}^{n\times n}$ regulär. Wir betrachten das gestörte lineare Gleichungssystem*

$$\begin{aligned} Ax &= b, \\ (A + \Delta A)(x + \Delta x) &= b + \Delta b \end{aligned}$$

bei dem zusätzlich die Matrix A gestört ist. Unter der Voraussetzung $\|A^{-1}\|\|\Delta A\| < 1$ gilt, dass

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\operatorname{cond}_{\|\cdot\|}(A)}{1 - \operatorname{cond}_{\|\cdot\|}(A)\frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right).$$

Beweis. Analog zur geometrischen Reihe gilt für alle $\|M\| < 1$ die Darstellung

$$(\mathrm{I} - M)^{-1} = \sum_{k=0}^{\infty} M^k.$$

Dabei ist $M^0 := \mathrm{I}$, die Reihe konvergiert in dem Banach-Raum der Matrizen, und es gilt

$$\left\|(\mathrm{I} - M)^{-1}\right\| = \left\|\sum_{k=0}^{\infty} M^k\right\| \leq \sum_{k=0}^{\infty} \|M\|^k = \frac{1}{1 - \|M\|}. \tag{2.8}$$

Wir setzen jetzt $M := -A^{-1}(\Delta A)$. Nach Annahme gilt $\|M\| \leq \|\Delta A\|\|A^{-1}\| < 1$. Also mit (2.8):

$$\left\|(\mathrm{I} + A^{-1}(\Delta A))^{-1}\right\| \leq \frac{1}{1 - \|A^{-1}(\Delta A)\|}.$$

Weiterhin gilt $A + \Delta A = A(\mathrm{I} + A^{-1}\Delta A)$ und somit $(A + \Delta A)^{-1} = (\mathrm{I} + A^{-1}\Delta A)^{-1}A^{-1}$. Wir erhalten

$$\|(A + \Delta A)^{-1}\| \leq \|(\mathrm{I} + A^{-1}\Delta A)^{-1}\| \cdot \|A^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\Delta A\|} \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\Delta A\|}.$$

Aus $(A+\Delta A)\Delta x = \Delta b - (\Delta A)\, x$ folgt mit Hilfe der Dreiecksungleichung die Abschätzung

$$\|\Delta x\| \leq \|(A + \Delta A)^{-1}\|\|\Delta b - (\Delta A)\, x\| \leq \frac{\|A^{-1}\|(\|\Delta b\| + \|\Delta A\|\|x\|)}{1 - \|A^{-1}\|\|\Delta A\|}.$$

Dann liefern

$$\begin{aligned}\frac{\|\Delta x\|}{\|x\|} &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\Delta A\|}\left(\frac{\|\Delta b\|}{\|x\|} + \|\Delta A\|\right) \\ &= \frac{\mathrm{cond}_{\|\cdot\|}(A)}{1 - \mathrm{cond}_{\|\cdot\|}(A)\frac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta b\|}{\|A\|\|x\|} + \frac{\|\Delta A\|}{\|A\|}\right)\end{aligned}$$

und $\|b\| = \|Ax\| \leq \|A\|\|x\|$ die Behauptung. □

2.5 Turing's Konditionszahl

Im vorherigen Abschnitt haben wir gezeigt, dass $\mathrm{cond}_{\|\cdot\|}(A)$ (2.6) die Konditionszahl des Problems $b \mapsto A^{-1}b$ beschränkt. Die Zahl $\mathrm{cond}_2(A)$, also $\mathrm{cond}_{\|\cdot\|}(A)$ für die 2-Norm, nennt man auch *Turing's Konditionszahl*, da diese zuerst von Turing in [10] beschrieben wurde. Wir beweisen im nächsten Satz, dass Turing's Konditionszahl die Konditionszahl der Matrixinversion ist.

Satz 2.29. *Sei* $E := \{A \in \mathbb{K}^{n\times n} : \det(A) \neq 0\}$ *und* $f : E \to E,\ A \mapsto A^{-1}$. *Dann ist*

$$\kappa_{\mathrm{rel}}(f, A) = \mathrm{cond}_2(A) = \|A^{-1}\|_2 \cdot \|A\|_2.$$

Beweis. Wir verwenden wieder Satz 2.4. Es ist allerdings kompliziert die Jacobimatrix der Abbildung f direkt hinzuschreiben. Es ist einfacher ihre Wirkung auf Vektoren zu beschreiben. Sei dazu $A(t) \in \mathbb{K}^{n\times n}$ eine differenzierbare Kurve mit $A(0) = A$ und $A'(t) = B$. Sei $\mathrm{vec}(B)$ die Vektorisierung von B bezüglich einer fest gewählten Basis. Dann gilt:

$$Jf(A) \cdot \mathrm{vec}(B) = \frac{\mathrm{d}}{\mathrm{d}t}(A(t))^{-1}\,\Big|_{t=0}.$$

Wir berechnen die rechte Seite. Sei $C(t) := (A(t))^{-1}$. Dann gilt $A(t)C(t) = \mathrm{I}$. Ableiten beider Seiten ergibt $A'(t)C(t) + A(t)C'(t) = 0$. Daraus folgt

$$\frac{\mathrm{d}}{\mathrm{d}t}(A(t))^{-1}\,\Big|_{t=0} = C'(0) = -A^{-1}BA.$$

Daraus folgt nun, dass

$$\kappa_{\text{rel}}(f, A) = \max_{\|B\|_2=1} \|A^{-1}BA^{-1}\|_2 \cdot \frac{\|A\|_2}{\|A^{-1}\|_2}.$$

Einerseits impliziert die Submultiplikativität der 2-Norm

$$\kappa_{\text{rel}}(f, A) \leq \|A^{-1}\|_2^2 \cdot \frac{\|A\|_2}{\|A^{-1}\|_2} = \|A^{-1}\|_2 \cdot \|A\|_2.$$

Für die andere Richtung sei $\lambda^2, \lambda > 0$, der betragsmäßig größte Eigenwert von $A^{-*}A^{-1}$ mit Eigenvektor v, $\|v\|_2 = 1$. Dann gilt mit 2.24 a), dass

$$\|A^{-1}\|_2 = \sqrt{\rho(A^{-*}A^{-1})} = \lambda.$$

Sei weiterhin $w := A^{-1}v$ und $B = \lambda^{-1}\, vw^*$. Beachte, dass

$$\|B\|_2 = \lambda^{-1} \cdot \|w\|_2 = \lambda^{-1} \cdot \sqrt{v^*A^{-*}A^{-1}v} = 1.$$

Wir verwenden nun wieder Satz 2.24 a) und erhalten

$$\|A^{-1}BA^{-1}\|_2^2 = \rho(M), \quad M := A^{-*}B^*A^{-*}A^{-1}BA^{-1}.$$

Aus $BA^{-1}v = Bw = \lambda^{-1}\,\|w\|_2^2\, v = \lambda\, v.$ folgt

$$Mv = \lambda^3\, A^{-*}B^*v = \lambda^2\, A^{-*}w = \lambda^2\, A^{-*}A^{-1}v = \lambda^4\, v,$$

also $\rho(M) \geq \lambda^4 = \|A^{-1}\|_2^4$. Hieraus folgt schließlich, dass

$$\kappa_{\text{rel}}(f, A) \geq \sqrt{\rho(M)} \cdot \frac{\|A\|_2}{\|A^{-1}\|_2} \geq \|A^{-1}\|_2^2 \cdot \frac{\|A\|_2}{\|A^{-1}\|_2} = \|A^{-1}\|_2 \cdot \|A\|_2$$

Insgesamt haben wir also gezeigt, dass $\kappa_{\text{rel}}(f, A) = \|A^{-1}\|_2 \cdot \|A\|_2$. □

Beispiel 2.30. Wir kommen noch einmal zu Beispiel 2.2 zurück. Die Matrizen waren

$$A = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \quad \text{und } B = \begin{pmatrix} 1 & 1 \\ 1 & 1+\delta \end{pmatrix}.$$

Konkret setzen wir $\delta = 10^{-8}$. Dann gilt $\text{cond}_2(A) = 1$ und $\text{cond}_2(B) \approx 4 \cdot 10^8$. Zusammen mit Satz 2.29 erklärt dies das unterschiedliche Verhalten der Ausgaben in Bezug auf Fehler in der Eingabe im Beispiel 2.2. ◇

Zuletzt geben wir eine metrische Interpretation der Konditionszahl. Sei dazu

$$\Omega := \{B \in \mathbb{K}^{n\times n} \mid \det(B) = 0\}$$

die algebraische Varietät der singulären Matrizen. Der Abstan der der Matrix $A \in \mathbb{K}^{n\times n}$ zu Ω ist dann

$$d(A, \Omega) := \min_{B\in\Omega} \|A - B\|_2.$$

Beachte, dass das Minimum existiert, da Ω abgeschlossen ist. Wir haben folgenden Satz.

Satz 2.31. *Sei $A \in \mathbb{K}^{n\times n}$. Dann gilt*

$$\operatorname{cond}_2(A) = \frac{\|A\|_2}{d(A,\Omega)}.$$

Beweis. Wir haben in (2.7) gezeigt, dass $\operatorname{cond}_2(A) = \sigma_1/\sigma_n$, wobei σ_1, σ_n der größte und kleinste Singulärwert von A sind. Laut Satz 2.24 gilt $\|A\|_2 = \sigma_1$. Es ist also zu zeigen, dass $\sigma_n = d(A,\Omega)$. Sei dazu $A = U\Sigma V^*$ die Singulärwertzerlegung von A, wobei $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ und U, V unitär. Für alle $B \in \Omega$ gilt $U^*BV \in \Omega$, so dass:

$$d(A,\Omega) = d(\Sigma,\Omega) = \min_{B\in\Omega} \|\Sigma - B\|_2.$$

Sei nun $B_0 = \operatorname{diag}(\sigma_1, \ldots, \sigma_{n-1}, 0)$. Dann ist $B_0 \in \Omega$ und $\|A - B_0\|_2 = \sigma_n$. Dies zeigt $d(A,\Omega) \leq \sigma_n$. Für die andere Ungleichung sei $B \in \Omega$ beliebig und $V \in \mathbb{K}^n$, $\|v\|_2 = 1$, mit $Bv = 0$. Dann gilt:

$$\|\Sigma - B\|_2 \geq \|(\Sigma - B)v\|_2 = \|\Sigma v\|_2 = \sqrt{\sigma_1^2 v_1^2 + \cdots + \sigma_n^2 v_1^2} \geq \sigma_n,$$

die letzte Ungleichung, da $\sigma_1^2 v_1^2 + \cdots + \sigma_n^2 v_1^2$ eine Konvexkombination der σ_i ist. Es gilt also auch $d(A,\Omega) = d(\Sigma,\Omega) \geq \sigma_n$. Insgesamt gilt damit Gleichheit. □

3 Lineare Gleichungssysteme

In diesem Abschnitt beschäftigen wir uns mit dem Lösen linearer Gleichungssysteme. Speziell wollen wir für die Daten $A \in \mathbb{K}^{m\times n}$, $x \in \mathbb{K}^n$ und $b \in \mathbb{K}^m$ das Gleichungsystem

$$Ax = b \tag{3.1}$$

lösen. Die Kondition dieses Problems haben wir bereits in den Abschnitten 2.4 und 2.5 studiert. Falls es. bei überbestimmten Gleichungen mit gestörten Daten nicht möglich ist eine Lösung zu finden, so sind wir auch an der Lösung des *Ausgleichsproblems*

$$\|Ax - b\| = \min\{\|Ay - b\| \mid y \in \mathbb{K}^n\} \tag{3.2}$$

interessiert.

Das Ziel dieses Abschnitts ist es effiziente numerische Algorithmen für das Lösen von (3.1) und (3.2) zu entwickeln. Wir fokussieren uns also auf die Punkte c) (*Laufzeit*) und d) (*Einfache Beschreibung*) in der Tabelle am Anfang von Kapitel 1. Um die Laufzeit zu beschreiben, werden wir die Anzahl der jeweils benötigten Multiplikationen und Divisionen angeben (Multiplikationen und Divisionen zweier Zahlen sind aufwendiger als Additionen und Subtraktionen, daher werden bei der Aufwandsberechnung meist nur Erstere gezählt.). Dazu verwenden wir die *Landau-Symbole* O und o.

Definition 3.1. Für zwei Folgen $(f_n)_{n\in\mathbb{N}}$ und $(g_n)_{n\in\mathbb{N}}$ sind die *Landau-Symbole* O und o folgendermaßen definiert:

$$(1) \quad f_n = O(g_n) \quad :\Longleftrightarrow \quad \exists K > 0 : \limsup_{n\to\infty} \frac{|f_n|}{|g_n|} < K.$$

$$(2) \quad f_n = o(g_n) \quad :\Longleftrightarrow \quad \lim_{n\to\infty} \frac{|f_n|}{|g_n|} = 0.$$

3.1 Gauß-Elimination und LR-Zerlegung

Sei $m = n$ und A invertierbar. Zur Lösung des linearen Gleichungssystems $Ax = b$ kann man Gauß-Elimination auf die erweiterte $n \times (n + 1)$-Matrix $(A|b)$ anwenden. Dabei überführt man die Matrix $(A|b)$ durch elementare Zeilenumformungen zunächst in eine Matrix $(R|Ub)$, wobei

$$R = UA \tag{3.3}$$

eine rechte obere Dreiecksmatrix und U eine unipotente untere Dreiecksmatrix ist. Wir nennen diesen Schritt „Vorwärtselimination“.

Definition 3.2. Eine Matrix $A \in \mathbb{K}^{n\times n}$ heißt *unipotent*, wenn $(A - \mathrm{I})^k = 0$ für ein $k \in \mathbb{N}$.

Lemma 3.3. *Eine untere Dreiecksmatrix $U \in \mathbb{K}^{n\times n}$ ist genau dann unipotent, wenn alle Diagonaleinträge gleich 1 sind.*

Beweis. Angenommen alle Diagonaleinträge von U sind gleich 1. Dann ist $U = N + \mathrm{I}$, wobei die Einträge von N gleich 0 sind. Dann existiert aber ein $k \in \mathbb{N}$ mit $N^k = 0$, also $(U - \mathrm{I})^k = 0$. Ist andererseits U nilpotent, so müssen die Diagonaleinträge von $N = U - \mathrm{I}$ alle gleich 0 sein. Die Diagonaleinträge von $U = N + \mathrm{I}$ sind dann alle gleich 1. □

Zurück zum Lösen von $Ax = b$ durch Gauß-Elimination. Nach der Vorwärtselimination erhalten wir das äquivalente Gleichungssystem $(R|Ub)$. Dann löst man das gestaffelte Gleichungssystem $Rx = Ub$ durch Rückwärtssubstitution. Alternativ kann man $(R|Ub)$ durch weitere Zeilenumformungen in die Matrix $(I|A^{-1}b)$ („Rückwärtselimination“ und anschließende Normierung) überführen und anschließend $x := A^{-1}b$ berechnen. Muss man das Gleichungssystem bei festgehaltener Koeffizientenmatrix A für viele rechte Seiten b lösen, so lässt sich viel Aufwand sparen, wenn man zunächst *entweder* die Dreiecksmatrizen R und U berechnet *oder* A^{-1} direkt bestimmt.

Wir wollen einen Algorithmus (die „LR-Zerlegung“) angeben, der zur Lösung von Gleichungssystemen im Wesentlichen auf die Dreiecksmatrizen R und U zurückgreift. Setzen wir $L := U^{-1}$ so erhalten wir die zu 3.3 äquivalente folgende Formulierung.

Definition 3.4. Lässt sich $A \in \mathbb{K}^{n\times n}$ als

$$A = LR$$

schreiben mit einer unipotenten unteren Dreiecksmatrix L und einer oberen Dreiecksmatrix R, so nennt man dies auch *LR-Zerlegung.*

Satz 3.5. *Falls $A \in \mathbb{K}^{n\times n}$ eine LR-Zerlegung besitzt, so ist diese eindeutig bestimmt.*

Beweis. Seien $A = LR = L'R'$ zwei LR-Zerlegungen von A. Dann gilt $L_0R_0 = \mathrm{I}$, wobei $L_0 := L^{-1}L'$ und $R_0 := R'R^{-1}$. Da die unipotenten unteren Dreiecksmatrizen eine Gruppe bilden, ist L_0 eine unipotente untere Dreiecksmatrix. Genauso ist R_0 eine obere Dreiecksmatrix. Da $R_0 = L_0^{-1}$ muss R_0 auch eine unipotente untere Dreiecksmatrix sein. Es folgt, dass $R_0 = L_0 = \mathrm{I}$. □

Bemerkung 3.6. a) Nicht alle regulären Matrizen lassen eine LR-Zerlegung zu. Z.B. besitzt die Matrix $A = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ keine LR-Zerlegung.

b) Sei $\epsilon > 0$ klein. Die LR-Zerlegung

$$A = \begin{pmatrix} \epsilon & 1 \\ 1 & \epsilon \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1/\epsilon & 1 \end{pmatrix} \begin{pmatrix} \epsilon & 1 \\ 0 & \epsilon - 1/\epsilon \end{pmatrix} = LR$$

zerlegt die gut konditionierte Matrix A in das Produkt zweier schlecht konditionierter Matrizen. Dies ist ein Beispiel, wie man ein gut konditioniertes Problem

(das Lösen von $Ax = b$) mit Hilfe eines spezifischen Verfahrens in zwei schlecht konditionierte Teilprobleme (zuerst das Lösen von $Ly = b$ und anschließend das von $Rx = y$) zerlegen kann.

Möchte man nun $Ax = b$ mit Hilfe der LR-Zerlegung lösen, so lösen wir sukzessiv Gleichungsysteme mit den einzelnen Dreiecksmatrizen. Dies führt zu folgendem Algorithmus:

Algorithmus : Lösen mittels LR-Zerlegung
Eingabe : $A \in \mathbb{K}^{n\times n}$ und $b \in \mathbb{K}^n$
Berechne LR-Zerlegung $A = LR$;
Löse $Ly = b$ (Vorwärtssubstitution);
Löse $Rx = y$ (Rückwärtssubstitution);
Ergebnis : Lösung x von $Ax = b$

Es bleibt noch zu klären, wie die LR-Zerlegung berechnet wird. Bei unserer numerischen Umsetzung beginnen wir mit $(A|\mathrm{I})$ und führen sukzessive Schritte per Gauß-Elimination durch.

Definition 3.7. Für $k \in \{1, \dots, n-1\}$ und $x = (x_1, \dots, x_n)^T$ mit $x_k \neq 0$ definieren wir die *Eliminationsmatrix* oder *Frobenius-Matrix* L_k durch

$$L_k = \mathrm{I} - \ell_k e_k^T,$$

wobei $\ell_k = (0, \dots, 0, \ell_{k+1,k} \dots, \ell_{n,k})^T$ und $\ell_{i,k} = x_i/x_k$.

Die Eliminationsmatrix hat die Eigenschaft, dass

$$L_k x = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ 0 & \ddots & & & & \\ 0 & \vdots & 1 & & & \\ \vdots & \vdots & -\ell_{k+1,k} & 1 & & \\ \vdots & \vdots & \vdots & & \ddots & \\ \vdots & \vdots & -\ell_{n,k} & 0 & & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Ist hingegen $y = (y_1, \dots, y_{k-1}, 0, \dots, 0)^T \in \mathbb{K}^n$, so erhält man $L_k y = y$, d.h. der Vektor y bleibt unter Linksmultiplikation mit L_k unverändert.

Jeden Eliminationsschritt der oben erwähnten Vorwärtselimination lässt sich als Multiplikation von links mit einer geeigneten Eliminationsmatrix L_k auffassen. Diese erzeugt Nullen in der k-ten Spalte unterhalb der Diagonale. Genauer lässt sich die *Vorwärtselimination* wie folgt beschreiben:

a) Setze $A_1 := A$.

b) Für alle $1 \le k \le n-1$: Ist $(A_k)_{k,k} \neq 0$, so setzen wir $A_{k+1} = L_k A_k$.

Beispiel 3.8. Ausgangspunkt ist also die Matrix $A_1 := A$. Ist $a_{1,1} \neq 0$, so erhalten wir:

$$\underbrace{\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ 0 & a_{2,2}^{(2)} & \dots & a_{2,n}^{(2)} \\ 0 & a_{3,2}^{(2)} & \dots & a_{3,n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n,2}^{(2)} & \dots & a_{n,n}^{(2)} \end{pmatrix}}_{=: A_2} = \underbrace{\begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ -\ell_{2,1} & 1 & 0 & \dots & 0 \\ -\ell_{3,1} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -\ell_{n,1} & 0 & \dots & 0 & 1 \end{pmatrix}}_{= L_1} \underbrace{\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & \dots & a_{3,n} \\ \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix}}_{= A_1}.$$

Multiplikation von A_1 mit L_1 von links bewirkt also, dass die erste Spalte in A_2 Nulleinträge unterhalb der Diagonalen hat. Der nächste Schritt ist dann

$$\underbrace{\begin{pmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & -\ell_{3,2} & 1 & 0 & \dots & 0 \\ \vdots & -\ell_{4,2} & 0 & 1 & 0 & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & -\ell_{n,2} & 0 & \dots & 0 & 1 \end{pmatrix}}_{= L_2} \underbrace{\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ 0 & a_{2,2}^{(2)} & \dots & a_{2,n}^{(2)} \\ 0 & a_{3,2}^{(2)} & \dots & a_{3,n}^{(2)} \\ 0 & a_{4,2}^{(2)} & \dots & a_{4,n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n,2}^{(2)} & \dots & a_{n,n}^{(2)} \end{pmatrix}}_{= A_2} = \underbrace{\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ 0 & a_{2,2}^{(2)} & a_{2,3}^{(2)} & \dots & a_{2,n}^{(2)} \\ 0 & 0 & a_{3,3}^{(3)} & \dots & a_{3,n}^{(3)} \\ 0 & 0 & a_{4,3}^{(3)} & \dots & a_{4,n}^{(3)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n,3}^{(3)} & \dots & a_{n,n}^{(3)} \end{pmatrix}}_{=: A_3}$$

Jetzt haben die ersten zwei Spalten von A_2 Nulleinträge unterhalb der Diagonalen. Diese Eigenschaft setzt sich induktiv fort.

Die nach dem $(n-1)$-Schritt resultierende Matrix A_n ist dann die gesuchte obere Dreiecksmatrix R, und U ist das Produkt der L_k, $k = n-1, \dots, 1$:

$$R := A_n, \quad U := L_{n-1} \cdots L_1 \quad \Longrightarrow \quad R = UA.$$

Lemma 3.9. *Sei $L_k = \mathrm{I} - \ell_k e_k^T$ wie in Definition 3.7. Dann gilt $L_k^{-1} = \mathrm{I} + \ell_k e_k^T$.*

Beweis. Dies folgt aus $L_k \cdot (\mathrm{I} + \ell_k e_k^T) = (\mathrm{I} - \ell_k e_k^T) \cdot (\mathrm{I} + \ell_k e_k^T) = \mathrm{I} - \ell_k e_k^T \ell_k e_k^T = \mathrm{I}$, wobei wir für die letzte Gleichung $e_i^T \ell_j = 0$ für $1 \le i \le j \le n$ benutzt haben. □

Dieses Lemma zeigt, dass man die Inverse der Eliminationsmatrix $L_k = \mathrm{I} - \ell_k e_k^\top$ erhält, indem man die Einträge unterhalb der Diagonale negiert. Es folgt, dass

$$L = U^{-1} = L_1^{-1} \cdots L_{n-1}^{-1} = (\mathrm{I} + \ell_1 e_1^\top) \cdots (\mathrm{I} + \ell_{n-1} e_{n-1}^\top) = \mathrm{I} + \sum_{j=1}^{n-1} \ell_j e_j^\top.$$

wobei die letzte Identität aus $e_i^T \ell_j = 0$ für $1 \leq i \leq j \leq n$ folgt.

Insbesondere lässt sich L leicht im Algorithmus berechnen und effizient speichern, indem man sukzessive die im k-ten Schritt zu Null werdenden Einträge $a_{j,k}^{(k)}$ der Matrix A_k für $j = k+1, \ldots, n$ mit $\ell_{j,k}$ überschreibt.

In jedem der n Schritte müssen wir $O(n^2)$ Multiplikationen und Divisionen durchführen. Die Lösung der Gleichungssysteme $Ly = b$ und $Rx = y$ benötigt $O(n^2)$ Operationen. Dies führt zu folgendem Satz.

Satz 3.10. *Der Algorithmus benötigt $O(n^3)$ Multiplikationen und Divisionen für die LR Zerlegung und $O(n^2)$ Operationen für die Vor-und Rückwärtselimination. Das wiederholte Lösen mit neuem b benötigt deshalb nur $O(n^2)$ Operationen.*

Zum Abschluss dieses Abschnitts diskutieren wir noch Strategien um die numerische Stabilität des Verfahrens zu verbessern. Sind die Pivotelemente $a_{k,k}^{(k)}$ betragsmäßig sehr klein, so werden die Faktoren $\ell_{j,k} = a_{j,k}^{(k)} / a_{k,k}^{(k)}$ betragsmäßig sehr groß und können zur Fehlerverstärkung, evtl. kombiniert mit Auslöschung, beitragen. Zum Erreichen höherer numerischer Stabilität, ist es in diesem Fall sinnvoll zu *Pivotisieren*, d.h. durch Zeilen- und ggf. Spaltenvertauschungen einen geeigneteren Eintrag für die Vorwärtselimination zu erhalten. Hier gibt es verschiedene Strategien.

Keine Pivotisierung: Kann zur Fehlerverstärkung und/oder einem Abbruch des Algorithmus führen (weil z.B. A selbst gar keine LR-Zerlegung besitzt oder betragsmäßig kleine Einträge zu null gerundet werden). Bei manchen Matrizen weiss man jedoch im Voraus, dass die LR-Zerlegung ohne Pivotisierung zum Ziel führt. Ist z.B. *A strikt diagonal dominant*, d.h. gilt für alle $j = 1, \ldots, n$ $|a_{jj}| > \sum_{k\,:\,k \neq j} |a_{jk}|$, so behalten die Matrizen A_k in jedem Schritt der LR-Zerlegung diese Eigenschaft.

Spaltenpivotsuche: Gehe Spalte für Spalte vor. Das *Pivotelement*, mit dem die unteren Einträge der jeweiligen Spalte eliminiert werden, ist das relativ zur Betragssumme (also der $\|\cdot\|_1$-Norm) seiner Zeile betragsgrößte. Die entsprechende Zeile wird weiter nach oben getauscht.

Totalpivotsuche: Eliminiere im i-ten Schritt mit dem Eintrag der verbleibenden rechten unteren $(n-i+1) \times (n-i+1)$-Matrix, der maximalen Betrag hat. Die Totalpivotsuche in der Praxis nur in Ausnahmefällen eingesetzt, weil die Suche das Totalpivots vergleichsweise aufwändig ist ($O(n^3)$ Vergleiche).

Beispiel 3.11 (Bandmatrizen)**.** Nach Satz 3.10 ist der Aufwand der LR-Zerlegung $O(n^3)$. Dies gilt für allgemeine Matrizen $A \in \mathbb{K}^{n \times n}$. Besitzen die betrachteten Matrizen aber eine spezielle Struktur, so lässt sich diese u.U. nutzen, um den Aufwand deutlich kleiner zu gestalten. Ein wichtiges Beispiel sind *Bandmatrizen*.

Wir nennen $A \in \mathbb{K}^{n\times n}$ eine *Bandmatrix der Breite* r, falls

$$a_{j,k} = 0 \text{ für alle } |j-k| > r.$$

Eine Bandmatrix der Breite 0 ist eine Diagonalmatrix. Eine Bandmatrix der Breite 1 heißt auch *Tridiagonalmatrix*. Das Produkt einer Bandmatrix der Breite r_1 mit einer Bandmatrix der Breite r_2 ist eine Bandmatrix der Breite (höchstens) $r_1 + r_2$.

Ist A eine Bandmatrix der Breite r, so führt die LR-Zerlegung zu Matrizen L, R der Breite r. Für $r = 1$ folgt zum Beispiel

$$\begin{pmatrix} a_1 & c_1 & 0 & \cdots & 0 \\ b_1 & a_2 & c_2 & \ddots & \vdots \\ 0 & b_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & c_{n-1} \\ 0 & \cdots & 0 & b_{n-1} & a_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ \beta_1 & 1 & 0 & \ddots & \vdots \\ 0 & \beta_2 & \ddots & \ddots & 0 \\ \dots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \beta_{n-1} & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 & c_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & c_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & c_{n-1} \\ 0 & \cdots & \cdots & 0 & \alpha_n \end{pmatrix}$$

Die Koeffizienten können wie folgt berechnet werden.

Algorithmus : LR-Zerlegung einer Tridiagonalmatrix
Eingabe : Tridiagonalmatrix $A \in \mathbb{K}^{n\times n}$.
$c_0 = 0$, $\beta_0 = 0$;
für $j = 1, \dots, n-1$ **tue**
  $\alpha_j = a_j - c_{j-1} \cdot \beta_{j-1}$;
  $\beta_j = \frac{b_j}{\alpha_j}$;
Ende
$\alpha_n = a_n - c_{n-1} \cdot \beta_{n-1}$;
Ergebnis : Koeffizienten der LR-Zerlegung

Die Laufzeit dieses Algorithmus für Tridiagonalmatrizen ist $O(n)$. Das anschließende Lösen von $Ly = b$ und $Rx = y$ ist von der Komplexität $O(n)$. Also benötigen wir nun $O(n)$ Schritte statt $O(n^3)$ wie im allgemeinen Fall.

3.2 Iterationsverfahren

In diesem Abschnitt werden wir die Gleichung $Ax = b$ mittels eines Fixpunktverfahrens lösen. Die exakte Lösung wird hierbei möglicherweise niemals erreicht. Allerdings nähern wir uns der Lösung in jedem Schritt.

Die Idee einen Fixpunktverfahren ist es, eine Abbildung

$$\Phi : \mathbb{K}^n \to \mathbb{K}^n$$

zu finden, so dass ein Fixpunkt von Φ (d.h. $\Phi(x) = x$) eine Lösung von $Ax = b$ ist. Anstatt $Ax = b$ zu lösen wird dann ein Fixpunkt von Φ gesucht.

Iterative Verfahren sind dann Fixpunktverfahren für sogenannte Kontraktionen.

Definition 3.12. Sei (X, d) ein metrischer Raum. Eine Abbildung $\Phi : X \to X$ heißt *Kontraktion*, falls es eine Lipschitz-Konstante $q \in [0, 1)$ gibt, so dass

$$\forall x, y \in X : d(\Phi(x), \Phi(y)) \leq q\, d(x, y).$$

Wir erinnern uns: Ein metrischer Raum X heißt vollständig, wenn jede Cauchy-Folge in X auch einen Grenzwert in X besitzt.

Folgenden Satz zitieren wir ohne Beweis.

Satz 3.13 (Fixpunktsatz von Banach)**.** *Sei (X, d) ein vollständiger metrischer Raum und sei $\Phi : X \to X$ eine Kontraktion mit Lipschitz-Konstante $q \in [0, 1)$. Dann liefert die Iteration*

$$x_{n+1} := \Phi(x_n)$$

mit beliebigem Startpunkt $x_0 \in X$ eine konvergente Folge, die gegen den eindeutigen Fixpunkt x_∞ von Φ konvergiert.

Bemerkung 3.14. Man kann die Aussage des Fixpunktsatzes von Banach (Satz 3.13) erweitern, indem man Aussagen zur Geschwindigkeit der Konvergenz beweist. Es gilt in der Notation von Satz 3.13:

a) A-priori-Schranke (Abschätzung des Fehlers eines numerischen Verfahrens vor der Durchführung des Verfahrens.): $d(x_k, x_\infty) \leq \frac{q^k}{1-q} d(x_1, x_0)$,

b) A-posteriori-Schranke (Abschätzung des Fehlers eines numerischen Verfahrens nach Durchführung des Verfahrens): $d(x_k, x_\infty) \leq \frac{q}{1-q} d(x_k, x_{k-1})$.

Betrachten wir nun das Problem $Ax = b$. Wir zerlegen A als

$$A = M - N,$$

wobei M invertierbar ist. Dann gilt für

$$T = M^{-1}N \quad \text{und} \quad c = M^{-1}b,$$

dass

$$\Phi(x) := Tx + c = M^{-1}(Nx + b) = M^{-1}(Mx - Ax + b) = x - M^{-1}(Ax - b).$$

Also ist x genau dann ein Fixpunkt von $\Phi(x) = Tx + c$, wenn $Ax = b$. Diese Beobachtung nutzen wir im folgenden Algorithmus aus.

Algorithmus : Allgemeine Fixpunktiteration für lineare Gleichungssysteme
Eingabe : $A \in \mathbb{K}^{n \times n}$, $b \in \mathbb{K}^n$
Zerlege $A = M - N$ mit M invertierbar;
Wähle x_0 beliebig und $n := 0$;
wiederhole
  $n := n + 1$;
  $x_n := Tx + c = M^{-1}(Nx_{n-1} + b)$;
bis $\|Ax_n - b\|$ *klein genug*;
Ergebnis : Approximation x_n der Lösung x_∞ von $Ax_\infty = b$.

Damit dieser Algorithmus effizient genutzt werden kann, sollte M einfach zu invertieren sein. Z.B. könnte M die Diagonale oder der untere Dreiecksteil von A sein. Außerdem muss $\|T\| < 1$ gelten, wie der folgende Satz zeigt. Die Matrix $T = M^{-1}N$ heißt in diesem Kontext auch *Iterationsmatrix*.

Satz 3.15. *Sei $\|\cdot\|$ eine Norm auf $\mathbb{K}^n$ und sei $A = M - N$ mit invertierbarem M. Sei weiterhin $T = M^{-1}N$, $c = M^{-1}b$ und*

$$\Phi : \mathbb{K}^n \to \mathbb{K}^n, \quad x \mapsto Tx + c.$$

Falls $\|T\| < 1$, ist Φ eine Kontraktion bzgl. der Metrik $d(x, y) = \|x - y\|$ mit eindeutigem Fixpunkt x_∞. Es gilt $Ax_\infty = b$ und $x_n - x_\infty = T^n(x_0 - x_\infty)$.

Beweis. Es gilt

$$d(\Phi(x), \Phi(y)) = \|\Phi(x) - \Phi(y)\| = \|T(x - y)\| \leq \|T\| \, \|x - y\|.$$

Da $q := \|T\| < 1$, ist Φ eine Kontraktion. Nach Satz 3.13 konvergiert die Fixpunktiteration gegen den Fixpunkt x_∞. Wie oben beschrieben ist $\Phi(x_\infty) = x_\infty$ äquivalent zu $Ax_\infty = b$. Zuletzt betrachten wir

$$x_1 - x_\infty = \Phi(x_0) - \Phi(x_\infty) = (Tx_0 - c) - (Tx_\infty - c) = T(x_0 - x_\infty).$$

Induktiv folgt $x_n - x_\infty = T^n(x_0 - x_\infty)$. □

Im Folgenden geben wir zwei konkrete Beispiele für Zerlegungen $A = M - N$: Zunächst zerlegen wir A in

$$A = L + D + R$$

mit einer Diagonalmatrix D, einer strikt unteren Dreiecksmatrix L und einer strikt oberen Dreiecksmatrix R.

- **Gesamtschrittverfahren/Jacobi-Verfahren**: Wir wählen beim

  $$M = D \quad \text{and} \quad N = M - A = -(L + R).$$

  Dann ist die Iterationsmatrix $T = -D^{-1}(L + R)$. Ist nun A strikt diagonal dominant (d.h. $\sum_{k \neq j} |a_{jk}| < |a_{jj}|$ für alle j), so gilt

  $$\|T\|_\infty = \|D^{-1}N\|_\infty = \max_j \sum_k \left|(D^{-1}N)_{jk}\right| = \max_j \sum_{k \neq j} \frac{|a_{jk}|}{|a_{jj}|} =: q < 1.$$

  Somit konvergiert das Verfahren nach Korollar 3.15.

- **Einzelschrittverfahren/Gauß-Seidel-Verfahren**: Hier wählen wir

$$M := L + D \quad \text{und} \quad N := M - A = -R.$$

Die Iterationsmatrix ist $T = -(L + D)^{-1}R$. Sei nun wieder A strikt diagonal dominant und

$$q := \max_j \frac{1}{|a_{jj}|} \sum_{k \neq j} |a_{jk}| < 1.$$

Wir behaupten, dass $\|T\|_\infty \leq q$. Zunächst beobachten wir, dass

$$\|T\|_\infty = \|M^{-1}N\|_\infty = \|(D + L)^{-1}R\|_\infty.$$

Sei dann $\|x\|_\infty \leq 1$ und $y := (L+D)^{-1}Rx$. Wir zeigen, dass $\|y\|_\infty \leq q$. Wir zeigen induktiv induktiv $|y_j| \leq q$. Aus $(L + D)y = Rx$ folgt

$$a_{jj}y_j + \sum_{k<j} a_{jk}y_k = \sum_{k>j} a_{jk}x_k,$$

also

$$y_j = \frac{1}{a_{jj}}\Big(- \sum_{k<j} a_{jk}y_k + \sum_{k>j} a_{jk}x_k \Big).$$

Nun folgt

$$|y_j| \leq \frac{1}{|a_{jj}|}\Big(\sum_{k<j} |a_{jk}|q + \sum_{k>j} |a_{jk}| \Big) \leq \frac{1}{|a_{jj}|}\Big(\sum_{k\neq j} |a_{jk}| \Big) \leq q,$$

was also wie gewünscht $\|y\|_\infty \leq q$ impliziert. Daraus folgt $\|T\|_\infty \leq q < 1$, so dass das Verfahren nach Korollar 3.15 konvergiert.

Beispiel 3.16. Betrachte $Ax = b$ mit

$$A := \begin{pmatrix} 2 & 0 & 1 \\ 1 & -4 & 1 \\ 0 & -1 & 2 \end{pmatrix}, \qquad b := \begin{pmatrix} 1 \\ 4 \\ -1 \end{pmatrix}.$$

Die Matrix A ist strikt diagonal dominant. Die Iterationsmatrizen des Gesamtschrittverfahrens und des Einzelschrittverfahrens sind

$$T_{\text{gesamt}} = \begin{pmatrix} 0 & 0 & -\frac{1}{2} \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{2} & 0 \end{pmatrix}, \qquad T_{\text{einzel}} = \begin{pmatrix} 0 & 0 & -\frac{1}{2} \\ 0 & 0 & \frac{1}{8} \\ 0 & 0 & \frac{1}{16} \end{pmatrix}$$

mit $\|T_{\text{gesamt}}\|_\infty = \frac{1}{2}$ und $\|T_{\text{einzel}}\|_\infty = \frac{1}{2}$.

3.3 Cholesky-Zerlegung

Die Cholesky-Zerlegung einer Matrix $A \in \mathbb{K}^{n\times n}$ ist ähnlich der LR-Zerlegung eine Faktorisierung in eine untere und eine obere Dreiecksmatrix.

Definition 3.17. Sei $A \in \mathbb{K}^{n\times n}$. Eine Faktorisierung der Form

$$A = CC^*$$

mit einer unteren Dreiecksmatrix C mit Diagonalelementen $c_{jj} > 0$ für alle $j = 1, \dots, n$, heißt *Cholesky-Zerlegung* von A.

Ist $A = CC^*$ eine Cholesky-Zerlegung, so schreibt man auch

$$C = A^{\frac{1}{2}}.$$

Die Schreibweise ist berechtigt, das die Cholesky-Zerlegung eindeutig ist. Dies beweisen wir in Satz 3.21.

Jedoch lässt sich nicht jeder Matrix eine Cholesky-Zerlegung zuordnen. In jedem Fall muss, da wir C als invertierbare Matrix annehmen, $A = CC^*$ positiv definit sein. Der nächste Satz zeigt, dass das auch ein hinreichendes Kriterium ist.

Satz 3.18. *Eine Matrix $A \in \mathbb{K}^{n\times n}$ besitzt genau dann eine Cholesky-Zerlegung, wenn A positiv definit ist.*

Um den Beweis konzeptionell anzugehen, stellen wir vor dem eigentlichen Beweis einige Vorüberlegungen an.

Lemma 3.19. *Seien $d \in \{1, \dots, n-1\}$ und*

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \in \mathbb{K}^{n\times n}, \tag{3.4}$$

mit $A_{11} \in \mathbb{K}^{d\times d}$ invertierbar, $A_{12} \in \mathbb{K}^{d\times(n-d)}$, $A_{21} \in \mathbb{K}^{(n-d)\times d}$ und $A_{22} \in \mathbb{K}^{(n-d)\times(n-d)}$. Dann gilt

$$A = \begin{pmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ 0 & S \end{pmatrix},$$

wobei

$$S := A_{22} - A_{21}A_{11}^{-1}A_{12} \in \mathbb{K}^{(n-d)\times(n-d)} \tag{3.5}$$

das sogenannte Schur-Komplement *von A_{11} in A ist.*

Beweis. Man multipliziert das Produkt der zwei Matrizen und erhält A. □

Lemma 3.20. *Sei $A \in \mathbb{K}^{n\times n}$ positiv definit. Stellt man A in der Form (3.4) dar, so ist A_{11} positiv definit und somit invertierbar. Daher ist das Schur Komplement S aus (3.5) wohldefiniert und ebenfalls positiv definit.*

Beweis. Aus

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = A = A^* = \begin{pmatrix} A_{11}^* & A_{21}^* \\ A_{12}^* & A_{22}^* \end{pmatrix}$$

folgt $A_{11}^* = A_{11}$, $A_{22}^* = A_{22}$ und $A_{12}^* = A_{21}$. Insbesondere ist A_{11} hermitesch und für beliebiges $x \in \mathbb{K}^d \setminus \{0\}$ gilt

$$0 < \begin{pmatrix} x \\ 0 \end{pmatrix}^* \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x \\ 0 \end{pmatrix} = x^* A_{11} x = \langle x, A_{11} x \rangle.$$

Also ist A_{11} positiv definit und somit insbesondere invertierbar. Ferner gilt

$$S^* = A_{22}^* - (A_{21} A_{11}^{-1} A_{12})^* = A_{22}^* - A_{12}^* (A_{11}^{-1})^* A_{21}^* = A_{22} - A_{21} A_{11}^{-1} A_{12} = S.$$

Also ist S hermitesch. Zu beliebigem $y \in \mathbb{K}^{n-d} \setminus \{0\}$ setze $x := -A_{11}^{-1} A_{12} y \in \mathbb{K}^d$. Dann

$$0 < \begin{pmatrix} x \\ y \end{pmatrix}^* \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}^* \begin{pmatrix} A_{11}x + A_{12}y \\ A_{21}x + A_{22}y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}^* \begin{pmatrix} 0 \\ Sy \end{pmatrix} = y^* S y = \langle y, Sy \rangle.$$

Das zeigt, dass S positiv definit ist. □

Beweis des Satzes 3.18. Wir haben bereits bemerkt, dass, falls A eine Cholesky-Zerlegung hat, A auch positiv definit ist. Es reicht als die andere Implikation zu zeigen.

Sei dazu A positiv definit. Dann gilt insbesondere

$$a_{jj} = \langle e_j, A e_j \rangle > 0.$$

Wir wollen nun eine Cholesky-Zerlegung von A konstruieren und gehen dabei per Induktion vor.

Induktionsanfang: Für $n = 1$ nehmen wir $C = (\sqrt{a_{11}})$, so dass $A = (a_{11}) = C^2 = CC^*$.

Induktionsschritt: Sei $n > 1$ und für positiv definite $(n-1) \times (n-1)$-Matrizen existiere immer eine Cholesky-Zerlegung. Wir betrachten die Blockzerlegung

$$A = \begin{pmatrix} a_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

mit $A_{22} \in \mathbb{K}^{(n-1)\times(n-1)}$. Aus $A = A^*$ folgt $A_{22} = A_{22}^*$ und $A_{12} = A_{21}^*$. Nach Lemma 3.20 ist nun das Schur-Komplement $S = A_{22} - \frac{1}{a_{11}} A_{21} A_{12}$ von (a_{11}) in A positiv definit und besitzt nach Induktionsannahme eine Cholesky-Zerlegung $S = C_S C_S^*$. Setze

$$c_{11} := \sqrt{a_{11}} \quad \text{und} \quad C := \begin{pmatrix} c_{11} & 0 \\ \frac{1}{c_{11}} A_{21} & C_S \end{pmatrix}.$$

Damit erhält man

$$CC^* = \begin{pmatrix} c_{11} & 0 \\ \frac{1}{c_{11}} A_{21} & C_S \end{pmatrix} \begin{pmatrix} c_{11} & \frac{1}{c_{11}} A_{12} \\ 0 & C_S^* \end{pmatrix} = \begin{pmatrix} c_{11}^2 & A_{12} \\ A_{21} & \frac{1}{c_{11}^2} A_{21} A_{12} + C_S C_S^* \end{pmatrix} = A\,.$$

Somit besitzt A eine Cholesky-Zerlegung. □

Satz 3.21. *Die Cholesky-Zerlegung einer positiv definiten Matrix ist eindeutig.*

Beweis. Die Einträge der Cholesky-Matrix C lassen sich durch zeilenweisen Vergleich bestimmen: Aus

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} c_{11} & 0 & \cdots & 0 \\ c_{21} & c_{22} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix} \begin{pmatrix} c_{11} & \overline{c_{21}} & \cdots & \overline{c_{n1}} \\ 0 & c_{22} & \cdots & \overline{c_{n2}} \\ 0 & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & c_{nn} \end{pmatrix}$$

folgt die Implikationskette

$$\begin{aligned} a_{11} &= c_{11}^2 && \Longrightarrow && c_{11} &= \sqrt{a_{11}}\,, \\ a_{21} &= c_{21}c_{11} && \Longrightarrow && c_{21} &= a_{21}/c_{11}\,, \\ a_{22} &= |c_{21}|^2 + c_{22}^2 && \Longrightarrow && c_{22} &= \sqrt{a_{22} - |c_{21}|^2}\,, \\ a_{31} &= c_{31}c_{11} && \Longrightarrow && c_{31} &= a_{31}/c_{11} \qquad \text{etc.} \end{aligned}$$

Zusammengefasst kann man die Einträge von C direkt mit folgendem Algorithmus berechnen:

$$c_{jj} := \sqrt{a_{jj} - \sum_{k=1}^{j-1} |c_{jk}|^2}, \qquad c_{ij} := \frac{1}{c_{jj}} \Big(a_{i,j} - \sum_{k=1}^{j-1} c_{ik} c_{jk}^* \Big).$$

Man muss hierbei die c_{ij}, $j = 1, \dots, i-1$, spaltenweise von links nach rechts berechnen. □

Satz 3.22. *Die Anzahl der benötigten Operationen des Algorithmus im Beweis von Satz 3.21 ist $O(n^3)$.*

Bemerkung 3.23. Hat man schon die LR-Zerlegung $A = LR$, so kann man die Cholesky-Zerlegung $A = CC^*$ daraus berechnen. Die Matrix L hat schon Einsen auf der Diagonale. Nun schreibt man $R = D\tilde{R}$ mit einer Diagonalmatrix D und einer unipotenten oberen Dreiecksmatrix $\tilde{R}$. Wir haben also $A = LD\tilde{R}$ mit einer unipotenten unteren Dreiecksmatrix L, einer Diagonalmatrix D und einer unipotenten oberen Dreiecksmatrix $\tilde{R}$. Die Diagonaleinträge d_{jj} von D müssen positiv sein. Aus $A = A^*$ folgt $\tilde{R} = L^*$. Also ist $A = LDL^*$. Definiert man nun $\sqrt{D}$ als Diagonalmatrix mit den Diagonaleinträgen $\sqrt{d_{jj}}$, so gilt

$$A = LDL^* = L\sqrt{D}\sqrt{D}^* L^* = (L\sqrt{D})(L\sqrt{D})^*.$$

Also ist $A = CC^*$ mit $C = L\sqrt{D}$.

3.4 Gradientenverfahren

Das Gradientenverfahren ist eine aus der Optimierung bekannte Methode um glatte Funktionen der Form $f : \mathbb{R}^n \to \mathbb{R}$ zu minimieren. Dabei wird eine Lösung der kritischen Gleichung $\nabla f(x) = 0$ gesucht, wobei $\nabla f(x) = (\partial f/\partial x_i)_i$ der Vektor der partiellen Ableitungen, auch *Gradient* genannt, ist.

Die Idee ist es ein iteratives Verfahren aufzusetzen, in dem in jedem Schritt lokal der Wert der Funktion f verringert wird. Für alle $v \in \mathbb{R}^n$ ist die Richtungsableitung von f in Richtung v, $\|v\|_2 = 1$, gegeben durch $\langle \nabla f(x), v\rangle$. Somit wird die Richtungsableitung minimal für $v = -\nabla f(x)/\|\nabla f(x)\|_2$. D.h. lokal wird der Wert der Funktion am meisten verringert, wenn wir uns in Richtung des negativen Gradienten bewegen. Diese Beobachtung führt zu folgendem Algorithmus.

Algorithmus : Gradientenverfahren
Eingabe : $f : \mathbb{R}^n \to \mathbb{R}$, Startwert $x_0 \in \mathbb{R}^n$
wiederhole
  $g_k := \nabla f(x_k)$;
  Ist $g_k = 0$, so gebe x_k aus;
  Sonst wähle α_k und setze $x_{k+1} := x_k - \alpha_k g_k$.
bis g_k *klein genug*;
Ergebnis : Approximation x_{k+1} der Lösung x von $\nabla f(x) = 0$.

Per Konstruktion berechnet die Methode kritische Punkte, also Lösungen der Gleichung $\nabla f(x) = 0$. Damit ein kritischer Punkt x ein lokales Minimum ist, muss zusätzlich die *Hessematrix* $\nabla^2 f(x) = (\partial f/\partial x_i)_i$ positiv definit sein. Beachte, dass das obige Verfahren ein Abstiegsverfahren ist (die Schrittrichtung ist immer in Richtung des Gradientenabstiegs) und daher Konvergenz nur in Richtung lokaer Minima gegeben ist.

Sei nun A eine symmetrische Matrix. Wir wollen in diesem Abschnitt das Gradientenverfahren auf die glatte Funktion

$$f(x) = \tfrac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle.$$

anwenden. Zunächst beobachten wir, dass

$$\nabla f(x) = \frac{1}{2}(Ax + A^*x) - b = Ax - b.$$

Die Funktion f hat also nur einen einzigen kritischen Punkt und dieser ist die Lösung des Gleichungssystems $Ax = b$. Die Hessematrix von f ist

$$\nabla^2 f(x) = A.$$

Die Lösung x ist also genau dann ein Minimum von f, wenn A positiv definit ist. Dies halten wir fest.

Lemma 3.24. *Ist $A \in \mathbb{R}^{n\times n}$ symmetrisch, so ist die Lösung von $Ax = b$ gleich dem eindeutigen kritischen Punkt der Funktion $f(x) := \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$. Ist A zudem positiv definit, so ist die Lösung von $Ax = b$ ein Minimum der Funktion f.*

Sei nun wie in obigen Algorithmus

$$g_k := \nabla f(x_k) = Ax_k - b.$$

Mit Hilfe der Kettenregel bestimmen wir nun die optimale Schrittweite α_k:

$$\begin{aligned} 0 &= \frac{\mathrm{d}}{\mathrm{d}\alpha}\big(f(x_k - \alpha_k g_k)\big)\Big|_{\alpha=\alpha_k} \\ &= -\langle \nabla f(x_k - \alpha_k g_k), g_k\rangle \\ &= -\langle A(x_k - \alpha_k g_k) - b, g_k\rangle \\ &= -\langle g_k, g_k\rangle + \alpha_k \langle Ag_k, g_k\rangle, \end{aligned}$$

Daraus folgt

$$\alpha_k = \frac{\langle g_k, g_k\rangle}{\langle Ag_k, g_k\rangle}.$$

Diese Wahl der Schrittweite führt zum Gradientenverfahren.

Algorithmus : Gradientenverfahren für $Ax = b$
Eingabe : $A \in \mathbb{R}^{n\times n}$ positiv definit, $b \in \mathbb{R}^n$, Startwert $x_0 \in \mathbb{R}^n$
wiederhole
  $g_k := Ax_k - b$;
  Ist $g_k = 0$, so ist x_k schon Lösung ($\Rightarrow$ Abbruch);
  Bestimme $\langle Ag_k, g_k\rangle$;
  $\alpha_k := \dfrac{\langle g_k, g_k\rangle}{\langle Ag_k, g_k\rangle}$;
  Setze $x_{k+1} := x_k - \alpha_k g_k$.
bis g_k *klein genug*;
Ergebnis : Approximation x_{k+1} der Lösung x von $Ax = b$.

Wir wollen jetzt die Konvergenzgeschwindigkeit des Verfahrens untersuchen. Dazu beobachten wir, dass, wenn A positiv definit ist, $\langle Ay, y\rangle$ eine Norm auf $\mathbb{R}^n$ ist. Diese bezeichnen wir mit

$$\|y\|_A := \sqrt{\langle Ay, y\rangle}.$$

Lemma 3.25. *Sei x die Lösung von $Ax = b$. Für alle $y \in \mathbb{R}^n$ gilt dann*

$$f(y) - f(x) = \tfrac{1}{2}\|x - y\|_A^2.$$

Beweis. Wir rechnen

$$\begin{aligned} f(y) - f(x) &= \tfrac{1}{2}\langle Ay, y\rangle - \tfrac{1}{2}\langle Ax, x\rangle + \langle x, b\rangle - \langle y, b\rangle \\ &= \tfrac{1}{2}\langle Ay, y\rangle - \tfrac{1}{2}\langle Ax, x\rangle + \langle x, Ax\rangle - \langle y, Ax\rangle \\ &= \tfrac{1}{2}\langle Ay, y\rangle + \tfrac{1}{2}\langle Ax, x\rangle - \tfrac{1}{2}\langle y, Ax\rangle - \tfrac{1}{2}\langle Ay, x\rangle \\ &= \tfrac{1}{2}\langle A(x - y), x - y\rangle = \tfrac{1}{2}\|x - y\|_A^2. \end{aligned}$$

□

Mit dieser Notation haben wir nun folgenden Satz.

Satz 3.26. *Das Gradientenverfahren konvergiert gegen die Lösung von $Ax = b$. Bezeichnet $x_0 \in \mathbb{R}^n$ den Startwert und x die Lösung, so gilt für die k-te Iterierte x_k die Fehlerabschätzung*

$$\|x_k - x\|_A \leq (1-\theta)^{k/2}\|x_0 - x\|_A,$$

wobei

$$\theta = \inf_{y \neq 0} \frac{\langle y, y\rangle^2}{\langle Ay, y\rangle \langle A^{-1}y, y\rangle}.$$

Es gilt $1/\mathrm{cond}_2(A) \leq \theta \leq 1$.

Beweis. Wir erinnern uns, dass $f(y) = \frac{1}{2}\langle Ay, y\rangle - \langle b, y\rangle$ und $g_k = Ax_k - b$.

Gilt $g_k = 0$, so ist x_k bereits die gesuchte Lösung und obige Ungleichung ist erfüllt. Wir nehmen daher nun an, dass für alle $\ell \leq k$ jeweils $g_\ell \neq 0$ gilt.

Unter Benutzung von $x_{k+1} = x_k - \alpha_k g_k$ folgt dann weiterhin

$$\begin{aligned}
f(x_k) - f(x_{k+1}) &= \tfrac{1}{2}\langle Ax_k, x_k\rangle - \langle b, x_k\rangle - \tfrac{1}{2}\langle A(x_k - \alpha_k g_k), x_k - \alpha_k g_k\rangle + \langle b, x_k - \alpha_k g_k\rangle \\
&= \tfrac{1}{2}\left(\langle Ax_k, \alpha_k g_k\rangle + \langle A(\alpha_k g_k), x_k\rangle - \langle A(\alpha_k g_k), \alpha_k g_k\rangle\right) - \alpha_k\langle b, g_k\rangle \\
&= \alpha_k\langle Ax_k, g_k\rangle - \tfrac{1}{2}\alpha_k^2\langle Ag_k, g_k\rangle - \alpha_k\langle b, g_k\rangle \\
&= \frac{\langle g_k, g_k\rangle}{\langle Ag_k, g_k\rangle}\langle Ax_k - b, g_k\rangle - \tfrac{1}{2}\frac{\langle g_k, g_k\rangle^2}{\langle Ag_k, g_k\rangle^2}\langle Ag_k, g_k\rangle \\
&= \tfrac{1}{2}\frac{\langle g_k, g_k\rangle^2}{\langle Ag_k, g_k\rangle}.
\end{aligned}$$

Mit Hilfe von Lemma 3.25 haben wir weiterhin

$$f(x_k) - f(x) = \tfrac{1}{2}\langle A(x_k - x), x_k - x\rangle = \tfrac{1}{2}\langle A^{-1}g_k, g_k\rangle,$$

so dass insgesamt

$$\frac{f(x_k) - f(x_{k+1})}{f(x_k) - f(x)} \geq \theta.$$

Für $0 < \theta \leq 1$ folgt

$$f(x_{k+1}) - f(x) = \big(f(x_k) - f(x)\big) - \big(f(x_k) - f(x_{k+1})\big) \leq (1-\theta)\big(f(x_k) - f(x)\big)$$

und induktiv

$$f(x_k) - f(x) \leq (1-\theta)^k \left(f(x_0) - f(x)\right).$$

Lemma 3.25 impliziert dann

$$\|x_k - x\|_A \leq (1-\theta)^{k/2}\|x_0 - x\|_A$$

umwandeln und somit folgt also auch $x_k \to x$.

Wir schätzen noch θ ab. Dazu sei $A = Q^T DQ$ mit einer Diagonalmatrix D und einer Orthogonalmatrix Q. Wir setzen $z = Qy$, so dass

$$\theta = \inf_{z \neq 0} \frac{\langle z, z\rangle^2}{\langle Dz, z\rangle \langle D^{-1}z, z\rangle} \leq \frac{\langle e_1, e_1\rangle^2}{\langle De_1, e_1\rangle \langle D^{-1}e_1, e_1\rangle} = 1.$$

Um θ abzuschätzen, können wir $\langle Dz, z\rangle \langle D^{-1}z, z\rangle$ unter der Bedingung $\|z\|_2 = 1$ maximieren. Sind λ_j die Eigenwerte von A und somit die Einträge von D, so können wir abschätzen

$$\langle Dz, z\rangle \langle D^{-1}z, z\rangle = \Big(\sum_j z_j^2 \lambda_j\Big)\Big(\sum_k z_k^2 \frac{1}{\lambda_k}\Big) \leq \max_j \lambda_j \max_k \frac{1}{\lambda_k} = \mathrm{cond}_2(D) = \mathrm{cond}_2(A)$$

für $\|z\|_2 = 1$ und somit $\theta \geq 1/\mathrm{cond}_2(A)$. □

Bemerkung 3.27. Obige Abschätzung von θ lässt sich durch die sogenenannte Kantorovich-Ungleichung verfeinern. Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch positiv definit mit größtem Eigenwert λ_n und kleinstem Eigenwert λ_1. Dann gilt

$$\begin{aligned}\theta = \inf_{y \neq 0} \frac{\langle y, y\rangle^2}{\langle Ay, y\rangle \langle A^{-1}y, y\rangle} \geq \left(\frac{\lambda_1 + \lambda_n}{2} \frac{1/\lambda_1 + 1/\lambda_n}{2}\right)^{-1} &= 4\frac{\lambda_1 \lambda_n}{(\lambda_1 + \lambda_n)^2} \\ &= 4\,\frac{\mathrm{cond}_2(A)}{(1 + \mathrm{cond}_2(A))^2}.\end{aligned}$$

Für große Konditionszahlen $\mathrm{cond}_2(A)$, liefert die Kantorovich-Ungleichung ein fast um den Faktor 4 besseres θ als oben. Mit der Abschätzung aus obigem Beweis gilt dann

$$\|x_k - x\|_A \leq (1-\theta)^{k/2} \|x_0 - x\|_A \leq \left(\frac{\mathrm{cond}_2(A) - 1}{\mathrm{cond}_2(A) + 1}\right)^k \|x_0 - x\|_A$$

Beispiel 3.28. Für

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad x_0 = \begin{pmatrix} 2 \\ \frac{1}{2} \end{pmatrix}$$

ergibt sich die Iterationsfolge in Abbildung 3.1. Die Richtung des steilsten Abstiegs ist stets senkrecht zur Niveaulinie durch die aktuelle Iterierte, somit entsteht eine Zickzacklinie mit rechtwinkligen Knicken und die Minimalstelle wird im Allgemeinen niemals nach endlich vielen Schritten erreicht.

In der Abbildung 3.1 lässt sich die Ursache für die vielen benötigten Iterationen im Beispiel 3.28 gut erkennen: Die Niveaulinien des Funktionals $y \mapsto f(y) - f(x)$ bzgl. der Euklidischen Norm sind Ellipsen mit Zentrum x, der Vektor $x_k - x_{k-1}$ (bzw. g_{k-1}) ist tangential und der Gradient g_k normal zur entsprechenden Ellipse im Punkt x_k; g_k weist also i.A. nicht in Richtung des Minimums x.

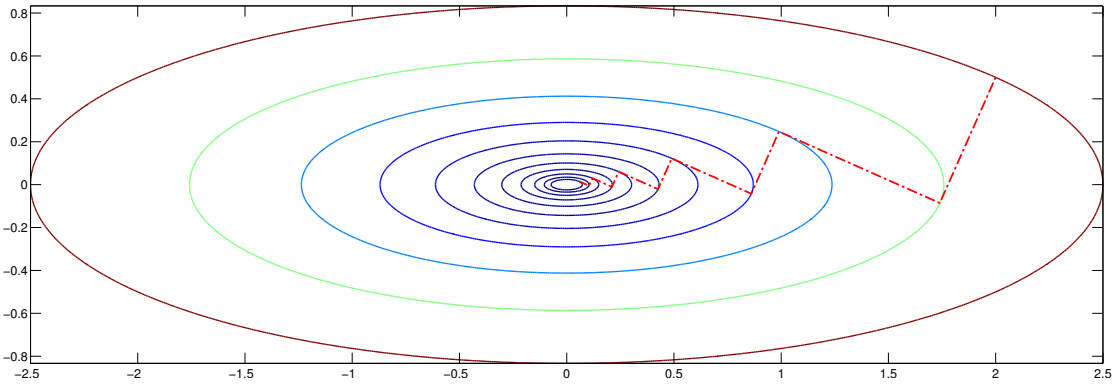


Abbildung 3.1: Iterationsfolge des Gradientenverfahren.

3.5 Lineare Ausgleichsprobleme

Wir wollen das überbestimmte Gleichungssystem

$$Ax = b \qquad \text{mit } A \in \mathbb{K}^{m\times n} \text{ und } m > n.$$

„lösen“. Da dies Gleichungssystem im Allgemeinen unlösbar ist, suchen wir stattdessen die Lösung des *kleinste Quadrate Problems*

$$\min_{x\in\mathbb{K}^n} \|Ax - b\|_2^2.$$

Die algebraische Lösung dieses Problems ist durch die *Pseudo-Inverse* von A gegeben.

Definition 3.29. Sei $A \in \mathbb{K}^{m\times n}$ mit $m > n$, so dass A^*A invertierbar ist. Dann heißt

$$A^\dagger := (A^*A)^{-1}A^*$$

die Pseudo-Inverse von A.

Lemma 3.30. *Es ist* $\ker A^*A = \ker A$. *Die Pseudo-Inverse von A existiert also genau dann, wenn* $\ker A = \{0\}$, *wenn also A vollen Rang hat.*

Beweis. Für alle $x \in \mathbb{K}^n$ gilt $\langle A^*Ax, x\rangle = \langle Ax, Ax\rangle = \|Ax\|_2^2$. Somit ist $A^*Ax = 0$ genau dann, wenn $Ax = 0$. D.h., $\ker A^*A = \ker A$. □

Wir beschreiben nun die allgemeine Lösung des Problems der kleinsten Quadrate. Sei dazu für einen linearen Unterraum $U \subseteq \mathbb{K}^n$ das *orthogonale Komplement* definiert als

$$U^\perp := \{y \in \mathbb{K}^n \mid \forall x \in U : \langle x, y\rangle = 0\}.$$

Wir benötigen folgendes Lemma.

Lemma 3.31. *Sei $A \in \mathbb{K}^{m\times n}$. Dann gilt*

$$\text{Bild}(A)^\perp = \ker(A^*) \quad \textit{und} \quad \ker(A)^\perp = \text{Bild}(A^*).$$

Beweis. Nach Definition gilt $\ker(A^*) = \{y \in \mathbb{K}^m \mid A^*y = 0\}$. Der Vektor $A^*y \in \mathbb{K}^n$ ist der Nullvektor genau dann, wenn $\langle x, A^*y\rangle = 0$ für alle $x \in \mathbb{K}^n$. Da $\langle x, A^*y\rangle = \langle Ax, y\rangle$, gilt insbesondere

$$\ker(A^*) = \{y \in \mathbb{K}^m \mid \forall x \in \mathbb{K}^n : \langle x, A^*y\rangle = 0\} = \mathrm{Bild}(A)^\perp.$$

Die zweite Aussage folgt, wenn wir A durch A^* ersetzen. □

Satz 3.32. *Sei $A \in \mathbb{K}^{m\times n}$ und $b \in \mathbb{K}^m$, so minimiert $x \in \mathbb{K}^n$ genau dann $\|Ax - b\|_2^2$, falls es die* Gaußsche Normalengleichung

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

erfüllt. Die Lösung dieses Gleichungssystem ist genau dann eindeutig, wenn A vollen Rang hat. In diesem Fall ist die eindeutige Lösung

$$x = A^\dagger b.$$

Beweis. Sei $x \in \mathbb{K}^n$ ein Minimum von $\|Ax - b\|_2^2$. Sei $b_0 := Ax$. Dann ist $b_0 \in \mathrm{Bild}(A)$ der Punkt im Bild von A, der den Abstand zu b minimiert (siehe die Illustration in Abbildung 3.2), was impliziert, dass $b - b_0 \in \mathrm{Bild}(A)^\perp$. Nach Lemma 3.31 gilt dann $A^*(b - b_0) = 0$, so dass

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

Also erfüllt x die Normalengleichung.

Sei andererseits x eine Lösung der Normalengleichung, d.h. es gelte $A^*(b - Ax) = 0$. Dann gilt für alle $y \in \mathbb{K}^n$

$$\begin{aligned}\|b - Ay\|_2^2 &= \|b - Ax\|_2^2 + \|A(x - y)\|_2^2 + \langle b - Ax, A(x - y)\rangle + \langle A(x - y), b - Ax\rangle \\ &\geq \|b - Ax\|_2^2.\end{aligned}$$

Dies zeigt, dass die Lösungen der Normalengleichung Minima der Funktion $f(x)$ sind.

Hat A vollen Rank, so gilt nach Lemma 3.30 für jeden kritischen Punkt x, dass $x = A^\dagger b$. Das Minimum ist also eindeutig. □

Abbildung 3.2 illustriert die Bedeutung der Pseudo-Inversen $A^\dagger = (A^*A)^{-1}A^*$. Wie im Beweis von Satz 3.32 sei $x = A^\dagger b$ und $b_0 = Ax$. Dann ist $b - b_0 \in \ker(A^*) = \mathrm{Bild}(A)^\perp$. D.h. b_0 ist der eindeutige Punkt in $\mathrm{Bild}(A)$, der den Abstand zu b_0 minimiert. In anderen Worten, b_0 ist die orthogonale Projektion von b auf $\mathrm{Bild}(A)$.

Die Pseudo-Inverse $A^\dagger \in \mathbb{K}^{m\times n}$ von $A \in \mathbb{K}^{m\times n}$ hat die folgende Wirkung angewandt auf einen Vektor $b \in \mathbb{K}^n$. Sie projiziert zuerst $b \in \mathbb{R}^n$ orthogonal auf $b_0 \in \mathrm{Bild}(A)$ und bildet danach b_0 auf den eindeutigen Punkt $x = A^\dagger b \in \mathrm{Bild}(A^*)$ mit $Ax = b_0$ ab.

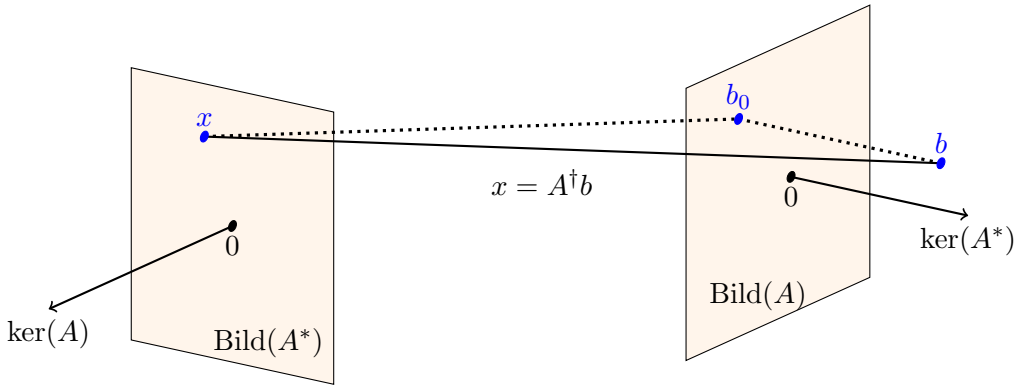


Abbildung 3.2: Geometrische Interpretation der Pseudo-Inversen

Zuletzt wollen wir eine explizite Methode angeben, die Pseudo-Inverse zu berechnen. Sei dazu $A \in \mathbb{K}^{m\times n}$ von vollem Rang und

$$A = U\Sigma V^*$$

die Singulärwertzerlegung von A und seien $\sigma_1 \geq \ldots \geq \sigma_n$ die Singulärwerte von A. Dann gilt $A^*A = V^*\Sigma^*\Sigma V$. Somit

$$A^\dagger = (A^*A)^{-1}A^* = V\Sigma^\dagger U^*.$$

Beachte, dass $\Sigma^\dagger = (\Sigma^*\Sigma)^{-1}\Sigma$ eine Matrix mit den Werten $\sigma_i^{-1}, i = 1,\ldots,n$ auf der Hauptdiagonalen ist. Somit sind die Singulärwerte von $A^\dagger$ genau $\sigma_n^{-1} \geq \ldots \geq \sigma_1^{-1}$.

3.6 Die QR-Zerlegung

Häufig ist das Problem der Lösung der Normalengleichung $A^*Ax = A^*b$ schlechter konditioniert, als das eigentliche kleinste Quadrate Minimierungsproblem. Dies liegt daran, dass, wie in der Diskussion am Ende des letzten Abschnitts gezeigt, die Singulärwerte von A^*A genau $\sigma_1^2,\ldots,\sigma_n^2$ sind, wobei $\sigma_1,\ldots,\sigma_n$ die Singulärwerte von A sind. Nach Satz 2.27 ist die Konditions des Normalengleichung bestimmt durch $\mathrm{cond}_2(A^*A) = (\sigma_1/\sigma_n)^2 = \mathrm{cond}(A)^2$. Andereseits hat $A^\dagger$ die Singulärwerte σ_i^{-1}. D.h., die Kondition des Problems $x \mapsto A^\dagger b$ ist bestimmt durch $\mathrm{cond}(A)$. Als Faustregel kann man also festhalten, dass die Konditionszahl die Normalengleichung zu lösen ungefähr das Quadrat der Kondition des linearen Ausgleichproblems ist.

Indem wir die Normalengleichung lösen, um das Ausgleichsproblem lösen, transformieren wir ein gut gestelltes Problem in ein (möglicherweise) schlecht gestelltes Problem.

Daher wollen wir ein Verfahren angeben, das die Normalengleichung vermeidet. Eine Möglichkeit wäre es die Singulärwertzerlegung von A zu berechnen und daraus $A^\dagger$ zu bestimmen. Die Lösung des Problems ist dann, wie in Satz 3.32 gezeigt, $x = A^\dagger b$. Eine alternative Möglichkeit ist es, die *QR-Zerlegung* von A zu berechnen. Dies erfordert weniger Rechenoperationen als die Bestimmung der SVD.

Wir benötigen den folgenden Darstellungssatz.

Satz 3.33 (QR-Zerlegung)**.** *Sei $A \in \mathbb{K}^{m\times n}$, $m \geq n$ mit vollem Rang n. Dann existiert eine Darstellung $A = QR$ mit einem unitären $Q \in \mathbb{K}^{m\times m}$, d.h. $Q^*Q = \mathrm{I}$, und einer oberen Dreiecksmatrix $R \in \mathbb{K}^{m\times n}$ mit $r_{jj} > 0$ für $j = 1, \ldots, n$.*

Beweis. Seien $a_1, \ldots, a_n$ die Spaltenvektoren von A. Da A vollen Spaltenrang hat, sind die Vektoren linear unabhängig. Wir können sie durch geeignete Vektoren $a_{n+1}, \ldots, a_m$ zu einer Basis des $\mathbb{K}^m$ ergänzen und auf diese das Gram-Schmidt-Orthonormalisierungs-Verfahren anwenden:

$$\begin{aligned} u_1 &:= a_1, & e_1 &:= \frac{u_1}{\|u_1\|_2}, \\ u_k &:= a_k - \sum_{j<k} \langle e_j, a_k\rangle e_j, & e_k &:= \frac{u_k}{\|u_k\|_2}. \end{aligned}$$

Dies führt zu

$$a_k = \sum_{j\leq k} \langle e_j, a_k\rangle e_j.$$

Daraus folgt für $k = 1, \ldots, n$

$$r_{kk} := \langle e_k, a_k\rangle = \langle e_k, a_k\rangle - \sum_{j<k} \langle e_j, a_k\rangle\langle e_k, e_j\rangle = \langle e_k, u_k\rangle = \langle e_k, \|u_k\|_2 e_k\rangle = \|u_k\|_2 > 0.$$

Nun rechnet man direkt nach, dass die Matrizen

$$Q := \begin{pmatrix} e_1, & \ldots, & e_n, & e_{n+1} & \ldots, & e_m \end{pmatrix}, \quad R := \begin{pmatrix} \langle e_1, a_1\rangle & \langle e_1, a_2\rangle & \ldots & \langle e_1, a_n\rangle \\ 0 & \langle e_2, a_2\rangle & \ldots & \langle e_2, a_n\rangle \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ldots & \langle e_n, a_n\rangle \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & 0 \end{pmatrix}$$

die gewünschte Eigenschaft haben. □

Sei $A = QR$ wie in Satz 3.33. Dann hat R die Gestalt

$$R = \begin{pmatrix} \widetilde{R} \\ 0 \end{pmatrix} \tag{3.6}$$

mit $\widetilde{R} \in \mathbb{K}^{n\times n}$ regulär. Zerlegt man analog $Q = (Q_1 Q_2)$ mit $Q_1 \in \mathbb{K}^{m\times n}$, so hat man

$$A = QR = Q_1 \widetilde{R}.$$

Es folgt $\mathrm{Bild}(A) = \mathrm{Bild}(Q_1)$ und $\mathrm{Kern}(A^*) = \mathrm{Kern}(Q_1^*)$, d.h.

$$\mathrm{Bild}(A)^\perp = \mathrm{Bild}(Q_1)^\perp = \mathrm{Bild}(Q_2).$$

Satz 3.34. *Sei $A = QR$ wie in Satz 3.33, $\widetilde{R} \in \mathbb{K}^{n\times n}$ wie in (3.6) und $Q = (Q_1\ Q_2)$ mit $Q_1 \in \mathbb{K}^{m\times n}$. Sei*

$$d_1 := Q_1^* b \quad \textit{und} \quad d_2 := Q_2^* b.$$

Dann ist die Lösung des kleinste Quadrate Problems gegeben durch

$$x = \widetilde{R}^{-1} d_1.$$

Das Residuum erfüllt $b - Ax = Q_2 d_2$.

Beweis. Da Q unitär gilt

$$\|b - Ax\|_2^2 = \|Q^*(b - Ax)\|_2^2 = \left\| \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} - \begin{pmatrix} \widetilde{R}x \\ 0 \end{pmatrix} \right\|_2^2 = \|d_1 - \widetilde{R}x\|_2^2 + \|d_2\|_2^2.$$

Das Minimum wird für $\widetilde{R}x = d_1$ angenommen. Weiterhin gilt

$$\begin{aligned} b = QQ^*b &= \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} Q_1^* \\ Q_2^* \end{pmatrix} b = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = Q_1 d_1 + Q_2 d_2 \\ &= Q_1 \widetilde{R}x + Q_2 d_2 \\ &= Ax + Q_2 d_2. \end{aligned}$$

□

Satz 3.34 führt zu folgendem Algorithmus zur Lösung des Ausgleichproblems.

Algorithmus : Lösung des Ausgleichproblems mittels QR Zerlegung
Eingabe : $A \in \mathbb{K}^{m\times n}$, $m \geq n$, von Rang n und $b \in \mathbb{K}^m$.
Berechne die QR-Zerlegung $A = QR$;
Setze $Q_1 \in \mathbb{K}^{m\times n}$ als den linken $m \times n$ Block von Q;
Setze $d_1 = Q_1^* b$;
Löse das Gleichungssystem $\tilde{R}x = d_1$, wobei $\tilde{R}$ wie in (3.6) definiert ist;
Ergebnis : Die eindeutige Lösung x des Ausgleichproblems $\min_x \|Ax - b\|_2$.

Eine QR-Zerlegung wie in Satz 3.33 kann mit Hilfe des Gram-Schmidt-Verfahrens berechnet werden (wie im Beweis von Satz 3.33) ausgeführt. In der Praxis wird die QR-Zerlegung jedoch häufig mit Hilfe sogenannter *Householder-Transformationen* durchgeführt. Die zugrunde liegende geometrische Idee dieses Ansatzes ist es, dass jede Rotation oder Spiegelung, beschrieben durch eine unitäre/orthogonale Matrix $Q \in \mathbb{K}^{n\times n}$, sich Hintereinanderausführung von *Reflektionen* ausdrücken lässt.

Definition 3.35. Sei $v \in \mathbb{K}^n \setminus \{0\}$. Dann ist die *Householder-Matrix* von v definiert als

$$H_v := \mathrm{I} - \frac{2}{\langle v, v\rangle} vv^*.$$

Die Householder Matrix von v hat die Eigenschaft, dass $H_v v = -v$ und $H_v u = u$ für $u \in (\mathbb{K}v)^\perp$. Insbesondere beschreibt also H_v eine Reflektion an $(\mathbb{K}v)^\perp$. Reflektionen sind normerhaltend, also ist die Matrix H_v unitär/orthogonal. Wir beweisen dies formal.

Lemma 3.36. *Sei $v \in \mathbb{K}^n \setminus \{0\}$. Dann ist H_v unitär/orthogonal.*

Beweis. Es ist $H_v^* H_v = \mathrm{I} - \frac{4}{\langle v,v\rangle} vv^* + \frac{4}{\langle v,v\rangle^2} vv^*vv^* = \mathrm{I}$. □

Eine wichtige Eigenschaft für die Berechnung von QR-Zerlegung mittels Householder-Transformationen ist das folgende Lemma. Hier müssen wir $\mathbb{K} = \mathbb{R}$ annehmen.

Lemma 3.37. *Seien $u, v \in \mathbb{R}^n, u \neq v$ mit $\|u\|_2 = \|v\|_2$. Dann gilt $H_{u-v} u = v$.*

Beweis. Es ist $H_{u-v}\, u = u - 2\alpha(u - v)$, wobei

$$\alpha = \frac{\langle u - v, u\rangle}{\langle u - v, u - v\rangle} = \frac{\|u\|_2^2 - \langle v, u\rangle}{\|u\|_2^2 + \|v\|_2^2 - 2\langle u, v\rangle} = \frac{1}{2}. \tag{3.7}$$

Es folgt, dass $H_{u-v}\, u = v$. □

Bemerkung 3.38. Die letzte Gleichheit in (3.7) gilt nicht im Fall $\mathbb{K} = \mathbb{R}$. Im komplexen Fall müssen wir daher mit einer anderen Klasse von Matrizen arbeiten. Diese sind wie folgt definiert: $\langle u, v\rangle = e^{i\phi} \cdot |\langle u, v\rangle|$, wobei ϕ das Argument der komplexen Zahl ist, und $w := e^{i\phi} \cdot u - v$. Wir setzen $U_w := e^{i\phi} H_w$. Dann gilt, falls $\|u\|_2 = \|v\|_2$, dass $U_w\, u = v$

Lemma 3.37 wird nun für die Berechnung der QR-Zerlegung von $A \in \mathbb{R}^{n\times n}$ wie folgt ausgenutzt. Sei a_1 der erste Spaltenvektor von A. Dann setzen wir $w_1 := \|a_1\|_2 \cdot e_1 \in \mathbb{R}^n$ und berechnen die Matrix

$$P_1 := H_{a_1 - w_1} \in \mathbb{R}^{n\times n}.$$

Dann hat die erste Spalte von

$$A_1 := P_1 A$$

nur Nulleinträge unterhalb der Diagonalen. Sei nun $a_2 \in \mathbb{R}^{n-1}$ der untere $(n-1)$-Vektor der zweiten Spalte von A_1. Im nächsten Schritt definieren wir $w_2 := \|a_2\|_2 \cdot e_1 \in \mathbb{R}^{n-1}$, $P_2' := H_{a_2 - w_2} \in \mathbb{R}^{(n-1)\times(n-1)}$ und

$$P_2 = \begin{pmatrix} 1 & 0 \\ 0 & P_2' \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

Dann haben die erste und zweite Spalte von

$$A_2 = P_2 A_1 = P_2 P_1 A$$

nur Nulleinträge unterhalb der Diagonalen. Dieses Verfahren wird induktiv fortgesetzt. Am Ende erhalten wir eine obere Dreiecksmatrix R mit

$$A = QR, \qquad Q = (P_n \cdots P_1)^*.$$

Um Auslöschung zu vermeiden, kann bei der Definition der w_i das Vorzeichen angepasst werden.

Ist nur nach der Matrix R gefragt, brauchen wir die Matrizen P_i nicht zu speichern. Stattdessen kann iterativ in jedem Schritt jeweils nur *eine* Addition und eine Norm (bei der Berechnung von $w_i - a_i$) sowie ein Matrix-Vektor Produkt und ein äußeres Produkt (bei der Berechnung von $P_i A_i$) berechnet werden. Diese Strategie macht die Berechnung der QR-Zerlegung mittels Householder Matrizen numerisch außerordentlich stabil. Für mehr Details siehe [5, Kapitel 19.3].

3.7 Nichtlineare Gleichungen

Zum Abschluss dieses Kapitels wollen wir noch Algorithmen zum Lösen *nichtlinearer* Gleichungen untersuchen. Dies sind Gleichungen der Form

$$f(x) = 0,$$

wobei

$$f : U \to \mathbb{K}^m, \qquad U \subseteq \mathbb{K}^n,$$

eine Abbildung ist. Wir werden drei Verfahren kennenlernen, die jeweils unterschiedliche Anforderungen an f stellen.

Das erste Verfahren ist das *Bisektionsverfahren* für den Fall $\mathbb{K} = \mathbb{R}$ und $m = n = 1$. Hierfür müssen wir annehmen, dass $f : [a, b] \to \mathbb{R}$ stetig mit Vorzeichenwechsel ist, d.h. $f(a)f(b) < 0$. Der Zwischenwertsatz garantiert dann die Existenz eines Punktes $\zeta \in [a, b]$ mit $f(\zeta) = 0$. Das Bisektionsverfahren funktioniert dann wie folgt.

Algorithmus : Bisektionsverfahren
Eingabe : Startintervall $[a_0, b_0] := [a, b]$
wiederhole
  $x_n := \frac{a_n + b_n}{2}$;
  Falls $f(x_n) = 0$, so sind wir fertig;
  wenn $f(a_n)$ *und* $f(x_n)$ *verschiedene Vorzeichen haben* **dann**
    $[a_{n+1}, b_{n+1}] := [a_n, x_n]$.
  sonst
    $[a_{n+1}, b_{n+1}] := [x_n, b_n]$.
  Ende
  $n := n + 1$;
bis $|b_n - a_n|$ *oder* $|f(a_n)|$ *klein genug*;
Ergebnis : $[a_n, b_n]$ enthält Nullstelle.

Per Konstruktion gilt im Bisektionsverfahren $a_n, b_n \to \zeta$ mit $f(\zeta) = 0$. Um die Qualität der Konvergenz zu beschreiben führen wir folgende Terminologie ein.

Definition 3.39. Sei $(x_n)_{n\in\mathbb{N}}$ eine konvergente Folge in $\mathbb{K}^n$ mit Grenzwert ζ. Wir nennen

$$\kappa := \limsup_{k\to\infty} \|x_k - \zeta\|^{1/k}$$

den *asymptotischen Konvergenzfaktor* der Folge.

$$\text{Falls } \begin{cases} \kappa = 1, \\ 0 \le \kappa < 1, \\ \kappa = 0, \end{cases} \quad \text{sprechen wir von } \begin{cases} \text{sublinearer Konvergenz.} \\ \text{linearer Konvergenz.} \\ \text{superlinearer Konvergenz.} \end{cases}.$$

Existieren im superlinear konvergenten Fall zudem Konstanten $c > 0$, $p > 1$ mit

$$\|x_{k+1} - x\| \le c\|x_k - x\|^p,$$

so sagen wir, die Folge x_n hat die *Konvergenzordnung* p.

Lemma 3.40. *Sei $(x_n)_{n\in\mathbb{N}_0}$ eine Folge, die gegen $\zeta \in \mathbb{K}^n$ konvergiert.*

a) Falls $c \in (0,1)$ existiert mit

$$\|x_{n+1} - \zeta\| \le c\|x_n - \zeta\|, \tag{3.8}$$

dann konvergiert (x_n) linear gegen ζ.

b) Falls $c > 0$ und $p > 1$ existieren mit $\|x_0 - \zeta\| < c^{-\frac{1}{p-1}}$ und

$$\|x_{k+1} - \zeta\| \le c\|x_k - \zeta\|^p,$$

dann konvergiert (x_n) superlinear mit Konvergenz der Ordnung p gegen ζ.

Beweis. Aus (3.8) folgt $\|x_k - \zeta\| \le c^k\|x_0 - x\|$ und somit gilt für den asymptotischen Konvergenzfaktor $\kappa \le \limsup_k c\|x_0 - \zeta\|^{1/k} = c$.

Die die zweite Aussage sei $c_1 := c^{\frac{1}{p-1}}$ und $\alpha_k := c_1\|x_k - \zeta\|$. Dann gilt

$$\alpha_{k+1} = c_1\|x_{k+1} - \zeta\| \le c_1 c\|x_k - \zeta\|^p = c_1 c_1^{p-1}\|x_k - \zeta\|^p = c_1^p\|x_k - \zeta\|^p = \alpha_k^p.$$

Mit Induktion folgt $\alpha_k \le \alpha_0^{p^k}$. Anders formuliert gilt $\|x_k - \zeta\| \le \frac{1}{c_1}(c_1\|x_0 - \zeta\|)^{p^k}$ für alle $k \in \mathbb{N}$. Falls $\|x_0 - \zeta\| < c^{-\frac{1}{p-1}} = \frac{1}{c_1}$ gilt somit $\kappa \le \limsup_k(\frac{1}{c_1}\big(c_1\|x_0 - \zeta\|\big)^{p^k})^{1/k} = 0$, also haben wir in diesem Fall superlineare Konvergenz. □

Korollar 3.41. *Das Bisektionsverfahren hat immer (zumindest) lineare Konvergenz.*

Beweis. Für das Bisektionsverfahren gilt $|a_{n+1} - \zeta| \le (1/2)|a_n - x|$. Die Aussage folgt dann mit Lemma 3.40 a). □

Als Nächstes betrachten wir das Sekantenverfahren Wir suchen wieder die Nullstelle einer Funktion $f : \mathbb{R} \to \mathbb{R}$. Beim Sekantenverfahren beginnt man mit zwei Startwerten x_0 und x_1 und nimmt die Nullstelle der Sekante als nächsten Wert, d.h.

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

Dieses Verfahren konvergiert (lokal) mit Ordnung $\frac{1+\sqrt{5}}{2} \approx 1,6180$.

Zuletzt studieren wir das *Newton-Verfahren*. In diesem Fall nehmen wir eine Funktion

$$f : U \to \mathbb{C}^n, \quad U \subseteq \mathbb{C}^n \text{ offen.}$$

Wir suchen wieder eine Nullstelle von f und haben eine approximierte Position x gegeben. Falls f komplex differenzierbar in x ist, ist die Idee, dass f lokal durch die Tangente approximiert wird, d.h.

$$f(x + h) \approx f(x) + Jf(x)h,$$

wobei $Jf(x)$ die Jacobi-Matrix von f an der Stelle x ist. Dadurch liegt es nahe, die Nullstelle von f durch die Nullstelle der Tangente zu approximierten. D.h. wir suchen h, so dass $f(x + h) = 0$. Falls die Jacobi-Matrix invertierbar ist, können wir eindeutig lösen: $h = -Jf(x)^{-1}f(x)$. Wir erhalten eine Folge von Punkten rekursiv definiert durch $x_{n+1} = x_n - Jf(x_n)^{-1}f(x_n)$, falls $Jf(x_n)$ invertierbar, und $x_{n+1} = x_n$, falls die Jacobi-Matrix nicht invertierbar ist.

Definition 3.42. Sei $f : U \to \mathbb{C}^n, U \subseteq \mathbb{C}^n$, analytisch[1]. Der *Newton-Operator* von f ist

$$N_f : U \to \mathbb{C}^n, \quad x \mapsto N_f(x) = x - Jf(x)^{-1}f(x),$$

falls $Jf(x)$ invertierbar ist und $N_f(x) = x$ sonst.

Falls $N_f(\mathbb{C}^n) \subseteq U$, liefert dies das Newton-Verfahren.

Algorithmus : Newton-Verfahren
Eingabe : Startwert x_0
wiederhole
  Setze $x_{n+1} = N_f(x_n)$;
  $n := n + 1$;
bis $f(x_n)$ *oder* $x_{n+1} - x_n$ *klein genug*;
Ergebnis : x_n approximiert die Nullstelle.

Bemerkung 3.43. Wir nennen f eine reelle Abbildung, falls $f(\mathbb{R}^n) \subseteq \mathbb{R}^n$. Ein Beispiel sind reelle Polynome. Falls f eine reelle Abbildung ist, so ist auch N_f eine reelle Abbildung. In diesem Fall ergibt das Newton-Verfahren mit Startpunkt in $\mathbb{R}^n$ eine Folge von reellen Punkten und, falls das Newton-Verfahren konvergiert, konvergiert es somit gegen eine reelle Nullstelle.

[1] Erinnerung: $f : U \to \mathbb{C}^n$ ist genau dann analytisch, wenn f in allen Punkten in U komplex differenzierbar ist.

Beispiel 3.44. (Das Heron-Verfahren) Im Fall $n = 1$ wollen wir $\sqrt{a}$ für gegebenes $a \geq 0$ ausrechnen, indem wir eine Nullstelle von $f(x) := x^2 - a$ mit Hilfe des Newton-Verfahrens berechnen. In diesem Spezialfall heißt das Newton-Verfahren auch *Heron-Verfahren*.

Es gilt $Jf(x) = f'(x) = 2x$. Falls $x_n \neq 0$ ist $f'(x_n)$ invertierbar. In diesem Fall ist die Folge im Newton-Verfahren gegeben durch

$$x_{n+1} = x_n - \frac{f(x)}{f'(x)} = x_n - \frac{x_n^2 - a}{2x_n} = \frac{1}{2}\Big(x_n + \frac{a}{x_n}\Big).$$

Sei z.B. $a = 9$ und $x_0 = 9$. Dann ist

$$\begin{aligned} x_1 &= \tfrac{1}{2}(9 + \tfrac{9}{9}) = 5, \\ x_2 &= \tfrac{1}{2}(5 + \tfrac{9}{5}) = \tfrac{34}{10} = 3,4, \\ x_3 &= \tfrac{1}{2}(\tfrac{34}{10} + \tfrac{9 \cdot 10}{34}) \approx 3,0235 \\ x_4 &\approx 3,0009. \end{aligned}$$

Angenommen, es gelte $x_n \in [\frac{\sqrt{a}}{2}, 2\sqrt{a}]$. Dann ist auch $\frac{a}{x_n}$ aus $[\frac{\sqrt{a}}{2}, 2\sqrt{a}]$ und somit auch der Mittelwert x_{n+1} . Es gilt

$$\frac{x_{n+1} - \sqrt{a}}{x_{n+1} + \sqrt{a}} = \frac{x_n + \frac{a}{x_n} - 2\sqrt{a}}{x_n + \frac{a}{x_n} + 2\sqrt{a}} = \frac{x_n^2 - 2\sqrt{a}x_n + a}{x_n^2 + 2\sqrt{a}x_n + a} = \left(\frac{x_n - \sqrt{a}}{x_n + \sqrt{a}}\right)^2.$$

Insbesondere gilt für den absoluten Fehler

$$|x_{n+1} - \sqrt{a}| = \left|\frac{x_n - \sqrt{a}}{x_n + \sqrt{a}}\right|^2 |x_{n+1} + \sqrt{a}| \leq \frac{4}{3\sqrt{a}} |x_n - \sqrt{a}|^2$$

Nach Lemma 3.40 konvergiert das Heron-Verfahren für $x_0 \in (\frac{1}{2}\sqrt{a}, \frac{7}{4}\sqrt{a})$ superlinear mit Ordnung 2 (wir sagen auch es konvergiert *quadratisch*). Zum Schluss berechnen wir noch den relativen Fehler:

$$\frac{|x_{n+1} - \sqrt{a}|}{\sqrt{a}} \leq \frac{4}{3}\left(\frac{|x_n - \sqrt{a}|}{\sqrt{a}}\right)^2.$$

Tatsächlich konvergiert das Heron-Verfahren für jeden Startwert $x_0 \neq 0$ immer (wenn auch am Anfang nicht unbedingt mit quadratischer Konvergenzgeschwindigeit). Dies ist im Allgemeinen für das Newton-Verfahren nicht erfüllt.

Bemerkung 3.45 (Horner Schema)**.** Betrachten wir den Spezialfall der Nullstellenbestimmung eines Polynoms $p(x) = a_0 + a_1 x + \cdots + a_n x^n$. Im Newton-Verfahren müssen $p(x)$ und $p'(x)$ ausgerechnet werden. Dies geschieht am besten mit dem *Horner Schema*

$$p(x) = a_0 + x(a_1 + x(a_2 + x(\ldots(a_{n-1} + xa_n)))).$$

Dies kann man analog für $p'(x)$ machen. Dies führt zu folgendem Schema: Definiert man $\alpha_{n+1} := \beta_{n+1} := 0$ und

$$\begin{aligned} \alpha_k &:= a_k + \alpha_{k+1} x, \\ \beta_k &:= \alpha_k + \beta_{k+1} x, \end{aligned}$$

dann ist $\alpha_0 = p(x)$ und $\beta_1 = p'(x)$. Diese Method funktioniert verbatim für System von Polynomen.

Beispiel 3.46. Sei $A \in \mathbb{C}^{n\times n}$. Wir wollen A^{-1} mittels Newton Verfahren berechnen. Sei hierzu $f(X) := X^{-1} - A$ für $X \in \mathbb{C}^{n\times n}$ invertierbar. Wir haben im Beweis von Satz 2.29 gezeigt, dass $Df(X)\,Y = -X^{-1}YX^{-1}$. Auflösen der Gleichung $Z = -X^{-1}YX^{-1}$ nach Y liefert

$$Df(X)^{-1}\,Z = -XZX.$$

Hieraus erhalten wir die Formel für die Folge im Newton-Verfahren:

$$\begin{aligned} X_{n+1} &= X_n - Df(X_n)^{-1}(f(X_n)) \\ &= X_n + X_n\, f(X_n)\, X_n \\ &= X_n + X_n(X_n^{-1} - A)X_n \\ &= 2X_n - X_nAX_n \\ &= X_n(2\,\mathrm{I} - AX_n). \end{aligned}$$

Kommen wir nun zurück zum allgemeinen Newton-Verfahren. Wir wollen zeigen, dass das Newton-Verfahren für gut gewählte Startpunkte konvergiert. Dazu nehmen wir an, dass f analytisch ist (insbesondere also unendlich oft differenzierbar). In diesem Fall definieren wir den *Tensor* $D^k f(x)$ der höheren Ableitungen von f. Der Tensor ist eine multilineare Abbildung

$$D^k f(x) : (\mathbb{C}^n)^k \to \mathbb{C}^n,$$

definiert durch

$$D^k f(x)(e_{i_1}, \dots, e_{i_k}) = \frac{\partial^k f}{\partial x_{i_1} \cdots \partial x_{i_k}},$$

wobei die e_{i_j} die Standardbasisvektoren bezeichnen. Analog zur Operatornorm definieren wir die Tensornorm via

$$\|D^k f(x)\| := \max_{\|v_1\|=\cdots\|v_k\|=1} \|D^k f(x)(v_1, \dots, v_k)\|.$$

Die entscheidende Größe, um die Konvergenz des Newton-Verfahrens zu untersuchen, ist dann *Smale's Gamma Zahl*:

$$\gamma(f, x) := \sup_{k\geq 2} \frac{1}{k!}\|Df(x)^{-1}D^k f(x)\|^{\frac{1}{k-1}},$$

wobei, wie zuvor, $Df(x)$ die Ableitung von f als lineare Ableitung beschreibt (die Matrixdarstellung von $Df(x)$ ist die Jacobi-Matrix $Jf(x)$). Falls $Df(x)$ nicht invertierbar ist, $\gamma(f, x) := \infty$. Für den Beweis des nächsten Satzes verweisen wir auf [1, Kapitel 15].

Satz 3.47. *Sei $f : \mathbb{C}^n \to \mathbb{C}^n$ analytisch und $\zeta \in \mathbb{C}^n$ eine Nullstelle von f mit $Df(\zeta)$ invertierbar. Falls*

$$\|x_0 - \zeta\| \leq \frac{3 - \sqrt{7}}{2\,\gamma(f, \zeta)},$$

konvergiert das Newton-Verfahren mit Startpunkt x_0 quadratisch gegen ζ.

Korollar 3.48. *Seien ζ, ξ zwei verschiedene Nullstellen von f. Dann gilt*

$$\|\zeta - \xi\| > \frac{3-\sqrt{7}}{2\,\min\{\gamma(f,\zeta),\gamma(f,\xi)\}}.$$

Beweis. Ohne Einschränkung sei $\min\{\gamma(f,\zeta),\gamma(f,\xi)\} = \gamma(f,\zeta)$. Angenommen die Ungleichung ist nicht erfüllt. Dann gilt nach Satz 3.47, dass das Newton-Verfahren mit Startpunkt $x_0 = \xi$ nach ζ konvergiert. Es gilt aber $x_1 = \xi - Jf(\xi)^{-1}f(\xi) = \xi$ und entsprechend $x_n = \xi$ für alle n. Daraus folgt $\zeta = \xi$, was ein Widerspruch zur Annahme ist. □

Korollar 3.49. *Angenommen f ist eine reelle Abbildung ($f(\mathbb{R}^n) \subseteq \mathbb{R}^n$). Sei $\zeta \in \mathbb{C}^n \setminus \mathbb{R}^n$ eine nicht-reelle Nullstelle von f. Dann gilt*

$$\min_{x\in\mathbb{R}^n} \|\zeta - x\| > \frac{3-\sqrt{7}}{2\,\gamma(f,\zeta)}.$$

Beweis. Angenommen es existiert $x \in \mathbb{R}^n$ mit $\|\zeta - x\| \leq \frac{3-\sqrt{7}}{2\gamma(f,\zeta)}$. Dann konvergiert nach Satz 3.47 das Newton-Verfahren mit Startpunkt x gegen ζ. Andererseits ergibt das Newton-Verfahren mit Startpunkt x eine Folge reeller Punkte (siehe Bemerkung 3.43). Dies ist ein Widerspruch. □

Wir etablieren noch eine Verbindung zu Konditionszahlen. Sei $\mathcal{F}$ ein (endlicher) normierter Vektorraum analytischer Funktionen und sei $f \in \mathcal{F}$. Beachte, dass die Gleichung $F(f,\zeta) = f(\zeta) = 0$ eine implizite Gleichung in f und ζ ist. Im Fall, dass $Df(\zeta)$ invertierbar ist, liefert der Satz von der impliziten Funktion eine Abbildung

$$s : U \to \mathbb{C}^n,\ f \mapsto \zeta, \qquad \text{wobei } f(\zeta) = 0,$$

und $V \subset \mathcal{F}$ offen ist mit $f \in V$. Die Ableitung von s ist gegeben durch

$$Ds(f)(g) = Df(\zeta)^{-1}g(\zeta), \quad g \in \mathcal{F}.$$

Nach Satz 2.4 ist $\kappa_{\text{rel}}(s,f) = \max_{\|g\|=1} \|Ds(f)g\| \frac{\|f\|}{\|\zeta\|}$. Insbesondere ist $\kappa_{\text{rel}}(s,f)$ durch $Df(\zeta)^{-1}$ bestimmt. Korollar 3.48 und 3.49 zeigen also, dass der Abstand von zwei Nullstellen und der Abstand einer Nullstelle zu $\mathbb{R}^n$ mit der Konditionszahl $\kappa_{\text{rel}}(s,f)$ zusammenhängen.

Bemerkung 3.50. Beim Newton-Verfahren könnnen verschiedene Probleme auftreten:

a) Hoher Aufwand für die Berechnung von $Jf(x)^{-1}$. Manchmal genügt es, $Jf(c)$ mit festem c nahe der Nullstelle zu nehmen. Dies muss dann nur einmal berechnet werden. Eine andere Idee ist es, $Jf(x)$ in niedriger Präzision auszuwerten.

b) Zu kleiner Konvergenzbereich:

   Hierzu führt man einen Dämpfungsparameter $\lambda_n \in (0,1]$ ein, d.h.

$$x_{n+1} = x_n - \lambda_n Jf(x_n)^{-1}f(x_n).$$

   Kleinere Schritte λ_n führen zu langsamer, aber stabiler Konvergenz.

4 Interpolation

In diesem Kapitel wollen wir eine uns unbekannte Funktionen $f : \mathbb{R} \to \mathbb{R}$ approximieren. Die Daten, die uns gegeben sind, sind endlich viele Funktionswerte $y_j := f(x_j)$ für $j = 0, \ldots, n$. Die Strategie ist es, eine Menge von *Basisfunktionen* $\mathcal{B} := \{u_0, \ldots, u_n\}$ zu wählen und eine Linearkombination $\varphi = \sum_{k=0}^{n} c_k u_k$ zu berechnen, welche mit f an den Stützstellen x_j übereinstimmt. D.h. wir suchen $c_0, \ldots, c_n$ mit

$$y_j = \varphi(x_j) = \sum_{k=0}^{n} c_k u_k(x_j) \qquad \text{für alle } j = 0, \ldots, n.$$

Dies führt zum $(n+1) \times (n+1)$-Gleichungssystem

$$Ac := \begin{pmatrix} u_0(x_0) & \ldots & u_n(x_0) \\ \vdots & & \vdots \\ u_0(x_n) & \ldots & u_n(x_n) \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} \varphi(x_0) \\ \vdots \\ \varphi(x_n) \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix} = y.$$

Je nach Wahl von $\mathcal{B}$ führt dies zu unterschiedlichen Lösungen.

Mögliche Anwendungen von Interpolation sind:

a) Approximation von Funktionen durch einfachere Funktionen;

b) Visualisierung diskreter Daten;

c) Ansätze zur Differentation;

d) Ansätze zur numerischen Integration;

e) Oberflächen- und Kurvenrekonstruktion im computergestützten Design (CAD).

Beispiel 4.1. In diesem Beispiel haben wir 3 Auswertung von f:

x_j	-1	0	2
y_j	1	2	2

Wir berechnen die Interpolation für verschiedene Wahlen von $\mathcal{B}$.

a) Für $\mathcal{B}_1 = \{1, x, x^2\}$ erhalten wir das Gleichungssystem

$$Ac := \begin{pmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} = y.$$

Die Lösung ist $c = (2, \frac{2}{3}, -\frac{1}{3})^T$. Also ist

$$\varphi_1(x) = 2 + \frac{2}{3}x - \frac{1}{3}x^2.$$

b) Sei $\mathcal{B}_2 = \{u_0, u_1, u_2\}$ gegeben durch

$$u_0(x) := \begin{cases} x+2 & \text{für } x \in [-2,-1], \\ -x & \text{für } x \in [-1,0], \\ 0 & \text{sonst,} \end{cases}$$

$$u_1(x) := \begin{cases} x+1 & \text{für } x \in [-1,0], \\ 1-x/2 & \text{für } x \in [0,2], \\ 0 & \text{sonst,} \end{cases}$$

$$u_2(x) := \begin{cases} x/2 & \text{für } x \in [0,2], \\ 3-x & \text{für } x \in [2,3], \\ 0 & \text{sonst,} \end{cases}$$

D.h. die Funktionen u_k sind auf den Intervallen $[-2,-1]$, $[-1,0]$, $[0,2]$ und $[2,3]$ jeweils linear, außerhalb von $[-2,3]$ gleich Null und $u_k(x_j) = \delta_{j,k}$. Dies führt zum trivialen Gleichungssystem $c = y$ (also zur Matrix $A = I$). Also ist $c = (1,2,2)^T$ und die interpolierte Funktion ist

$$\varphi_2(x) := \begin{cases} (x+2) & \text{für } x \in [-2,-1], \\ -x + 2(x+1) & \text{für } x \in [-1,0], \\ 2(1-x/2) + 2(x/2) & \text{für } x \in [0,2], \\ 2(3-x) & \text{für } x \in [2,3], \\ 0 & \text{sonst.} \end{cases}$$

c) Sei $\mathcal{B}_3 := \{u_0, u_1, u_2\}$ mit $u_k(x) := \exp(-(x-x_k)^2)$. Dies führt zu

$$\begin{pmatrix} 1 & e^{-1} & e^{-9} \\ e^{-1} & 1 & e^{-4} \\ e^{-9} & e^{-4} & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}.$$

Also ist $c \approx (0.3206, 1.8460, 1.9661)^T$ und damit

$$\varphi_3(x) \approx 0.3206 \cdot \exp(-(x+1)^2) + 1.8460 \cdot \exp(-x^2) + 1.9661 \cdot \exp(-(x-2)^2).$$

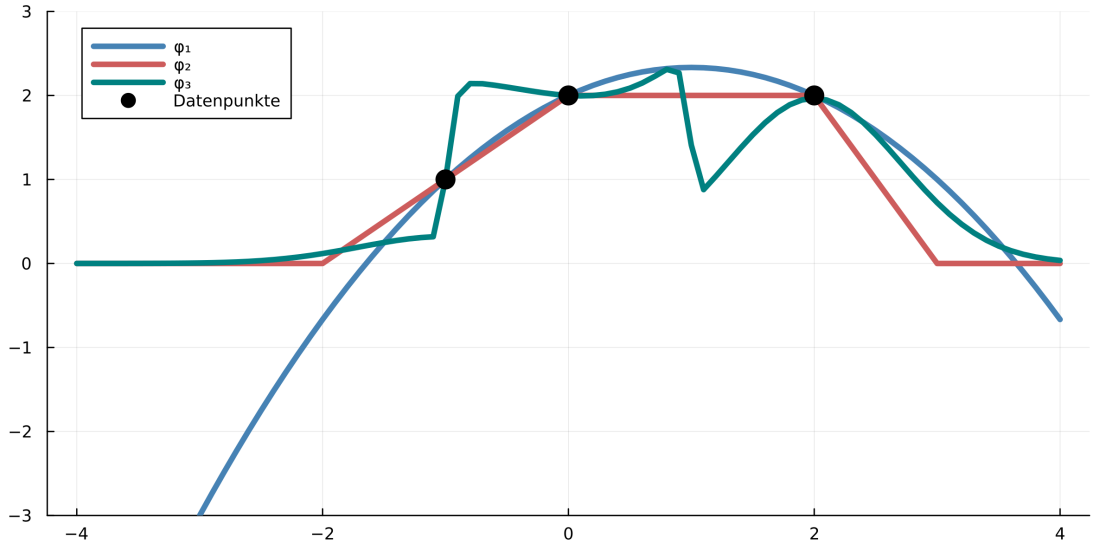


Abbildung 4.1: Die drei interpolierten Funktionen φ_1, φ_2 und φ_3 aus Beispiel 4.1.

Die drei interpolierten Funktionen sind in Abbildung 4.1 zu sehen.

Es bezeichne $\mathcal{P}_n$ den Vektorraum der Polynome vom Grad höchstens n. Unter *Polynominterpolation* verstehen wir Interpolation mit einer Basis von $\mathcal{P}_n$.

Wir wählen zunächst die Monombasis $\mathcal{B} = \{1, x, \dots, x^n\}$ von $\mathcal{P}_n$. Dies führt zum Gleichungssystem $Ac = y$ mit

$$A = \begin{pmatrix} 1 & x_0 & \dots & x_0^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^n \end{pmatrix}.$$

Diese Matrix heißt *Vandermonde-Matrix*. Es gilt

$$\det(A) = \prod_{0 \le i < j \le n} (x_j - x_i). \tag{4.1}$$

Satz 4.2. *Interpolation mit einer Basis in $\mathcal{P}_n$ ist genau dann eindeutig lösbar, wenn die Stützstellen $x_0, \dots, x_n$ paarweise verschieden sind.*

Beweis. Für die Vandermonde-Matrix A gilt genau dann $\det(A) \neq 0$, wenn die Stützstellen paarweise verschieden sind, s. (4.1). □

Vandermonde Matrizen sind häufig schlecht konditioniert. Ist z.B. $x_0 = 100$, $x_1 = 101$, $x_2 = 102$, so ist $\text{cond}_2(A) \approx 2.14353 \cdot 10^8$. Daher sollte Interpolation mit der Monombasis vermieden werden. Im Folgenden stellen wir alternative Basen für Polynominterpolation vor.

4.1 Lagrange Interpolation

Wir definieren für paarweise verschiedene Stützstellen $x_0, \dots, x_n \in \mathbb{R}^n$

$$\ell_k(x) := \prod_{j \neq k} \frac{x - x_j}{x_k - x_j} \in \mathcal{P}_n.$$

Dann gilt $\ell_k(x_j) = \delta_{k,j}$. Insbesondere folgt, dass die ℓ_k eine Basis für $\mathcal{P}_n$ bilden. Das Interpolationspolynom ist dann

$$\varphi(x) = \sum_{j=0}^{n} y_j \, \ell_j(x). \tag{4.2}$$

Interpolation des konstanten Polynoms $f(x) = 1$ zeigt, dass

$$\sum_k \ell_k(x) = 1 \quad \text{für alle } x \in \mathbb{R}.$$

Beispiel 4.3. Für die Stützstellen $(x_0, x_1, x_2) = (-1, 0, 2)$ erhalten wir

$$\ell_0(x) = \frac{x(x-2)}{3}, \qquad \ell_1(x) = -\frac{(x+1)(x-2)}{2}, \qquad \ell_2(x) = \frac{(x+1)x}{6}.$$

Sind nun die Daten $f(x_0) = y_0 = 1, f(x_1) = y_1 = 2, f(x_2) = y_2 = 2$ aus Beispiel 4.1 gegeben, so erhalten wir

$$\varphi(x) = \frac{x(x-2)}{3} - (x+1)(x-2) + \frac{(x+1)x}{3} = \frac{-1}{3}x^2 + \frac{2}{3}x + 2.$$

Nachteilig ist, dass für die Auswertung von $\varphi(x)$ die $\frac{(n+1)n}{2} = O(n^2)$ Faktoren $x_k - x_j$, $k > j$, für die Lagrange-Basispolynome zu berechnen sind. Ein weiterer großer Nachteil ist, dass das Hinzufügen einer neuen Stützstelle alle Basispolynome ändert.

4.2 Newton Interpolation

Wir wollen nun eine Basis finden, welche sich beim Hinzufügen einer Stützstelle nicht mehr komplett ändert. Wir definieren die *Newton-Basispolynome* $\mathcal{N}_k$ durch

$$\mathcal{N}_0(x) := 1, \quad \mathcal{N}_k(x) := \prod_{j=0}^{k-1} (x - x_j).$$

Da der Grad aufsteigend ist, ist $\mathcal{B} = \{\mathcal{N}_0, \dots, \mathcal{N}_n\}$ eine Basis von $\mathcal{P}_n$.

Weiterhin definieren wir die *dividierten Differenzen* (für paarweise verschiedene x_k) rekursiv durch

$$f[x_j] := f(x_j) = y_j, \quad f[x_i, \dots, x_{i+k}] := \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}.$$

Satz 4.4. *Es gilt für paarweise verschiedene Stützstellen:*

a) Das Interpolationspolynom ist gegeben durch

$$\varphi(x) = \sum_{k=0}^{n} f[x_0, \ldots, x_k]\, \mathcal{N}_k(x).$$

b) Dividierte Differenzen sind invariant unter Permutationen, d.h. für jede Permutation σ von $\{0, \ldots, n\}$ gilt

$$f[x_0, \ldots, x_n] = f[x_{\sigma(0)}, \ldots, x_{\sigma(n)}].$$

Beweis. Beweis durch vollständige Induktion über n von a) und b). Für $n = 0$ ist b) trivial und a) folgt aus

$$f[x_0]\mathcal{N}_0(x) = f(x_0) \cdot 1 = f(x_0).$$

Kommen wir nun zum Induktionsschritt $n-1 \to n$. Wir konstruieren die zwei Polynome $p, q \in \mathcal{P}_{n-1}$ zu den Stützstellen $x_0, \ldots, x_{n-1}$ bzw. $x_1, \ldots, x_n$, d.h.

$$\begin{aligned} p(x_j) &= y_j \qquad \text{für } j = 0, \ldots, n-1, \\ q(x_j) &= y_j \qquad \text{für } j = 1, \ldots, n. \end{aligned}$$

Nach Induktionsvoraussetzung gelingt dies mit

$$\begin{aligned} p(x) &= \sum_{k=0}^{n-1} f[x_0, \ldots, x_k] \prod_{j=0}^{k-1} (x - x_j), \\ q(x) &= \sum_{k=1}^{n-1} f[x_1, \ldots, x_k] \prod_{j=1}^{k-1} (x - x_j) + f[x_1, \ldots, x_{n-1}, x_n] \prod_{j=1}^{n-1} (x - x_j). \end{aligned}$$

Wir definieren nun

$$\varphi(x) := p(x) + \frac{q(x) - p(x)}{x_n - x_0}(x - x_0) \in \mathcal{P}_n.$$

Da p und q bei $x_1, \ldots, x_{n-1}$ übereinstimmen, folgt

$$\varphi(x_j) = p(x_j) = y_j \qquad \text{für } j = 1, \ldots, n-1.$$

Außerdem gilt

$$\varphi(x_0) = p(x_0) = y_0, \quad \varphi(x_n) = q(x_n) = y_n.$$

Also ist φ das Interpolationspolynom.

Da p eindeutig bestimmt ist, gilt nach Induktionsvoraussetzung a) mit Hilfe der Stützstellen $x_1, \ldots, x_{n-1}, x_0$ (Reihenfolge geändert)

$$p(x) = \sum_{k=1}^{n-1} f[x_1, \ldots, x_k] \prod_{j=1}^{k-1} (x - x_j) + f[x_1, \ldots, x_{n-1}, x_0] \prod_{j=1}^{n-1} (x - x_j).$$

Dies ergibt

$$\varphi(x) = p(x) + \frac{f[x_1, \ldots, x_n] - f[x_1, \ldots, x_{n-1}, x_0]}{x_n - x_0} \prod_{j=0}^{n-1} (x - x_j).$$

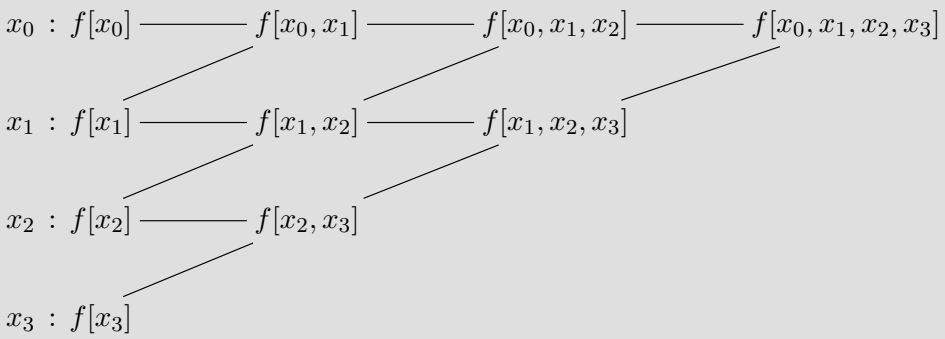
Nach Induktionsvoraussetzung b) gilt $f[x_1, \ldots, x_{n-1}, x_0] = f[x_0, x_1, \ldots, x_{n-1}]$. Also haben wir

$$\varphi(x) = p(x) + f[x_0, \ldots, x_n] \prod_{j=0}^{n-1} (x - x_j) = \sum_{k=0}^{n} f[x_0, \ldots, x_k] \prod_{j=0}^{k-1} (x - x_j).$$

Dies beweist a).

Nach a) ist der führende Koeffizient des Interpolationspolynoms $f[x_0, \ldots, x_n]$. Vertauscht man die Stützstellen mittels einer Permutation, so erhält man als führenden Koeffizienten $f[x_{\sigma(0)}, \ldots, x_{\sigma(n)}]$. Da das Interpolationspolynom eindeutig ist, stimmen beide Koeffizienten überein, d.h. $f[x_0, \ldots, x_n] = f[x_{\sigma(0)}, \ldots, x_{\sigma(n)}]$. Dies beweist b). □

Berechnung der dividierten Differenzen nach grafischem Schema:



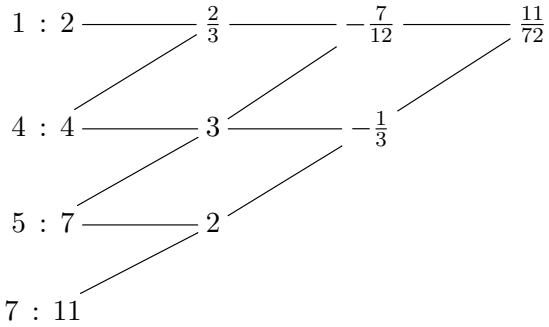
Bemerkung 4.5. Im allgemeinen Fall ist die Anzahl der zu berechnenden dividierten Differenzen $(n+1)(n+2)/2$; der Gesamtaufwand zur Berechnung aller dieser dividierten Differenzen ist also von der Größenordnung $O(n^2)$.

Beispiel 4.6. Seien die folgenden Werte gegeben:

x_i	1	4	5	7
y_i	2	4	7	11

. Dann haben

wir als dividierte Differenzen



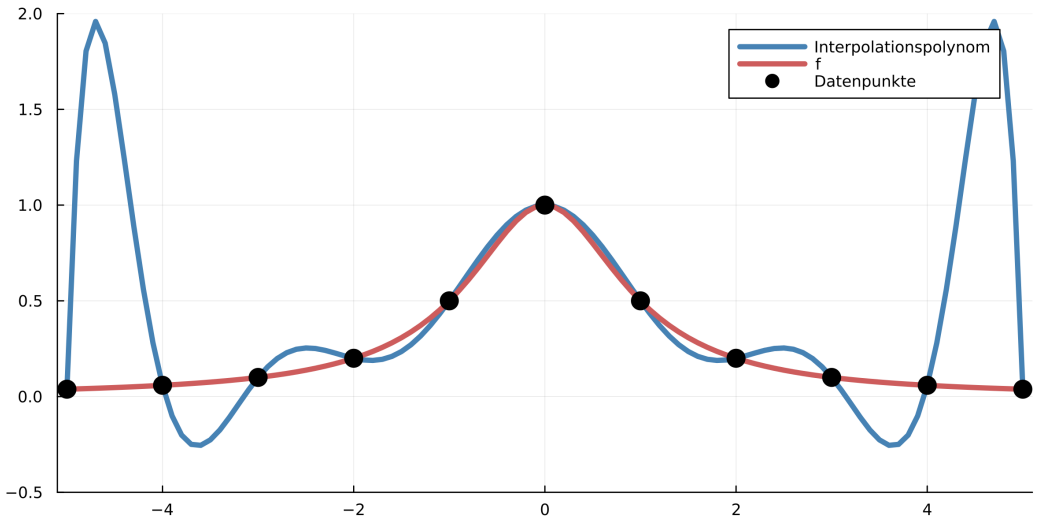
Wir erhalten als Interpolationspolynom

$$\varphi(x) = 2 + \frac{2}{3}(x-1) + \frac{7}{12}(x-1)(x-4) - \frac{11}{72}(x-1)(x-4)(x-5).$$

4.3 Fehleranalyse

Bisher haben wir drei Methoden kennengelernt, um Funktionen mittels Polynomen zu interpolieren. In diesem Abschnitt wollen wir verstehen, wie gut unsere approximation ist.

Beispiel 4.7. Betrachten wir die Funktion $f(t) := \frac{1}{1+t^2}$ auf dem Intervall $[-5, 5]$. Wir nehmen $n+1$ äquidistante Punkte $x_i = 10\frac{i}{n} - 5$ in $[-5, 5]$. Das Interpolationspolynom φ konvergiert nicht gleichmäßig gegen f. An den Rändern beginnt das Polynom zu „flattern“. Hier das Polynom für $n = 10$.



Dies ist bekannt unter dem Namen *Runge-Phänomen*. Ein Problem ist, dass die Ableitungen von f schnell wachsen. Ein weiteres Problem ist die äquidistante Wahl der Stützstellen.

Wir wollen im Folgenden das Phänomen in Beispiel 4.7 besser verstehen. Dazu zunächst ein Lemma.

Lemma 4.8. *Seien $f \in C^n([a,b])$ und $x_0, \dots, x_n$ paarweise verschiedene Stützstellen mit $f(x_i) = y_i$ für $i = 0, 1, \dots, n$. Dann gibt es*

$$\zeta \in \operatorname{conv}(x_0, \dots, x_n) = [\min\{x_0, \dots, x_n\}, \max\{x_0, \dots, x_n\}]$$

mit

$$f[x_0, \dots, x_n] = \frac{1}{n!} f^{(n)}(\zeta).$$

Beweis. Nach Satz 4.4 ist $\varphi(x) = \sum_{k=0}^{n} f[x_0, \dots, x_k]\mathcal{N}_k(x)$. Sei $g(x) := f(x) - \varphi(x)$. Dann gilt

$$g(x_0) = \cdots = g(x_n) = 0.$$

Zwischen den $n+1$-Nullstellen finden wir nach dem Satz von Rolle n Nullstellen von g'. Dazwischen finden wir $(n-1)$ Nullstellen von g'' usw. Am Ende bleibt eine Nullstelle ζ von $g^{(n)}$. Also ist $0 = g^{(n)}(\zeta) = f^{(n)}(\zeta) - n!\, f[x_0, \dots, x_n]$, was zu zeigen war. □

Der nächste Satz bereitet die Identität (4.4) vor, auf der unsere Fehleranalyse fußt. Dazu definieren wir für paarweise verschiedene Stützstellen $x_0, \dots, x_n$ das *Knotenpolynom*.

$$\omega(x) := \prod_{j=0}^{n} (x - x_j).$$

(Erinnerung: $\mathcal{N}_k(x) = \prod_{j=0}^{k-1}(x - x_j)$.)

Satz 4.9. *Sei $f \in C([a,b])$ mit paarweise verschiedenen Stützstellen $x_0, \dots, x_n$ und $y_i = f(x_i)$. Dann gilt für das Interpolationspolynom φ und $x \in [a,b]$*

$$f(x) - \varphi(x) = f[x_0, \dots, x_n, x]\, \omega(x). \tag{4.3}$$

Beweis. Wir nehmen zunächst an, dass $x \notin \{x_0, \dots, x_n\}$. Laut Satz 4.4 ist das Polynom

$$\psi(t) := \varphi(t) + f[x_0, \dots, x_n, x]\, \omega(t),$$

das (eindeutige) Interpolationspolynom in t, welches an den Stellen $x_0, \dots, x_n, x$ die Werte $f(x_0), \dots, f(x_n), f(x)$ annimmt. Setzt man $t := x$, so erhalten wir eine Funktion $\psi(x)$ mit der Eigenschaft, dass $f(x) - \varphi(x) = \psi(x) - \varphi(x) = f[x_0, \dots, x_n, x]\, \omega(x)$. Dies zeigt (4.3).

Beachte, dass $f(x_i) - \varphi(x_i) = y_i - y_i = 0$. Da $f - \varphi$ stetig ist, gilt (4.3) also genau dann, wenn für alle $0 \le i \le n$ gilt, dass

$$\lim_{x \to x_i} f[x_0, \dots, x_n, x]\, \omega(x) = 0.$$

Dazu schreiben wir unter Benutzung von Satz 4.4:

$$f[x_0,\dots,x_n,x] = f[x_i,x_0,\dots,x_n,x] = \frac{f[x_0,\dots,x_n,x]-f[x_i,x_0,\dots,x_n]}{x-x_i}.$$

Da dividierte Differenzen stetig sind, gilt

$$\lim_{x\to x_i} f[x_0,\dots,x_n,x] - f[x_i,x_0,\dots,x_n] = f[x_0,\dots,x_n,x_i] - f[x_i,x_0,\dots,x_n] = 0.$$

Außerdem $\lim_{x\to x_i} \omega(x)/(x-x_i) = \prod_{j\neq i}(x_i-x_j) =: c$. Insgesamt erhalten wir somit

$$\lim_{x\to x_i} f[x_0,\dots,x_n,x]\ \omega(x) = 0\cdot c = 0,$$

was zu zeigen war. □

Aus der Kombination von Lemma 4.8 mit Satz 4.9 ergibt sich sofort:

Korollar 4.10. *Sei $f \in C^{n+1}([a,b])$ mit Stützstellen $x_0,\dots,x_n$. Sei φ das zugehörige Interpolationspolynom. Dann gibt es für jedes x ein $\zeta \in \operatorname{conv}\{x_0,\dots,x_n,x\}$ mit*

$$f(x) - \varphi(x) = \frac{f^{(n+1)}(\zeta)}{(n+1)!}\,\omega(x). \tag{4.4}$$

Für die folgenden Beispiele definiern wir die ∞-Norm für eine Funktion $g \in C([a,b])$:

$$\|g\|_{L^\infty([a,b])} := \max_{x\in[a,b]} |g(x)|.$$

Wenn der Definitionsbereich der Funktion g aus dem Kontext ersichtlich ist, schreiben wir auch $\|g\|_{L^\infty} := \|g\|_{L^\infty([a,b])}$.

Beispiel 4.11. Nehmen wir $f(x) := \exp(\frac{x+1}{2})$ auf $[a,b] := [-1,1]$. Wir approximieren nun f an den drei Stellen $-1,0,1$. Sei φ das zugehörige Interpolationspolynom. Dann gilt auf $[-1,1]$ die Fehlerabschätzung

$$\|f-\varphi\|_{L^\infty} \le \frac{\|f^{(3)}\|_{L^\infty}}{3!}\|\omega\|_{L^\infty} \approx \frac{\frac{1}{8}e}{3!}\cdot 0.3849 \approx 0.0218,$$

Damit haben wir eine gute Genauigkeit erreicht.

Beispiel 4.12. Wir kommen noch einmal zurück zu Beispiel 4.7. Hier haben wir die Funktion $f(t) := \frac{1}{1+t^2}$ auf dem Intervall $[-5,5]$. Beachte, dass

$$f^{(11)}(t) = \frac{-159667200t(3t^{10}-55t^8+198t^6-198t^4+55t^2-3)}{(t^2+1)^{12}}.$$

Der Absolutbetrag von $f^{(11)}(t)$ wird bei $t \approx 0.12$ maximiert, so dass

$$\frac{\|f^{(11)}\|_{L^\infty}}{11!} \approx 0.909.$$

Weiterhin gilt für $n+1 = 11$ äquidistante Punkte auf $[-5,5]$, dass

$$\|\omega\|_{L^\infty} \approx 416614.$$

Sei φ das Interpolationspolynom. Dann gilt nach Korollar 4.10 die Fehlerabschätzung $\|f-\varphi\|_{L^\infty} \le \frac{\|f^{(11)}\|_{L^\infty}}{11!}\|\omega\|_{L^\infty} \approx 378702$. Die verhältnismäßig große Norm von ω führt zum oszillatorischen Verhalten in Beispiel 4.7.

4.4 Chebyshev Polynome

In Beispiel 4.12 haben wir gesehen, dass die Wahl äquidistanter Stützstellen den Effekt hat, dass $\|\omega\|_{L^\infty}$ großwird, womit das Interpolationspolynom φ von der zu interpolierenden Funktion f stark abweicht.

Wir wollen nun die Stützstellen so wählen, dass der Fehler $\|f - \varphi\|_{L^\infty}$ möglichst klein wird, indem wir ω klein wird. Dies erreichen wir mit Hilfe der sogenannten *Chebyshev-Polynome*.

Definition 4.13. Wir definieren die *Chebyshev-Polynome* $T_n : [-1, 1] \to \mathbb{R}$ für $n \in \mathbb{N}_0$ durch

$$T_n(x) := \cos(n \arccos(x)).$$

Das folgende Lemma zeigt, dass die T_n Polynome sind.

Satz 4.14. *Die wesentlichen Eigenschaften der Chebyshev-Polyome sind*

a) $T_0(x) = 1, T_1(x) = x$ *und* $T_{n+1}(x) = 2x\,T_n(x) - T_{n-1}(x)$ *für* $n \geq 1$.

b) $T_n \in \mathcal{P}_n$ *und* $T_n(x) = 2^{n-1}x^n + \ldots$ *für* $n \geq 1$..

c) Die Nullstellen von T_n *sind*

$$x_j = \cos\left(\frac{(2j+1)\pi}{2n}\right) \qquad \text{für } j = 0, \ldots, n-1.$$

d) Die Extrema von T_n *in* $[-1, 1]$ *sind*

$$\tilde{x}_j = \cos\left(\frac{j\pi}{n}\right), \quad \text{für } j = 0, \ldots, n$$

mit $T_n(\tilde{x}_j) = (-1)^j$.

Beweis.

a) Die ersten beiden Aussagen folgen direkt aus der Definition von T_0 und T_1. Mit dem Cosinusgesetz $\cos(a + b) + \cos(a - b) = 2\cos(a)\cos(b)$ folgt

$$T_{n+1}(x) + T_{n-1}(x) = 2\cos(\arccos(x))\cos(n \arccos(x)) = 2xT_n(x).$$

b) Dies folgt per Induktion aus a).

c) Einsetzen ergibt, dass dies Nullstellen sind. Die Anzahl der Nullstellen ist gleich n, was gleich dem Grad von T_n ist. Also haben wir alle Nullstellen gefunden.

d) Aus $T_n(x) = \cos(n \arccos(x)) = \pm 1$ folgt $n \arccos(x) = j\pi$, $j \in \mathbb{Z}$, und damit $x = \cos(\frac{j\pi}{n})$. Die Einschränkung $j \in \{0, 1, \ldots, n\}$ folgt aus $\arccos(x) \in [0, \pi]$.

□

Wählt man nun die Nullstellen von T_{n+1} als Stützstellen, so wird die Supremumsnorm von ω minimiert.

Lemma 4.15. *Für $\omega(x) = \prod_{j=0}^{n}(x - x_j)$ mit paarweise verschiedenen $x_0, \dots, x_n$ gilt*

a) $\omega \in \mathcal{P}_{n+1}$ und $\omega(x) = x^{n+1} + \dots$.

b) $\|\omega\|_{L^\infty([-1,1])} = \max_{x\in[-1,1]} |\omega(x)| \geq 2^{-n}$.

c) Gleichheit gilt in b) nur für $x_j = \cos\frac{(2j+1)\pi}{2(n+1)}$ mit $j = 0, \dots, n$. In diesem Fall gilt also $\omega = 2^{-n}T_{n+1}$.

Beweis. a): Folgt direkt aus der Definition von ω. b) und c): Nach Satz 4.14 ist $2^{-n}T_{n+1}$ ein normiertes Polynom mit Nullstellen $x_j = \cos(\frac{(2j+1)\pi}{2n})$. Es folgt, dass

$$\omega = \prod_{j=0}^{n}(x - x_j) = 2^{-n}T_{n+1}(x) \in \mathcal{P}_{n+1}.$$

Nach Satz 4.14 d) ist das Maximum des Absolutbetrags von $\omega = 2^{-n}T_{n+1}$ genau 1. Also erfüllt ω die Abschätzung in b) mit Gleichheit. Angenommen, es gäbe ein weiteres Polynom $v \in \mathcal{P}_{n+1}$ mit $v(x) = x^{n+1} + \dots$ mit $\|v\|_{L^\infty} \leq 2^{-n}$. Betrachte $q := 2^{-n}T_{n+1} - v$. Dann ist $q \in \mathcal{P}_n$, da der führende Term von $2^{-n}T_{n+1}$ und v jeweils x^{n+1} ist. Seien $\tilde{x}_j = \cos\frac{j\pi}{n+1}$ mit $j = 0, \dots, n+1$ die Extrema von $2^{-n}T_{n+1}$.

Fall 1: $\|v\|_{L^\infty} < 2^{-n}$. Da $2^{-n}T_{n+1}$ bei den Extrema $\tilde{x}_j$ die Werte $(-1)^j 2^{-n}$ hat und $\|v\|_{L^\infty} < 2^{-n}$, hat q wechselnde Vorzeichen an diesen Stellen. Damit befinden sich dazwischen $n+1$ verschiedene Nullstellen. Wegen $q \in \mathcal{P}_n$, muss $q = 0$ gelten. Damit wäre $v = 2^{-n}T_{n+1}$, was im Widerspruch zu $\|v\|_{L^\infty} < 2^{-n}$ steht.

Fall 2: $\|v\|_{L^\infty} = 2^{-n}$. Hier muss die Argumentation aus Fall 1 modifiziert werden – Übungsaufgabe! □

Bemerkung 4.16. Mittels Skalierung kann man die Resultate von $[-1, 1]$ auf ein beliebiges Intervall $[a, b]$ übertragen. So ist

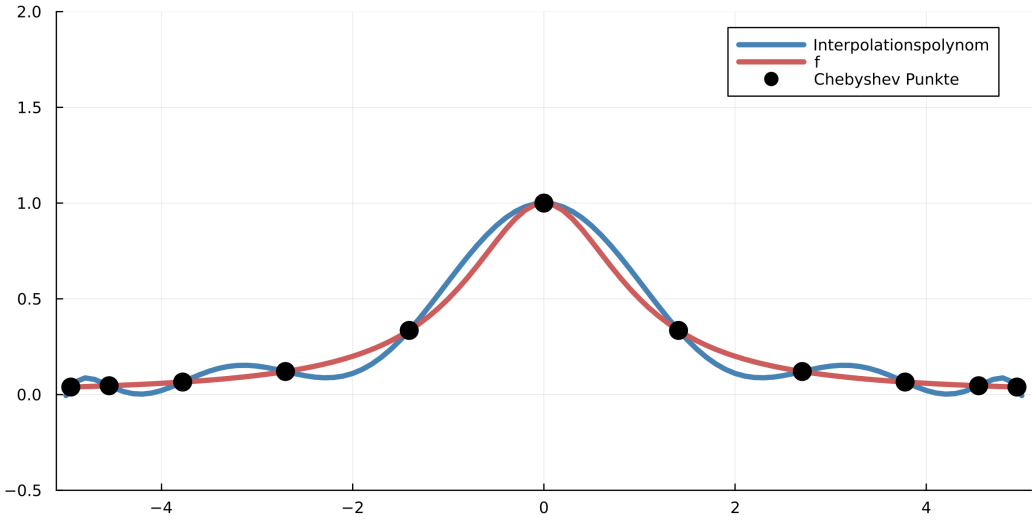
$$\phi : [-1, 1] \to [a, b],\ x \mapsto \frac{a+b}{2} + x\frac{b-a}{2}$$

eine affin-lineare Bijektion mit Inverser $\phi^{-1}(z) = \frac{2z-(a+b)}{b-a}$. Für $a = z_0 < \dots < z_n = b$ betrachten wir $\omega(z) = \prod_{j=0}^{n}(z - z_j)$. Dann ist $q := (\frac{b-a}{2})^{-n-1}\omega \circ \phi \in \mathcal{P}_{n+1}$ mit führendem Koeffizienten 1. Also gilt

$$\begin{aligned}\|\omega\|_{L^\infty([a,b])} = \|\omega \circ \phi\|_{L^\infty([-1,1])} &= \left(\frac{b-a}{2}\right)^{n+1} \left\| \left(\frac{b-a}{2}\right)^{-n-1} \omega \circ \phi \right\|_{L^\infty([-1,1])} \\ &\geq \left(\frac{b-a}{2}\right)^{n+1} 2^{-n}.\end{aligned}$$

Gleichheit gilt genau dann, wenn $z_j = \varphi(x_j)$, $j = 0, 1, \dots, n$, wobei x_j die Nullstellen von T_{n+1} sind.

Beispiel 4.17. Verwenden wir in Beispiel 4.7 die Nullstelle von Chebychev Polynomen als Interpolationspunkte erhalten wir folgendes Interpolationspolynom:



Im Vergleich zum Interpolationspolynom in Beispiel 4.7 für äquidistante Punkte oszilliert dieses Interpolationspolynom deutlich weniger.

4.5 Spline Interpolation

Die Idee von Spline-Interpolation ist es, das Intervall $[a, b]$, auf dem die zu interpolierende Funktion f definiert ist, in Teilintervalle

$$[a, b] = [x_0, x_1] \cup [x_1, x_2] \cup \cdots \cup [x_{n-1}, x_n]$$

aufzuteilen und für jedes Teilintervall $[x_i, x_{i+1}]$ eine Funktionen zu wählen, die außerhalb von $[x_i, x_{i+1}]$ verschwindet. Die Übergänge zwischen diesen Funktionen soll dabei genügend oft stetig differenzierbar sein. Wir nennen $\Delta = \{a = x_0 < x_1 < \cdots < x_n = b\}$ die *Spline-Knoten*.

Definition 4.18. Zu den Spline-Knoten $\Delta = \{a = x_0 < x_1 < \cdots < x_n = b\}$ heißt

$$s : [a, b] \to \mathbb{R}$$

Spline bzw. *Spline-Funktion* vom Grad $m \in \mathbb{N}$, falls

a) die Einschränkung von s auf das Intervall

$$I_j = [x_j, x_{j+1})$$

ein Polynom vom Grad höchstens m ist, d.h. $s|_{I_j} \in \mathcal{P}_m$ für $j = 0, \ldots, n-1$.

b) $s \in C^{m-1}([a, b])$.

Den Raum der Splines bezeichnen wir mit $S(m, \Delta)$. Splines der Ordnung $m = 1$ heißen *linear*, Splines der Ordnung $m = 2$ heißen *quadratisch* und Splines der Ordnung $m = 3$ heißen *kubisch*.

Die Linearkombination zweier m-Splines ist wieder ein m-Spline. Auf jedem Intervall I_j können wir $s|_{I_j}$ aus einem $n+1$-dimensionalen Vektorraum wählen; d.h.,

$$s(x) = \sum_{i=0}^{n-1} p_i(x) \cdot \mathbf{1}_{I_j}(x),$$

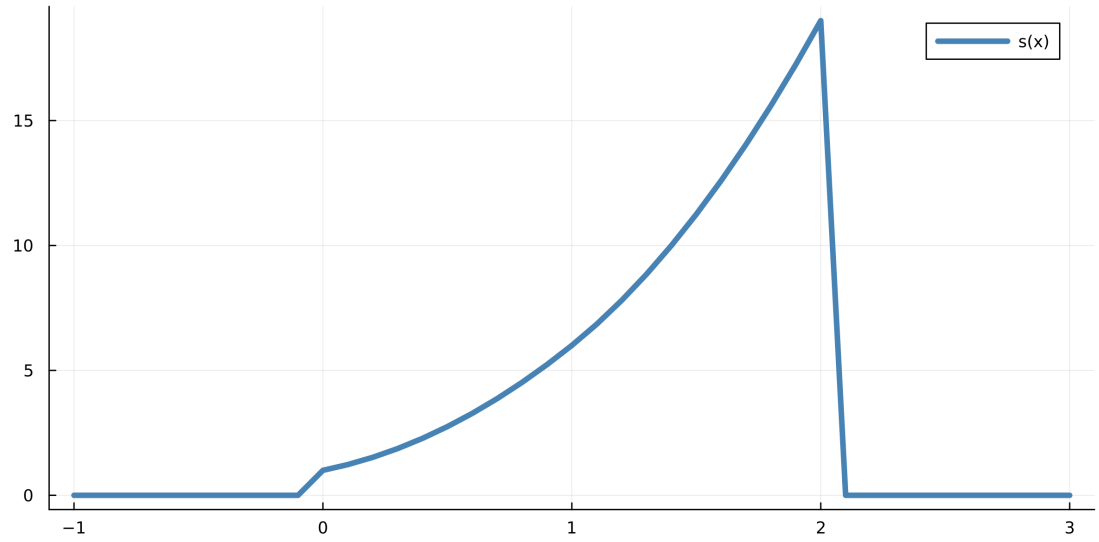
wobei $p_i \in \mathcal{P}_m$ und $\mathbf{1}_{I_j}$ die Indikatorfunktion von I_j ist. Dabei induziert jeder Übergang zwischen zwei Intervallen m lineare Gleichungen in den Koeffizienten der s_i. Insgesamt haben wir $n-1$ solche Übergänge. Wir haben somit folgenden Satz bewiesen.

Satz 4.19. *$S(m, \Delta)$ mit den üblichen Verknüpfungen ist ein reeller Vektorraum der Dimension $(m+1)n - m(n-1) = n + m$.*

Beispiel 4.20. Seien $[a, b] = [0, 2]$ und $x_0 = 0$, $x_1 = 1$, $x_2 = 2$ sowie $m = 2$. Dann ist

$$s(x) = \begin{cases} p_0(x) = 3x^2 + 2x + 1 & \text{für } x \in [0, 1), \\ p_1(x) = 5x^2 - 2x + 3 & \text{für } x \in [1, 2], \end{cases}$$

ein quadratischer Spline, denn es gelten $p_0(1) = 6 = p_1(1)$ und $p_0'(1) = 8 = p_1'(1)$.



Zu den Spline-Knoten $\Delta = \{x_0 < x_1 < \cdots < x_n\}$ fügen wir noch zwei weiteren Knoten $x_{-1} < x_0$ und $x_{n+1} > x_n$ am Anfang und am Ende hinzu und definieren die *Basis der*

Hutfunktionen für $i = 0, \dots, n$ durch

$$\Lambda_i(x) := \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & \text{für } x \in [x_{i-1}, x_i] \cap [a, b], \\ \dfrac{x - x_{i+1}}{x_i - x_{i+1}} & \text{für } x \in [x_i, x_{i+1}] \cap [a, b] \\ 0 & \text{sonst.} \end{cases}$$

Satz 4.21. *Zu den Knotenpunkten $\Delta = \{a = x_0 < x_1 < \cdots < x_n = b\}$ mit den Stützwerten $y_0 = f(x_0), \dots, y_n = f(x_n)$ ist*

$$s(x) = \sum_{i=0}^{n} y_i \, \Lambda_i(x), \qquad x \in [a, b],$$

der eindeutig bestimmte interpolierende lineare Spline.

Ist $f \in C^2([a, b])$ und sind $x_i = a + ih$ mit $h = \frac{b-a}{n}$ äquidistant, so gilt

$$\|f - s\|_\infty \le \frac{h^2}{8} \|f''\|_\infty, \quad \text{und} \quad \max_{x \in [a,b] : x \notin \{x_0, \dots, x_n\}} |f'(x) - s'(x)| \le \frac{h}{2} \|f''\|_\infty.$$

(Beachte, dass s in $x_0, \dots, x_n$ nicht differenzierbar ist.)

Beweis. s hat per Konstruktion interpolierend. Außerdem ist der Spline s eindeutig, da $s|_{I_j}$ linear ist, und es eine eindeutige lineare Funktion gibt, die zwei Punkte interpoliert.

Im Fall von äquidistanten Knoten gilt für jedes Teilintervall $I_i = [x_i, x_{i+1}]$ und jedes $z \in I_i$ nach Korollar 4.10:

$$|f(z) - s(z)| \le \frac{\|f''\|_\infty}{2!} \max_{x \in [x_i, x_{i+1}]} |(x - x_i)(x - x_{i+1})| \le \frac{h^2}{8} \|f''\|_\infty.$$

Weiterhin gilt für $x \in (x_i, x_{i+1})$, dass

$$s'(x) = \frac{y_{i+1} - y_i}{h} = \frac{f(x_{i+1}) - f(x_i)}{h} = \frac{1}{h} \int_{x_i}^{x_{i+1}} f'(z) \, \mathrm{d}z$$

und somit

$$\begin{aligned} |f'(x) - s'(x)| \le \frac{1}{h} \int_{x_i}^{x_{i+1}} |f'(x) - f'(z)| \, \mathrm{d}z &= \frac{1}{h} \int_{x_i}^{x_{i+1}} \Big| \int_x^z f''(\xi) \, d\xi \Big| \, \mathrm{d}z \\ &\le \frac{\|f''\|_\infty}{h} \int_{x_i}^{x_{i+1}} |x - z| \, \mathrm{d}z \le \frac{h}{2} \|f''\|_\infty. \end{aligned}$$

□

Lineare Splines sind stetig. Im Gegensatz dazu sind Splines von Ordnung $m \ge 2$ $m-1$ mal stetig differenzierbar. Es werden daher, insbesondere im Kontext von grafischen

Darstellungen, bevorzugt Splines von höherer Ordnung verwendent, da diese dem Auge als glatt erscheinen. Kubische Splines $s \in S(3, \Delta)$ bieten dabei einen guten Kompromiss zwischen Glattheit und Simplizität.

Wir wollen nun verstehen, wie wir kubische Splines für gegebene Datenpunkte $x_0, \dots, x_n$ mit $y_j = f(x_j)$ berechnen können. Für $x \in I_j = [x_j, x_{j+1})$ haben wir

$$s(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3. \tag{4.5}$$

Da $y_j = f(x_j)$, gilt

$$a_j = y_j.$$

Die übrigen Koeffizienten lassen sich mit Hilfe des folgenden Lemmas berechnen.

Satz 4.22. *Für $j = 0, 1, \dots, n-1$ sei $h_j := x_{j+1} - x_j$ und*

$$g_j := 6\frac{y_{j+1} - y_j}{h_j} - 6\frac{y_j - y_{j-1}}{h_{j-1}}.$$

Angenommen s_0'', s_1'', ..., $s_n'' \in \mathbb{R}$ erfüllen die $n-1$ linearen Gleichungen

$$h_{j-1}s_{j-1}'' + 2(h_{j-1} + h_j)s_j'' + h_j s_{j+1}'' = g_j, \quad 1 \le j \le n-1;$$

in Matrix-Vektor Schreibweise:

$$\begin{pmatrix} h_0 & 2(h_0+h_1) & h_1 & 0 & \dots & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \end{pmatrix} \begin{pmatrix} s_0'' \\ s_1'' \\ \vdots \\ s_{n-1}'' \\ s_n'' \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_{n-1} \end{pmatrix}.$$

Dann liefert der Ansatz (4.5) mit

$$c_j := \frac{s_j''}{2}, \quad a_j := y_j, \quad d_j := \frac{s_{j+1}'' - s_j''}{6h_j}, \quad b_j := \frac{y_{j+1} - y_j}{h_j} - \frac{h_j}{6}(s_{j+1}'' + 2s_j'') \tag{4.6}$$

ein $s \in S(3, \Delta)$, welches f interpoliert.

Beweis. Es sei

$$p_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \in \mathcal{P}_3,$$

so dass

$$s(x) = \sum_{j=0}^{n-1} p_j(x) \cdot \mathbf{1}_{I_j}(x).$$

Dann gilt insbesondere $s(x_j) = p_j(x_j) = a_j = y_j$ für $0 \le j \le n-1$. Die Gleichung (4.7) unten impliziert außerdem, dass $s(x_n) = y_n$. D.h., s interpoliert f.

Wir müssen noch zeigen, dass $s \in S(3, \Delta)$. Dazu betrachten wir zunächst die Stetigkeit von s. Hierfür verwenden wir lediglich die Gleichungen (4.6). Es gilt

$$\begin{aligned} p_j(x_{j+1}) &= a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \\ &= y_j + b_j h_j + \left(\frac{s''_j}{2} h_j^2 + \frac{s''_{j+1} - s''_j}{6} h_j^2 \right) \\ &= y_j + \left(y_{j+1} - y_j - \frac{h_j^2}{6}(s''_{j+1} + 2s''_j) \right) + \frac{h_j^2}{6}(s''_{j+1} + 2s''_j) \\ &= y_{j+1} = p_{j+1}(y_j). \end{aligned} \tag{4.7}$$

Als Nächstes haben wir die ersten Ableitungen. Jetzt müssen wir sowohl (4.6) als auch das lineare Gleichungssystem in den s''_i verwenden. Für $j = 1, 2, \ldots, n-1$ gilt:

$$\begin{aligned} p'_{j-1}(x_j) &= b_{j-1} + 2c_{j-1}h_{j-1} + 3d_{j-1}h_{j-1}^2 \\ &= \frac{y_j - y_{j-1}}{h_{j-1}} - \frac{h_{j-1}}{6}(s''_j + 2s''_{j-1}) + s''_{j-1}h_{j-1} + \frac{s''_j - s''_{j-1}}{2} h_{j-1} \\ &= \frac{1}{6}\left(6\frac{y_j - y_{j-1}}{h_{j-1}} + 2h_{j-1}s''_j + h_{j-1}s''_{j-1} \right) \\ &= \frac{1}{6}\left(6\frac{y_{j+1} - y_j}{h_j} - 2h_j s''_j - h_j s''_{j+1} \right) \\ &= b_j \\ &= p'_j(x_j). \end{aligned}$$

Zuletzt die zweiten Ableitungen. Hier verwenden wir wieder nur (4.6). Für $0 \leq j \leq n-2$ gilt dann

$$p''_{j+1}(x_{j+1}) = 2c_{j+1} = s''_{j+1} = s''_j + 6h_j d_j = 2c_j + 6d_j(x_{j+1} - x_j) = p''_j(x_{j+1}). \tag{4.8}$$

Somit ist $s \in S(3, \Delta)$. □

Die Variablen s''_i in Satz 4.22 nennen wir auch *Momente*. Dies hat folgenden Hintergrund. Nach (4.8) gilt $s''_j = p''_j(x_j) = s''(x_j)$ für $1 \leq j \leq n-1$. Außerdem gilt

$$p''_0(x_0) = 2c_0 = s''_0$$

und

$$p''_{n-1}(x_n) = 2c_{n-1} + 6d_{n-1}(x_n - x_{n-1}) = s''_{n-1} + 6h_{n-1}d_{n-1} = s''_n.$$

Somit legen die s''_i die zweiten Ableitungen von $s \in S(3, \Delta)$ in den Stützstellen fest.

In Satz 4.22 haben wir $n-1$ lineare Gleichungen, aber $n+1$ festzulegende Momente. Zur eindeutigen Festlegung kann man die Randbedingung $s''_0 = s''_n = 0$ fordern. Dies hat

den Effekt, dass die erste und letzte Spalte im linearen Gleichungssystem in Satz 4.22 eliminiert werden. Wir erhalten folgendes lineares Gleichungssystem in $s''_1, \dots, s''_{n-1}$:

$$\begin{pmatrix} 2(h_0+h_1) & h_1 & 0 & \dots & \dots & 0 \\ h_1 & 2(h_1+h_2) & h_2 & 0 & \dots & 0 \\ 0 & h_2 & 2(h_2+h_3) & h_3 & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{pmatrix} \begin{pmatrix} s''_1 \\ s''_2 \\ \vdots \\ s''_{n-1} \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_{n-1} \end{pmatrix}$$

Da obige Matrix als strikt diagonal dominante Matrix invertierbar ist, sind die Momente durch das lineare Gleichungssystem eindeutig bestimmt.

Lemma 4.23. *Jede strikt diagonal dominante Matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ ist invertierbar.*

Beweis. Zu $x \in \mathbb{R}^n$ sei $i \in \{1, \dots, n\}$ so, dass $|x_i| = \|x\|_\infty$. Es gilt

$$\begin{aligned} \|Ax\|_\infty \geq |(Ax)_i| &= \left| \sum_{j=1}^n a_{ij} x_j \right| \geq |a_{ii}||x_i| - \sum_{j=1; j\neq i}^n |a_{ij}||x_j| \\ &\geq |a_{ii}||x_i| - \sum_{j=1; j\neq i}^n |a_{ij}| \|x\|_\infty \geq \left(|a_{ii}| - \sum_{j=1; j\neq i}^n |a_{ij}| \right) \|x\|_\infty. \end{aligned}$$

Diagonaldominanz

$$|a_{ii}| - \sum_{j=1; j\neq i}^n |a_{ij}| > 0$$

impliziert, dass $\operatorname{Kern} A = \{0\}$ ist. Da A eine quadratische Matrix ist, ist dies äquivalent zur Invertierbarkeit von A. □

5 Numerische Integration

Gegeben seien relle Zahlen $a < b$ und eine integrierbare Funktionen $f : [a, b] \to \mathbb{R}$. Das Ziel dieses Kapitels ist es, das Integral

$$\mathcal{I}[f] := \int_a^b f(x)\,\mathrm{d}x$$

numerisch zu approximieren. Wir verwenden dabei den Ansatz, f durch ein Polymom φ zu approximieren und dann φ statt f zu integrieren. Der Vorteil dieses Ansatzes ist es, dass sich Polynome systematisch integrieren lassen. Dazu rufen wir uns noch einmal Satz 4.4 in Erinnerung: Für Stützstellen $x_0, \ldots, x_n \in [a, b]$ ist

$$\varphi(x) = \sum_{k=0}^{n} f[x_0, \ldots, x_k]\, \mathcal{N}_k(x) \in \mathcal{P}_n.$$

Beispiel 5.1 (Mittelpunktsformel)**.** Wir approximieren f durch $\varphi(x) = f(\frac{a+b}{2}) \in \mathcal{P}_0$ (Interpolation mit Stützstelle $\frac{a+b}{2}$). Das Integral von φ ist dann

$$\mathcal{I}[\varphi] = \int_a^b f\Big(\frac{a+b}{2}\Big)\,\mathrm{d}x = (b-a) f\Big(\frac{a+b}{2}\Big).$$

Dies ist die sogenannte Mittelpunktsformel.

Beispiel 5.2 (Trapezregel)**.** Wir approximieren f durch $\varphi \in \mathcal{P}_1$ per Interpolation zu den Punkten a und b. Das Integral von φ ist dann

$$\begin{aligned}\mathcal{I}[\varphi] = \int_a^b f[a] + f[a,b](x-a)\,\mathrm{d}x &= \int_a^b f(a) + \frac{f(b)-f(a)}{b-a}(x-a)\,\mathrm{d}x \\ &= \frac{f(b)+f(a)}{2}(b-a).\end{aligned}$$

Man könnte erwarten, dass eine Approximation durch Interpolation mit $\varphi \in \mathcal{P}_{n+1}$ sukzessive zu besseren Formeln für das Integral von f führt. Dies ist im Allgemeinen aber nicht der Fall, da hier die gleichen Probleme wie beim Runge-Phänomen auftauchen. Deshalb unterteilt man das Intervall $[a, b]$ meist in Teilintervalle, so dass

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{j=1}^{n} \int_{x_{j-1}}^{x_j} f(x)\,\mathrm{d}x.$$

Wendet man nun die Trapezregel auf die einzelnen Integrale an, so erhält man die folgende Approximation des Integrals:

Definition 5.3 (Trapezsumme)**.** Wir definieren die *Trapezsumme* für die Unterteilung $a = x_0 < x_1 < \ldots < x_n = b$ als

$$\mathcal{T}_n[f] := \sum_{j=1}^{n} \left((x_j - x_{j-1}) \frac{f(x_j) + f(x_{j-1})}{2} \right)$$

Für äquidistante Intervallgrenzen $x_j := a + jh$ mit $h := \frac{b-a}{n}$ und $j = 0, \ldots, n$ erhalten wir folgende Formel für die Trapezsumme:

$$\mathcal{T}_n[f] = h\left(\frac{1}{2} f(a) + \sum_{j=1}^{n-1} f(x_j) + \frac{1}{2} f(b) \right). \tag{5.1}$$

Satz 5.4. *Sei $f \in C^2([a,b])$. Dann gilt für die n-te Trapezsumme $\mathcal{T}_n[f]$ in (5.1) die Fehlerabschätzung*

$$\left| \mathcal{I}[f] - \mathcal{T}_n[f] \right| \leq \frac{b-a}{12} \|f''\|_\infty h^2.$$

Beweis. Für $n = 1$ folgt mit Korollar 4.10

$$\left| \mathcal{I}[f] - \mathcal{T}_1[f] \right| \leq \|f''\|_\infty \int_{x_0}^{x_1} \frac{1}{2!} |(x - x_0)(x - x_1)| \, \mathrm{d}x = \|f''\|_\infty \frac{(x_1 - x_2)^3}{12}.$$

Da $h = x_1 - x_0$, folgt daraus $|\mathcal{I}[f] - \mathcal{T}_1[f]| \leq \frac{1}{12}(x_1 - x_0)\|f''\|_\infty h^2$. Der allgemeine Fall folgt durch Anwendung des Falls $n = 1$ auf jedes Teilintervall. □

5.1 Newton-Cotes-Formeln

Etwas allgemeiner betrachten wir nun im Folgenden gewichtete Integrale der Form

$$\mathcal{I}[f, w] := \int_a^b f(x)\, w(x) \, \mathrm{d}x$$

mit einem positiven, integrierbaren *Gewicht* $w : [a,b] \to (0, \infty)$. Das Gewicht wird auch als *Dichtefunktion* oder nur als *Dichte* bezeichnet. Zur Approximation betrachten wir *Quadraturformeln* der Form

$$Q[f] := \sum_{i=0}^{n} w_i \, f(x_i),$$

wobei $w_0, \ldots, w_n$ Gewichte sind. Wir können z.B. $w_i = w(x_i)$ wählen, erlauben uns aber hier die Freiheit w_i auch anders zu wählen.

Bemerkung 5.5. Um Auslöschung bei der Quadratur von Funktionen mit Vorzeichenwechsel zu vermeiden, geht man in manchen Fällen zu *zusammengesetzten Quadraturformeln* über.

Definition 5.6. Die Quadraturformel Q für das Integrationsfunktional $\mathcal{I}[\cdot, w]$ nennen wir *exakt vom Grad d*, falls

$$Q[p] = \mathcal{I}[p, w] \qquad \text{für alle } p \in \mathcal{P}_d.$$

Bemerkung 5.7.

a) Sowohl $\mathcal{I}[\cdot, w]$ als auch $Q[\cdot]$ und $Q_n[\cdot]$ sind linear.

b) Aufgrund der Linearität der Quadraturformel und des gewichteten Integrals genügt es, die Exaktheit für eine Basis von $\mathcal{P}_d$ zu prüfen.

Mit Hilfe der Polynom-Interpolation lassen sich sogenannte *interpolatorische Quadraturformeln* konstruieren. Dabei ersetzen wir zuvor die Funktion f vor dem Integrieren durch ihr Interpolationspolynom.

Definition 5.8. Eine Quadraturformel $Q[f]$ mit den Knoten $x_0, \ldots, x_n$ für das Integral $\mathcal{I}[f, w]$ heißt *interpolatorisch*, falls

$$Q[f] = \mathcal{I}[\varphi_n, w],$$

wobei φ_n das Interpolationspolynom von f zu den Punkten $x_0, \ldots, x_n$ sein soll.

a) Unterteilt man $[a, b]$ gleichmäßig in n Intervalle und nimmt als Knoten alle Intervallgrenzen, d.h. $x_i = a + ih$, $i = 0, 1, \ldots, n$, mit $h = \frac{b-a}{n}$, und ist $w(x) = 1$, so heißt Q_n *geschlossene Newton-Cotes-Formel*.

b) Unterteilt man $[a, b]$ gleichmäßig in $n + 2$ Intervalle und nimmt als Knoten alle inneren Intervallgrenzen, d.h. $x_i = a + (i + 1)h$, $i = 0, 1, \ldots, n$, mit $h = \frac{b-a}{n+2}$, und ist $w(x) = 1$, so heißt Q_n *offene Newton-Cotes-Formel*.

Lemma 5.9. *Seien ℓ_i die Lagrange-Basispolynome zu den Knoten $x_0, \ldots, x_n$ und die Gewichte gegeben durch*

$$w_i = \int_a^b \ell_i(x) w(x) \, \mathrm{d}x = \mathcal{I}[\ell_i, w] \qquad \textit{für } i = 0, \ldots, n,$$

Dann ist Q interpolatorisch.

Beweis. wir haben in (4.2) gezeigt, dass $\varphi_n(x) = \sum_{i=0}^n f(x_i)\ell_i$. Somit:

$$\begin{aligned} Q[f] = \sum_{i=0}^n w_i f(x_i) &= \sum_{i=0}^n \mathcal{I}[\ell_i, w]\, f(x_i) \\ &= \mathcal{I}\Big[\sum_{i=0}^n f(x_i)\ell_i, w\Big] \\ &= \mathcal{I}[\varphi_n, w]. \end{aligned}$$

Also ist Q interpolatorisch, □

Satz 5.10. *Die Quadratur Q sei interpolatorisch zu den Punkten $x_0, \dots, x_n$. Dann gilt:*

a) Q hat Exaktheitsgrad $d \geq n$.

b) Ist $f \in C^{n+1}$, so gilt

$$\big|\mathcal{I}[f,w] - Q[f]\big| \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \int_a^b |\omega(x)|\, w(x)\,\mathrm{d}x,$$

wobei $\omega(x) = \prod_{j=0}^n (x - x_j)$ wieder das Knotenpolynom ist.

Beweis.

a) Sei $p \in \mathcal{P}_k, k \leq n$. Dann stimmt p mit seinem Interpolationspolynom überein, sodass die Quadratur exakt ist.

b) Sei $\varphi_n \in \mathcal{P}_n$ das Interpolationspolyom von f zu den Knotenpunkten $x_0, \dots, x_n$. Da Q interpolatorisch ist, gilt

$$Q[f] = \mathcal{I}[\varphi_n, w].$$

Also ist

$$\mathcal{I}[f,w] - Q[f] = \mathcal{I}[f,w] - \mathcal{I}[\varphi_n,w] = \mathcal{I}[f-\varphi_n,w] = \int_a^b \big(f(x) - \varphi_n(x)\big) w(x)\,\mathrm{d}x.$$

Wegen Korollar 4.10 haben wir

$$|f(x) - \varphi_n(x)| \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} |\omega(x)|.$$

Hieraus folgt die Behauptung.

□

Im Folgenden wollen wir drei Beispiele von interpolatorischen Quadraturen kennenlernen. Wir verwenden Lemma 5.9, um die Gewichte zu setzen.

Beispiel 5.11 (Offene Newton-Cotes-Formel $n = 0$; Mittelpunktsregel)**.** Sei die Gewichtsfunktion $w(x) = 1$. Die Wahl des Mittelpunktes $x_0 = \frac{a+b}{2}$ führt nach Lemma 5.9 zum Gewicht

$$w_0 = \int_a^b \ell_0(x)\,\mathrm{d}x = \int_a^b 1\,\mathrm{d}x = b - a.$$

Dies führt dann zur oben beschriebenen Mittelpunktsregel

$$Q[f] = (b-a) f\Big(\frac{a+b}{2}\Big).$$

Diese Quadratur ist mindestens exakt vom Grad 0. Wegen

$$\mathcal{I}[x] = \int_a^b x\,\mathrm{d}x = \Big[\frac{1}{2}x^2\Big]_a^b = \frac{1}{2}(b^2 - a^2) = (b-a)\frac{a+b}{2} = Q[x]$$

ist diese Quadratur sogar exakt vom Grad 1. Das Beispiel $f(x) = (x - \frac{a+b}{2})^2$ zeigt, dass Q genau Exaktheitsgrad 1 besitzt, denn $Q[f] = 0$, aber $\mathcal{I}[f] > 0$.

Beispiel 5.12 (Geschlossene Newton-Cotes-Formel $n = 1$; Trapezregel)**.** Sei wieder $w(x) = 1$. Die Wahl $x_0 = a$ und $x_1 = b$ führt nach Lemma 5.9 zu

$$w_0 = \int_a^b \ell_0(x)\,\mathrm{d}x = \int_a^b \frac{x-b}{a-b}\,\mathrm{d}x = \frac{b-a}{2}, \quad w_1 = \int_a^b \ell_0(x)\,\mathrm{d}x = \int_a^b \frac{x-a}{b-a}\,\mathrm{d}x = \frac{b-a}{2}.$$

Dies ergibt dann die oben beschriebenens Trapezformel

$$Q[f] = (b-a)\frac{f(a)+f(b)}{2}.$$

Diese Quadratur ist mindestens exakt vom Grad 1. Das Beispiel $f(x) = (b-x)(x-a)$ zeigt, dass Q genau Exaktheitsgrad 1 besitzt, denn $Q[f] = 0$, aber $\mathcal{I}[f] > 0$.

Beispiel 5.13 (Geschlossene Newton-Cotes-Formel $n = 2$; Simpson-Regel)**.** Sei wieder $w(x) = 1$. Die Wahl $x_0 = a$, $x_1 = \frac{a+b}{2}$ und $x_2 = b$ führt zur *Simpson-Regel*. In diesem Fall haben wir

$$\begin{aligned}
w_0 &= \int_a^b \ell_0(x)\,\mathrm{d}x = \int_a^b \frac{2x-(a+b)}{a-b}\cdot\frac{x-b}{a-b}\,\mathrm{d}x = \frac{b-a}{6}\\
w_1 &= \int_a^b \ell_1(x)\,\mathrm{d}x = \int_a^b \frac{2(x-a)}{b-a}\cdot\frac{2(x-b)}{a-b}\,\mathrm{d}x = \frac{2(b-a)}{3}\\
w_2 &= \int_a^b \ell_2(x)\,\mathrm{d}x = \int_a^b \frac{x-a}{b-a}\cdot\frac{2x-(a+b)}{b-a}\,\mathrm{d}x = \frac{b-a}{6}
\end{aligned}$$

Wir erhalten folgende Quadratur

$$Q[f] = (b-a)\frac{1}{6}\left(f(a) + 4\,f\left(\frac{a+b}{2}\right) + f(b)\right).$$

Diese Quadratur ist mindestens exakt von der Ordnung 2. Wegen

$$Q[x^3] = \frac{b-a}{6}\left(a^3 + 4\frac{(a+b)^3}{8} + b^3\right) = \frac{1}{4}(b^4-a^4) = \mathcal{I}[x^3]$$

ist die Formel sogar exakt von der Ordnung 3.

Das Beispiel $f(x) = (b-x)(x-a)(x-\frac{a+b}{2})^2$ zeigt, dass Q genau Exaktheitsgrad 3 besitzt, denn $Q[f] = 0$, aber $\mathcal{I}[f] > 0$.

5.2 Gauß Quadratur

Wir haben schon festgestellt, dass manche Integrationsformeln einen höheren Exaktheitsgrad haben, als man durch die Anzahl der Stützstellen vermuten würden. Können wir durch geschickte Wahl der Stützstellen den Exaktheitsgrad noch weiter erhöhen? Dies wollen wir systematisch untersuchen.

Im Nachfolgenden sei Q die Quadratur, welche durch Integration des Interpolationspolynoms zu den Stützstellen $x_0, \ldots, x_n$ entsteht. Wir haben also $n+1$ Stützstellen. Sei wieder $\omega(x) = \prod_{j=0}^n (x - x_j)$ das Knotenpolynomund $w : [a,b] \to (0,\infty)$ ein positives Gewicht.

Lemma 5.14. *Die Quadratur ist höchstens exakt vom Grad $2n+1 = 2(n+1)-1$, also einen weniger als die doppelte Stützstellenanzahl.*

Beweis. Wir integrieren das Knotenpolynom $\omega^2 \in \mathcal{P}_{2(n+1)}$. Da $\omega(x_0) = \cdots = \omega(x_n) = 0$ folgt $Q[\omega^2] = 0$. Allerdings ist

$$\mathcal{I}[\omega^2, w] = \int_a^b \big(\omega(x)\big)^2 w(x)\,\mathrm{d}x > 0.$$

Damit ist $\mathcal{I}[\omega^2] \neq Q[\omega^2]$. □

Im Folgenden definieren wir das (gewichtete) Skalarprodukt

$$\langle f, g\rangle_w := \int_a^b f(x)\, g(x)\, w(x)\,\mathrm{d}x$$

mit der zugehörigen Norm

$$\|f\|_w := \sqrt{\langle f, f\rangle_w}$$

auf dem Vektorraum $L^2([a,b], w(x)\,\mathrm{d}x)$ der quadratintegrierbaren Funktionen bzgl. der Dichtefunktion w.

Satz 5.15. *Zu jeder Gewichtsfunktion w existieren (eindeutig bestimmte) Polynome $u_0, u_1, \ldots$ mit $u_k(x) = \gamma_k x^k + \ldots$, wobei $\gamma_k > 0$, und*

$$\langle u_j, u_k\rangle_w = \delta_{j,k}.$$

Insbesondere ist

$$u_0(x) = \gamma_0 = \left(\int_a^b w(x)\,\mathrm{d}x\right)^{-\frac{1}{2}}.$$

Mit $u_{-1} := 0$ gilt die Dreiterm-Rekursion

$$\begin{aligned}
u_{n+1}(x) &:= \frac{v_{n+1}(x)}{\|v_{n+1}\|_w} \\
v_{n+1}(x) &= x\,u_n(x) - \alpha_n u_n(x) - \beta_n u_{n-1}(x), \\
\alpha_n &= \langle u_n, x u_n\rangle_w, \\
\beta_n &= \langle u_n, x u_{n-1}\rangle_w, \\
\beta_0 &:= 0.
\end{aligned}$$

Beweis. Wir setzen $u_0, \ldots, u_n$ wie oben beschrieben. Dann gilt $\langle u_k, u_\ell\rangle = \delta_{k,\ell}$ und

$$u_n(x) := \frac{v_n(x)}{\|v_n\|_w} =: \gamma_n x^n + \ldots, \text{ wobei } \gamma_n > 0.$$

Sei dann für $n = 1, 2, \ldots,$

$$X_n = \{p \in \mathcal{P}_n \mid \ p \text{ ist orthogonal bezüglich } \langle\cdot,\cdot\rangle_w \text{ auf } u_0, \ldots, u_{n-1}\}.$$

Dann ist $u_n \in X_n$. Außerdem ist $\dim X_n = 1$ und daher ist u_n eindeutig bestimmt. □

Beispiel 5.16. Sei $[a,b] = [-1,1]$ und $w(x) = \frac{1}{\sqrt{1-x^2}}$. So gilt für die Chebyshev-Polynome T_n

$$\begin{aligned}\langle T_j, T_k\rangle_w &= \int_{-1}^{1} T_j(x)T_k(x)\frac{1}{\sqrt{1-x^2}}\,\mathrm{d}x \\ &= \int_{-1}^{1} \cos\big(j\arccos x\big)\cos\big(k\arccos(x)\big)\frac{1}{\sqrt{1-x^2}}\,\mathrm{d}x \\ &= \int_{0}^{\pi} \cos(jt)\cos(kt)\,dt \\ &= \frac{1}{2}\int_{-\pi}^{\pi} \cos(jt)\cos(kt)\,dt\end{aligned}$$

mit der Substitution $x = \cos t$, $dx = -\sin t\,dt = -\sqrt{1-\cos^2 t}\,dt$.

Für $j = k$ ist (unter Berücksichtigung der Periodizität)

$$\begin{aligned}\langle T_0, T_0\rangle_w &= \frac{1}{2}\int_{-\pi}^{\pi} 1\,dt = \pi, \\ \langle T_k, T_k\rangle_w &= \frac{1}{2}\int_{-\pi}^{\pi} \cos^2(kt)\,dt \\ &= \frac{1}{2}\frac{1}{k}\int_{-k\pi}^{k\pi} \cos^2(s)\,ds \\ &= \frac{1}{2}\int_{-\pi}^{\pi} \cos^2(s)\,ds \\ &= \frac{\pi}{2} \qquad \text{für } k \geq 1.\end{aligned}$$

Für $j \neq k$ ergibt sich mittels zweifacher partielle Integration

$$\langle T_j, T_k\rangle_w = \cdots = \frac{j^2}{k^2}\langle T_j, T_k\rangle_w.$$

Hieraus folgt

$$\langle T_j, T_k\rangle_w = 0 \qquad \text{für } j \neq k.$$

D.h. die *normierten Chebyshev-Polynome*

$$\begin{aligned}\widetilde{T}_0(x) &= \frac{1}{\sqrt{\pi}}, \\ \widetilde{T}_n(x) &= \sqrt{\frac{2}{\pi}}T_n(x) \qquad \text{für } n \geq 1\end{aligned}$$

sind die gesuchten paarweise $\langle\cdot,\cdot\rangle_w$-orthormalen Polynome.

Für die Höchstkoeffizienten und die Rekursionskoeffizienten gilt

$$\begin{aligned}\gamma_0 &= \frac{1}{\sqrt{\pi}} \\ \gamma_k &= \frac{2^k}{\sqrt{2\pi}}, \qquad \text{für } k \in \mathbb{N} \\ \beta_1 &= \frac{1}{\sqrt{2}} \\ \beta_k &= \frac{1}{2} \qquad \text{für } k \geq 2, \\ \alpha_k &= 0 \qquad \text{für } k \in \mathbb{N}.\end{aligned}$$

Beispiel 5.17. Betrachen wir $[a, b] := [-1, 1]$ mit $w(x) = 1$. So gilt für die *Legendre-Polynome*

$$P_k(x) = \frac{1}{2^k k!} \frac{d^k}{dx^k}(x^2 - 1)^k$$

die Orthogonalitätsrelation

$$\langle P_j, P_k \rangle = \frac{2}{2k+1} \delta_{j,k}.$$

Es gilt die Dreiterm-Rekursion

$$\begin{aligned}P_0(x) &= 1, \\ P_1(x) &= x, \\ nP_n(x) &= (2n-1)xP_{n-1}(x) - (n-1)P_{n-2}(x) \qquad \text{für } n \geq 2.\end{aligned}$$

Die normierten Legendre-Polynome

$$\widetilde{P}_k(x) := \sqrt{\frac{2k+1}{2}} P_k(x)$$

sind die gesuchten $\langle \cdot, \cdot \rangle$-orthonormalen Polynome.

Für die Höchstkoeffizienten und die Rekursionskoeffizienten gilt

$$\begin{aligned}\gamma_k &= \sqrt{\frac{2k+1}{2}} \frac{(2k)!}{2^k (k!)^2}, \qquad \text{für } k \in \mathbb{N}_0 \\ \beta_k &= \frac{k}{\sqrt{4k^2 - 1}} \qquad \text{für } k \in \mathbb{N}, \\ \alpha_k &= 0 \qquad \text{für } k \in \mathbb{N}.\end{aligned}$$

Die ersten Legendre-Polynome sind

$$\begin{aligned}P_0(x) &= 1, \\ P_1(x) &= x, \\ P_2(x) &= \frac{3x^2 - 1}{2}, \\ P_3(x) &= \frac{5x^3 - 3x}{2}.\end{aligned}$$

Satz 5.18. *Die Nullstellen der u_k sind einfach.*

Beweis. Betrachte u_k. Seien $a_1 \dots, a_m$ die Nullstellen ungerader Ordnung von u_k. Wir zeigen zunächst, dass $m \geq k$. Hieraus folgt dann durch abzählen und $\operatorname{grad}(u_k) = k$, dass alle Nullstellen von u_k einfach sein müssen.

Wir nehmen nun das Gegenteil nämlich $m < k$ an. Sei

$$q(x) := \prod_{j=1}^{m} (x - a_j) \in \mathcal{P}_m.$$

Dann hat $u_k(x)q(x)$ nur Nullstellen gerader Ordnung mit positiven führendem Koeffizienten und ist damit nicht-negativ. Da $d := \operatorname{grad}(q) < k$, können wir $q = \sum_{i=0}^{d} b_i u_i$ für Koeffizienten b_i schreiben. Aus der Orthogonalität der u_i folgt dann:

$$0 = \sum_{i=1}^{d} b_i \langle u_k, u_i \rangle_w = \langle u_k, q \rangle_w = \int_a^b \underbrace{u_k(x) \prod_{j=1}^{m} (x - a_j) w(x)}_{\geq 0} \, \mathrm{d}x > 0.$$

Dies ist der gewünschte Widerspruch. □

Definition 5.19. Zu $m \in \mathbb{N}$ seien $u_0, \dots, u_m$ die $\langle \cdot, \cdot \rangle_w$-Orthonormalbasis von $\mathcal{P}_m$ aus Satz 5.15. Seien nun $x_1, \dots, x_m$ die m Nullstellen von u_m. Dann definieren wir die m-stufige Gauß-Quadratur $\mathcal{G}_m$ durch

$$\mathcal{G}_m[f, w] := \sum_{i=1}^{m} w_i f(x_i),$$

dabei sind die Gewichte $w_1, \dots, w_m$ gegeben durch

$$w_j = \int_a^b \ell_j(x) w(x) \, \mathrm{d}x,$$

wobei $\ell_1, \dots, \ell_m$ die zugehörigen Lagrange-Basispolynome sind.

Satz 5.9 impliziert folgendes Lemma.

Lemma 5.20. *Die oben definierte Gauß-Quadratur ist interpolatorisch.*

Wir haben in Lemma 5.14 bewiesen, dass eine Quadratur mit m Stützstellen exakt von Grad maximal $2m - 1$ sein kann. Der folgende Satz zeigt, dass die GaußQuadratur dieses Maximum erreicht.

Satz 5.21. *Die Gauß-Quadratur $\mathcal{G}_m$ ist exakt von der Ordnung $2m - 1$. Die Gewichte $w_1, \dots, w_m$ sind alle positiv.*

Beweis. (i) *Exaktheitsgrad* $2m-1$: Seien $x_1, \dots, x_m$ wie in der Definition von $\mathcal{G}_m$. Sei $\omega(x) := (x-x_1)\cdots(x-x_m) \in \mathcal{P}_m$ das Knotenpolynom. Dann ist $u_m(x) = \gamma_m \omega(x)$.

Die Polynome $1, x, \dots, x^{m-1}, \omega(x), x\omega(x), \dots, x^{m-1}\omega(x)$ sind eine Basis von $\mathcal{P}_{2m-1}$. Da $\mathcal{G}_m$ interpolatorisch ist, werden $1, \dots, x^{m-1}$ exakt integriert, d.h.

$$\mathcal{I}[x^k, w] = \mathcal{G}_m[x^k, w], \quad k = 0, \dots, m-1.$$

Weiterhin gilt

$$\mathcal{G}_m[x^k\omega, w] = \sum_{i=1}^{m} w_i x_i^k \omega(x_i) = 0.$$

Mit $u_m(x) = \gamma_m\omega(x)$ und $u_m \perp \mathcal{P}_{m-1}$ bzgl. $\langle\cdot,\cdot\rangle_w$ folgt

$$\mathcal{I}[x^k\omega, w] = \int_a^b x^k\omega(x)w(x)\,\mathrm{d}x = \langle x^k, \omega\rangle_w = \frac{1}{\gamma_m}\langle x^k, u_m\rangle_w = 0.$$

Also ist $\mathcal{G}_m[x^k\omega, w] = \mathcal{I}[x^k\omega, w]$. Insgesamt haben wir gezeigt, dass

$$\mathcal{G}_m[q, w] = \mathcal{I}[q, w] \qquad \text{für alle } q \in \mathcal{P}_{2m-1}.$$

Insbesondere ist $\mathcal{G}_m$ exakt von der Ordnung $2m-1$.

(ii) *Formel für die Gewichte*: Wegen der Exaktheit von $\mathcal{G}_m$ der Ordnung $2m-1$ und $(\ell_i)^2 \in \mathcal{P}_{2m-2}$ gilt

$$0 < \int_a^b \big(\ell_i(x)\big)^2 w(x)\,\mathrm{d}x = \mathcal{G}_m[\ell_i^2, w] = \sum_{j=1}^{m} w_j(\ell_i(x_j))^2 = w_i.$$

Dies beendet den Beweis. □

Bemerkung 5.22. Die Gauß-Quadratur hat ob ihrer positiven Gewichte den Vorteil, dass es bei positiven Integranden nicht zur Auslöschung kommen kann. Das Integral einer positiven Funktion ist außerdem auch immer positiv.

Beispiel 5.23. Wir wollen die Gauß-Quadratur für $[a, b] = [-1, 1]$ und $w(x) = 1$ herleiten. Hierzu benötigen wir die Nullstellen der Legendre-Polynome, vgl. Beispiel 5.17.

a) Wir beginnen mit $m = 1$. Dann ist $P_1(x) = x$. Die Nullstelle ist $x = 0$. Wir haben also $x_1 = 0$. Damit ist $\ell_1(x) = 1$. Es folgt

$$w_1 = \langle 1, 1\rangle = \int_{-1}^{1} \mathrm{d}x = 2.$$

Also ist

$$\mathcal{G}_1(f) = 2\, f(0).$$

Dies ist gerade die Mittelpunktsformel, von der wir aus Beispiel 5.11 wissen, dass sie Exaktheitsgrad 1 besitzt.

b) Sei nun $m = 2$. Dann ist $P_2(x) = \frac{3x^2-1}{2}$. Die Nullstellen sind $x_1 = -\frac{1}{\sqrt{3}}$ und $x_2 = \frac{1}{\sqrt{3}}$. Es ergibt sich

$$w_1 = \langle \ell_1, \ell_1\rangle = \int_{-1}^{1} \left(\frac{x - \frac{1}{\sqrt{3}}}{-2\frac{1}{\sqrt{3}}}\right)^2 \mathrm{d}x = \cdots = 1$$

sowie aus Symmetriegründen

$$w_2 = \langle \ell_2, \ell_2\rangle = \int_{-1}^{1} \left(\frac{x + \frac{1}{\sqrt{3}}}{2\frac{1}{\sqrt{3}}}\right)^2 \mathrm{d}x = 1.$$

Also ist

$$\mathcal{G}_2(f) = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right).$$

Wie bei der Trapezregel benötigen wir zwei Punkte; die Trapezregel ist jedoch nur exakt von der Ordnung 1 (s. Beispiel 5.12), $\mathcal{G}_2$ hingegen von der Ordnung 3.

c) Analog berechnet man für $m = 3$

$$\mathcal{G}_3(f) = \frac{5}{9} f\left(-\frac{\sqrt{15}}{5}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\frac{\sqrt{15}}{5}\right).$$

Die Formel ist exakt von der Ordnung 5. Die Simpson-Regel mit 3 Punkten hingegen ist nur exakt von der Ordnung 3 (s. Beispiel 5.13).

Konkret ergibt dies als Approximation für das Integral $\int_{-1}^{1} e^x \,\mathrm{d}x = e - e^{-1} \approx 2.35040$

$$\begin{aligned}
&\mathcal{G}_1(e^x) = 2,\\
&\mathcal{G}_2(e^x) \approx 2.34270\\
&\mathcal{G}_3(e^x) \approx 2.35034.
\end{aligned}$$

Beispiel 5.24. Wir wollen die Gauß-Quadratur zu $[a,b] = [-1,1]$ und $\omega(x) = \frac{1}{\sqrt{1-x^2}}$ bestimmen. Es gilt

$$\mathcal{I}[f,\omega] = \int_{-1}^{1} f(x)\frac{1}{\sqrt{1-x^2}}\,\mathrm{d}x \approx \mathcal{G}_m[f,\omega] = \sum_{i=1}^{m} w_i f(x_i)$$

mit den Quadraturknoten (Nullstellen der Chebychev-Polynome, vgl. Beispiel 5.16)

$$x_i = \cos\frac{(2i-1)\pi}{2m} \qquad \text{für } i = 1,\ldots,m,$$

und den Quadraturgewichten (ohne Beweis)

$$w_i = \frac{\pi}{m}.$$

Beispiel 5.25. Wir berechnen

$$\int_{-1}^{1} e^x \frac{1}{\sqrt{1-x^2}}\,\mathrm{d}x \approx 3.97746.$$

Sei $[a,b] = [-1,1]$, $\omega(x) = \frac{1}{\sqrt{1-x^2}}$ und $m = 2$. Wir brauchen zur Berechnung von $\mathcal{G}_2$ die Nullstellen des Chebychev-Polynoms T_2 vom Grad 2. Diese sind $x_1 = \cos(\frac{\pi}{4}) = \frac{1}{\sqrt{2}}$ und $x_2 = \cos(\frac{3\pi}{4}) = -\frac{1}{\sqrt{2}}$. Wir erhalten

$$\mathcal{G}_2(e^x,\omega) = \frac{\pi}{2}e^{\frac{1}{\sqrt{2}}} + \frac{\pi}{2}e^{-\frac{1}{\sqrt{2}}} \approx 3.96027.$$

Für $m = 3$ haben wir $x_1 = \frac{\sqrt{3}}{2}$, $x_2 = 0$, $x_3 = -\frac{\sqrt{3}}{2}$ und $\omega_1 = \omega_2 = \omega_3 = \frac{\pi}{3}$. Es folgt

$$\mathcal{G}_3[f,\omega] = \frac{\pi}{3}e^{\sqrt{3}/2} + \frac{\pi}{3}e^{0} + \frac{\pi}{3}e^{-\sqrt{3}/2} = 3.977322.$$

Literaturverzeichnis

[1] P. Bürgisser and F. Cucker. *Condition: The Geometry of Numerical Algorithms*. Springer, Heidelberg. 2013.

[2] P. Breiding, K. Kohn, B. Sturmfels. *Metric Algebraic Geometry*. Birkhäuser. 2024.

[3] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, 1996.

[4] M. Hanke-Bourgeois. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*, 3. aktualisierte Auflage, Vieweg + Teubner, Wiesbaden, 2009.

[5] N. J. Higham, *Accuracy and stability of numerical algorithms*, Society for Industrial and Applied Mathematics, 1996, 2. Auflage.

[6] R. Meinicken, E. Wagenführer. *Numerische Mathematik, Band 2*, Vieweg, 1977.

[7] M. L. Overton. *Numerical Computing with IEEE Floating Point Arithmetic.*, SIAM 2001.

[8] R. Plato. *Numerische Mathematik kompakt*, Vieweg-Verlag, Wiesbaden, 3. Auflage, 2006.

[9] L. N. Trefethen und D. Bau, *Numerical Linear Algebra*, SIAM, 1997.

[10] A. Turing. *Rounding-off errors in matrix processes*. The Quarterly Journal of Mechanics and Applied Mathematics 1(1), 1948, pages 287–308.