Vorlesungsskript
Einführung in die Numerik

Sommersemester 2022

Roland Herzog*

2022-06-24

*Interdisciplinary Center for Scientific Computing, Heidelberg University, 69120 Heidelberg, Germany (roland.herzog@iwr.uni-heidelberg.de, https://scoop.iwr.uni-heidelberg.de/team/roland-herzog).

Material für 13 Wochen.

Inhaltsverzeichnis

Kapitel 0 Einführung

§ 1 Was ist Numerische Mathematik?

Die **numerische Mathematik** (kurz: **Numerik**, englisch: *numerical analysis*) beschäftigt sich mit der Konstruktion und Analyse von Rechenverfahren (Algorithmen) zur Lösung von Aufgaben der «kontinuierlichen Mathematik» (im Gegensatz zur diskreten Mathematik) auf Computern. Das englische Originalzitat «Numerical analysis is the study of algorithms for the problems of continuous mathematics.» hinter dieser Definition stammt von Trefethen, 1992. Ein typisches Beispiel einer Grundaufgabe der Numerik ist etwa das Lösen linearer Gleichungssysteme $Ax = b$ mit Hilfe des Gauß-Algorithmus.

Die Numerik ist damit ein Teilgebiet des **wissenschaftlichen Rechnens** (englisch: *scientific computing, computational science*), bei dem es darum geht, eine konkrete Fragestellung (etwa aus der Physik) in ein Modell und anschließend in ein numerisches Verfahren zu übersetzen, dieses auf geeigneter Hardware effizient zu implementieren und die Simulationsergebnisse mit realen Beobachtungen oder Experimenten abzugleichen, um damit die ursprüngliche Fragestellung beantworten zu können.

Nicholas Higham hat 2016 eine Liste der «Top 10»-Algorithmen in der angewandten Mathematik veröffentlicht[1], indem er die Verweise im Index des über 1000-seitigen Buches *The Princeton Companion to Applied Mathematics* (Higham u. a., 2016) gezählt hat. Herausgekommen ist dabei folgende Liste:

(1) Newton and quasi-Newton methods
(2) Matrix factorizations (LU, Cholesky, QR)
(3) Singular value decomposition, QR and QZ algorithms
(4) Monte-Carlo methods
(5) Fast Fourier transform
(6) Krylov subspace methods (conjugate gradients, Lanczos, Gmres, Minres)
(7) JPEG
(8) PageRank
(9) Simplex algorithm
(10) Kalman filter

Wir behandeln im Verlauf dieser Lehrveranstaltung von dieser Liste voraussichtlich die Punkte (1) bis (3), (6) und (8). Für das Simplex-Verfahren (Punkt (9)) verweisen wir auf die Lehrveranstaltung *Grundlagen der Optimierung*.

Zur Einführung gehen wir hier schon einmal auf die PageRank-Aufgabe ein. Dabei handelt es sich um

[1] https://press.princeton.edu/ideas/nicholas-higham-on-the-top-10-algorithms-in-applied-mathematics

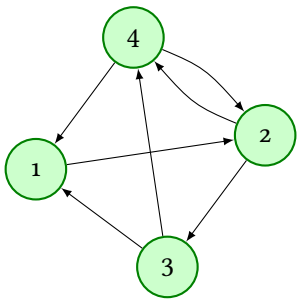


Abbildung 1.1: Ein Miniatur-Web zur Demonstration des PageRank-Verfahrens.

die von Sergey Brin und Lawrence Page in Brin, Page, 1998 vorgeschlagenen Ranking-Methode für Webseiten, die seit dem Gründungsjahr 1998 von Google bestimmt, in welcher Reihenfolge Suchergebnisse der Google-Suche erscheinen. Unsere Beschreibung folgt Higham u. a., 2016, Chapter VI.9.

Die in das Ranking einzubeziehenden Webseiten werden als sogenannter einfacher gerichteter Graph dargestellt, siehe Abbildung 1.1. Dabei sind die Webseiten die **Knoten** (englisch: *vertex*) des Graphen, und ein Pfeil (eine **Kante**, englisch: *edge, arc*) bedeutet, dass eine Webseite per Link auf eine andere verweist.

Der Graph wird mittels einer **Adjazenzmatrix** (englisch: *adjacency matrix*) codiert:

$$A = \begin{array}{c} \begin{matrix} ① & ② & ③ & ④ \end{matrix} \\ \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array} \begin{matrix} ① \\ ② \\ ③ \\ ④ \end{matrix} \tag{1.1}$$

Dabei sind in der ersten Spalte diejenigen Kanten aufgeführt, die in Knoten ① starten usw.[2] In der ersten Zeile finden wir diejenigen Kanten, die im Knoten ① enden.

Quizfrage: Welche Bedeutung haben die Spaltensummen von A? Welche Bedeutung haben die Zeilensummen von A?

Wir stellen uns nun eine Person vor, die sich aktuell auf einer dieser Webseiten j aufhält. Ihr Verhalten wird wie folgt modelliert. Mit einer Wahrscheinlichkeit α wird die Person einem (von der aktuellen Webseite wegführenden) Link folgen. Jedem zur Verfügung stehenden Link wird dabei mit derselben Wahrscheinlichkeit gefolgt. Andernfalls, also mit Wahrscheinlichkeit $1 - \alpha$, begibt sich die Person zu einer zufälligen Webseite aus ①–④.

Befindet sich die Person beispielsweise zu einem Zeitpunkt auf Webseite ②, so wird sie sich zum

[2] **Beachte:** Oft wird die Adjazenzmatrix gerichteter Graphen gerade als die Transponierte unserer Matrix A definiert. Dann stehen die Zeilen für die Anfangsknoten und die Spalten für die Endknoten. Für uns ist aber die Konvention wie in (1.1) günstig.

nächsten Zeitpunkt

$$\begin{aligned}
&\text{auf Webseite } ① \text{ mit Wahrscheinlichkeit } \alpha \cdot \frac{0}{2} + (1-\alpha) \cdot \frac{1}{4} \\
&\text{auf Webseite } ② \text{ mit Wahrscheinlichkeit } \alpha \cdot \frac{0}{2} + (1-\alpha) \cdot \frac{1}{4} \\
&\text{auf Webseite } ③ \text{ mit Wahrscheinlichkeit } \alpha \cdot \frac{1}{2} + (1-\alpha) \cdot \frac{1}{4} \\
&\text{auf Webseite } ④ \text{ mit Wahrscheinlichkeit } \alpha \cdot \frac{1}{2} + (1-\alpha) \cdot \frac{1}{4}
\end{aligned}$$

befinden. (**Quizfrage:** Klar?) Der Faktor $\frac{1}{2}$ kommt daher, dass die Webseite ② gerade zwei wegführende Links hat.

Insgesamt kann man diese Überlegungen in der Matrix Π der Übergangswahrscheinlichkeiten zusammenfassen, die in unserem Beispiel wie folgt aussieht:

$$\Pi = \alpha \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & \frac{1}{2} & & \\ & & \frac{1}{2} & \\ & & & \frac{1}{2} \end{bmatrix} + (1-\alpha)\frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{1.2}$$

Die Einträge π_{ij} dieser Matrix beschreiben, mit welcher Wahrscheinlichkeit die Person von Webseite ⓙ (Spalte) im nächsten Zeitpunkt auf die Webseite ⓘ (Zeile) gelangt.

Besteht das Netzwerk nun allgemeiner aus n Webseiten, ist weiterhin A die Adjazenzmatrix und D die Diagonalmatrix mit den Kehrwerten der Anzahlen der wegführenden Links[3] sowie $e = (1, 1, \ldots, 1)^\mathsf{T} \in \mathbb{R}^n$, dann hat die Matrix der Übergangswahrscheinlichkeiten die Gestalt

$$\Pi = \alpha\, A\, D + (1-\alpha)\frac{1}{n} e\, e^\mathsf{T}. \tag{1.3}$$

Quizfrage: Welche Eigenschaften fallen Ihnen zu dieser Matrix ein? Was sind insbesondere die Spaltensummen $\sum_{i=1}^n \pi_{ij}$?

Welcher Zusammenhang besteht nun zwischen der Matrix Π und dem Ranking der Webseiten in unserem Netzwerk? Dazu betrachten wir folgendes Gedankenexperiment. Nehmen wir an, die Person startet zum Zeitpunkt 0 auf Webseite ②, also mit einer Wahrscheinlichkeitsverteilung von $x^{(0)} = (0, 1, 0, 0)^\mathsf{T}$. Zum Zeitpunkt 1 ist sie dann mit den Wahrscheinlichkeiten auf den Seiten ①–④ angekommen, die sich aus den Einträgen von

$$x^{(1)} = \Pi x^{(0)}$$

ergeben. Zum Zeitpunkt 2 haben wir dann die Verteilung $x^{(2)} = \Pi x^{(1)} = \Pi^2 x^{(0)}$ usw. Man kann im Fall von $\alpha \in (0, 1)$ zeigen, dass diese Folge $(x^{(n)})$ der Wahrscheinlichkeiten gegen eine stationäre Verteilung konvergiert, und zwar unabhängig davon, mit welcher Anfangsverteilung $x^{(0)}$ wir beginnen.[4] Die Eigenschaft, dass $x^{(n)}$ eine Wahrscheinlichkeitsverteilung beschreibt (also $x^{(n)} \geqslant 0$ und $e^\mathsf{T} x^{(n)} = 1$) überträgt sich von der Anfangsverteilung $x^{(0)}$ auf alle Nachfolger. **Quizfrage:** Warum?

[3] Wir gehen hier der Einfachheit davon aus, dass jede Webseite im betrachteten Netzwerk wenigstens einen von der Seite wegführenden Link hat.

[4] Wir haben es hier mit einer stationären Markow-Kette mit endlichem Zustandsraum zu tun.

Die Stationarität bedeutet, dass sich die Wahrscheinlichkeiten nicht mehr ändern, dass also gilt:

$$\Pi x = x. \tag{1.4}$$

Diese Gleichung (1.4) kann man auf verschiedene Arten interpretieren. Eine davon besagt, dass x ein Eigenvektor der Matrix Π zum Eigenwert $\lambda = 1$ des Eigenwertproblems

$$\Pi x = \lambda x \tag{1.5}$$

ist. Man kann unter geeigneten Annahmen zeigen (Stichwort: Perron-Frobenius-Theorem),

- dass tatsächlich $\lambda = 1$ immer ein Eigenwert der Übergangsmatrix Π ist und daher eine Lösung $x \neq 0$ von (1.5) existiert,
- dass alle anderen Eigenwerte von Π vom Betrag her $\leqslant 1$ sind,
- dass $\lambda = 1$ der einzige strikt positive, reelle Eigenwert von Π ist — die anderen Eigenwerte sind entweder negativ reell oder komplexwertig,
- dass die algebraische, also auch die geometrische Vielfachheit des Eigenwerts $\lambda = 1$ gleich eins ist,
- dass der zugehörige Eigenvektor x so gewählt werden kann, dass er nur Einträge $\geqslant 0$ hat und für die Gesamtsumme gilt: $e^\top x = 1$.

Die Größe der Einträge des gesuchten Eigenvektors x gibt gerade die Wichtigkeit der zugehörigen Webseite ①–④ im Sinne der PageRank-Idee an. Die PageRank-Aufgabe ist also letztendlich eine **Eigenwertaufgabe** (englisch: *eigenvalue problem*)!

Angesichts der zur Zeit (Stand: 2022) geschätzten Anzahl von etwa $n \approx 2 \cdot 10^9$ Webseiten und der entsprechenden Dimensionen der Matrizen A und Π ist neben der Frage, wie/ob man diese Matrizen speichert, auch die Frage relevant, wie man den gesuchten Eigenvektor effizient berechnet.

Quizfrage: Die Adjazenzmatrix A des heutigen Internet ist extrem dünn besetzt. Trifft das auch für die Matrix Π zu? Was hat das für Konsequenzen?

Wir nehmen hier vorweg, dass glücklicherweise für die Bestimmung von Eigenpaaren für betragsgrößte Eigenvektoren einer Matrix mit der **Potenzmethode** (auch **Vektoriteration**, **Von-Mises-Iteration**, englisch: *power iteration*) eine matrixfreie Methode zur Verfügung steht. Wir geben sie im Folgenden an und kommen darauf später in der Lehrveranstaltung zurück. Für die Matrix Π aus (1.2) kann man die Potenzmethode wie in Algorithmus 1.1 gezeigt aufschreiben.

Algorithmus 1.1 (PageRank-Algorithmus mittels Potenzmethode)**.**
Eingabe: *Adjazenzmatrix* $A \in \{0,1\}^{n\times n}$
Eingabe: *Link-Folge-Wahrscheinlichkeit* $\alpha \in (0,1)$
Ausgabe: *ungefährer Eigenvektor* $x \geqslant 0$ *der Matrix* Π *aus* (1.3) *zum Eigenwert* $\lambda = 1$

1: *Initialisiere* $x^{(0)} := e/n$ // $e = (1,1,\ldots,1)^\top \in \mathbb{R}^n$
2: *Setze* $k := 0$
3: ***while*** *noch nicht konvergiert* ***do***
4: *Setze* $x^{(k+1)} := \alpha A D x^{(k)} + (1-\alpha)\frac{1}{n}e$ // $x^{(k+1)} = \Pi x^{(k)}$
5: *Setze* $x^{(k+1)} := x^{(k+1)}/e^\top x^{(k+1)}$ // *Normalisierung*

6: *Setze* $k := k + 1$
7: ***end while***

Als Konvergenzkriterium in Zeile 3 kann man etwa heranziehen, ob sich die Einträge von $x^{(k+1)}$ nur noch sehr wenig von denen von $x^{(k)}$ unterscheiden. Die Initialisierung in Zeile 1 kann auch anders erfolgen, etwa indem man $x^{(0)}$ mit Werten in $[0, 1]$ zufällig wählt und dann normalisiert, sodass $e^\mathsf{T} x^{(0)} = 1$ gilt, oder durch die Wahl eines Einheitsvektors (ein Eintrag eins, Rest Nullen).

Quizfrage: Wie kommt man auf die Anweisung in Zeile 4? Was ist mit der Matrix in (1.3) passiert, die als lauter Einsen besteht?

Quizfrage: Wozu könnte der Schritt in Zeile 5 dienen? Sollte dieser nicht eigentlich überflüssig sein?

Für unser obiges Beispiel aus Abbildung 1.1 ergibt sich bei $\alpha = 0.85$ und Startvektor $x^{(0)} = (0, 1, 0, 0)^\top$ auf vier Stellen gerundet die Folge

$$\begin{aligned}
x^{(0)} &= (0.0000, 1.0000, 0.0000, 0.0000)^\mathsf{T} \\
x^{(1)} &= (0.0375, 0.0375, 0.4625, 0.4625)^\mathsf{T} \\
x^{(1)} &= (0.4306, 0.2659, 0.0534, 0.2500)^\mathsf{T} \\
&\vdots \quad \vdots \\
x^{(23)} &= (0.2240, 0.3374, 0.1809, 0.2578)^\mathsf{T} \\
x^{(24)} &= (0.2239, 0.3374, 0.1809, 0.2578)^\mathsf{T} \\
x^{(25)} &= (0.2239, 0.3374, 0.1809, 0.2578)^\mathsf{T} \\
&\vdots \quad \vdots
\end{aligned}$$

Die Abhängigkeit der PageRank:s von der Link-Folge-Wahrscheinlichkeit α für das Beispiel-Netzwerk aus Abbildung 1.1 wird in Abbildung 1.2 gezeigt. (**Quizfrage:** Wie kann man Abbildung 1.2, vor allem für α in der Nähe von 0 und 1 erklären?) Im Original-Paper Brin, Page, 1998 wird eine Link-Folge-Wahrscheinlichkeit von $\alpha = 0.85$ angenommen.

§ 2 Wiederholung und Notation

Wir gehen davon aus, dass der Leser mit Grundkonzepten der mehrdimensionalen Analysis und der linearen Algebra vertraut ist, darunter insbesondere Stetigkeit und Differenzierbarkeit von Funktionen sowie Matrix- und Vektorrechnung im Vektorraum $\mathbb{R}^n$.[5]

Wir bezeichnen die nicht-negativen bzw. die positiven reellen Zahlen mit

$$\mathbb{R}_{\geqslant 0} := \{x \in \mathbb{R} \mid x \geqslant 0\} \quad \text{und} \quad \mathbb{R}_{>0} := \{x \in \mathbb{R} \mid x > 0\}.$$

Die natürlichen Zahlen sind $\mathbb{N} = \{1, 2, \dots, \}$, und wir setzen $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.

[5] Vieles kann drekt auch auf den komplexen Vektorraum $\mathbb{C}^n$ übertragen werden.

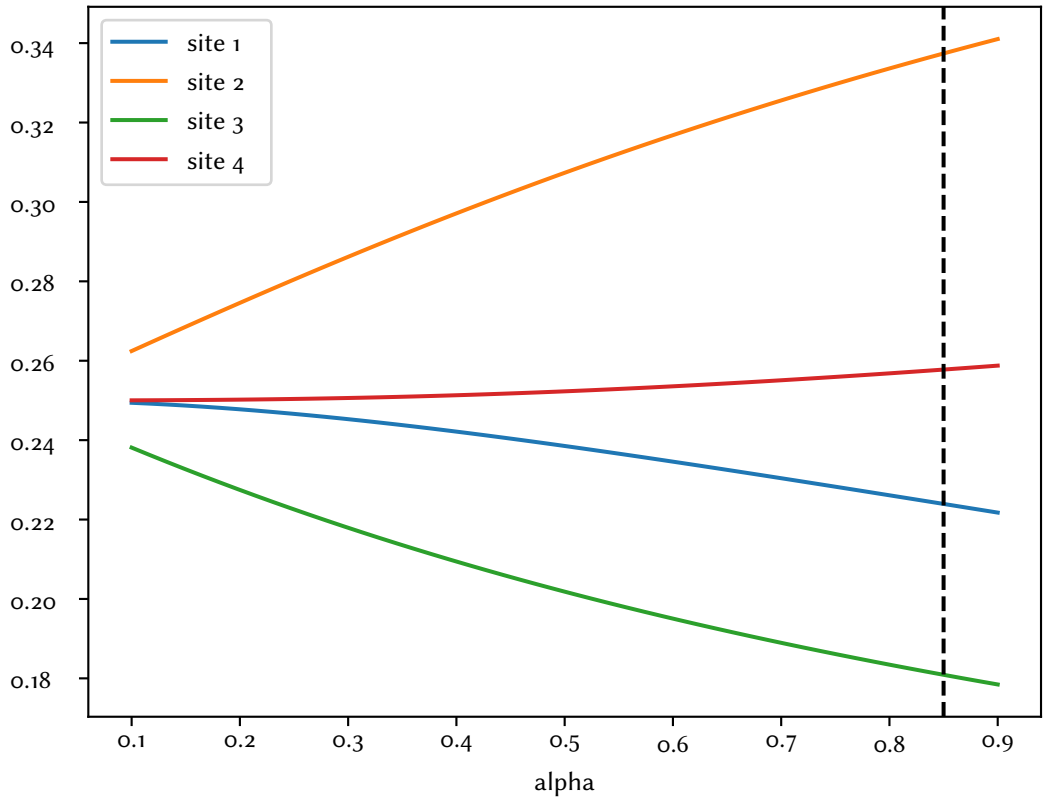


Abbildung 1.2: Gewichte (Einträge im Eigenvektor zum Eigenwert $\lambda = 1$) der Webseiten für das Beispiel-Netzwerk aus Abbildung 1.1 in Abhängigkeit der Link-Folge-Wahrscheinlichkeit α.

§ 2.1 Vektor- und Matrix-Normen

Definition 2.1 (Norm). *Es sei V irgendein Vektorraum über $\mathbb{R}$. Eine Abbildung $\|\cdot\| : V \to \mathbb{R}$ heißt eine* ***Norm*** *(englisch:* norm*) auf V, wenn für beliebige $x, y \in V$ und $\alpha \in \mathbb{R}$ folgende Eigenschaften gelten:*

(*i*) $\|x\| \geqslant 0$ *und* $\|x\| = 0 \Longrightarrow x = 0$.
*(**Definitheit**)*

(*ii*) $\|\alpha x\| = |\alpha| \, \|x\|$.
*(**positive Homogenität**)*

(*iii*) $\|x + y\| \leqslant \|x\| + \|y\|$.
*(**Subadditivität** / **Dreiecksungleichung**)*

Vektornormen

Wir arbeiten in dieser Lehrveranstaltung im Vektorraum $\mathbb{R}^n$ der Spaltenvektoren, soweit nichts anderes gesagt wird, mit dem **Euklidischen Innenprodukt** (englisch: *Euclidean inner product*)

$$x^\mathsf{T} y = \sum_{i=1}^{n} x_i y_i \quad \text{für } x, y, \in \mathbb{R}^n,$$

der daraus erzeugten **Euklidischen Norm** (auch: 2-**Norm**)

$$\|x\|_2 := \sqrt{x^\mathsf{T} x} \quad \text{für } x \in \mathbb{R}^n$$

und dem wiederum daraus erzeugten Abstand (der **Metrik**, (englisch: *metric*))

$$\|x - y\|_2 \quad \text{für } x, y, \in \mathbb{R}^n.$$

Wie für jedes Innenprodukt gilt die **Cauchy-Schwarz-Ungleichung** (englisch: *Cauchy-Schwarz inequality*)

$$x^\mathsf{T} y \leqslant |x^\mathsf{T} y| \leqslant \|x\|_2 \, \|y\|_2 \quad \text{für alle } x, y, \in \mathbb{R}^n,$$

wobei die Gleichheit $|x^\mathsf{T} y| = \|x\|_2 \, \|y\|_2$ genau dann gilt, wenn x und y linear abhängig sind.

Quizfrage: Wie sehen das Euklidische Innenprodukt und die daraus erzeugte Euklidische Norm für **Zeilenvektoren** (englisch: *row vectors*) im $\mathbb{R}_n$ aus?

Im Folgenden gelten alle Aussagen für $\mathbb{R}^n$ sinngemäß auch für $\mathbb{R}_n$.

Wir benötigen außerdem die 1-**Norm** und die ∞-**Norm** von Vektoren im $\mathbb{R}^n$, definiert durch

$$\|x\|_1 := \sum_{i=1}^{n} |x_i| \quad \text{und} \quad \|x\|_\infty := \max_{i=1,\dots,n} |x_i|.$$

Beachte: In den endlich-dimensionalen Vektorräumen $\mathbb{R}^n$ sind alle Normen äquivalent. Es ist also unerheblich, in welcher Norm wie z. B. Konvergenz oder Fehler messen. Die Wahl erfolgt häufig danach, wie es am einfachsten geht. Für die 1-, 2- und ∞-Normen im $\mathbb{R}^n$ gelten die folgenden Abschätzungen, die die Äquivalenz dieser Normen untereinander bestätigen:

$$\|x\|_2 \leqslant \|x\|_1 \leqslant \sqrt{n} \, \|x\|_2, \tag{2.1a}$$

$$\|x\|_\infty \leqslant \|x\|_2 \leqslant \sqrt{n} \, \|x\|_\infty, \tag{2.1b}$$

$$\|x\|_\infty \leqslant \|x\|_1 \leqslant n \, \|x\|_\infty. \tag{2.1c}$$

Quizfrage: Für welche Vektoren im $\mathbb{R}^n$ sind die Abschätzungen im (2.1) jeweils scharf?

Matrixnormen

Matrizen kennzeichnen wir i. d. R. mit Großbuchstaben und ihre Einträge mit kleinen Buchstaben, beispielsweise hat eine Matrix $A \in \mathbb{R}^{m \times n}$ die Einträge a_{ij} mit $i = 1, \dots, m$ und $j = 1, \dots, n$. Manchmal

schreiben wir statt A dann auch (a_{ij}), um die gesamte Matrix zu addressieren. Die **Einheitsmatrix** bezeichnen wir mit Id. Wenn wir die Dimension betonen wollen, schreiben wir auch Id_n für die Einheitsmatrix der Größe $n \times n$.

Auch im Vektorraum der Matrizen $A \in \mathbb{R}^{m\times n}$ bzw. der durch sie repräsentierten linearen Abbildung $\mathbb{R}^n \ni x \mapsto Ax \in \mathbb{R}^m$ lassen sich Normen definieren. Wir betrachten hier vorwiegend die durch die Vektornormen $\|\cdot\|_2$, $\|\cdot\|_1$ und $\|\cdot\|_\infty$ induzierten Operatornormen:

Definition 2.2 (Operatornorm einer Matrix). *Es sei $A \in \mathbb{R}^{m\times n}$. Weiter seien $\|\cdot\|_X$ bzw. $\|\cdot\|_Y$ die in den Vektorräumen $\mathbb{R}^n$ und $\mathbb{R}^m$ verwendeten Vektornormen. Die Größe*

$$\|A\|_{X\to Y} := \sup_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Ax\|_Y}{\|x\|_X} \tag{2.2}$$

*heißt die (durch die Normen $\|\cdot\|_X$ im $\mathbb{R}^n$ und $\|\cdot\|_Y$ im $\mathbb{R}^m$ induzierte) **Operatornorm** (englisch: operator norm) von A.*

Lemma 2.3 (Eigenschaften der Operatornorm). *Es seien $A \in \mathbb{R}^{m\times n}$ und $B \in \mathbb{R}^{\ell\times m}$. In den Räumen $\mathbb{R}^n$, $\mathbb{R}^m$ und $\mathbb{R}^\ell$ verwenden wir die Normen $\|\cdot\|_X$, $\|\cdot\|_Y$ bzw. $\|\cdot\|_Z$. Für die Operatornorm (2.2) gelten folgende Eigenschaften:*

(*i*) *Die Operatornorm ist eine Norm auf $\mathbb{R}^{m\times n}$.*

(*ii*) $\|Ax\|_Y \leqslant \|A\|_{X\to Y}\,\|x\|_X$ *für alle $x \in \mathbb{R}^n$.*

(*iii*) *Es gilt*

$$\|A\|_{X\to Y} = \sup_{\substack{x\in\mathbb{R}^n\\ \|x\|_X=1}} \|Ax\|_Y = \max_{\substack{x\in\mathbb{R}^n\\ \|x\|_X=1}} \|Ax\|_Y.$$

(*iv*) *Es gilt*

$$\begin{aligned}\|A\|_{X\to Y} &= \inf\{c \geqslant 0 \mid \|Ax\|_Y \leqslant c\,\|x\|_X \text{ für alle } x \in \mathbb{R}^n\}\\ &= \min\{c \geqslant 0 \mid \|Ax\|_Y \leqslant c\,\|x\|_X \text{ für alle } x \in \mathbb{R}^n\}.\end{aligned}$$

(*v*) $\|BA\|_{X\to Z} \leqslant \|B\|_{Y\to Z}\,\|A\|_{X\to Y}$ (***Submultiplikativität***).

Beachte: Aussage (*iv*) bedeutet: Kennt man eine Zahl $c \geqslant 0$, mit der $\|Ax\|_Y \leqslant c\,\|x\|_X$ für alle $x \in \mathbb{R}^n$ gilt, so besteht die Abschätzung $\|A\|_{X\to Y} \leqslant c$.

Beweis. Findet sich als Übungsaufgabe 1 auf Übungsblatt 1. □

Lemma 2.4 (Operatornormen). *Für Matrizen $A \in \mathbb{R}^{m\times n}$ und ihre Operatornormen, induziert durch die Verwendung jeweils der 1-Norm, der 2-Norm bzw. der ∞-Norm im Definitionsraum $\mathbb{R}^n$ wie auch im*

Zielraum $\mathbb{R}^m$ gelten die folgenden Aussagen:

$$\|A\|_{1\to 1} = \max_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Ax\|_1}{\|x\|_1} = \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}| \quad (\textbf{\textit{Spaltensummennorm}}), \tag{2.3a}$$

$$\|A\|_{2\to 2} = \max_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Ax\|_2}{\|x\|_2} = \textit{Wurzel des maximalen Eigenwerts von } A^{\mathsf{T}}A \quad (\textbf{\textit{Spektralnorm}}), \tag{2.3b}$$

$$\|A\|_{\infty\to\infty} = \max_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Ax\|_\infty}{\|x\|_\infty} = \max_{i=1,\dots,m} \sum_{j=1}^{n} |a_{ij}| \quad (\textbf{\textit{Zeilensummennorm}}). \tag{2.3c}$$

Beweis. Zum Beweis von (2.3a) schätzen wir ab:

$$\begin{aligned}
\|Ax\|_1 &= \sum_{i=1}^{m} \Big| \sum_{j=1}^{n} a_{ij}x_j \Big| \\
&\leqslant \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|\,|x_j| && \text{(Dreiecksungleichung)} \\
&= \sum_{j=1}^{n} |x_j| \sum_{i=1}^{m} |a_{ij}| \\
&\leqslant \sum_{j=1}^{n} |x_j| \max_{k=1,\dots,n} \sum_{i=1}^{m} |a_{ik}| \\
&= \|x\|_1 \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}| \quad \text{für alle } x \in \mathbb{R}^n.
\end{aligned}$$

Es gilt also (siehe Text nach Lemma 2.3) $\|A\|_{1\to 1} \leqslant \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}|$. Um die Gleichheit zu zeigen, sei j^* einer derjenigen Indizes, für die

$$\sum_{i=1}^{m} |a_{ij^*}| = \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}|$$

gilt. Wir wählen $x^* := e_{j^*}$ als j^*-ten Einheitsvektor in $\mathbb{R}^n$. Dann gilt $\|x^*\|_1 = 1$ und

$$\|Ax^*\|_1 = \|Ae_{j^*}\|_1 = \left\| \begin{pmatrix} a_{1j^*} \\ \vdots \\ a_{mj^*} \end{pmatrix} \right\|_1 = \sum_{i=1}^{m} |a_{ij^*}| = \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}| = \|x^*\|_1 \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}|,$$

also $\|A\|_{1\to 1} \geqslant \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}|$. Zusammen mit der obigen umgekehrten Abschätzung ist (2.3a) gezeigt.

Zum Beweis von (2.3c) schätzen wir wie folgt ab:

$$\begin{aligned}
\|Ax\|_\infty &= \max_{i=1,\dots,m}\left|\sum_{j=1}^n a_{ij}x_j\right| \\
&\leqslant \max_{i=1,\dots,m}\sum_{j=1}^n |a_{ij}|\,|x_j| && \text{(Dreiecksungleichung)} \\
&\leqslant \max_{j=1,\dots,n}|x_j| \max_{i=1,\dots,m}\sum_{j=1}^n |a_{ij}| \\
&= \|x\|_\infty \max_{i=1,\dots,m}\sum_{j=1}^n |a_{ij}| \quad \text{für alle } x\in\mathbb{R}^n.
\end{aligned}$$

Es gilt also $\|A\|_{\infty\to\infty} \leqslant \max_{i=1,\dots,m}\sum_{j=1}^n |a_{ij}|$. Um die Gleichheit zu zeigen, sei i^* einer derjenigen Indizes, für die

$$\sum_{j=1}^n |a_{i^*j}| = \max_{i=1,\dots,m}\sum_{j=1}^n |a_{ij}|$$

gilt. Wir wählen $x^* := (\operatorname{sgn} a_{i^*1},\dots,\operatorname{sgn} a_{i^*n})^\mathsf{T}$.[6] Dann gilt $\|x^*\|_\infty = 1$ und

$$\|Ax^*\|_\infty = \max_{i=1,\dots,m}\left|\sum_{j=1}^n a_{ij}x_j^*\right| \geqslant \left|\sum_{j=1}^n a_{i^*j}x_j^*\right| = \left|\sum_{j=1}^n |a_{i^*j}|\right| = \sum_{j=1}^n |a_{i^*j}| = \max_{i=1,\dots,m}\sum_{j=1}^n |a_{ij}|,$$

also $\|A\|_{\infty\to\infty} \geqslant \max_{i=1,\dots,m}\sum_{j=1}^n |a_{ij}|$. Zusammen mit der obigen umgekehrten Abschätzung ist (2.3c) gezeigt.

Den Beweis von (2.3b) liefern wir in Abschnitt 4 nach. □

Für die Operatornormen von Spaltenvektoren $y\in\mathbb{R}^m \cong \mathbb{R}^{m\times 1}$ und Zeilenvektoren $y\in\mathbb{R}_n \cong \mathbb{R}^{1\times n}$ gilt Folgendes:

Folgerung 2.5 (Operatornormen von Zeilen- und Spaltenvektoren)**.**

(*i*) *Für Spaltenvektoren $y\in\mathbb{R}^m$, die wir auch als Matrix $y\in\mathbb{R}^{m\times 1}$ interpretieren, gilt:*

$$\|y\|_{1\to 1} = \|y\|_1, \quad \|y\|_{2\to 2} = \|y\|_2, \quad \|y\|_{\infty\to\infty} = \|y\|_\infty \tag{2.4}$$

(*ii*) *Für Zeilenvektoren $y\in\mathbb{R}_n$, die wir auch als Matrix $y\in\mathbb{R}^{1\times n}$ interpretieren, gilt:*

$$\|y\|_{1\to 1} = \|y\|_\infty, \quad \|y\|_{2\to 2} = \|y\|_2, \quad \|y\|_{\infty\to\infty} = \|y\|_1 \tag{2.5}$$

Beweis. Aussage (*i*): Für $y\in\mathbb{R}^m \cong \mathbb{R}^{m\times 1}$ gilt nach (2.3a) bzw. (2.3c)

$$\|y\|_{1\to 1} = \sum_{i=1}^m |y_{i1}| = \|y\|_1 \quad \text{und} \quad \|y\|_{\infty\to\infty} = \max_{i=1,\dots,m}|y_{i1}| = \|y\|_\infty.$$

[6] Das **Signum** einer reellen Zahl x ist definiert als $\operatorname{sgn} x = 1$ für $x > 0$, $\operatorname{sgn} x = -1$ für $x < 0$ und $\operatorname{sgn} x = 0$ für $x = 0$.

Da wir (2.3b) noch nicht bewiesen haben, untersuchen wir die 2-Norm direkt:

$$\|y\|_{2\to 2} = \max_{\substack{\alpha\in\mathbb{R}\\|\alpha|=1}} \|\alpha y\|_2 = \|y\|_2.$$

Aussage *(ii)*: Für $y \in \mathbb{R}^n \cong \mathbb{R}^{1\times n}$ gilt nach (2.3a) bzw. (2.3c)

$$\|y\|_{1\to 1} = \max_{j=1,\dots,n} |y_{1j}| = \|y\|_\infty \quad \text{und} \quad \|y\|_{\infty\to\infty} = \sum_{j=1}^{n} |y_{1j}| = \|y\|_1.$$

Die 2-Norm untersuchen wir wieder direkt:

$$\|y\|_{2\to 2} = \max_{\substack{x\in\mathbb{R}^n\\\|x\|_2=1}} |y\,x| \leqslant \max_{\substack{x\in\mathbb{R}^n\\\|x\|_2=1}} \|y\|_2\,\|x\|_2.$$

Im Fall $y = 0$ sind ohnehin beide Seiten gleich Null. Andernfalls erreichen wir durch die Wahl von $x = y/\|y\|_2$ die Gleichheit (Cauchy-Schwarz-Ungleichung). □

§ 2.2 Differenzierbarkeit und der Satz von Taylor

Zwei wesentliche Hilfsmittel zur Analyse von Aufgaben und Algorithmen sind die Kettenregel und der Satz von Taylor, die wir hier kurz wiederholen:

Definition 2.6 (Differenzierbarkeit, siehe z. B. Heuser, 2002, Definition 164). *Es sei $F\colon \mathbb{R}^n \to \mathbb{R}^m$ eine Funktion. F heißt **differenzierbar** (englisch: differentiable) an der Stelle $x_0 \in \mathbb{R}^n$, wenn es eine Matrix $A \in \mathbb{R}^{m\times n}$ gibt, sodass für die durch*

$$r(x_0;\Delta x) := F(x_0+\Delta x) - F(x_0) - A\,\Delta x$$

*definierte **Restgliedfunktion** (englisch: remainder function) gilt:*[7]

$$\lim_{\Delta x\to 0} \frac{r(x_0;\Delta x)}{\|\Delta x\|} = 0. \tag{2.6}$$

In diesem Fall heißt die durch die Definition eindeutig bestimmte Matrix A die **Ableitung** (englisch: *derivative*) von F an der Stelle x_0 und wird mit $F'(x_0)$ bezeichnet. Sie stimmt überein mit der **Jacobimatrix** (der Matrix aus allen partiellen Ableitungen erster Ordnung, englisch: *Jacobian*) von F an der Stelle x_0:

$$J_F(x_0) := \begin{bmatrix} \frac{\partial F_1(x_0)}{\partial x_1} & \cdots & \frac{\partial F_1(x_0)}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial F_m(x_0)}{\partial x_1} & \cdots & \frac{\partial F_m(x_0)}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m\times n}. \tag{2.7}$$

[7] Da diese Aussage wegen der Äquivalenz der Normen auf $\mathbb{R}^n$ für jede Vektornorm $\|\cdot\|$ gilt, wird die Norm hier nicht näher spezifiziert.

Partielle Ableitungen zweiter Ordnung werden, sofern sie existieren, nach dem Muster

$$\frac{\partial^2 F_j(x_0)}{\partial x_2\,\partial x_1} := \frac{\partial}{\partial x_2}\frac{\partial F_j}{\partial x_1}(x_0)$$

gebildet. Analog gibt es auch Ableitungen höherer Ordnung. **Quizfrage:** Wieviele verschiedene partielle Ableitungen der Ordnung $k \in \mathbb{N}_0$ gibt es?

Definition 2.7 (C^k-Funktion). *Es sei $k \in \mathbb{N}_0$ und $U \subseteq \mathbb{R}^n$ offen. Eine Funktion $F\colon \mathbb{R}^n \to \mathbb{R}^m$ heißt **k-mal stetig partiell differenzierbar** (englisch: k times continuously partially differentiable) auf U oder eine **C^k-Funktion** auf U (kurz: $C^k(U,\mathbb{R}^m)$), wenn F sowie alle partiellen Ableitungen bis einschließlich zur Ordnung k existieren und auf U stetig sind.*

Beachte: Eine Funktion $F\colon \mathbb{R}^n \to \mathbb{R}^m$, die auf der offenen Menge $U \subseteq \mathbb{R}^n$ eine C^1-Funktion ist, ist in jedem Punkt von U auch differenzierbar, erfüllt also die Restgliedbedingung (2.6) für jedes feste $x_0 \in U$. Kurz: Eine einmal stetig partiell differenzierbare Funktion ist differenzierbar.

Satz 2.8 (Kettenregel).
Es seien $F\colon \mathbb{R}^m \to \mathbb{R}^p$ und $G\colon \mathbb{R}^n \to \mathbb{R}^m$ zwei Funktionen, und es sei G an der Stelle $x_0 \in \mathbb{R}^n$ differenzierbar sowie F an der Stelle $G(x_0)$ differenzierbar. Dann ist auch die Komposition $F \circ G$ an der Stelle x_0 differenzierbar, und es gilt

$$(F\circ G)'(x_0) = F'(G(x_0))\,G'(x_0)$$

oder ausgeschrieben

$$\begin{bmatrix} \frac{\partial (F\circ G)_1(x_0)}{\partial x_1} & \cdots & \frac{\partial (F\circ G)_1(x_0)}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial (F\circ G)_p(x_0)}{\partial x_1} & \cdots & \frac{\partial (F\circ G)_p(x_0)}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial F_1(G(x_0))}{\partial x_1} & \cdots & \frac{\partial F_1(G(x_0))}{\partial x_m} \\ \vdots & & \vdots \\ \frac{\partial F_p(G(x_0))}{\partial x_1} & \cdots & \frac{\partial F_p(G(x_0))}{\partial x_m} \end{bmatrix} \begin{bmatrix} \frac{\partial G_1(x_0)}{\partial x_1} & \cdots & \frac{\partial G_1(x_0)}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial G_m(x_0)}{\partial x_1} & \cdots & \frac{\partial G_m(x_0)}{\partial x_n} \end{bmatrix}.$$

Die Ableitung (Jacobimatrix) der zusammengesetzten Funktion $F \circ G$ ergibt sich also aus dem Produkt der Jacobimatrizen $F'(G(x_0))$ und $G'(x_0)$.

Wir geben nun zwei Versionen des Satzes von Taylor für *reellwertige* Funktionen an, die sich in der Darstellung des Restglieds unterscheiden. Analog zur Schreibweise für Intervalle

$$[a,b] := \big\{x \in \mathbb{R} \,\big|\, a \leqslant x \text{ und } x \leqslant b\big\} \tag{2.8}$$

ist es günstig, eine ähnliche Schreibweise auch für die **Verbindungsstrecke** (englisch: *segment*) zwischen Punkten a, b im $\mathbb{R}^n$ einzuführen, also

$$\langle a\,,b\rangle := \big\{(1-t)\,a + t\,b \,\big|\, t \in [0,1]\big\} \tag{2.9}$$

für $a, b \in \mathbb{R}^n$. **Quizfrage:** Gilt für reelle Zahlen a und b immer $[a,b] = \langle a\,,b\rangle$?

Definition 2.9 (Konvexe Menge). *Eine Menge $A \subseteq \mathbb{R}^n$ heißt **konvex** (englisch: convex), wenn für $x, y \in A$ immer auch die gesamte Verbindungsstrecke in A liegt, also $\langle x\,,y\rangle \subseteq A$.*

Satz 2.10 (Taylor mit Restglied in Zwischenwertform, siehe z. B. Heuser, 2002, Satz 168.1)**.**
Es sei $U \subseteq \mathbb{R}^n$ offen, $k \in \{1, 2\}$ und $f : U \to \mathbb{R}$ eine C^k-Funktion auf U, kurz: $C^k(U, \mathbb{R})$. Die Punkte x_0 und $x_0 + d$ seien so gewählt, dass $\langle x_0 , x_0 + d\rangle$ in U liegt. Dann existiert eine Zahl $\xi \in [0, 1]$, sodass gilt:

$$\begin{aligned} &\text{im Fall } k = 1: \quad f(x_0 + d) = f(x_0) + f'(x_0 + \xi\, d)\, d, \\ &\text{im Fall } k = 2: \quad f(x_0 + d) = f(x_0) + f'(x_0)\, d + \frac{1}{2} d^\mathsf{T} f''(x_0 + \xi\, d)\, d. \end{aligned}$$

Die erste Aussage ist gerade der **Mittelwertsatz** (englisch: *mean value theorem*). Weiter sind $f'(x_0)$ und $f''(x_0)$ die Ableitungen erster bzw. zweiter Ordnung von f an der Stelle x_0, die folgende Gestalt besitzen:

$$f'(x_0) = \begin{bmatrix} \frac{\partial f(x_0)}{\partial x_1} & \cdots & \frac{\partial f(x_0)}{\partial x_n} \end{bmatrix} \quad \text{(Jacobimatrix)}$$

$$f''(x_0) = \begin{bmatrix} \frac{\partial^2 f(x_0)}{\partial x_1\, \partial x_1} & \cdots & \frac{\partial^2 f(x_0)}{\partial x_1\, \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(x_0)}{\partial x_n\, \partial x_1} & \cdots & \frac{\partial^2 f(x_0)}{\partial x_n\, \partial x_n} \end{bmatrix} \quad \text{(\textbf{Hessematrix}, englisch: \textit{Hessian})}.$$

Als Resultat des Satzes von Schwarz (Heuser, 2002, Satz 162.2) ist die Hessematrix $f''(x_0)$ dabei symmetrisch, also die Reihenfolge der partiellen Differentiationen unerheblich.

Quizfrage: Gilt der Satz von Taylor 2.10 auch für vektorwertige C^k-Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$?

Satz 2.11 (Taylor mit integralem Restglied, siehe z. B. Heuser, 2002, Satz 168.1)**.**
Unter denselben Voraussetzungen wie in Satz 2.10 gilt

$$\begin{aligned} &\text{im Fall } k = 1: \quad f(x_0 + d) = f(x_0) + \left[\int_0^1 f'(x_0 + t\, d)\, \mathrm{d}t\right] d \\ &\qquad\qquad = f(x_0) + \int_0^1 f'(x_0 + t\, d)\, d\, \mathrm{d}t, \\ &\text{im Fall } k = 2: \quad f(x_0 + d) = f(x_0) + f'(x_0)\, d + \frac{1}{2} d^\mathsf{T} \left[\int_0^1 (1 - t)\, f''(x_0 + t\, d)\, \mathrm{d}t\right] d \\ &\qquad\qquad = f(x_0) + f'(x_0)\, d + \frac{1}{2} \int_0^1 (1 - t)\, d^\mathsf{T} f''(x_0 + t\, d)\, d\, \mathrm{d}t. \end{aligned}$$

Die erste Aussage folgt direkt aus dem **Hauptsatz der Differential- und Integralrechnung**.[8]

Beachte: Unter dem jeweils ersten Integral steht eine vektorwertige bzw. matrixwertige Funktion, die einfach komponentenweise integriert wird. Im jeweils zweiten Integral wurden die konstanten (von t unabhängigen) Inkremente d in das Integral hineingezogen, wodurch der Integrand reellwertig wird.

Quizfrage: In dieser Form gilt Satz 2.11 auch für vektorwertige C^k-Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$. Wie genau sollte der Satz dann formuliert werden?

[8] «Eine differenzierbare Funktion ist das Integral ihrer Ableitung.»

Definition 2.12 (Lipschitz-Stetigkeit). *Eine Funktion $F\colon \mathbb{R}^n \to \mathbb{R}^m$ heißt **Lipschitz-stetig** (englisch: Lipschitz continuous) auf der Menge $A \subseteq \mathbb{R}^n$, falls es eine Zahl $L \geqslant 0$ (genannt: **Lipschitz-Konstante**, englisch: Lipschitz constant) gibt mit der Eigenschaft*[9]

$$\|F(x) - F(y)\|_2 \leqslant L\,\|x - y\|_2 \quad \text{für alle } x, y \in A. \tag{2.10}$$

*In diesem Fall heißt F dann auch **L-Lipschitz-stetig** auf A.*

Eine einfache aber wichtige Konsequenz aus dem Satz von Taylor 2.10 ist die Tatsache, dass C^1-Funktionen genau dann Lipschitz-stetig sind, wenn ihre Ableitung beschränkt ist, genauer:

Folgerung 2.13 (Lipschitz-Stetigkeit von C^1-Funktionen). *Es sei $F\colon \mathbb{R}^n \to \mathbb{R}$ eine C^1-Funktion auf der offenen, konvexen Menge $U \subseteq \mathbb{R}^n$. Dann sind folgende Aussagen äquivalent:*

(*i*) *F ist Lipschitz-stetig auf U.*

(*ii*) *Die Ableitung F' ist auf U beschränkt, also $\sup\left\{\|F'(x)\|_{2\to 2} \,\middle|\, x \in U\right\} < \infty$.*

Falls die Aussagen wahr sind, dann ist $L^* := \sup\left\{\|F'(x)\|_{2\to 2} \,\middle|\, x \in U\right\} < \infty$ die kleinstmögliche Lipschitz-Konstante in (2.10).

Beachte: $\|F'(x)\|_{2\to 2}$ ist die durch die 2-Norm induzierte Operatornorm der Jacobimatrix von F an der Stelle x, vgl. auch (2.3b).

Beweis. Wir nehmen zunächst Aussage (*i*) an. Es seien $x \in U$ und $d \in \mathbb{R}^n$ fest gewählt sowie $\alpha > 0$. Aus der Dreiecksungleichung folgt

$$\begin{aligned}\|F'(x)\,(\alpha d)\|_2 &= \|F'(x)\,(\alpha d) \pm [F(x+\alpha d) - F(x)]\|_2 \\ &\leqslant \|F(x+\alpha d) - F(x) - F'(x)\,(\alpha d)\|_2 + \|F(x+\alpha d) - F(x)\|_2.\end{aligned}$$

Es gilt also auch

$$\|F'(x)\,d\|_2 = \frac{\|F'(x)\,(\alpha d)\|_2}{\alpha} \leqslant \frac{\|F(x+\alpha d) - F(x) - F'(x)\,(\alpha d)\|_2}{\alpha} + \frac{\|F(x+\alpha d) - F(x)\|_2}{\alpha}.$$

Für hinreichend kleine $\alpha > 0$ liegt $x + \alpha d \in U$, und wir können für den ersten Term auf der rechten Seite die Voraussetzung der Differenzierbarkeit und für den zweiten Term die Voraussetzung der L-Lipschitz-Stetigkeit von F auf U mit einer Lipschitz-Konstanten L nutzen. Im Grenzübergang $\alpha \searrow 0$ folgt:

$$\|F'(x)\,d\|_2 \leqslant 0 + L\,\|d\|_2.$$

Da $d \in \mathbb{R}^n$ beliebig war, folgt aus Lemma 2.3 jetzt $\|F'(x)\|_2 \leqslant L$.

[9] Auf der linken und rechten Seite von (2.10) dürften wegen der Äquivalenz der Normen auch andere Vektornormen im $\mathbb{R}^n$ bzw. $\mathbb{R}^m$ stehen. Die konkrete Wahl beeinflusst nicht die Frage, *ob* eine Funktion Lipschitz-stetig ist, sondern nur, welche Lipschitz-Konstanten gültig sind. Die Wahl der 2-Norm ist hier günstig wegen der Formulierung der Folgerung 2.13.

Wir nehmen nun Aussage (ii) an. Es seien $x, y \in U$. Aufgrund der Konvexität von U liegt auch $\langle x, y\rangle$ in U. Aus dem Satz von Taylor 2.11 folgt

$$F(y) - F(x) = \int_0^1 F'(x + t\,(y - x))\,\mathrm{d}t\,(y - x),$$

also (**Quizfrage:** Warum gelten jeweils die Abschätzungen?)

$$\begin{aligned}\|F(y) - F(x)\|_2 &= \left\| \int_0^1 F'(x + t\,(y - x))\,(y - x)\,\mathrm{d}t \right\| \\ &\leqslant \int_0^1 \left\| F'(x + t\,(y - x))\,(y - x) \right\| \mathrm{d}t \\ &\leqslant \int_0^1 \left\| F'(x + t\,(y - x)) \right\|_2 \|(y - x)\|_2\,\mathrm{d}t.\end{aligned}$$

Der Integrand $\|F'(x + t\,(y - x))\|_2$ ist aber durch $\sup\{\|F'(x)\|_2 \mid x \in U\}$ beschränkt, da $x + t\,(y - x)$ auf der Verbindungsstrecke $\langle x, y\rangle$ und damit in U liegt. Damit ist $L^* := \sup\{\|F'(x)\|_2 \mid x \in U\} < \infty$ eine gültige Lipschitz-Konstante für F auf U, also gilt Aussage (i).

Wir müssen noch zeigen, dass L^* die kleinstmögliche Lipschitz-Konstante ist. Aus dem Beweisteil Aussage (i) $\Rightarrow$ Aussage (ii) erhalten wir aber

$$F \text{ ist } L\text{-Lipschitz} \quad\Rightarrow\quad \|F'(x)\|_2 \leqslant L \text{ für alle } x \in U \quad\Rightarrow\quad L^* := \sup\{\|F'(x)\|_2 \mid x \in U\} \leqslant L.$$

□

§ 2.3 Landau-Notation

Wir geben jetzt noch die (Teile der) Definition der **Landau-Notation** (englisch: *Landau notation*) an, mit deren Hilfe man asymptotisches Verhalten, z. B. von Fehlern, charakterisieren kann.

Definition 2.14 (Landau-Notation für Funktionen).
Es sei $g\colon \mathbb{R}^n \setminus \{0\} \to \mathbb{R}_{\geqslant 0}$ eine Funktion.

(i) *Es sei $m \in \mathbb{N}$. Wir bezeichnen die Menge von Funktionen*

$$O(g) := \left\{ F\colon \mathbb{R}^n \setminus \{0\} \to \mathbb{R}^m \;\middle|\; \begin{array}{l} \text{es existiert ein } C \geqslant 0 \text{ und ein } \varepsilon > 0, \text{ sodass gilt:} \\ \|F(x)\| \leqslant C\,g(x) \text{ für alle } x \text{ mit } 0 < \|x\| < \varepsilon \end{array} \right\}$$

*als «**Groß-O von g bei Null**» (englisch:* big-O of g near zero*).*

*Für $F \in O(g)$ sagt man auch, g sei eine **asymptotisch obere Schranke** für F bei Null.*

(ii) *Es sei $m \in \mathbb{N}$. Wir bezeichnen die Menge von Funktionen*

$$o(g) := \left\{ F\colon \mathbb{R}^n \setminus \{0\} \to \mathbb{R}^m \;\middle|\; \begin{array}{l} \text{für alle } c > 0 \text{ existiert ein } \varepsilon > 0, \text{ sodass gilt:} \\ \|F(x)\| \leqslant c\,g(x) \text{ für alle } x \text{ mit } 0 < \|x\| < \varepsilon \end{array} \right\}$$

*als «**Klein-o von g bei Null**» (englisch:* little-o of g near zero*).*

*Für $F \in o(g)$ sagt man auch, F sei gegenüber g **asymptotisch vernachlässigbar** bei Null.*

Beachte: Wir lassen in der obigen Definition offen, welche Norm im $\mathbb{R}^m$ wir verwenden, um $\|F(x)\|$ zu messen, weil es wegen der Äquivalenz der Normen unerheblich ist.

Quizfrage: Wie kann man die Restgliedbedingung (2.6) in der Landau-Notation ausdrücken?

Quizfrage: Welche Beziehung besteht zwischen den Funktionenklassen $o(\|\Delta x\|)$ und $O(\|\Delta x\|^2)$?

Beispiel 2.15 (Beispiele für Landau-Notation für Funktionen)**.**

(*i*) *Die Funktion $f : \mathbb{R} \to \mathbb{R}$, definiert durch $f(x) = \sin(x)$, gehört zu $O(g)$ für $g(x) = |x|$.*
Kurz schreibt man auch: $\sin(x) \in O(|x|)$.

(*ii*) *Die Funktion $f : \mathbb{R} \to \mathbb{R}$, definiert durch $f(x) = x^2$, gehört zu $o(g)$ für $g(x) = |x|$.*
Kurz schreibt man auch: $x^2 \in o(|x|)$.

(*iii*) *Die Funktion $F : \mathbb{R}^2 \to \mathbb{R}^2$, definiert durch $F(x) = \begin{pmatrix} x_1 - x_2 \\ x_1^2 + x_2 \end{pmatrix}$, gehört zu $O(g)$ für $g(x) = \|x\|$.*
Kurz schreibt man auch: $F(x) \in O(\|x\|)$.

Auch für die Beschreibung der Asymptotik von Folgen ist die Landau-Notation hilfreich. Dann ist nicht der Fall «bei Null», sondern der Fall «bei Unendlich» interessant.

Definition 2.16 (Landau-Notation für Folgen)**.**
Es sei $\left(y^{(n)}\right)$ eine $\mathbb{R}_{\geqslant 0}$-wertige Folge.

(*i*) *Es sei $m \in \mathbb{N}$. Wir bezeichnen die Menge von Folgen*

$$O\left(y^{(n)}\right) := \left\{ \left(x^{(n)}\right) : \mathbb{N} \to \mathbb{R}^m \;\middle|\; \begin{array}{l} \text{es existiert ein } C \geqslant 0 \text{ und ein } n_0 \in \mathbb{N}, \text{ sodass gilt:} \\ \|x^{(n)}\| \leqslant C\, y^{(n)} \text{ für alle } n \geqslant n_0 \end{array} \right\}$$

*als «**Groß-O von $\left(y^{(n)}\right)$ bei Unendlich**».*

(*ii*) *Es sei $m \in \mathbb{N}$. Wir bezeichnen die Menge von Folgen*

$$o\left(y^{(n)}\right) := \left\{ \left(x^{(n)}\right) : \mathbb{N} \to \mathbb{R}^m \;\middle|\; \begin{array}{l} \text{für alle } c > 0 \text{ existiert ein } n_0 \in \mathbb{N}, \text{ sodass gilt:} \\ \|x^{(n)}\| \leqslant c\, y^{(n)} \text{ für alle } n \geqslant n_0 \end{array} \right\}$$

*als «**Klein-o von $\left(y^{(n)}\right)$ bei Unendlich**».*

Beachte: Auch in dieser Definition ist wieder irrelevant, welche Norm im $\mathbb{R}^m$ wir verwenden, um $\|x^{(n)}\|$ zu messen.

Beispiel 2.17 (Beispiele für Landau-Notation für Folgen)**.**

(*i*) *Die reellwertige Folge* $x^{(n)} = 1/n$ *gehört zu* $O(y^{(n)})$ *für* $y^{(n)} = 1$.
Kurz schreibt man auch: $1/n \in O(1)$.

(*ii*) *Die reellwertige Folge* $x^{(n)} = 1/n$ *gehört zu* $o(y^{(n)})$ *für* $y^{(n)} = 1$.
Kurz schreibt man auch: $1/n \in o(1)$.

(*iii*) *Die* $\mathbb{R}^2$*-wertige Folge* $x^{(n)} = \begin{pmatrix} 1/n \\ 1/n^2 \end{pmatrix}$ *gehört zu* $O(y^{(n)})$ *für* $y^{(n)} = 1/n$.
Kurz schreibt man auch: $x^{(n)} \in O(1/n)$.

Quizfrage: Was bedeuten diese Aussagen?

Ende der Woche 1

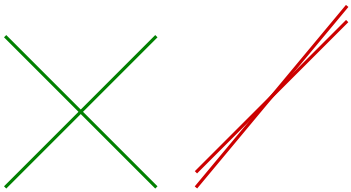
Kapitel 1 Sensitivätsanalyse und Kondition von Aufgaben

In diesem Kapitel beschäftigen wir uns zunächst mit Eigenschaften von Aufgaben, insbesondere mit dem Konzept der **Kondition** eines Problems. Der Grund dafür ist, dass jedes numerische Verfahren höchstens so gut sein kann, wie die Kondition der Aufgabe es zulässt.

§ 3 Sensitivitätsanalyse und Kondition

Zur Einstimmung zunächst eine anschauliche Situation.

Beispiel 3.1 (Schnitt zweier Geraden). *Wir bestimmen grafisch den Schnittpunkt zweier Geraden im $\mathbb{R}^2$. Im ersten Fall (links im Bild) ist das Ergebnis, also die Lage des Schnittpunkts, vergleichsweise unempfindlich gegenüber Störungen in den Daten (der Eingabe) der Aufgabe.*[1] *Wir sprechen von einer **gut konditionierten Aufgabe**. Im zweiten Fall (rechts im Bild) ist dagegen das Ergebnis viel empfindlicher gegenüber Veränderungen in der Eingabe. Wir sprechen dann von einer **schlecht konditionierten Aufgabe**.*



Quizfrage: Welche weiteren Beispiele für Aufgaben, für die Sie eine gute oder schlechte Kondition vermuten, fallen Ihnen ein?

Die **Sensitivitätsanalyse** (englisch: *sensitivity analysis*) beschäftigt sich mit der Frage, wie stark das Ergebnis $y \in \mathbb{R}^m$ einer mathematischen Funktion

$$y = F(x)$$

von Änderungen bzw. Störungen in der Eingabe (den Daten) $x \in \mathbb{R}^n$ abhängt. Die Darstellung $y = F(x)$ bedeutet nicht, dass y notwendigerweise durch eine endliche Abfolge von Operationen explizit

[1] Die Störungen können etwa durch die begrenzte Zeichengenauigkeit auftreten.

berechenbar ist. Vielmehr kann $y = F(x)$ zum Beispiel auch dafür stehen, dass y die Lösung eines nichtlinearen Gleichungssystems ist, in das x als Datum eingeht. Wie man aber bereits an einfachen Beispielen wie $y = x^2$ sieht, wird die Antwort *lokal* sein, also abhängig davon, welchen Wert die Eingabe x selbst hat; vgl. Abbildung 3.1.

Wir setzen voraus, dass F am interessierenden Punkt x differenzierbar ist. Dann erhalten wir mit Hilfe der Definition der Differenzierbarkeit, dass

$$F(x + \Delta x) \in F(x) + F'(x)\,\Delta x + o(\|\Delta x\|) \tag{3.1}$$

gilt.[2] Bezeichnen wir mit

$$\Delta y := F(x + \Delta x) - F(x)$$

die durch die Änderung Δx der Eingangsdaten hervorgerufenen Änderungen im Resultat an der Stelle x, so können wir (3.1) schreiben als

$$\Delta y \in F'(x)\,\Delta x + o(\|\Delta x\|). \tag{3.2}$$

Das heißt also, dass die durch die Ableitung $F'(x)$ definierte lineare Abbildung $\Delta x \mapsto F'(x)\,\Delta x$ eine derart gute Näherung für Δy ist, dass der dabei gemachte Fehler asymptotisch vernachlässigbar gegenüber $\|\Delta x\|$ ist. Man schreibt statt (3.2) auch:

$$\Delta y \doteq F'(x)\,\Delta x \tag{3.3}$$

und sagt, Δy sei gleich $F'(x)\,\Delta x$ «bis auf Terme, die gegenüber $\|\Delta x\|$ vernachlässigbar sind» oder «bis auf Terme höherer Ordnung in Δx». Letztere Sprechweise ist dadurch begründet, dass unter der Voraussetzung, dass F eine C^2-Funktion in einer Umgebung von x ist, aus dem Satz von Taylor 2.11 folgt, dass statt (3.2) sogar gilt: $\Delta y \in F'(x)\,\Delta x + O(\|\Delta x\|^2)$. Der letzte Term ist dabei «von höherer Ordnung» (nämlich quadratisch) in $\|\Delta x\|$.

In der **differentiellen Sensitivitätsanalyse** (englisch: *differential sensitivity analysis*) vernachlässigt man tatsächlich die Restgliedterme und stützt sich allein auf die Ableitung $F'(x)$. Daraus wollen wir nun im Folgenden verschiedene Informationen gewinnen.

§ 3.1 Absolute Betrachtungsweise

Zunächst lautet (3.3) in Komponenten ausgeschrieben für eine Funktion $F\colon \mathbb{R}^n \to \mathbb{R}^m$:

$$\Delta y = \begin{pmatrix} \Delta y_1 \\ \vdots \\ \Delta y_m \end{pmatrix} \doteq F'(x)\,\Delta x = \begin{bmatrix} \frac{\partial F_1(x)}{\partial x_1} & \cdots & \frac{\partial F_1(x)}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial F_m(x)}{\partial x_1} & \cdots & \frac{\partial F_m(x)}{\partial x_n} \end{bmatrix} \begin{pmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{pmatrix}. \tag{3.4}$$

[2] **Beachte:** Auf der rechten Seite von (3.1) steht die Summe aus einer Funktion, nämlich $F(x) + F'(x)\,\Delta x$, und einer Menge von Funktionen $o(\|\Delta x\|)$. Das ist im Sinne der Minkowski-Summe zu verstehen, d. h., auf der rechten Seite steht die Menge aller Funktionen der Bauart $\Delta x \mapsto F(x) + F'(x)\,\Delta x + G(\Delta x)$, wobei $G(\Delta x)$ zu $o(\|\Delta x\|)$ gehört. Alternativ können wir (3.1) auch lesen als: $F(x + \Delta x) - F(x) - F'(x)\,\Delta x \in o(\|\Delta x\|)$.

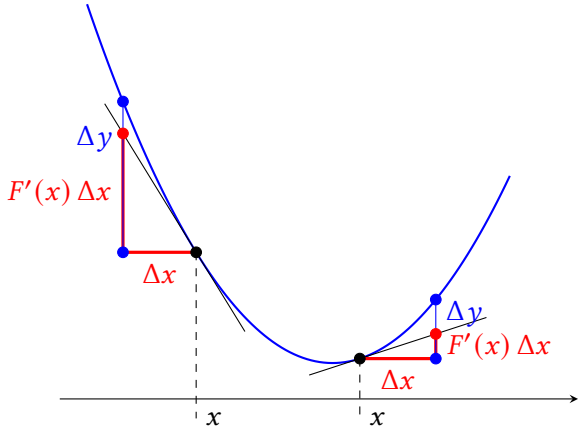


Abbildung 3.1: Wie stark das Ergebnis $y = F(x) := x^2$ von Änderungen Δx in der Eingabe x abhängt, ist i. d. R. abhängig vom Wert von x abhängig.

Der Eintrag

$$K_{ij}(x) := \frac{\partial F_i(x)}{\partial x_j} \tag{3.5}$$

in Zeile i und Spalte j der Jacobimatrix gibt also den Faktor an, mit dem sich Änderungen Δx_j in der j-ten Komponente der Eingabe x auswirken auf die i-te Komponente des Ergebnisses.

Definition 3.2 (Absolute partielle Konditionszahlen). *Die Zahlen $K_{ij}(x)$ heißen die **absoluten partiellen Konditionszahlen** (englisch:* absolute partial condition numbers*) der Funktion F an der Stelle x.*

Wollen wir Störungen in allen Komponenten der Eingabe gemeinsam betrachten, aber uns im Ergebnis auf die i-te Komponente beschränken, also in (3.4) nur die i-Zeile

$$\Delta y_i \doteq \underbrace{\begin{bmatrix} \frac{\partial F_i(x)}{\partial x_1} & \cdots & \frac{\partial F_i(x)}{\partial x_n} \end{bmatrix}}_{F_i'(x)} \begin{pmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{pmatrix} = \sum_{j=1}^{n} \frac{\partial F_i(x)}{\partial x_j} \Delta x_j \tag{3.6}$$

herausgreifen, so können wir (3.6) mit Hilfe der Cauchy-Schwarz-Ungleichung abschätzen durch[3]

$$|\Delta y_i| \mathrel{\dot\leqslant} \|F_i'(x)\|_2 \, \|\Delta x\|_2. \tag{3.7}$$

Dabei ist $\|F_i'(x)\|_2$ die 2-Norm des Zeilenvektors bestehend aus den partiellen Ableitungen der i-ten Komponentenfunktion F_i von F, also die Norm der i-ten Zeile der Jacobimatrix $F'(x)$. Diese Abschätzung gilt gleichmäßig, unabhängig von der Richtung des Störungsvektors Δx, es handelt sich also um eine *Worst-Case*-Abschätzung. Aus der Cauchy-Schwarz-Ungleichung wissen wir auch, dass die Abschätzung genau dann mit Gleichheit angenommen wird, also $|\Delta y_i| \doteq \|F_i'(x)\|_2 \, \|\Delta x\|_2$ gilt, wenn die Störungsrichtung Δx und die Ableitung $F_i'(x)^\mathsf{T}$ linear abhängig sind, also einer dieser Vektoren ein Vielfaches des anderen ist.

[3] Auch hier bedeutet der Punkt auf dem Symbol $\dot\leqslant$, dass die Abschätzung gilt bis auf Terme, die gegenüber $\|\Delta x\|$ vernachlässigbar sind.

Möchte man umgekehrt die Auswirkung der Störung in nur einer Komponente x_j der Eingabe, aber gleichzeitig auf alle Komponenten des Ergebnisses betrachten, so setzen wir in (3.4) alle $\Delta x_k = 0$ bis auf den Index $k = j$ und erhalten

$$\Delta y \doteq \underbrace{\begin{bmatrix} \frac{\partial F_1(x)}{\partial x_j} \\ \vdots \\ \frac{\partial F_m(x)}{\partial x_j} \end{bmatrix}}_{\frac{\partial F(x)}{\partial x_j}} \Delta x_j \tag{3.8}$$

und daraus die Abschätzung

$$\|\Delta y\|_2 \dot{\leqslant} \left\| \frac{\partial F(x)}{\partial x_j} \right\|_2 |\Delta x_j|. \tag{3.9}$$

Wollen wir nun abschließend gleichzeitig Störungen aller Komponenten in der Eingabe und die Auswirkung auf alle Komponenten im Ergebnis betrachten, so können wir (3.4) mit Hilfe von Lemma 2.3 abschätzen als

$$\|\Delta y\|_2 \dot{\leqslant} \|F'(x)\|_{2\to 2} \, \|\Delta x\|_2. \tag{3.10}$$

Dabei ist $\|F'(x)\|_{2\to 2}$ die durch die 2-Norm induzierte Operatornorm (Definition 2.2) der Jacobimatrix $F'(x)$. Die Abschätzung (3.10) besagt also, dass die Größe (Norm) $\|\Delta y\|_2$ der Störung Δy im Ergebnis nicht größer sein kann als die Operatornorm der Jacobimatrix, multipliziert mit der Größe (Norm) $\|\Delta x\|_2$ der Störung Δx in der Eingabe. Diese Abschätzung gilt wieder gleichmäßig, unabhängig von der Richtung des Störungsvektors Δx, es handelt sich also um eine *Worst-Case*-Abschätzung. In Abschnitt 4 werden wir sehen, für welche Richtungen Δx diese Abschätzung scharf ist.

Definition 3.3 (Absolute Konditionszahl). *Es sei $F\colon \mathbb{R}^n \to \mathbb{R}^m$ an der Stelle x differenzierbar.*

(*i*) *Dann heißt die Operatornorm*

$$K(x) := \|F'(x)\|_{2\to 2} = \|(K_{ij}(x))\|_{2\to 2} \tag{3.11}$$

*die **absolute Konditionszahl** (englisch: absolute condition number) von F an der Stelle x. Man spricht auch von der absoluten Konditionszahl der Aufgabe, $y = F(x)$ auszuwerten.*

(*ii*) *Die Funktion F heißt an der Stelle x **schlecht konditioniert im absoluten Sinne**, wenn $\|F'(x)\|_{2\to 2}$ «sehr groß» ist, ansonsten **gut konditioniert im absoluten Sinne**.*

Beachte: Was «sehr groß» im Einzelfall bedeutet, wird von den Anforderungen der jeweiligen Aufgabe bestimmt und kann nicht allgemeingültig festgelegt werden. Man beachte dabei auch, dass sich beim Umskalieren der Eingabe oder des Resultats — z. B. durch Wechsel der physikalischen Einheit von m zu mm — die absolute Konditionszahl ändert.

Beispiel 3.4 (Absolute Konditionierung). *Bei einer gleichmäßig beschleunigten Bewegung mit der Beschleunigung a und Anfangsgeschwindigkeit v_0 legt ein Körper innerhalb der Zeit t die Strecke*

$$s(t, v_0, a) = v_0\, t + \frac{1}{2} a\, t^2$$

zurück. Wir betrachten die Strecke s als Funktion der Eingangsgrößen $x = (t, v_0, a)$. Bei einer Beschleunigung von $a = 5\,\mathrm{m\,s^{-2}}$, einer Anfangsgeschwindigkeit von $v_0 = 0\,\mathrm{m\,s^{-1}}$ und einer Zeit von $t = 10\,\mathrm{s}$ beispielsweise ist $s = 250\,\mathrm{m}$. Die Jacobimatrix an dieser Stelle ist

$$\left[\frac{\partial s}{\partial t}(t, v_0, a) \quad \frac{\partial s}{\partial v_0}(t, v_0, a) \quad \frac{\partial s}{\partial a}(t, v_0, a)\right] = \left[v_0 + a\,t \quad t \quad \tfrac{1}{2}t^2\right] = \left[50\,\mathrm{m\,s^{-1}} \quad 10\,\mathrm{s} \quad 50\,\mathrm{s^2}\right].$$

Die Daten $x = (t, v_0, a)$ stammen vielleicht aus einer Messung und sind Unsicherheiten unterworfen. Änderungen einer einzelnen der drei Größen der Größenordnung $\Delta t = 0.1\,\mathrm{s}$, $\Delta v_0 = 0.5\,\mathrm{m\,s^{-1}}$ bzw. $\Delta a = 0.2\,\mathrm{m\,s^{-2}}$ rufen beispielsweise folgende Änderungen im Ergebnis hervor:

$$\begin{aligned}
\Delta s &\doteq \frac{\partial s}{\partial t}(t, v_0, a)\,\Delta t &&= 50\,\mathrm{m\,s^{-1}} \cdot 0.1\,\mathrm{s} = 5\,\mathrm{m},\\
\Delta s &\doteq \frac{\partial s}{\partial v_0}(t, v_0, a)\,\Delta v_0 &&= 10\,\mathrm{s} \cdot 0.5\,\mathrm{m\,s^{-1}} = 5\,\mathrm{m},\\
\Delta s &\doteq \frac{\partial s}{\partial a}(t, v_0, a)\,\Delta a &&= 50\,\mathrm{s^2} \cdot 0.2\,\mathrm{m\,s^{-2}} = 10\,\mathrm{m}.
\end{aligned}$$

Quizfrage: *Wie ändern sich die Zahlenwerte der absoluten partiellen Konditionszahlen, wenn man die zurückgelegte Strecke s in* mm *misst statt wie oben in* m*?* ***Quizfrage:*** *Und was ändert sich, wenn man die Anfangsgeschwindigkeit v_0 in* $\mathrm{km\,h^{-1}}$ *misst statt wie oben in* $\mathrm{m\,s^{-1}}$*?* ***Quizfrage:*** *Ist die Frage nach der absoluten Konditionszahl (3.11) überhaupt physikalisch sinnvoll?*

Beachte: Im Fall von $F\colon \mathbb{R} \to \mathbb{R}$ gibt es nur eine einzige absolute partielle Konditionszahl $K_{11} = F'(x)$, siehe (3.5). Diese ist vorzeichenbehaftet, während die absolute Konditionszahl nach Definition 3.3 immer nicht-negativ ist. Im betrachteten Fall gilt $\|F'(x)\|_{2\to 2} = |F'(x)| = |K_{11}|$.

Bemerkung 3.5 (Alternative Definition der absoluten Konditionszahl)**.** *In der Literatur findet man auch die folgende Definition[4] der absoluten Konditionszahl einer Funktion $F\colon \mathbb{R}^n \to \mathbb{R}^m$:*

$$\limsup_{\Delta x \to 0} \frac{\|F(x+\Delta x) - F(x)\|_2}{\|\Delta x\|_2} = \lim_{\varepsilon \searrow 0} \sup_{\substack{\|\Delta x\| < \varepsilon \\ \Delta x \neq 0}} \frac{\|F(x+\Delta x) - F(x)\|_2}{\|\Delta x\|_2}. \tag{3.12}$$

Bzgl. welcher Norm die Größe Δx im Supremum kleiner als ε genommen wird, ist hier unerheblich. Die Gleichheit der beiden Ausdrücke in (3.12) gilt nach Definition des Limes superior. Man betrachtet also beliebige Störungen Δx in immer kleiner werdenden Kugeln und bestimmt die größtmögliche Änderung, die solche Störungen im Ergebnis — bezogen auf ihre Größe $\|\Delta x\|_2$ — hervorrufen können.

Man kann zeigen, dass (3.12) genau dann endlich ist, wenn F an der Stelle x differenzierbar ist. In diesem Fall stimmt (3.12) mit (3.11) aus Definition 3.3, also mit $\|F'(x)\|_{2\to 2}$, überein. Wenn F an der Stelle x nicht differenzierbar ist, dann ist (3.12) gleich ∞.

Beachte: Die Wahl der $2 \to 2$-Norm (also der Wahl der 2-Norm in Zähler und Nenner von (3.12)) ist willkürlich. Stattdessen hätten wir z. B. auch die $\infty \to \infty$-Norm nehmen können, wie es im Anschluss bei der relativen Betrachtungsweise gleich der Fall sein wird. Aufgrund der Äquivalenz der Normen ändert sich dadurch die absolute Konditionszahl nur um einen (allerdings dimensionsabhängigen) konstanten Faktor.

[4] vgl. etwa Bornemann, 2018, Kapitel 11.2, S.41

§ 3.2 Relative Betrachtungsweise

Neben den bisherigen absoluten Betrachtungsweise kann man die Sensitivitäten auch relativ betrachten. Das bedeutet, dass wir die Störung in der Eingabe Δx_j auf x_j beziehen und die Änderung in der Ausgabe Δy_i auf y_i. Aus (3.4) wird also

$$\begin{pmatrix} \frac{\Delta y_1}{y_1} \\ \vdots \\ \frac{\Delta y_m}{y_m} \end{pmatrix} \doteq \begin{bmatrix} \frac{\partial F_1(x)}{\partial x_1} \cdot \frac{x_1}{y_1} & \dots & \frac{\partial F_1(x)}{\partial x_n} \cdot \frac{x_n}{y_1} \\ \vdots & & \vdots \\ \frac{\partial F_m(x)}{\partial x_1} \cdot \frac{x_1}{y_m} & \dots & \frac{\partial F_m(x)}{\partial x_n} \cdot \frac{x_n}{y_m} \end{bmatrix} \begin{pmatrix} \frac{\Delta x_1}{x_1} \\ \vdots \\ \frac{\Delta x_n}{x_n} \end{pmatrix}. \tag{3.13}$$

Die Größe

$$k_{ij}(x) := \frac{\partial F_i(x)}{\partial x_j} \cdot \frac{x_j}{y_i} = \frac{\partial F_i(x)}{\partial x_j} \cdot \frac{x_j}{F_i(x)} = K_{ij}(x) \cdot \frac{x_j}{F_i(x)} \tag{3.14}$$

gibt den Faktor an, mit dem sich relative Änderungen $\Delta x_j / x_j$ in der j-ten Komponente der Eingabe relativ auswirken auf die i-te Komponente des Ergebnisses. Die relative Betrachtungsweise setzt natürlich voraus, dass die Komponenten y_i des Ergebnisses $y = F(x)$ ungleich null sind und außerdem die Komponenten x_j der Eingabe ungleich null sind.

Definition 3.6 (Relative partielle Konditionszahlen). *Es sei $F\colon \mathbb{R}^n \to \mathbb{R}^m$ an der Stelle x differenzierbar. Die Zahlen $k_{ij}(x)$ heißen die* ***relativen partiellen Konditionszahlen***[5] *(englisch:* relative partial condition numbers*) der Funktion F an der Stelle x.*

In Analogie zu Definition 3.3 und Bemerkung 3.5 wollen wir noch eine **relative Konditionszahl** $k(x)$ für die Auswertung von F an einer Stelle x angeben. Es stellt sich dabei — um die relative Konditionszahl aus den relativen partiellen Konditionszahlen $k_{ij}(x)$ berechenbar zu machen — als praktisch heraus, bzgl. x komponentenweise vorzugehen.

Wir betrachten dazu zunächst eine reellwertige Funktion $f\colon \mathbb{R}^n \to \mathbb{R}$ und untersuchen die größtmögliche relative Änderung, die Störungen Δx im Ergebnis $f(x)$ hervorrufen können, wenn sie klein werden. Die Größe der Störung Δx messen wir dabei wie gesagt erstens komponentenweise und zweitens relativ zur Größe von x. Es sei dazu D die Diagonalmatrix $D = \operatorname{diag}(x_1, \dots, x_n)$. Dann hat $D^{-1}\Delta x$ die Gestalt

$$D^{-1}\Delta x = \begin{pmatrix} \frac{\Delta x_1}{x_1} \\ \vdots \\ \frac{\Delta x_n}{x_n} \end{pmatrix}.$$

Aufgrund der komponentenweise Betrachtung von Δx messen wir die Größe dieses Ausdrucks als

$$\|D^{-1}\Delta x\|_\infty = \max\left\{ \left| \frac{\Delta x_i}{x_i} \right| \,\middle|\, i = 1, \dots, n \right\}.$$

Die obige Frage nach der größtmöglichen relativen Änderung in $f(x)$ durch Störungen Δx führt uns nun zu dem Ausdruck

$$\limsup_{\Delta x \to 0} \frac{|f(x + \Delta x) - f(x)|}{|f(x)|\, \|D^{-1}\Delta x\|_\infty},$$

[5] In den Wirtschaftswissenschaften werden die relativen partiellen Konditionszahlen $k_{ij}(x)$ auch als **Elastizitäten** bezeichnet.

den wir jetzt untersuchen. Es gilt

$$\begin{aligned}
&\limsup_{\Delta x\to 0} \frac{|f(x+\Delta x)-f(x)|}{|f(x)|\,\|D^{-1}\Delta x\|_\infty} \\
&\quad = \lim_{\varepsilon\searrow 0} \sup_{\substack{\|\Delta x\|<\varepsilon \\ \Delta x\neq 0}} \frac{|f(x+\Delta x)-f(x)|}{|f(x)|\,\|D^{-1}\Delta x\|_\infty}
\end{aligned}$$

Bzgl. welcher Norm Δx im Supremum kleiner als ε genommen wird, ist hier unerheblich. Weiter:

$$\begin{aligned}
&= \lim_{\varepsilon\searrow 0} \sup_{\substack{\|\Delta x\|<\varepsilon \\ \Delta x\neq 0}} \frac{|f'(x)\,\Delta x|}{|f(x)|\,\|D^{-1}\Delta x\|_\infty} \\
&= \sup_{\Delta x\neq 0} \frac{|f'(x)\,\Delta x|}{|f(x)|\,\|D^{-1}\Delta x\|_\infty} \\
&= \sup_{\Delta y\neq 0} \frac{|f'(x)\,D\,\Delta y|}{|f(x)|\,\|\Delta y\|_\infty} \quad \text{(setze } \Delta y = D^{-1}\Delta x\text{)} \\
&= \frac{\|D\,f'(x)^{\mathsf T}\|_1}{|f(x)|} \quad \text{(wegen (2.5) mit der } \|\cdot\|_{\infty\to\infty}\text{-Norm des Zeilenvektors } f'(x)\,D\text{)} \\
&= \sum_{j=1}^n \left|\frac{\partial f(x)}{\partial x_j}\right| \frac{|x_j|}{|f(x)|} = \sum_{j=1}^n |k_{1j}|.
\end{aligned}$$

Ist nun $F\colon \mathbb{R}^n \to \mathbb{R}^m$ vektorwertig, so betrachten wir die obige Rechnung komponentenweise in F_i. Wir erhalten ganz analog wie oben

$$\max_{i=1,\dots,n} \limsup_{\Delta x\to 0} \frac{|F_i(x+\Delta x)-F_i(x)|}{|F_i(x)|\,\|D^{-1}\Delta x\|_\infty} = \max_{i=1,\dots,n} \sum_{j=1}^n \left|\frac{\partial f(x)}{\partial x_j}\right| \frac{|x_j|}{|f(x)|} = \max_{i=1,\dots,n} \sum_{j=1}^n |k_{ij}(x)| \tag{3.15}$$

als Maß für die relative Änderung im Ergebnis. Das ist gerade die Zeilensummennorm der Matrix, die aus den Einträgen $k_{ij}(x)$ besteht.

Quizfrage: Wie würde dieser Ausdruck aussehen, wenn wir statt der ∞-Normen überall die 2-Normen verwenden würden?

Definition 3.7 (Relative Konditionszahl)**.** *Es sei $F\colon \mathbb{R}^n \to \mathbb{R}^m$ an der Stelle x differenzierbar. Weiter sei $x \neq 0$ und $F(x) \neq 0$.*

(*i*) *Dann heißt der Ausdruck*

$$k(x) := \|(k_{ij}(x))\|_{\infty\to\infty} = \max_{i=1,\dots,m} \sum_{j=1}^n |k_{ij}(x)| \tag{3.16}$$

*die **relative Konditionszahl** (englisch:* relative condition number*) von F an der Stelle x.*

(*ii*) *Die Funktion F heißt an der Stelle x **schlecht konditioniert im relativen Sinne**, wenn (3.16) $\gg 1$ ist, ansonsten **gut konditioniert im relativen Sinne**.*

Beispiel 3.8 (Relative Konditionierung). *Wir greifen das Beispiel 3.4 nochmals auf, dieses Mal aber in relativer Betrachtungsweise. Die Matrix der relativen partiellen Konditionszahlen der Funktion $s(t, v_0, a)$ an der Stelle $x = (t, v_0, a) = \left(10\,\mathrm{s}, 0\,\mathrm{m\,s^{-1}}, 5\,\mathrm{m\,s^{-2}}\right)$ ist gleich*

$$\left[50\,\mathrm{m\,s^{-1}} \cdot \frac{10\,\mathrm{s}}{250\,\mathrm{m}} \quad 10\,\mathrm{s} \cdot \frac{0\,\mathrm{m\,s^{-1}}}{250\,\mathrm{m}} \quad 50\,\mathrm{s^2} \cdot \frac{5\,\mathrm{m\,s^{-2}}}{250\,\mathrm{m}}\right] = \begin{bmatrix} 2 & 0 & 1 \end{bmatrix}.$$

Diese Zahlen sind wie erwartet einheitenlos.

Eine Änderung von $\Delta t = 0.1\,\mathrm{s}$ wie in Beispiel 3.4 entspricht in relativer Betrachtungsweise gegenüber dem ungestörten Wert von $t = 10\,\mathrm{s}$ einer Änderung von $\frac{0.1\,\mathrm{s}}{10\,\mathrm{s}} = 0.01$. Diese erzeugt (bis auf Terme höherer Ordnung) eine relative Änderung von $2 \cdot 0.01 = 0.02$ im Ergebnis, also absolut gesehen $0.02 \cdot 250\,\mathrm{m} = 5\,\mathrm{m}$. Das stimmt (wie erwartet) mit dem aus Beispiel 3.4 bekannten Ergebnis überein. Analog ergibt die Änderung von $\Delta a = 0.2\,\mathrm{m\,s^{-2}}$, also in relativer Betrachtung von $\frac{0.2\,\mathrm{m\,s^{-2}}}{5\,\mathrm{m\,s^{-2}}} = 0.04$, eine relative Änderung von $1 \cdot 0.04 = 0.04$ im Ergebnis, also absolut gesehen $0.04 \cdot 250\,\mathrm{m} = 10\,\mathrm{m}$, was wiederum mit dem aus Beispiel 3.4 bekannten Ergebnis übereinstimmt.

Die relative Konditionszahl bzgl. der Anfangsgeschwindigkeit v_0 ist an der betrachteteten Stelle x ohne Aussagekraft, da $v_0 = 0\,\mathrm{m\,s^{-1}}$ ist. Änderungen Δv_0 in v_0 lassen sich also nicht relativ ausdrücken, vgl. (3.13).

In relativer Betrachtungsweise lassen sich in jedem Fall die partiellen relativen Konditionszahlen $k_{ij}(x)$ in physikalisch sinnvoller Weise zur relativen Konditionszahl gemäß (3.16) zusammenfassen, da die $k_{ij}(x)$ immer einheitenlos sind. ***Quizfrage:*** *Wie ändern sich die Zahlenwerte der relativen partiellen Konditionszahlen, wenn man die zurückgelegte Strecke s in* mm *misst statt wie oben in* m? ***Quizfrage:*** *Und was ändert sich, wenn man die Anfangsgeschwindigkeit v_0 in* $\mathrm{km\,h^{-1}}$ *misst statt wie oben in* $\mathrm{m\,s^{-1}}$? ***Quizfrage:*** *Wie groß ist die relative Konditionszahl (3.16) in diesem Beispiel?*

Im weiteren Verlauf der Vorlesung werden vor allem relative Konditionszahlen betrachtet. Das liegt (neben der Tatsache, dass sie vor allem bei Einbeziehung von Einheiten sinnvoller ist) auch daran, dass später wesentliche Fehlereinflüsse durch die Zahldarstellung im Computer eingebracht werden, die gut im relativen Sinn gemessen werden können, siehe Kapitel 2.

Wir geben nun die absoluten und relativen partiellen Konditionszahlen der Grundrechenarten und einiger weiterer elementarer Operationen an. Es handelt sich dabei jeweils um Funktionen $F\colon \mathbb{R}^2 \to \mathbb{R}$ bzw. $F\colon \mathbb{R} \to \mathbb{R}$.

Lemma 3.9 (Absolute und relative partielle Konditionszahlen für einige Grundoperationen). *Die absoluten und relativen Konditionszahlen (im Fall von $x \neq 0$) der folgenden Grundoperationen sind (auf ihren jeweiligen Definitionsbereichen) gegeben durch*

Operation	$K_{11}(x)$	$K_{12}(x)$	$k_{11}(x)$	$k_{12}(x)$	$k(x)$
$x_1 + x_2$	1	1	$\frac{x_1}{x_1+x_2}$	$\frac{x_2}{x_1+x_2}$	$\frac{\lvert x_1\rvert+\lvert x_2\rvert}{\lvert x_1+x_2\rvert}$
$x_1 - x_2$	1	-1	$\frac{x_1}{x_1-x_2}$	$\frac{-x_2}{x_1-x_2}$	$\frac{\lvert x_1\rvert+\lvert x_2\rvert}{\lvert x_1-x_2\rvert}$
$x_1 \cdot x_2$	x_2	x_1	1	1	2
$\frac{x_1}{x_2}$	$\frac{1}{x_2}$	$-\frac{x_1}{x_2^2}$	1	-1	2
$\frac{1}{x_1}$	$-\frac{1}{x_1^2}$		-1		1
$\sqrt{x_1}$	$\frac{1}{2\sqrt{x_1}}$		$\frac{1}{2}$		$\frac{1}{2}$

Beweis. Nachrechnen □

Bemerkung 3.10 (Bedeutung von Lemma 3.9)**.**

(*i*) *Die Ergebnisse von Lemma 3.9 besagen, dass die Multiplikation, Division, Kehrwerte und Quadratwurzeln auf ihrem gesamten Definitionsbereich im relativen Sinne überall gut konditioniert sind.*

(*ii*) *Für die Addition ist die relative Konditionszahl $k(x) = 1$, sofern x_1 und x_2 dasselbe Vorzeichen haben. Man bezeichnet diesen Fall auch als **echte Addition**. Dort wo x_1 und x_2 jedoch verschiedene Vorzeichen haben (also $x_1 + x_2$ einer **echten Subtraktion** zweier positiver Zahlen entspricht), ist die relative Konditionszahl $k(x) > 1$. Für $x_1 \approx -x_2$ kann die relative Konditionszahl beliebig groß werden, siehe Abbildung 3.2. Beispielsweise für festes $x_1 > 0$ und $x(\varepsilon) = (x_1, \varepsilon - x_2)^\mathsf{T}$ mit $\varepsilon > 0$ ist $k(x(\varepsilon)) \in O(\varepsilon^{-1})$. Ein analoges Resultat gilt natürlich auch für die Subtraktion zweier Zahlen, wobei dann der Fall kritisch ist, bei dem beide dasselbe Vorzeichen haben.*

***Beachte:** Die echte Subtraktion zweier fast gleicher Zahlen ist schlecht konditioniert!*

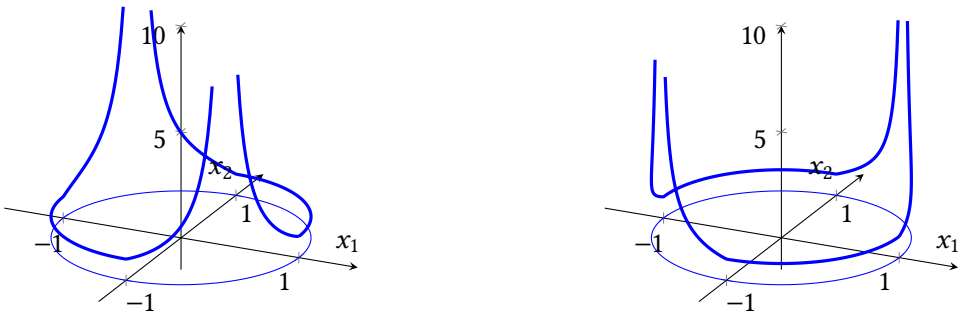


Abbildung 3.2: Relative Konditionszahl $k(x)$ der Addition (links) und der Subtraktion (rechts) zweier Zahlen x_1, x_2 auf dem Einheitskreis.

Das folgende Beispiel wurde nachträglich eingefügt.

Beispiel 3.11 (Auslöschung). *Die Subtraktion von*

$$x_1 = 1.000\,001$$
$$x_2 = 1.000\,000$$

ergibt das Ergebnis $y = x_1 - x_2 = 10^{-6}$. *Die relative partielle Konditionszahl* $k_{11}(x)$ *ist*

$$k_{11}(x) = \frac{x_1}{x_1 - x_2} \approx 1.0 \cdot 10^6.$$

Änderungen etwa von $\Delta x_1 = 10^{-6}$ *ergeben eine Änderung von* $\Delta y = 10^{-6}$ *im Ergebnis. Relativ gesehen sind diese Änderungen* $\Delta x_1/x_1 \approx 10^{-6}$ *und* $\Delta y/y \approx 10^{-6}/10^{-6} = 1$. *Wie erwartet verstärkt sich die relative Störung in der Eingabe um den Faktor* $k_{11}(x) \approx 10^6$.

Nochmal konkret: Statt $y = 1 \cdot 10^{-6}$ *ergibt sich bei Änderung von* x_1 *zu* $x_1 + \Delta x_1$ *nun das Ergebnis* $y + \Delta y = 2 \cdot 10^{-6}$. *Während sich also bei* x_1 *die siebte Dezimalstelle ändert, führt dies im Ergebnis* y *zu einer Änderung bereits der ersten Dezimalstelle. Wir verlieren damit sechs Dezimalstellen an Genauigkeit! Diesen Effekt nennt man* ***Auslöschung*** *(englisch:* cancellation*).*

Beachte: Gegenüber den Daten verliert man (im schlechtesten Fall) etwa $\log_{10} k(x)$ Dezimalziffern an Genauigkeit, wobei $k(x)$ die relative Konditionszahl der betrachteten Operation darstellt.

Der Klarheit halber liefern wir noch eine genaue Definition der Begriffe **echte Addition** und **echte Subtraktion** nach:

Definition 3.12 (Echte Addition, echte Subtraktion)**.**

(*i*) *Wir bezeichnen die Operation* $x_1 + x_2$ *als* ***echte Addition*** *der Zahlen* x_1 *und* x_2, *wenn* $|x_1 + x_2| = |x_1| + |x_2|$ *gilt. Andernfalls heißt sie* ***echte Subtraktion****.*

(*ii*) *Wir bezeichnen die Operation* $x_1 - x_2$ *als* ***echte Addition*** *der Zahlen* x_1 *und* x_2, *wenn* $|x_1 - x_2| = |x_1| + |x_2|$ *gilt. Andernfalls heißt sie* ***echte Subtraktion****.*

Das folgende Beispiel wurde nachträglich eingefügt.

Beispiel 3.13 (Echte Addition, echte Subtraktion)**.**

(*i*) *Die folgenden Operationen sind echte Additionen:*

$$3 + 5, \quad -3 - 5$$

(*ii*) *Die folgenden Operationen sind echte Subtraktionen:*

$$-3 + 5, \quad 5 - 3$$

§ 3.3 Kondition linearer Gleichungssysteme

Lineare Gleichungssysteme sind eine Grundaufgabe in der angewandten und insbesondere in der numerischen Mathematik. Wir betrachten hier Systeme

$$Ax = b \quad \Leftrightarrow \quad x = A^{-1}b$$

mit invertierbaren Matrizen $A \in \mathbb{R}^{n\times n}$ und rechten Seiten $b \in \mathbb{R}^n$. Zugunsten der für Gleichungssysteme üblichen Notation müssen wir in unserem allgemeinen Setting $y = F(x)$ also entsprechende Umbenennungen vornehmen.

Wir betrachten zunächst Störungen Δb der rechten Seite b. Dann übernimmt b die Rolle der Eingabe x, und die Lösung x übernimmt die Rolle des Ergebnisses y. Statt $y = F(x)$ haben wir also $x = A^{-1}b$. Diese Funktion ist linear in der Eingabe b, sodass wir an Stelle der Abschätzung (3.3) sogar die Gleichheit

$$\Delta x = A^{-1}\Delta b \tag{3.17}$$

bekommen. Hier könnten wir jetzt komponentenweise wie in Abschnitt 3.2 vorgehen. Es zeigt sich aber, dass wir das einfachere Resultat erhalten, wenn wir an Stelle der Norm der Vektoren der relativen Änderungen $(\Delta b_i/b_i)$ und $(\Delta x_i/x_i)$ nur die relativen Änderungen in den Normen $\|\Delta b\|/\|b\|$ und $\|\Delta x\|/\|x\|$ betrachten und dabei jeweils die 2-Norm verwenden. Die zugehörigen Abschätzungen

$$\begin{aligned} \|\Delta x\|_2 = \|A^{-1}\Delta b\|_2 &\leqslant \|A^{-1}\|_{2\to 2}\,\|\Delta b\|_2, \\ \|b\|_2 = \|Ax\|_2 &\leqslant \|A\|_{2\to 2}\,\|x\|_2 \end{aligned}$$

lassen sich kombinieren zu

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leqslant \|A\|_{2\to 2}\,\|A^{-1}\|_{2\to 2}\,\frac{\|\Delta b\|_2}{\|b\|_2}. \tag{3.18}$$

Wir betrachten nun Störungen ΔA der Matrix A. Jetzt übernimmt also A die Rolle der Eingabe x. Die differentielle Sensitivitätsanalyse, siehe (3.3), verlangt nun die Bestimmung der Ableitung der Funktion $A \mapsto F(A) := A^{-1}b$, also insbesondere die Bestimmung der Ableitung der Matrixinversion.

Lemma 3.14 (Ableitung der Matrixinversion)**.** *Es sei $\Phi(A) := A^{-1}$ die Matrixinversions-Funktion. Diese ist für jede reguläre (englisch:* non-singular*) Matrix $A \in \mathbb{R}^{n\times n}$ in einer ganzen Umgebung von A definiert und dort stetig differenzierbar.*[6] *Es gilt die Ableitungsformel*

$$\Phi'(A)\,\Delta A = -A^{-1}\Delta A\,A^{-1} \quad \textit{für alle regulären } A \in \mathbb{R}^{n\times n}. \tag{3.19}$$

Beweis. Wir wollen den Satz über implizite Funktionen anwenden auf die implizite Gleichung

$$A\,\Phi(A) = \mathrm{Id}$$

für $\Phi(A)$. Die Differentiation dieser Beziehung in Richtung einer beliebigen Matrix $\Delta A \in \mathbb{R}^{n\times n}$ ergibt nach Produktregel

$$\Delta A\,\Phi(A) + A\left(\Phi'(A)\,\Delta A\right) = 0_{n\times n}. \tag{$*$}$$

[6] Φ ist sogar eine C^∞-Funktion, was wir hier aber nicht benötigen.

Durch Umstellen erhalten wir zunächst

$$A\left(\Phi'(A)\,\Delta A\right) = -\Delta A\,\Phi(A)$$

und dann durch Multiplikation von links mit A^{-1}

$$\Phi'(A)\,\Delta A = -A^{-1}\Delta A\,\Phi(A) = -A^{-1}\Delta A\,A^{-1}.$$

Da also die Auflösung der Gleichung ($*$) nach der gesuchten Richtungsableitung $\Phi'(A)\,\Delta A$ eindeutig möglich war, können wir aus dem Satz über implizite Funktionen schließen, dass die Funktion Φ in einer Umgebung von A tatsächlich definiert und stetig differenzierbar ist und dass die Richtungsableitungen $\Phi'(A)\,\Delta A$ durch (3.19) gegeben sind. □

Mit dem Wissen aus Lemma 3.14 können wir nun die Gleichung (3.3) für unsere Aufgabe mit $F(A) = A^{-1}b$ auswerten, also

$$\Delta x \doteq F'(A)\,\Delta A = \left[\Phi'(A)\,\Delta A\right] b = -A^{-1}\Delta A\,A^{-1}b = -A^{-1}\Delta A\,x. \tag{3.20}$$

Wir schätzen ähnlich wie oben ab:

$$\|\Delta x\|_2 = \|A^{-1}\Delta A\,x\|_2 \leqslant \|A^{-1}\|_{2\to 2}\,\|\Delta A\|_{2\to 2}\,\|x\|_2$$

und erhalten schließlich

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leqslant \|A^{-1}\|_{2\to 2}\,\|A\|_{2\to 2}\,\frac{\|\Delta A\|_{2\to 2}}{\|A\|_{2\to 2}}. \tag{3.21}$$

Definition 3.15 (Konditionszahl einer Matrix)**.** *Es sei $A \in \mathbb{R}^{n\times n}$. Wenn A regulär ist, dann bezeichnen wir die Zahl*

$$\kappa(A) := \|A\|_{2\to 2}\,\|A^{-1}\|_{2\to 2} \tag{3.22}$$

*als die **Konditionszahl der Matrix** A. Wenn A singulär ist, dann setzen wir die Konditionszahl auf $\kappa(A) := \infty$.*

Nach (3.18) und (3.21) gibt die Konditionszahl eine obere Schranke für den Sensitivitätsfaktor an, mit dem sich eine relative Änderung der 2-Norm der rechten Seite bzw. der $2 \to 2$-Norm der Matrix überträgt auf die relative Änderung der 2-Norm der Lösung. In Abschnitt 4 kommen wir nochmal darauf und auch auf Möglichkeiten zur Berechnung von $\kappa(A)$ zurück.

Die Konditionszahl einer Matrix hat aber noch eine andere Interpretation:

Satz 3.16 (Kahan, 1966, vgl. Bornemann, 2018, Kapitel 11.9)**.** *Es sei $A \in \mathbb{R}^{n\times n}$ regulär. Dann gilt*

$$\frac{1}{\kappa(A)} = \min\left\{\frac{\|E\|_{2\to 2}}{\|A\|_{2\to 2}}\,\middle|\, E \in \mathbb{R}^{n\times n},\ A + E \text{ ist singulär}\right\}. \tag{3.23}$$

Der Kehrwert der Konditionszahl einer Matrix ist also ihr relativer Abstand zur Menge der singulären Matrizen in der $2 \to 2$-Norm.

Beweis. Wenn $A+E$ eine singuläre Matrix ist, dann gibt es ein $x \neq 0$ mit der Eigenschaft $(A+E)\,x = 0$, also auch $A^{-1}(A+E)\,x = x + A^{-1}Ex = 0$. Es gilt also

$$0 < \|x\|_2 = \|A^{-1}Ex\|_2 \leqslant \|A^{-1}\|_{2\to 2}\,\|E\|_{2\to 2}\,\|x\|_2 = \kappa(A)\,\frac{\|E\|_{2\to 2}}{\|A\|_{2\to 2}}\,\|x\|_2.$$

Daraus folgt

$$\frac{1}{\kappa(A)} \leqslant \frac{\|E\|_{2\to 2}}{\|A\|_{2\to 2}}\,\cancel{\|x\|_2}.$$

Um den Beweis abzuschließen, geben wir jetzt eine spezielle Matrix E an, für die hier die Gleichheit gilt, also $\kappa(A) = \frac{\|A\|_{2\to 2}}{\|E\|_{2\to 2}}$ oder äquivalent $\|E\|_{2\to 2} = 1/\|A^{-1}\|_{2\to 2}$. Dazu sei $y \in \mathbb{R}^n$ ein Vektor mit $\|y\|_2 = 1$, der das Maximum in der Definition der $2 \to 2$-Operatornorm für A^{-1} realisiert, also

$$\|A^{-1}\|_{2\to 2} = \max_{\substack{z\in\mathbb{R}^n\\ \|z\|_2=1}} \|A^{-1}z\|_2 = \|A^{-1}y\|_2,$$

vgl. Lemma 2.3 *(iii)*. Außerdem sei $x := A^{-1}y \neq 0$. Wir wählen $E := -\frac{y\,x^{\mathsf T}}{\|x\|_2^2}$. Es gilt

$$\|E\|_{2\to 2} = \frac{\|x\|_2\,\|y\|_2}{\|x\|_2^2} = \frac{1}{\|x\|_2} = \frac{1}{\|A^{-1}y\|_2} = \frac{1}{\|A^{-1}\|_{2\to 2}}.$$

Für den Beweis der ersten Gleichheit (Spektralnorm für Rang-1-Matrizen) betrachten wir

$$\begin{aligned}
\|E\|_{2\to 2} &= \max_{\substack{z\in\mathbb{R}^n\\ \|z\|_2=1}} \|Ez\|_2 \quad \text{(Lemma 2.3)}\\
&= \max_{\substack{z\in\mathbb{R}^n\\ \|z\|_2=1}} \frac{\|y\,x^{\mathsf T}z\|_2}{\|x\|_2^2} \quad \text{(Definition von } E\text{)}\\
&= \max_{\substack{z\in\mathbb{R}^n\\ \|z\|_2=1}} \frac{\|y\|_2}{\|x\|_2^2}|x^{\mathsf T}z| \quad \text{(positive Homogenität der Norm)}\\
&= \frac{\|y\|_2}{\|x\|_2^2}\,\frac{x^{\mathsf T}x}{\|x\|_2} \quad (z = x/\|x\|_2 \text{ maximiert den Ausdruck, Cauchy-Schwarz-Ungleichung})\\
&= \frac{\|x\|_2\,\|y\|_2}{\|x\|_2^2}.
\end{aligned}$$

□

§ 3.4 Kondition zusammengesetzter Funktionen

In diesem Abschnitt geht es darum, dass wir eine Funktionsauswertung $y = F(x)$ für $F\colon \mathbb{R}^n \to \mathbb{R}^m$ in kleinere Einheiten zerlegen, im einfachsten Fall

$$y = F(x) = (H \circ G)(x) = H(G(x))$$

mit Funktionen

$$G\colon \mathbb{R}^n \to \mathbb{R}^\ell \quad \text{und} \quad H\colon \mathbb{R}^\ell \to \mathbb{R}^m.$$

Die Funktionen G und H seien an der Stelle x bzw. an der Stelle $G(x)$ differenzierbar. Nach Kettenregel gilt für die differentielle Störungsanalyse, vgl. (3.3),

$$\Delta y \doteq (H \circ G)'(x)\,\Delta x = H'(G(x))\,G(x)\,\Delta x. \tag{3.24}$$

Wir können also die absolute Konditionszahl von F an der Stelle x mit Hilfe von Lemma 2.3 (*v*) (Submultiplikativität der Matrixnormen) abschätzen durch

$$K(x) = \|F'(x)\|_{2\to 2} \leqslant \|H'(G(x))\|_{2\to 2}\,\|G'(x)\|_{2\to 2}. \tag{3.25}$$

Eine solche Abschätzung wird i. A. nicht scharf sein (**Quizfrage:** Warum?) Dennoch ist die Betrachtung später hilfreich für die Stabilitätsanalyse von Algorithmen.

Wir können eine zu (3.24) analoge Beziehung auch für die relative Sensitivätsanalyse finden. Wir schreiben dazu (3.24) einmal komponentenweise auf:

$$\begin{pmatrix} \Delta y_1 \\ \vdots \\ \Delta y_m \end{pmatrix} \doteq \begin{bmatrix} \frac{\partial H_1(G(x))}{\partial z_1} & \dots & \frac{\partial H_1(G(x))}{\partial z_\ell} \\ \vdots & & \vdots \\ \frac{\partial H_m(G(x))}{\partial z_1} & \dots & \frac{\partial H_m(G(x))}{\partial z_\ell} \end{bmatrix} \begin{bmatrix} \frac{\partial G_1(x)}{\partial x_1} & \dots & \frac{\partial G_1(x)}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial G_\ell(x)}{\partial x_1} & \dots & \frac{\partial G_\ell(x)}{\partial x_n} \end{bmatrix} \begin{pmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{pmatrix}$$

und ergänzen wie in (3.13) die Terme, die zu einer relativen Betrachtungsweise führen:

$$\begin{pmatrix} \frac{\Delta y_1}{y_1} \\ \vdots \\ \frac{\Delta y_m}{y_m} \end{pmatrix} \doteq \underbrace{\begin{bmatrix} \frac{\partial H_1(G(x))}{\partial z_1}\cdot\frac{z_1}{y_1} & \dots & \frac{\partial H_1(G(x))}{\partial z_\ell}\cdot\frac{z_\ell}{y_1} \\ \vdots & & \vdots \\ \frac{\partial H_m(G(x))}{\partial z_1}\cdot\frac{z_1}{y_m} & \dots & \frac{\partial H_m(G(x))}{\partial z_\ell}\cdot\frac{z_\ell}{y_m} \end{bmatrix}}_{=:(k^H_{ik}(G(x)))} \underbrace{\begin{bmatrix} \frac{\partial G_1(x)}{\partial x_1}\cdot\frac{x_1}{z_1} & \dots & \frac{\partial G_1(x)}{\partial x_n}\cdot\frac{x_n}{z_\ell} \\ \vdots & & \vdots \\ \frac{\partial G_\ell(x)}{\partial x_1}\cdot\frac{x_1}{z_1} & \dots & \frac{\partial G_\ell(x)}{\partial x_n}\cdot\frac{x_n}{z_\ell} \end{bmatrix}}_{=:(k^G_{kj}(x))} \begin{pmatrix} \frac{\Delta x_1}{x_1} \\ \vdots \\ \frac{\Delta x_n}{x_n} \end{pmatrix} \tag{3.26}$$

Auch hier können wir also wieder die Submultiplikativität der Matrixnormen (Lemma 2.3 (*v*)) verwenden, um abzuschätzen:

$$\left\| \begin{pmatrix} \frac{\Delta y_1}{y_1} \\ \vdots \\ \frac{\Delta y_m}{y_m} \end{pmatrix} \right\|_\infty \leqslant \left\|(k^H_{ik}(G(x)))\right\|_{\infty\to\infty} \left\|(k^G_{kj}(x))\right\|_{\infty\to\infty} \left\| \begin{pmatrix} \frac{\Delta x_1}{x_1} \\ \vdots \\ \frac{\Delta x_n}{x_n} \end{pmatrix} \right\|_\infty . \tag{3.27}$$

Auch diese Abschätzung ist i. A. nicht scharf, aber eine ähnliche Betrachtung wird später bei der Stabilitätsanalyse von Algorithmen noch nützlich sein.

Bemerkung 3.17. *Wir betonen nochmal zur Verdeutlichung, dass die Kondition eine Eigenschaft der betrachteten Aufgabe ist. Sie hat nichts damit zu tun, mit welchem Verfahren wir einmal diese Aufgabe versuchen werden zu lösen.*

Bei einer schlecht konditionierten Aufgabe muss man sich die Frage stellen, ob die Aufgabe überhaupt sachgerecht formuliert ist.

Ende der Woche 2

§ 4 Singulärwertzerlegung von Matrizen

§ 4.1 Spektralzerlegung

Eine Zahl $\lambda \in \mathbb{C}$ heißt **Eigenwert** (englisch: *eigenvalue*) und ein Vektor $v \in \mathbb{C}^n \setminus \{0\}$ zugehöriger **Eigenvektor** (englisch: *eigenvector*) einer Matrix $A \in \mathbb{R}^{n\times n}$ (oder $A \in \mathbb{C}^{n\times n}$), wenn gilt:

$$A v = \lambda v, \tag{4.1}$$

d. h., dass v unter der Abbildung $v \mapsto A v$ auf ein Vielfaches von sich selbst abgebildet wird. Zusammen heißt (λ, v) ein **Eigenpaar** von A. Im Allgemeinen muss man damit rechnen, dass Eigenwerte und Eigenvektoren komplex sind, selbst wenn A eine reelle Matrix ist.

In der linearen Algebra untersucht man u. a. die Frage, unter welchen Umständen eine Basis aus lauter Eigenvektoren von A existiert und wann diese als Orthonormalbasis gewählt werden kann.

Definition 4.1 (Orthogonale Matrix). *Eine Matrix $V \in \mathbb{R}^{n\times n}$ heißt **orthogonal**, wenn $V^\mathsf{T} V = \mathrm{Id}$ ist.*

Diese Bedingung bedeutet, dass die Spaltenvektoren v_i von V paarweise senkrecht (bzgl. des Euklidischen Innenprodukts) aufeinander stehen und außerdem $\|v_i\|_2 = 1$ ist, denn es gilt

$$(V^\mathsf{T} V)_{ij} = v_i^\mathsf{T} v_j = \begin{cases} \|v_i\|_2^2 = 1 & \text{im Fall } i = j \\ 0 & \text{im Fall } i \neq j. \end{cases}$$

Eine Matrix $V \in \mathbb{R}^{n\times n}$ ist also genau dann orthogonal, wenn ihre Spaltenvektoren eine (reelle) **Orthonormalbasis** von $\mathbb{R}^n$ bilden, also eine Menge linear unabhängiger Vektoren, die paarweise senkrecht stehen und deren 2-Norm gleich eins ist. Für orthogonale Matrizen gilt offenbar $V^{-1} = V^\mathsf{T}$ und daher auch $V V^\mathsf{T} = \mathrm{Id}$.

Außerdem ist das Produkt orthogonaler Matrizen wieder eine orthogonale Matrix.[7]

Beachte: Der Begriff «orthogonale Matrix» ist für reelle Matrizen reserviert. Das komplexe Analogon heißt **unitäre Matrix**.

Wir interessieren uns hier nur für die Frage, wann eine Matrix $A \in \mathbb{R}^{n\times n}$ die Eigenschaft besitzt, eine Orthonormalbasis von $\mathbb{R}^n$ aus lauter reellen Eigenvektoren zusammenstellen zu können. Die Antwort gibt folgender Satz.

Satz 4.2 (Spektralsatz). *Es sei $A \in \mathbb{R}^{n\times n}$. Der Raum $\mathbb{R}^n$ besitzt genau dann eine reelle Orthonormalbasis aus lauter Eigenvektoren von A, wenn $A = A^\mathsf{T}$ ist. In diesem Fall besitzt A die folgende **Spektralzerlegung** (englisch: spectral decomposition):*

$$A = V \Lambda V^\mathsf{T}, \tag{4.2}$$

[7] D. h., die orthogonalen Matrizen bilden die algebraische Struktur einer Gruppe, die sog. **orthogonale Gruppe** (englisch: *orthogonal group*). Diese ist eine Untergruppe der **allgemeinen linearen Gruppe** (englisch: *general linear group*), die aus den invertierbaren $n \times n$-Matrizen besteht.

wobei $\Lambda \in \mathbb{R}^{n\times n}$ die Diagonalmatrix der Eigenwerte ist und die orthogonale Matrix $V \in \mathbb{R}^{n\times n}$ aus Eigenvektoren von A besteht.

Die Gleichung (4.2) können wir auch lesen als

$$A \begin{bmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{bmatrix} = \begin{bmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix},$$

also spaltenweise: $A\,v_i = \lambda_i\,v_i$. (**Quizfrage:** Warum?)

§ 4.2 Singulärwertzerlegung

Die Spektralzerlegung mit Hilfe der Eigenpaare ist nicht immer die geeignete Darstellung einer Matrix. Insbesondere für rechteckige Matrizen $A \in \mathbb{R}^{m\times n}$ existieren überhaupt keine Eigenpaare. Daher führen wir jetzt die Singulärwertzerlegung ein, die immer existiert.

Definition 4.3 (Singulärwertzerlegung). *Unter einer **Singulärwertzerlegung** (englisch:* singular value decomposition*, kurz: **SVD**) einer Matrix $A \in \mathbb{R}^{m\times n}$ versteht man eine Darstellung der Gestalt*

$$A = U\Sigma V^{\mathsf{T}}, \tag{4.3}$$

wobei

- *$U \in \mathbb{R}^{m\times m}$ und $V \in \mathbb{R}^{n\times n}$ orthogonale Matrizen sind und*
- *$\Sigma \in \mathbb{R}^{m\times n}$ eine (rechteckige) Diagonalmatrix ist, deren Diagonaleinträge $\sigma_1, \ldots, \sigma_{\min\{m,n\}} \geqslant 0$ sind.*

*Die Spaltenvektoren u_i von U heißen **Linkssingulärvektoren** (englisch:* left singular vectors*) von A. Die Spaltenvektoren v_i von V heißen **Rechtssingulärvektoren** (englisch:* right singular vectors*) von A. Die Zahlen σ_i heißen **Singulärwerte** (englisch:* singular values*) von A.*

Im Falle von $n > m$ hat die Singulärwertzerlegung beispielsweise die folgende Gestalt:

$$A = \begin{bmatrix} | & & | \\ u_1 & \cdots & u_m \\ | & & | \end{bmatrix} \left[\begin{array}{ccc|ccc} \sigma_1 & & & 0 & \cdots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & \sigma_m & 0 & \cdots & 0 \end{array}\right] \begin{bmatrix} \text{—} & v_1^{\mathsf{T}} & \text{—} \\ & \vdots & \\ & \vdots & \\ \text{—} & v_n^{\mathsf{T}} & \text{—} \end{bmatrix} \tag{4.4a}$$

und im Falle von $m > n$

$$A = \begin{bmatrix} | & & | \\ u_1 & \cdots & u_m \\ | & & | \end{bmatrix} \left[\begin{array}{ccc} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ \hline 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{array}\right] \begin{bmatrix} — & v_1^\mathsf{T} & — \\ & \vdots & \\ — & v_n^\mathsf{T} & — \end{bmatrix}. \tag{4.4b}$$

Jede Matrix $A \in \mathbb{R}^{m\times n}$ besitzt eine Singulärwertzerlegung. Um das zu begründen (**Quizfrage:** Können Sie die Details ausarbeiten?), kann man Spektralzerlegungen

$$A^\mathsf{T}A = V\Lambda V^\mathsf{T} \quad \text{und} \quad AA^\mathsf{T} = U\widetilde{\Lambda}U^\mathsf{T} \tag{4.5}$$

der reellen, symmetrischen, positiv semi-definiten Matrizen $A^\mathsf{T}A \in \mathbb{R}^{n\times n}$ und $AA^\mathsf{T} \in \mathbb{R}^{m\times m}$ gemäß Satz 4.2 betrachten. Die Singulärwerte sind gerade die Wurzeln der Diagonaleinträge der kleineren der beiden Diagonalmatrizen $\Lambda \in \mathbb{R}^{n\times n}$ und $\widetilde{\Lambda} \in \mathbb{R}^{m\times m}$.

Bemerkung 4.4 (Eindeutigkeit und Anordnung der Singulärwerte)**.** *Die Singulärwerte sind eindeutig und werden i. d. R. absteigend angeordnet:*

$$\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > 0 = \sigma_{r+1} = \cdots = \sigma_{\min\{m,n\}}. \tag{4.6}$$

Wir gehen ab jetzt immer von absteigend sortierten Singulärwerten aus.

Die Singulärwertzerlegung enthält sehr viel Information über die Matrix A und die zu ihr gehörende lineare Abbildung $x \mapsto Ax$. Beispielsweise ist die Anzahl $0 \leqslant r \leqslant \min\{m,n\}$ der echt positiven Singulärwerte gerade gleich $\operatorname{Rang}(A)$, also gleich der Dimension des Bildraumes $\operatorname{Bild} A$. Die «späten» Rechtssingulärvektoren $v_{r+1}, \ldots, v_n$ bilden eine orthonormale Basis von $\ker A$, und die «frühen» Rechtssingulärvektoren $v_1, \ldots, v_r$ bilden eine orthonormale Basis des orthogonalen Komplements $(\ker A)^\perp$. Weiterhin bilden die «frühen» Linkssingulärvektoren $u_1, \ldots, u_r$ eine Basis von $\operatorname{Bild} A$, und die «späten» Linkssingulärvektoren $u_{r+1}, \ldots, u_m$ bilden eine Basis von $(\operatorname{Bild} A)^\perp$. **Quizfrage:** Klar?

Quizfrage: Was vermuten Sie: Ist die Singulärwertzerlegung einer Matrix eindeutig?

Bemerkung 4.5 (Dünne Singulärwertzerlegung)**.** *Im Falle von $n > m$ sehen wir aus der Darstellung (4.4a), dass die «späten» Zeilen von V^T (Spalten von V) mit den Indizes $m+1, \ldots, n$ zur Darstellung von A nicht benötigt werden. Es gilt also auch*

$$A = \begin{bmatrix} | & & | \\ u_1 & \cdots & u_m \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \end{bmatrix} \begin{bmatrix} —— & v_1^\mathsf{T} & —— \\ & \vdots & \\ —— & v_m^\mathsf{T} & —— \end{bmatrix}. \tag{4.7a}$$

Im Falle von $m > n$ werden hingegen die «späten» Spalten von U mit den Indizes $n+1, \dots, m$ nicht benötigt. Es gilt also auch

$$A = \begin{bmatrix} | & & | \\ u_1 & \cdots & u_n \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} — & v_1^\mathsf{T} & — \\ & \vdots & \\ — & v_n^\mathsf{T} & — \end{bmatrix}. \tag{4.7b}$$

*Man spricht dann von einer **dünnen Singulärwertzerlegung** (englisch:* thin SVD, economy-size SVD*).*

Beispiel 4.6 (Singulärwertzerlegung). *Die Matrix*

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

besitzt die Singulärwertzerlegung (aus Darstellungsgründen auf zwei Nachkommastellen gerundet)

$$A = \left[\begin{array}{cc|c} -0.45 & 0.02 & -0.89 \\ -0.89 & 0.04 & 0.45 \\ -0.05 & -1.00 & 0.00 \\ \multicolumn{2}{c|}{\text{Bild}\, A} & (\text{Bild}\, A)^\perp \end{array}\right] \begin{bmatrix} 12.26 & 0 & 0 & 0 \\ 0 & 1.30 & 0 & 0 \\ 0 & 0 & 0.00 & 0 \end{bmatrix} \left[\begin{array}{cc|cc} -0.19 & -0.37 & -0.55 & -0.73 \\ -0.69 & -0.61 & 0.23 & 0.31 \\ -0.70 & 0.70 & -0.08 & -0.11 \\ 0.00 & 0.00 & -0.80 & 0.60 \\ \multicolumn{2}{c|}{(\ker A)^\perp} & \multicolumn{2}{c}{\ker A} \end{array}\right]^\mathsf{T}.$$

Der Rang der Matrix ist also gleich 2. Mit Hilfe von NumPy kann diese Singulärwertzerlegung wie folgt berechnet werden:

```
import numpy as np
A = np.array([[1, 2, 3, 4], [2, 4, 6, 8], [1, 1, 0, 0]])
U, Sigma, VT = np.linalg.svd(A)
```

Die Darstellung (4.3) erlaubt es uns, die Matrix A als eine Summe von r Rang-1-Matrizen darzustellen:

$$A = U\Sigma V^\mathsf{T} = \sum_{i=1}^{r} \sigma_i \underbrace{u_i\, v_i^\mathsf{T}}_{\text{Rang-1-Matrix}}. \tag{4.8}$$

Damit können wir auch Matrix-Vektor-Produkte genau analysieren:

$$Ax = U\Sigma V^\mathsf{T} x = \sum_{i=1}^{r} \sigma_i\, u_i\, (v_i^\mathsf{T} x). \tag{4.9}$$

(**Quizfrage:** Wie können wir diese Darstellung interpretieren? Was also «macht» $v_i^\mathsf{T} x$? Was bedeutet die Multiplikation mit σ_i? Und was die Multiplikation mit u_i?) Insbesondere folgt aus (4.9) sofort

$$Av_i = \begin{cases} \sigma_i\, u_i & \text{für } i = 1, \dots, r, \\ 0 & \text{für } i = r+1, \dots, n. \end{cases}$$

Mit Hilfe der Darstellung (4.9) können wir nun die Frage auflösen, warum die Matrixnorm $\|A\|_{2\to 2}$ durch (2.3b) gegeben ist. Die (quadrierte) Norm des Bildes Ax lässt sich nun unter Ausnutzung der Orthonomalität der Vektoren u_i wie folgt bestimmen:

$$\begin{aligned}
\|Ax\|_2^2 &= \Big\|\sum_{i=1}^r \sigma_i\, u_i\, (v_i^\mathsf{T} x)\Big\|_2^2 = \Big(\sum_{i=1}^r \sigma_i\, u_i\, (v_i^\mathsf{T} x), \sum_{j=1}^r u_j\, \sigma_j\, (v_j^\mathsf{T} x)\Big) \quad \text{(wegen (4.9))} \\
&= \sum_{i=1}^r \sum_{j=1}^r \sigma_i\, \sigma_j\, u_i^\mathsf{T} u_j\, (v_i^\mathsf{T} x)\, (v_j^\mathsf{T} x) \\
&= \sum_{i=1}^r \sigma_i^2\, \|u_i\|_2^2\, (v_i^\mathsf{T} x)^2 \quad \text{(wegen der Orthogonalität von } u_i \text{ und } u_j \text{ für } i \neq j) \\
&= \sum_{i=1}^r \sigma_i^2\, (v_i^\mathsf{T} x)^2 \quad \text{(wegen } \|u_i\|_2 = 1) \\
&\leqslant \sigma_1^2 \sum_{i=1}^r (v_i^\mathsf{T} x)^2 \\
&= \sigma_1^2 \|x\|_2^2.
\end{aligned}$$

Die obige Abschätzung ist mit Gleichheit erfüllt, wenn x ein Vielfaches von v_1 ist. In dem Fall gilt $\|Ax\|_2 = \sigma_1 \|x\|_2$. Da der größte Singulärwert σ_1 gleichzeitig die Wurzel des größten Eigenwertes von $A^\mathsf{T}A$ ist, haben wir damit (2.3b) gezeigt und außerdem:

Satz 4.7 (Die $\|\cdot\|_{2\to 2}$-Norm einer Matrix ist ihr größter Singulärwert)**.**

(*i*) *Für jede Matrix $A \in \mathbb{R}^{m\times n}$ gilt:*

$$\|A\|_{2\to 2} = \sigma_1, \tag{4.10}$$

also ist die Spektralnorm gerade der größte Singulärwert der Matrix.

(*ii*) *Für symmetrische Matrizen $A \in \mathbb{R}^{n\times n}$ gilt gleichzeitig:*

$$\|A\|_{2\to 2} = \max\big\{|\lambda| \,\big|\, \lambda \text{ ist ein Eigenwert von } A\big\}, \tag{4.11}$$

also ist die Spektralnorm gerade der Betrag des betragsgrößten Eigenwerts der Matrix.

Beweis. Aussage (*i*) haben wir oben bereits gezeigt. Es gilt nach Konstruktion der Singulärwertzerlegung

$$\sigma_1^2 = \max\big\{\lambda \,\big|\, \lambda \text{ ist ein Eigenwert von } A^\mathsf{T}A\big\}.$$

Wenn A symmetrisch ist, dann existiert die Spektralzerlegung $A = V\Lambda V^\mathsf{T}$, siehe (4.2) mit einer orthogonalen Matrix V. Es gilt also $A^\mathsf{T}A = V\Lambda V^\mathsf{T} V\Lambda V^\mathsf{T} = V\Lambda^2 V^\mathsf{T}$. Die Eigenwerte von $A^\mathsf{T}A$ sind demnach — wenn A symmetrisch ist — gerade die Quadrate der Eigenwerte von A. Wir haben also

$$\begin{aligned}
\sigma_1^2 &= \max\big\{\lambda \,\big|\, \lambda \text{ ist ein Eigenwert von } A^\mathsf{T}A\big\} \\
&= \max\big\{\lambda^2 \,\big|\, \lambda \text{ ist ein Eigenwert von } A\big\}.
\end{aligned}$$

Durch Wurzelziehen folgt Aussage (*ii*). □

Im Beweis von Satz 3.16 haben wir die Norm $\|y\,x^\mathsf{T}\|_{2\to 2}$ der Rang-1-Matrix $y\,x^\mathsf{T}$ benötigt. **Quizfrage:** Können Sie diese Norm mit dem jetzigen Wissen über die Singulärwertzerlegung bestimmen?

Falls $A \in \mathbb{R}^{n\times n}$ invertierbar ist (**Quizfrage:** Wie drückt man das mit Hilfe der Singulärwerte von A aus?), dann erlaubt es uns die Singulärwertzerlegung $A = U\Sigma V^\mathsf{T}$, auch direkt eine Singulärwertzerlegung von A^{-1} anzugeben:

$$A^{-1} = V\Sigma^{-1}U^\mathsf{T}. \tag{4.12}$$

Hieraus sieht man sofort, dass $1/\sigma_n$ der maximale Singulärwert von A^{-1} ist.

Weiter ergibt sich aus diesen Überlegungen, dass die Konditionszahl (3.22) einer regulären Matrix $A \in \mathbb{R}^{n\times n}$ die Beziehung

$$\kappa(A) = \|A\|_{2\to 2}\,\|A^{-1}\|_{2\to 2} = \frac{\sigma_1}{\sigma_n} \tag{4.13}$$

erfüllt.

Quizfrage: Was ist die Konditionszahl einer orthogonalen Matrix? **Quizfrage:** Welche quadratischen (notwendigerweise invertierbaren) Matrizen haben Konditionszahl gleich eins? **Quizfrage:** Wie sieht die Singulärwertzerlegung einer Diagonalmatrix aus?

Auf Basis der SVD kann man sehr schön sehen, warum die Abschätzung

$$\|BA\|_{2\to 2} \leqslant \|B\|_{2\to 2}\,\|A\|_{2\to 2}$$

aus Lemma 2.3 (*v*) (Submultiplikativität) insbesondere für die Spektralnorm i. A. nicht scharf ist. Wir betrachten dazu Singulärwertzerlegungen beider Matrizen:

$$B = U^{(2)}\Sigma^{(2)}V^{(2)\mathsf{T}} \quad \text{und} \quad A = U^{(1)}\Sigma^{(1)}V^{(1)\mathsf{T}}.$$

Es gilt also $\|A\|_{2\to 2} = \sigma_1^{(1)}$ und $\|B\|_{2\to 2} = \sigma_1^{(2)}$. Es sei nun x ein Rechtssingulärvektor von A zum größten Singulärwert $\sigma_1^{(1)}$ mit $\|x\|_2 = 1$. Dann gilt $Ax = \sigma^{(1)}u_1^{(1)}$ mit $\|Ax\|_2 = \sigma^{(1)}$. Falls nun $u^{(1)}$ ein Rechtssingulärvektor von B zum größten Singulärwert $\sigma_1^{(2)}$ ist, so gilt weiter

$$BAx = \sigma_1^{(1)}Bu^{(1)} = \sigma_1^{(1)}\sigma_1^{(2)}u^{(2)} \text{ mit } \|BAx\|_2 = \sigma_1^{(1)}\sigma_1^{(2)}.$$

In diesem Fall haben wir also tatsächlich einen Vektor gefunden, für den $\|BAx\|_2 = \|B\|_{2\to 2}\|A\|_{2\to 2}\|x\|_2$ gilt und damit $\|BA\|_{2\to 2} = \|B\|_{2\to 2}\|A\|_{2\to 2}$. Im anderen Fall besitzt $u^{(1)}$ auch Anteile von Rechtssingulärvektoren der Matrix B, die nicht zum größten Singulärwert von B gehören. Das impliziert $\|Bu^{(1)}\|_2 < \sigma_1^{(1)}\|u^{(1)}\|_2$.

Mit diesen Überlegungen kann man folgenden Satz beweisen:

Satz 4.8 (Submultiplikativität der $2 \to 2$-Norm)**.** *Es seien $A \in \mathbb{R}^{m\times n}$ und $B \in \mathbb{R}^{\ell\times m}$. Dann sind die folgenden Aussagen äquivalent:*

(*i*) *Es gilt* $\|BA\|_{2\to 2} = \|B\|_{2\to 2}\|A\|_{2\to 2}$.

(*ii*) *Es gibt einen Vektor* $x \in \mathbb{R}^n$, $x \neq 0$, *mit der Eigenschaft* $\|Ax\|_2 = \sigma_1^{(1)} \|x\|_2$ *und* $\|BAx\|_2 = \sigma_1^{(2)} \|Ax\|_2$.

(*iii*) *Der Unterraum von* $\mathbb{R}^m$, *der durch die Linkssingulärvektoren von* A *aufgespannt wird, die zum größten Singulärwert* $\sigma_1^{(1)}$ *gehören, schneidet den Unterraum von* $\mathbb{R}^m$, *der durch die Rechtssingulärvektoren von* B *aufgespannt wird, die zum größten Singulärwert* $\sigma_1^{(2)}$ *gehören, in mehr als nur dem Nullvektor.*

Folgerung 4.9 (Submultiplikativität der $2 \to 2$-Norm)**.** *Insbesondere in folgenden Fällen sind die Aussagen (i) bis (iii) wahr:*

- *Die «mittlere» Dimension ist* $m = 1$.
- *Mindestens eine der Matrizen* A *oder* B *ist die Nullmatrix.*
- *Mindestens eine der Matrizen* A *oder* B *ist eine orthogonale Matrix.*

Quizfrage: Beweis?

Eine weitere wichtige Folgerung daraus ist:

Folgerung 4.10 (Multiplikation mit einer orthogonalen Matrix ändert die Konditionszahl nicht)**.** *Es sei* $A \in \mathbb{R}^{n\times n}$ *beliebig und* $V \in \mathbb{R}^{n\times n}$ *orthogonal. Dann gilt* $\|AV\|_{2\to 2} = \|VA\|_{2\to 2} = \|A\|_{2\to 2}$.

Beweis. □

Schließlich geben wir noch eine Eigenschaft der Singulärwertzerlegung an.

Satz 4.11 (Die SVD löst die Aufgabe der Best-Approximation mit konstantem Rang, vgl. Golub, van Loan, 1983, Theorem 2.5.3)**.** *Es sei* $A \in \mathbb{R}^{m\times n}$ *eine Matrix mit Singulärwertzerlegung* $A = U\Sigma V^\mathsf{T}$. *Weiter sei* $r = \operatorname{Rang}(A)$ *und* $0 \leqslant k \leqslant r$. *Dann löst die Matrix*

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^\mathsf{T} = \begin{bmatrix} | & & | \\ u_1 & \cdots & u_k \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix} \begin{bmatrix} — & v_1^\mathsf{T} & — \\ & \vdots & \\ — & v_k^\mathsf{T} & — \end{bmatrix} \in \mathbb{R}^{m\times n} \tag{4.14}$$

die folgende Optimierungsaufgabe:

$$\begin{aligned} \textit{Minimiere} \quad & \|A - B\|_{2\to 2}, \quad B \in \mathbb{R}^{m\times n} \\ \textit{unter der Bedingung} \quad & \operatorname{Rang}(B) = k. \end{aligned} \tag{4.15}$$

Der Optimalwert ist $\|A - A_k\|_2 = \sigma_{k+1}$.

Beweis. Wir überzeugen uns zunächst davon, dass die Matrix A_k den Rang k hat und dadurch als Kandidat in der Optimierungsaufgabe (4.15) zur Verfügung steht. Unter Ausnutzung der Orthogonalität von U und V ergibt sich aus (4.14)

$$U^\mathsf{T} A_k V = \left[\begin{array}{ccc|ccc} \sigma_1 & & & 0 & \cdots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & \sigma_k & 0 & \cdots & 0 \\ \hline 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{array}\right] \in \mathbb{R}^{m\times n}.$$

Da die Multiplikation von A_k mit invertierbaren Matrizen den Rang nicht verändert und $\sigma_1 \geqslant \cdots \geqslant \sigma_k > 0$ gilt, gilt tatsächlich $\operatorname{Rang}(A_k) = k$. **Quizfrage:** Warum lässt die Multiplikation einer Matrix mit einer invertierbaren Matrix den Rang unverändert? **Quizfrage:** Warum ist $\sigma_k > 0$?

Weiter gilt (Illustration für den Fall $m > n$)

$$U^\mathsf{T}(A - A_k)V = \left[\begin{array}{ccc|ccc} 0 & & & 0 & \cdots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & 0 & 0 & \cdots & 0 \\ \hline 0 & \cdots & 0 & \sigma_{k+1} & & \\ \vdots & & \vdots & & \ddots & \\ 0 & \cdots & 0 & & & \sigma_p \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{array}\right] \in \mathbb{R}^{m\times n}$$

mit $p = \min\{m, n\}$. Aus Folgerung 4.10 folgt $\|A - A_k\|_{2\to 2} = \|U^\mathsf{T}(A - A_k)V\|_{2\to 2}$, und letztere Matrix hat offenbar σ_{k+1} als größten Singulärwert, also gilt $\|A - A_k\|_{2\to 2} = \sigma_{k+1}$.

Wir zeigen jetzt noch, dass für jede beliebige Matrix $B \in \mathbb{R}^{m\times n}$ mit $\operatorname{Rang}(B) = k$ gilt: $\|A - B\|_{2\to 2} \geqslant \sigma_{k+1}$. Damit ist dann die behauptete Optimalität von A_k gezeigt und der Beweis abgeschlossen. Die Dimensionsformel $k = \dim \operatorname{Bild} B = n - \dim \ker B$ zeigt, dass der Kern von B die Dimension $n - k$ besitzt. Wir können also eine Menge orthonormaler Vektoren $x_1, \ldots, x_{n-k}$ finden, sodass

$$\ker B = \operatorname{span}\{x_1, \ldots, x_{n-k}\}$$

gilt. Aus Dimensionsgründen gilt

$$\operatorname{span}\{x_1, \ldots, x_{n-k}\} \cap \operatorname{span}\{v_1, \ldots, v_{k+1}\} \neq \{0\}.$$

Es sei nun z ein Vektor in diesem Durchschnitt mit $\|z\|_2 = 1$. Wegen $Bz = 0$ und

$$Az = \sum_{i=1}^{r} \sigma_i u_i (v_i^\mathsf{T} z) = \sum_{i=1}^{k+1} \sigma_i u_i (v_i^\mathsf{T} z)$$

erhalten wir

$$\begin{aligned}
\|A-B\|_{2\to 2}^2 &\geqslant \|(A-B)z\|_{2\to 2}^2 \quad \text{siehe Lemma 2.3}\\
&= \|Az\|_{2\to 2}^2 \quad \text{wegen } Bz = 0\\
&= \Big\|\sum_{i=1}^{k+1} \sigma_i\, u_i\, (v_i^\mathsf{T} z)\Big\|_2^2\\
&= \sum_{i=1}^{k+1} \sigma_i^2\, (v_i^\mathsf{T} z)^2 \quad \text{wegen der Orthonormalität der Vektoren } u_i\\
&\geqslant \sigma_{k+1}^2 \sum_{i=1}^{k+1} (v_i^\mathsf{T} z)^2 \quad \text{wegen } \sigma_1 \geqslant \cdots \geqslant \sigma_{k+1}\\
&= \sigma_{k+1}^2 \quad \text{wegen } 1 = \|z\|_2^2 = \Big\|\sum_{i=1}^{k+1} (v_i^\mathsf{T} z)\, v_i\Big\|_2^2 = \sum_{i=1}^{k+1} (v_i^\mathsf{T} z)^2.
\end{aligned}$$

□

Dieses Resultat zeigt, dass die führenden Terme in einer Singulärwertzerlegung die beste Approximation der gegebenen Matrix im Sinne der $2 \to 2$-Norm unter allen Matrizen mit gegebenem Rang ~~Rank~~[GM] darstellen. Der dabei gemachte Approximationsfehler ist gerade der erste Singulärwert, der nicht berücksichtigt wird.

Folgerung 4.12 (Abstand zu Matrizen mit Rangdefekt)**.** *Es sei $A \in \mathbb{R}^{m\times n}$ eine Matrix und $p = \min\{m, n\}$. Dann gilt*

$$\min\big\{\|A-B\|_{2\to 2} \,\big|\, B \in \mathbb{R}^{m\times n},\ \operatorname{Rang}(B) < p\big\} = \sigma_p > 0. \tag{4.16}$$

Beweis. □

Der kleinste Singulärwert gibt also den Abstand einer Matrix A zur Menge der Matrizen gleicher Dimensionen an, die nicht vollen Rang haben.

Ende der Woche 3

Kapitel 2 Zahldarstellung und Rechnerarithmetik

§ 5 Zahldarstellung

Computer rechnen mit bestimmten Zahldarstellungen. Man unterscheidet zwischen **Ganzzahlen** (englisch: *integer numbers*), **Festkommazahlen** (englisch: *fixed-point numbers*) und **Fließkommazahlen** (englisch: *floating-point numbers*). Da sich die numerische Mathematik mit Aufgaben der «kontinuierlichen Mathematik» beschäftigt, sind Ganzzahlen i. d. R. nur als Zähler usw. interessant. Wir behandeln deshalb jetzt hier vorrangig Fließkommazahlen.

Definition 5.1 (Fließkommazahl, vgl. Bornemann, 2018, Kapitel 12.1)**.** *Eine **Fließkommazahl** zur **Basis** (englisch:* base, radix*) $\beta \in \{2, 3, \ldots\} \subset \mathbb{N}$ ist eine Zahl der Form*

$$m\,\beta^e. \tag{5.1}$$

Dabei heißt

$$m = \pm\big((m_0.m_{-1}\cdots m_{1-r})\big)_\beta = \pm\big(m_0\,\beta^0 + m_{-1}\,\beta^{-1} + \cdots + m_{1-r}\,\beta^{1-r}\big) \tag{5.2}$$

*mit Ziffern $m_0, \ldots, m_{1-r} \in \{0, 1, \ldots, \beta - 1\}$ die (rationale) **Mantisse** (englisch:* significand, mantissa*) der Zahl und*

$$e = \pm\big(e_{s-1}e_{s-2}\cdots e_0\big)_\beta = \pm\big(e_{s-1}\,\beta^{s-1} + e_{s-2}\,\beta^{s-2} + \cdots + e_1\,\beta^1 + e_0\,\beta^0\big) \tag{5.3}$$

*mit Ziffern $e_{s-1}, \ldots, e_0 \in \{0, 1, \ldots, \beta - 1\}$ ihr (ganzzahliger) **Exponent** (englisch:* exponent*). Es gilt die Konvention, dass die führende Stelle der Mantisse $m_0 \neq 0$ ist.[1] Die einzige Ausnahme ist die Zahl 0, die durch die Mantisse $m = 0$ und beliebigen Exponenten e dargestellt wird.*

*Wir bezeichnen die Menge der darstellbaren Fließkommazahlen (zu gegebener Basis, Mantissenlänge und Wertebereich für den Exponenten) mit $\mathbb{F}$. Man spricht manchmal auch von einem **Fließkommagitter** (englisch:* floating point grid*).*

Die Notation $(\cdot)_\beta$ in (5.2) und (5.3) bedeutet, dass die Zahl zur Basis β notiert ist. Im Fall der Basis $\beta = 10$ lässt man den tiefgestellten Index $(\cdot)_{10}$ häufig weg.

Beachte: Die Fließkommazahlen $\mathbb{F}$ sind eine Teilmenge der rationalen Zahlen $\mathbb{Q}$. (**Quizfrage:** Klar?)

Mit Hilfe von Fließkommazahlen lassen sich Zahlen sehr unterschiedlicher Größenordnung darstellen, etwa (zur Basis $\beta = 10$)

[1] Zur Verdeutlichung kann man dann auch von einer **normalisierten Fließkommazahl** sprechen.

- die Ruhemasse eines Elektrons, etwa $9.109\,383 \cdot 10^{-31}\,\mathrm{kg}$,
- die Lichtgeschwindigkeit im Vakuum, etwa $2.997\,924\,58 \cdot 10^{8}\,\mathrm{m\,s^{-1}}$,
- die Avogrado-Konstante, etwa $6.022\,140\,76 \cdot 10^{23}\,\mathrm{mol}$.

Wie oben bereits bemerkt: Wird eine Zahl nicht zur Basis $\beta = 10$ notiert, so deutet man die Basis häufig durch einen tiefgestellten Index an. Beispielsweise ist

$$1001_2 = 9 \quad \text{und} \quad 1001.11_2 = 9 + \frac{1}{2} + \frac{1}{4} = 9.75.$$

Die Anzahl r der Stellen der Mantisse sowie die Anzahl s der Stellen des Exponenten in einem System von Fließkommazahlen sind in der Regel fest. Die Menge der damit darstellbaren Zahlen ist somit endlich.

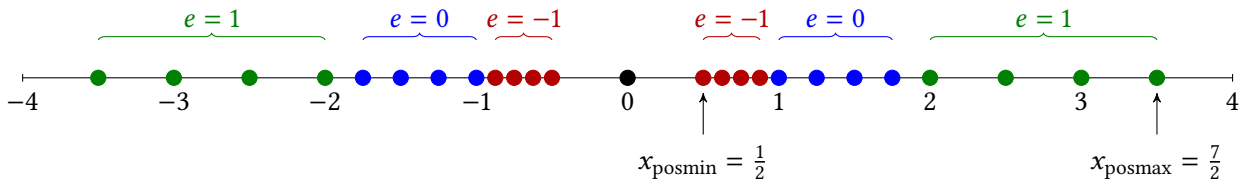
Beispiel 5.2 (Darstellbare Fließkommazahlen). *Im Fall der Basis $\beta = 2$, Mantissenlänge $r = 3$ und Exponentenlänge $s = 1$ sind die folgenden positiven Mantissen möglich:*

$$\begin{aligned} m = 1.00_2 &= \frac{4}{4}, \\ m = 1.01_2 &= \frac{5}{4}, \\ m = 1.10_2 &= \frac{6}{4}, \\ m = 1.11_2 &= \frac{7}{4}. \end{aligned}$$

Durch Kombination mit den zur Verfügung stehenden Exponenten $e \in \{-1, 0, 1\}$, also den Faktoren 2^{-1}, 2^0 und 2^1, sind also die positiven Zahlen

$$\left\{ \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, \ \frac{4}{4}, \frac{5}{4}, \frac{6}{4}, \frac{7}{4}, \ \frac{4}{2}, \frac{5}{2}, \frac{6}{2}, \frac{7}{2} \right\}$$

*darstellbar, dazu noch ihre negativen Gegenstücke sowie schließlich die Zahl 0. (**Quizfrage:** Ist die Darstellung einer Fließkommazahl immer eindeutig?)*



Wie das Beispiel 5.2 zeigt, sind Fließkommazahlen nicht gleichmäßig verteilt. Weiterhin gibt es eine kleinste und eine größte positive Zahl in $\mathbb{F}$. Die kleinste positive Zahl x_{posmin} wird dargestellt durch die Mantisse $m = (1.0\cdots0)_\beta = 1$ und den Exponenten

$$e = -\big((\beta-1)(\beta-1)\cdots(\beta-1)\big)_\beta = -(\beta-1)\sum_{k=0}^{s-1}\beta^k = -(\beta-1)\frac{\beta^s-1}{\beta-1} = 1-\beta^s.$$

Es ergibt sich also

$$x_{\text{posmin}} = \beta^{1-\beta^s} \tag{5.4}$$

Die größte positive Zahl x_{posmax} dagegen wird dargestellt durch die Mantisse

$$m = \big((\beta-1).(\beta-1)\cdots(\beta-1)\big)_\beta = (\beta-1)\sum_{k=1-r}^{0}\beta^k = (\beta-1)\beta^{1-r}\sum_{k=0}^{r-1}\beta^k = (\beta-1)\beta^{1-r}\frac{\beta^r-1}{\beta-1} = \beta^{1-r}(\beta^r-1)$$

und den Exponenten

$$e = \big((\beta-1)(\beta-1)\cdots(\beta-1)\big)_\beta = (\beta-1)\sum_{k=0}^{s-1}\beta^k = (\beta-1)\frac{\beta^s-1}{\beta-1} = \beta^s-1.$$

Daraus ergibt sich

$$x_{\text{posmax}} = \beta^{1-r}(\beta^r-1)\beta^{\beta^s-1} = (\beta-\beta^{1-r})\beta^{\beta^s-1}. \tag{5.5}$$

Im Beispiel 5.2 etwa mit $\beta = 2$, $r = 3$ und $s = 1$ gilt

$$x_{\text{posmin}} = \beta^{1-\beta^s} = 2^{1-2} = \frac{1}{2},$$
$$x_{\text{posmax}} = (\beta-\beta^{1-r})\beta^{\beta^s-1} = (2-2^{1-3})2^{2^1-1} = \frac{7}{4}\cdot 2^1 = \frac{7}{2}.$$

Das stimmt mit unseren Beobachtungen in Beispiel 5.2 überein.

Definition 5.3 (Zulässiger Bereich)**.** *Die Menge*

$$\mathbb{D} := [-x_{\text{posmax}}, -x_{\text{posmin}}] \cup \{0\} \cup [x_{\text{posmin}}, x_{\text{posmax}}] \subset \mathbb{R} \tag{5.6}$$

heißt der ***zulässige Bereich*** *eines Fließkommagitters $\mathbb{F}$.*

Quizfrage: Was ist die Bedeutung des zulässigen Bereichs?

Definition 5.4 (IEEE[2]-Standard 754)**.** *Der 1985 eingeführte IEEE-Standard 754 ist auch heute noch die Grundlage der Darstellung von Fließommazahlen auf sehr vielen Computerarchitekturen. In diesem Standard spielt das Format* ***double precision*** *(auch:* ***binary64****) die wichtigste Rolle. Es verwendet die Basis $\beta = 2$, Mantissenlänge $r = 53$ und ganzzahlige Exponenten $e \in [-1022, 1023]$. Daraus können wir ähnlich wie oben die kleinste und größte positive darstellbare (normalisierte) Fließkommazahl herleiten:*

$$x_{\text{posmin}} = 1\cdot 2^{-1022} = 2^{-1022} \approx 2.2\cdot 10^{-308},$$
$$x_{\text{posmax}} = 2^{1-53}(2^{53}-1)\cdot 2^{1023} = (2-2^{-52})\cdot 2^{1023} \approx 1.8\cdot 10^{308}.$$

An Stelle eines vorzeichenbehafteten Exponenten e wird dieser als $e = c - 1023$ interpretiert, wobei die Zahl $c \in [0, 2047]$ mit 11 Bits gespeichert wird. Da die Extremfälle $c = 0$ und $c = 2047$ besondere Bedeutung haben, ergeben sich die nutzbaren Werte $e \in [-1022, 1023]$ für den Exponenten. Da die Mantisse normalisiert ist, ist die führende Ziffer für Fließkommazahlen $x \neq 0$ immer $m_0 = 1$ und muss nicht gespeichert werden. Zusammen mit dem Vorzeichen der Mantisse benötigt eine Zahl nach dem double precision*-Standard daher $1 + 52 + 11 = 64$ Bits oder 8 Bytes an Speicher.*

Zusätzlich definiert der IEEE-Standard die erweiterten Fließkommazahlen $+\infty$, $-\infty$ und `NaN` *(«not a number»), die mit Hilfe von $c = 2047$ kodiert werden.*

[2] *Institute of Electrical and Electronics Engineers*

Definition 5.5 (Rundung). *Es sei $\mathbb{F}$ eine Menge von Fließkommazahlen. Eine Funktion* $\mathrm{rd}\colon \mathbb{R}\cup\{\pm\infty\} \to \mathbb{F}\cup\{\pm\infty\}$ *heißt* ***Rundung****, wenn sie folgende Eigenschaften erfüllt:*

(*i*) *Sie ist monoton, d. h.,*

$$\mathrm{rd}(x) \leqslant \mathrm{rd}(y) \quad \text{für } x, y \in \mathbb{R}\cup\{\pm\infty\} \text{ mit } x \leqslant y.$$

(*ii*) *Sie ist idempotent, d. h.,*

$$\mathrm{rd}(x) = x \quad \text{für } x \in \mathbb{F}\cup\{\pm\infty\}.$$

Quizfrage: Wofür könnten die beiden Eigenschaften der Rundung nützlich sein?

Wir betrachten hier nur die gängstige Strategie des IEEE-Standards, die **Rundung zur nächstgelegenen Fließkommazahl** (englisch: *round to nearest*). Diese Funktion bildet Zahlen im Bereich $[-x_{\mathrm{posmax}}, x_{\mathrm{posmax}}]$ auf die jeweils nächstgelegene Zahl in $\mathbb{F}$ ab, die bis auf endlich viele Ausnahmewerte für x eindeutig ist. Die Ausnahmewerte sind gerade diejenigen Zahlen x, die genau in der Mitte zwischen zwei benachbarten Zahlen in $\mathbb{F}$ liegen. Diese Grenzfälle werden bei der Unterstrategie *round to nearest, ties to even* so gerundet, dass das niederwertigste Bit der Mantisse = 0 wird. Weiterhin werden Zahlen $|x| > x_{\mathrm{posmax}} + C$ auf $+\infty$ bzw. $-\infty$ gerundet, wobei $C = \frac{1}{2}0.0\cdots01_\beta \cdot \beta^E$ mit dem maximalen Exponenten E ist. **Quizfrage:** Was könnte der Grund für diese Regel sein? (Diese Frage kann erst nach Lemma 5.6 beantwortet werden.)

Wenn wir in Zukunft von Rundung sprechen, so ist immer diese Strategie gemeint.

Lemma 5.6 (Relativer Fehler bei der Rundung). *Der relative Fehler bei der Rundung in einem System von Fließkommazahlen $\mathbb{F}$ mit Mantissenlänge r erfüllt die Abschätzung*

$$\frac{|x - \mathrm{rd}(x)|}{|x|} \leqslant \varepsilon_{\mathrm{mach}} := \frac{1}{2}\beta^{1-r} \tag{5.7}$$

für alle Zahlen $x \neq 0$, die im zulässigen Bereich $\mathbb{D}$ liegen. Dabei ist $\varepsilon_{\mathrm{mach}}$ die sogenannte ***Maschinengenauigkeit*** *(englisch:* machine precision, unit roundoff*).*

Beweis. Wir müssen nur den Fall $x > 0$ betrachten, da der Beweis im Fall $x < 0$ analog läuft. Aufgrund der Annahme an x existieren zwei benachbarte Zahlen $\underline{x}$ und $\overline{x}$ in $\mathbb{F}$ mit der Eigenschaft $0 < \underline{x} \leqslant x < \overline{x}$. Nach Definition der Rundung gilt

$$|x - \mathrm{rd}(x)| \leqslant \frac{1}{2}(\overline{x} - \underline{x}).$$

Aus der Darstellung (4.2)–(5.3) sieht man, dass sich $\underline{x}$ und $\overline{x}$ um die Zahl

$$(0.00\cdots01)_\beta \cdot \beta^e = \beta^{1-r}\cdot\beta^e$$

unterscheiden, wobei e der Exponent von $\underline{x}$ ist. (**Quizfrage:** Warum stimmt das auch dann, wenn $\underline{x} = \big((\beta-1).(\beta-1)(\beta-1)\cdots(\beta-1)\big)_\beta \cdot \beta^e$ ist?)

Es gilt also

$$|x - \mathrm{rd}(x)| \leqslant \frac{1}{2}\beta^{1-r} \cdot \beta^e$$

und

$$x \geqslant \underline{x} \geqslant (1.00\cdots00)_\beta \cdot \beta^e,$$

also

$$\frac{|x - \mathrm{rd}(x)|}{|x|} \leqslant \frac{1}{2}\beta^{1-r} \cdot \beta^e \cdot \frac{1}{\beta^e} = \frac{1}{2}\beta^{1-r}$$

wie behauptet. □

Quizfrage: Gilt die Abschätzung (5.7) auch für Zahlen x mit $|x| < x_{\mathrm{posmin}}$? Falls nicht, wie groß kann der relative Fehler bei der Rundung für diese Zahlen höchstens werden?

Folgerung 5.7 (zur Maschinengenauigkeit). *Für alle $x \in \mathbb{D}$ (einschließlich $x = 0$) gilt*

$$\mathrm{rd}(x) = x\,(1 + \varepsilon) \tag{5.8}$$

mit einer Zahl $\varepsilon \in \mathbb{R}$, die $|\varepsilon| \leqslant \varepsilon_{\mathrm{mach}}$ erfüllt.

Beispiel 5.8 (Maschinengenauigkeit).

(*i*) *Für das Fließkommagitter aus Beispiel 5.2 gilt*

$$\varepsilon_{\mathrm{mach}} = \frac{1}{2}2^{1-3} = 2^{-3} = \frac{1}{8}.$$

(*ii*) *Im Falle des* double precision*-Standards haben wir*

$$\varepsilon_{\mathrm{mach}} = \frac{1}{2}2^{1-53} = 2^{-53} \approx 1.11 \cdot 10^{-16}.$$

(*iii*) *Im Zehnersystem ($\beta = 10$) mit zwei Nachkommastellen, also für Zahlen der Bauart* 1.23 *mit Mantissenlänge $r = 3$, ergibt sich die «Maschinen»genauigkeit $\varepsilon_{\mathrm{mach}} = \frac{1}{2}10^{1-3} = 0.005$.*

Bemerkung 5.9 (Maschinengenauigkeit).

(a) Die Maschinengenauigkeit ist gerade die Hälfte des Stellenwertes der letzten Ziffer der Mantisse; vgl. (5.2). Auf die darstellbaren Exponenten kommt es dabei nicht an.

(b) Die doppelte Fließkommagenauigkeit $2\varepsilon_{\mathrm{mach}}$ ist gerade der Abstand zwischen der Fließkommazahl 1 *und der nächst größeren Fließkommazahl.*

Bemerkung 5.10 (Alternative Definition der Maschinengenauigkeit). *Manche Autoren definieren die Maschinengenauigkeit als das Doppelte von unserer Definition. Das liegt darin begründet, dass (5.7) dann auch für andere Rundungsarten als* round to nearest *gilt, etwa* round toward 0. *Siehe auch die Diskussion auf* https://en.wikipedia.org/wiki/Machine_epsilon.

§ 6 Rechnerarithmetik

Die Grundoperationen +, −, · und / liefern, selbst für Argumente in $\mathbb{F}$, im Allgemeinen Ergebnisse, die keine Fließkommazahlen in $\mathbb{F}$ sind. (**Quizfrage:** Können Sie Beispiele dafür angeben, wann das der Fall ist, etwa für das Fließkommasystem $\mathbb{F}$ aus Beispiel 5.2?) Die Grundoperationen lassen sich also nicht fehlerfrei in $\mathbb{F}$ ausführen. Stattdessen werden Ersatzoperationen, genannt **Maschinenoperationen** $\oplus$, $\ominus$, $\odot$ und $\oslash$ implementiert (in Hardware), die Ergebnisse in $\mathbb{F}$ liefern. Der IEEE-Standard sieht vor, dass diese Grundoperationen mit korrekter Rundung ausgeführt werden. Das heißt konkret, dass folgende Beziehungen gelten:

$$x \oplus y = \mathrm{rd}(x + y), \tag{6.1a}$$
$$x \ominus y = \mathrm{rd}(x - y), \tag{6.1b}$$
$$x \odot y = \mathrm{rd}(x \cdot y), \tag{6.1c}$$
$$x \oslash y = \mathrm{rd}(x / y). \tag{6.1d}$$

Dabei sind jeweils $x, y \in \mathbb{F}$, und das Ergebnis liegt in $\mathbb{F} \cup \{\pm\infty\}$ bzw. ergibt `NaN` im Falle von $0 \oslash 0$. Aus Lemma 5.6 bzw. Folgerung 5.7 ergibt sich direkt folgendes Resultat:

Satz 6.1 (Relativer Fehler der Maschinenoperationen $\oplus$, $\ominus$, $\odot$ und $\oslash$)**.** *Es seien $x, y \in \mathbb{F}$.*

(*i*) *Falls $x + y \in \mathbb{D}$ liegt, so gilt mit einer Zahl $\varepsilon \in \mathbb{R}$, $|\varepsilon| \leqslant \varepsilon_{\mathrm{mach}}$:*

$$x \oplus y = (x + y)(1 + \varepsilon).$$

(*ii*) *Falls $x - y \in \mathbb{D}$ liegt, so gilt mit einer Zahl $\varepsilon \in \mathbb{R}$, $|\varepsilon| \leqslant \varepsilon_{\mathrm{mach}}$:*

$$x \ominus y = (x - y)(1 + \varepsilon).$$

(*iii*) *Falls $x \cdot y \in \mathbb{D}$ liegt, so gilt mit einer Zahl $\varepsilon \in \mathbb{R}$, $|\varepsilon| \leqslant \varepsilon_{\mathrm{mach}}$:*

$$x \odot y = (x \cdot y)(1 + \varepsilon).$$

(*iv*) *Falls $y \neq 0$ ist und $x/y \in \mathbb{D}$ liegt, so gilt mit einer Zahl $\varepsilon \in \mathbb{R}$, $|\varepsilon| \leqslant \varepsilon_{\mathrm{mach}}$:*

$$x \oslash y = (x / y)(1 + \varepsilon).$$

Beachte: Die Voraussetzung $x + y \in \mathbb{D}$ besagt, dass das korrekte Resultat im Bereich der von den zur Verfügung stehenden Fließkommazahlen überdeckten Intervalle (5.6) liegt.

Wie passt die Aussage von Satz 6.1, dass $x \ominus y$ nur um einen kleinen relativen Faktor vom echten Ergebnis $x - y$ abweicht, mit der in Abschnitt 3 gemachten Beobachtung zusammen, dass die Aufgabe, $x - y$ zu berechnen, eine beliebig große relative Konditionszahl haben kann?

Das sind zwei verschiedene Paar Schuhe. Die Aussage von Satz 6.1 bezieht sich auf die Berechnung von $x \ominus y$, wobei x und y bereits Fließkommazahlen sind und keiner Rundung unterliegen. (Eine Rundung erfolgt erst nach der Berechnung von $x - y$.) Die Aussage zur Kondition bezieht sich darauf, dass das Ergebnis von $x - y$ stark abweichen kann, wenn die Daten x und y nicht exakt sind, etwa durch Rundung vor der Berechnung, um x und y in Fließkommazahlen zu verwandeln. Allerdings bestehen Algorithmen in der Regel aus einer Kette von Operationen, bei denen zwischendurch immer wieder Rundungen auf Fließkommazahlen stattfinden. Dadurch werden «früh» im Verfahren gemachte Rundungsfehler in allen späteren Schritten sichtbar, wobei die Konditionszahlen als (Verstärkungs-) Faktoren eingehen. Das untersuchen wir genauer in Kapitel 3.

Bemerkung 6.2 (Assoziativ- und Distributivgesetz)**.**

(a) Das Assoziativgesetz und das Distributivgesetz gelten für die Maschinenoperationen nicht mehr! Es ist also i. A.

$$(x \oplus y) \oplus z \neq x \oplus (y \oplus z),$$
$$(x \oplus y) \odot z \neq (x \odot z) \oplus (y \odot z)$$

für $x, y, z \in \mathbb{F}$.

(b) Das Kommutativgesetz der Addition und der Multiplikation gelten für $\oplus$ und $\odot$ aber weiter, d. h.,

$$x \oplus y = y \oplus x,$$
$$x \odot y = y \odot x$$

für $x, y \in \mathbb{F}$.

(c) Diese Beobachtung führt dazu, dass man Algorithmen, bevor man den Einfluss von Rundungsfehlern untersucht, zweifelsfrei notieren muss, weil bereits die Reihenfolge einen Einfluss auf das Ergebnis haben kann. Dabei helfen Klammersetzung und Einführung von Zwischenresultaten. Beispielsweise sind zur Auswertung von $y = x_1^2 - x_2^2 = (x_1 + x_2)(x_1 - x_2)$ folgende Realisierungen in Algorithmen denkbar:

$u_1 \coloneqq x_1 \odot x_1$	$u_1 \coloneqq x_1 \oplus x_2$	$u_1 \coloneqq x_1 \ominus x_2$
$u_2 \coloneqq x_2 \odot x_2$	$u_2 \coloneqq x_1 \ominus x_2$	$u_2 \coloneqq x_1 \oplus x_2$
$y \coloneqq u_1 \ominus u_2$	$y \coloneqq u_1 \odot u_2$	$y \coloneqq u_1 \odot u_2$

*Mit den Operationen $+, -, \cdot$ in $\mathbb{R}$ führen diese Algorithmen zu identischen Ergebnissen, in Maschinenarithmetik mit $\oplus, \ominus, \odot$ sind die zweite und dritte Variante identisch (**Quizfrage:** Warum?), aber verschieden von der ersten.*

Ende der Woche 4

Kapitel 3 Stabilität von Algorithmen

§ 7 Stabilitätsanalyse

Bei der Stabilitätsanalyse von Algorithmen geht es darum, zu beurteilen, welchen Einfluss die in einer konkreten Implementierung mit Maschinenoperationen unvermeidbar auftretenden Rundungsfehler auf das Endergebnis haben. Diese Beurteilung kann man auf unterschiedliche Arten vornehmen, was zu verschiedenen Stabilitätsbegriffen führt. Wir beginnen mit dem Konzept im Sinne der Vorwärtsanalyse.

§ 7.1 Vorwärtsanalyse

Wir bezeichnen mit

$F(x)$ die auszuwertende Funktion,

$\widehat{F}(x)$ die durch den implementierten Algorithmus realisierte Funktion.

Wir beginnen mit einem einfachen Beispiel.

Beispiel 7.1 (Fehlerfortpflanzung bei $(a+b)+c$)**.** *Wir wollen $y = F(x) = a+b+c$ für $x = (a,b,c)^\mathsf{T} \in \mathbb{F}^3$ auswerten. Wir gehen hier zunächst der Einfachheit halber davon aus, dass die Summanden bereits als Fließkommazahlen vorliegen. Wir betrachten den Algorithmus $\widehat{F}(x) = ((a \oplus b) \oplus c)$. Um Fehlerabschätzungen zu erhalten, gehen wir davon aus, dass die Zwischenergebnisse (vor der Rundung) $a+b$ sowie $(a \oplus b) + c$ im zulässigen Bereich $\mathbb{D}$ liegen. Dann erhalten wir aus Satz 6.1 die Aussage, dass Zahlen $\varepsilon^{(1)}, \varepsilon^{(2)} \in \mathbb{R}$ mit $|\varepsilon_i| \leqslant \varepsilon_{\text{mach}}$ existieren, sodass gilt:*

$$\begin{aligned} a \oplus b &= (a+b)(1+\varepsilon^{(1)}) \\ (a \oplus b) \oplus c &= ((a \oplus b) + c)(1+\varepsilon^{(2)}) \\ &= ((a+b)(1+\varepsilon^{(1)}) + c)(1+\varepsilon^{(2)}) \\ &= (a+b+c)\left[1 + \frac{a+b}{a+b+c}\varepsilon^{(1)}(1+\varepsilon^{(2)}) + \varepsilon^{(2)}\right]. \end{aligned}$$

Für den relativen Fehler gilt daher

$$\frac{\widehat{F}(x) - F(x)}{F(x)} = \frac{(a \oplus b) \oplus c - (a+b+c)}{a+b+c} = \frac{a+b}{a+b+c}\varepsilon^{(1)}(1+\varepsilon^{(2)}) + \varepsilon^{(2)}.$$

Unter Vernachlässigung des gegenüber $\varepsilon^{(i)}$ typischerweise kleinen Terms $\varepsilon^{(1)}\varepsilon^{(2)}$ vereinfacht sich dieser Ausdruck zu

$$\frac{(a \oplus b) \oplus c - (a+b+c)}{a+b+c} \doteq \frac{a+b}{a+b+c}\varepsilon^{(1)} + \varepsilon^{(2)}. \tag{7.1}$$

Da wir die Fehlerbeiträge als voneinander unabhängig betrachten wollen, verwenden wir die Dreiecksungleichung und erhalten schließlich

$$\frac{|(a \oplus b) \oplus c - (a+b+c)|}{|a+b+c|} \dot{\leqslant} \frac{|a+b|}{|a+b+c|}|\varepsilon^{(1)}| + |\varepsilon^{(2)}|. \tag{7.2}$$

*Wir sehen also hier, dass der erste (früher) auftretende relative Rundungsfehler $\varepsilon^{(1)}$ mit dem Faktor $\frac{|a+b|}{|a+b+c|}$ fortgepflanzt wird. Dieser Faktor ist der Betrag der relativen partiellen Konditionszahl k_{11} der Additionsaufgabe $(a+b)+c$, vgl. Lemma 3.9. (**Quizfrage:** Warum ist hier nur die eine relative partielle Konditionszahl k_{11} relevant, während die andere k_{12} nicht vorkommt?) Der zweite (später) auftretende Rundungsfehler $\varepsilon^{(2)}$ taucht dagegen ohne einen solchen Faktor in der Abschätzung auf. **Quizfrage:** Warum ist das so?*

***Quizfrage:** Wie sieht im Vergleich zu (7.1) die Analyse für den alternativen Algorithmus $\widehat{F}(x) = (a \oplus (b \oplus c))$ aus?*

Das Beispiel 7.1 zeigt bereits wesentliche Bestandteile der Stabilitätsanalyse eines Algorithmus im Vorwärtssinne. In jedem Schritt kommen neue Rundungsfehler hinzu, und bereits vorhandene Fehler werden — mit der entsprechenden relativen Konditionszahl multipliziert — weitergegeben. Wir wollen dies nun allgemein analysieren. Dazu betrachten wir eine auszuwertende Funktion $F\colon \mathbb{R}^n \to \mathbb{R}^m$ und einen Algorithmus, der aus der Eingabe $x =: x^{(0)} \in \mathbb{R}^{n^{(0)}}$ ein erstes Zwischenergebnis $x^{(1)} := F^{(0)}(x^{(0)}) \in \mathbb{R}^{n^{(1)}}$ erzeugt, anschließend ein weiteres $x^{(2)} := F^{(1)}(x^{(1)}) \in \mathbb{R}^{n^{(2)}}$ usw., bis schließlich das Endergebnis

$$y := x^{(N+1)} := F^{(N)}(x^{(N)}) \in \mathbb{R}^{n^{(N+1)}}$$

vorliegt. Für die erste und letzte Dimension gilt $n^{(0)} = n$ und $n^{(N+1)} = m$. Die durch den Algorithmus realisierte Funktion wird also in Teilabbildungen zerlegt:

$$F = F^{(N)} \circ \cdots \circ F^{(0)}.$$

Dabei bestehen die Komponenten der vektorwertigen Teilabbildungen $F^{(k)}$ jeweils aus elementaren Abbildungen wie $+$, $\cdot$, cos etc..

Demgegenüber steht in der tatsächlichen Implementierung des Verfahrens die Realisierung der Teilabbildungen durch Maschinenoperationen $\widehat{F}^{(k)}$. Dabei nehmen wir an, dass (komponentenweise)

$$\widehat{F}^{(k)} := \mathrm{rd}\big(F^{(k)}\big) \tag{7.3}$$

gilt, d. h., die implementierten Operationen sind bis auf die nachfolgende Rundung exakt.

Beispiel 7.2. *Für den «Algorithmus* $(a+b)+c$*» aus Beispiel 7.1 gilt*

$$x^{(0)} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \overset{F^{(0)}}{\longmapsto} \begin{pmatrix} a+b \\ c \end{pmatrix} \overset{F^{(1)}}{\longmapsto} (a+b)+c = x^{(2)} =: y \quad \textit{und}$$

$$\widehat{x}^{(0)} = \begin{pmatrix} \widehat{a} \\ \widehat{b} \\ \widehat{c} \end{pmatrix} \overset{\widehat{F}^{(0)}}{\longmapsto} \begin{pmatrix} \widehat{a} \oplus \widehat{b} \\ \widehat{c} \end{pmatrix} \overset{\widehat{F}^{(1)}}{\longmapsto} (\widehat{a} \oplus \widehat{b}) \oplus \widehat{c} = x^{(2)} =: \widehat{y}.$$

Wir untersuchen jetzt den Einfluss der Rundungsfehler, also den Unterschied zwischen

$$y = (F^{(N)} \circ \cdots \circ F^{(0)})(x^{(0)}) \quad \text{und}$$
$$\widehat{y} = (\widehat{F}^{(N)} \circ \cdots \circ \widehat{F}^{(0)})(\widehat{x}^{(0)})$$

Wir lassen im Gegensatz zu unserem anfänglichen Beispiel 7.1 nun auch (Rundungs-)fehler in der Eingabe zu, die zu einem anfänglichen Fehler

$$\Delta x^{(0)} := \widehat{x}^{(0)} - x^{(0)}$$

führen. Wegen Lemma 5.6 gilt dabei

$$\Delta x^{(0)} = \begin{bmatrix} \varepsilon_1^{(0)} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \varepsilon_{n^{(0)}}^{(0)} \end{bmatrix} x^{(0)}$$

mit Zahlen $\varepsilon_1^{(0)}, \ldots, \varepsilon_{n^{(0)}}^{(0)}$, die $|\varepsilon_j^{(0)}| \leqslant \varepsilon_{\text{mach}}$ erfüllen. Also gilt insbesondere $\|\Delta x^{(0)}\|_\infty \leqslant \varepsilon_{\text{mach}} \|x^{(0)}\|_\infty$.

Im nächsten Schritt erhalten wir die Zwischenergebnisse

$$x^{(1)} = F^{(0)}(x^{(0)}) \quad \text{und}$$
$$\widehat{x}^{(1)} = \widehat{F}^{(0)}(\widehat{x}^{(0)}).$$

Für den Fehler

$$\Delta x^{(1)} := \widehat{x}^{(1)} - x^{(1)}$$

gilt

$$\begin{aligned} \Delta x^{(1)} &= \widehat{F}^{(0)}(\widehat{x}^{(0)}) - F^{(0)}(x^{(0)}) \\ &= \left[\widehat{F}^{(0)}(\widehat{x}^{(0)}) - F^{(0)}(\widehat{x}^{(0)})\right] + \left[F^{(0)}(\widehat{x}^{(0)}) - F^{(0)}(x^{(0)})\right] \\ &= \underbrace{\left[\operatorname{rd}\big(F^{(0)}(\widehat{x}^{(0)})\big) - F^{(0)}(\widehat{x}^{(0)})\right]}_{\text{neu hinzukommender Rundungsfehler}} + \underbrace{\left[F^{(0)}(\widehat{x}^{(0)}) - F^{(0)}(x^{(0)})\right]}_{\text{Fortpflanzung des bereits vorhandenen Fehlers}}. \end{aligned} \tag{7.4}$$

Für den Term in der ersten Klammer gilt wegen Lemma 5.6:

$$\operatorname{rd}\big(F^{(0)}(\widehat{x}^{(0)})\big) - F^{(0)}(\widehat{x}^{(0)}) = \underbrace{\begin{bmatrix} \varepsilon_1^{(1)} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \varepsilon_{n^{(1)}}^{(1)} \end{bmatrix}}_{=:E^{(1)}} F^{(0)}(\widehat{x}^{(0)}) = E^{(1)} F^{(0)}(\widehat{x}^{(0)}) \tag{7.5}$$

mit Zahlen $\varepsilon_1^{(1)}, \dots, \varepsilon_{n^{(1)}}^{(1)}$, die $|\varepsilon_j^{(1)}| \leqslant \varepsilon_{\text{mach}}$ erfüllen. Nach dem Satz von Taylor 2.10 haben wir außerdem

$$F^{(0)}(\widehat{x}^{(0)}) \doteq F^{(0)}(x^{(0)}) \tag{7.6}$$

bis auf einen Fehlerterm, in den die Größe $\|\Delta x^{(0)}\|_\infty \leqslant \varepsilon_{\text{mach}} \|x^{(0)}\|_\infty$ eingeht sowie eine Schranke für die Ableitung von F in einer Umgebung von $x^{(0)}$. Wenn wir jetzt (7.6) in die rechte Seite von (7.5) einsetzen, so erhalten wir also schließlich

$$\operatorname{rd}\bigl(F^{(0)}(\widehat{x}^{(0)})\bigr) - F^{(0)}(\widehat{x}^{(0)}) \doteq E^{(1)} F^{(0)}(x^{(0)}) = E^{(1)} x^{(1)} \tag{7.7}$$

bis auf Terme, in die die komponentenweisen relativen Rundungsfehler $\varepsilon_j^{(1)}$ quadratisch eingehen.

Der zweite Term in (7.4) lässt sich nach dem Satz von Taylor 2.10 approximieren durch[1]

$$F^{(0)}(\widehat{x}^{(0)}) - F^{(0)}(x^{(0)}) \doteq DF^{(0)}(x^{(0)}) \bigl(\widehat{x}^{(0)} - x^{(0)}\bigr) = DF^{(0)}(x^{(0)})\, \Delta x^{(0)},$$

wobei die Restterme wiederum quadratisch in den relativen Rundungsfehlern $\varepsilon_j^{(1)}$ sind.

Zusammenfassend können wir also (7.4) in einer Approximation erster Ordnung schreiben als

$$\Delta x^{(1)} \doteq \overbrace{E^{(1)} x^{(1)}}^{\text{neu hinzukommender Rundungsfehler}} + \underbrace{DF^{(0)}(x^{(0)})\, \Delta x^{(0)}}_{\text{Fortpflanzung des bereits vorhandenen Fehlers}}. \tag{7.8}$$

Diese Beziehung gilt nicht nur für den ersten Schritt $\cdot^{(0)} \rightsquigarrow \cdot^{(1)}$ im Algorithmus, sondern auch für alle weiteren Schritte $\cdot^{(k)} \rightsquigarrow \cdot^{(k+1)}$, also

$$\Delta x^{(k+1)} \doteq E^{(k+1)} x^{(k+1)} + DF^{(k)}(x^{(k)})\, \Delta x^{(k)} \tag{7.9}$$

für $k = 0, \dots, N$. Durch Einsetzen erhalten wir schließlich für den absoluten Fehler $\Delta y = \Delta x^{(N+1)}$ in der Ausgabe die Beziehung

$$\begin{aligned} \Delta y \doteq\ & D(F^{(N)} \circ \cdots \circ F^{(0)})(x^{(0)})\, \Delta x \\ & + D(F^{(N)} \circ \cdots \circ F^{(1)})(x^{(1)})\, E^{(1)} x^{(1)} \\ & + \cdots \\ & + DF^{(N)}(x^{(N)})\, E^{(N)} x^{(N)} \\ & + E^{(N+1)} x^{(N+1)}, \end{aligned} \tag{7.10}$$

wobei wie eingangs gesagt $\Delta x := \Delta x^{(0)}$ den Fehler in der Eingabe bezeichnet. (**Quizfrage:** Können Sie die einzelnen Zeilen in (7.10) interpretieren? Wo entstehen sie jeweils?) Wir können an der Gleichung (7.10) sehen, dass die Norm der Jacobimatrix der «Restabbildung» $F^{(N)} \circ \cdots \circ F^{(k)}$ eine obere Schranke für den Einfluss des im Schritt $\cdot^{(k-1)} \rightsquigarrow \cdot^{(k)}$ neu hinzukommenden Rundungsfehlers $E^{(k)} x^{(k)}$ auf den Fehler Δy im Endergebnis angibt.

[1] Wir schreiben hier jetzt $DF^{(0)}$ für die Ableitung (Jacobimatrix) von $F^{(0)}$, um Ausdrücke wie $(F^{(0)})'$ zu vermeiden.

Beispiel 7.3 (Fehlerfortpflanzung bei $(a+b)+c$). *Wir führen die Rechnung aus Beispiel 7.1 für den «Algorithmus $(a+b)+c$» mit der soeben entwickelten Technik nochmal durch. Wir definieren das Verfahren mit Hilfe der Zwischenergebnisse*

$$x = x^{(0)} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \quad x^{(1)} = F^{(0)}(x^{(0)}) = \begin{pmatrix} a+b \\ c \end{pmatrix} \quad \text{und} \quad y = x^{(2)} = F^{(1)}(x^{(1)}) = a+b+c.$$

Es gilt also $N = 1$ (Anzahl der Zwischenschritte), und wir benötigen die Teilabbildungen

$$\begin{aligned} F^{(0)}(x) &= \begin{pmatrix} x_1 + x_2 \\ x_3 \end{pmatrix} \quad \Rightarrow \quad DF^{(0)}(x) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ F^{(1)}(x) &= x_1 + x_2 \quad \Rightarrow \quad DF^{(1)}(x) = \begin{bmatrix} 1 & 1 \end{bmatrix}. \end{aligned}$$

Weiter benötigen wir die Jacobimatrix der Abbildung $F^{(1)} \circ F^{(0)}$, also

$$D(F^{(1)} \circ F^{(0)})(x) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

*Die Matrizen für die auf jeder Ebene entstehenden relativen Fehler sind $E^{(1)} = \begin{bmatrix} \varepsilon_1^{(1)} & 0 \\ 0 & \varepsilon_2^{(1)} \end{bmatrix}$ und $E^{(2)} = \varepsilon_1^{(2)}$. Für $\varepsilon_2^{(1)}$ dürfen wir sogar Null annehmen. (**Quizfrage:** Warum?) Somit ergibt sich für den absoluten Fehler im Ergebnis also wie in (7.10):*

$$\begin{aligned} \Delta y \doteq D(F^{(1)} \circ F^{(0)})(x^{(0)})\,\Delta x \quad &+ \quad DF^{(1)}(x^{(1)})\,E^{(1)}x^{(1)} \quad + \quad E^{(2)}x^{(2)} \\ = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \Delta x \quad &+ \quad \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_1^{(1)} & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} a+b \\ c \end{pmatrix} \quad + \quad \varepsilon_1^{(2)}(a+b+c) \\ = \Delta a + \Delta b + \Delta c \quad &+ \quad \varepsilon_1^{(1)}(a+b) \quad + \quad \varepsilon_1^{(2)}(a+b+c). \end{aligned}$$

In relativer Betrachtungsweise haben wir daher

$$\frac{\Delta y}{y} \doteq \frac{\Delta a + \Delta b + \Delta c}{a+b+c} + \frac{a+b}{a+b+c}\varepsilon_1^{(1)} + \varepsilon_1^{(2)}. \tag{7.11}$$

Quizfrage: Wodurch kommen in (7.11) die zusätzlichen Terme im Vergleich zu unserer früheren Rechnung (7.1) zustande?

Die oben besprochene Technik zur Analyse des Einflusses von Rundungsfehlern (7.10) heißt **(differentielle) Vorwärtsanalyse**, weil man die Eingabefehler und zwischendurch auftretende Rundungsfehler vorwärts durch das Verfahren transportiert und weil die Faktoren, die dabei auftreten, aus der Ableitung gewonnen werden.

Bemerkung 7.4 (zur differentiellen Vorwärtsanalyse).

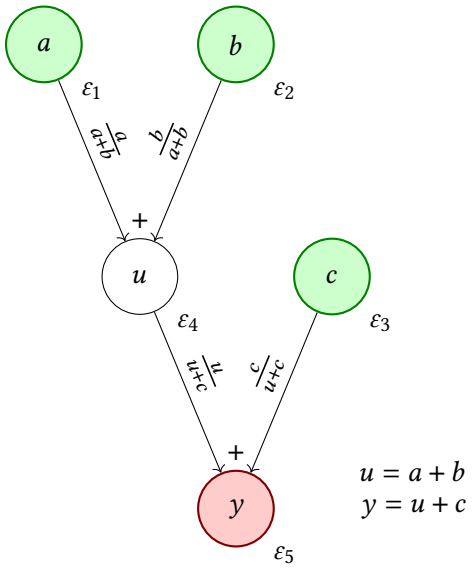
(1) Die Vorwärtsanalyse setzt natürlich voraus, dass alle Teilabbildungen $F^{(k)}$ und deren Realisierungen durch Maschinenoperationen $\widehat{F}^{(k)}$ überall definiert sind, wo sie benötigt werden.

(2) *Wie man bereits an einfachen Beispielen wie Beispiel 7.3 sieht, führen verschiedene Realisierungen mathematisch äquivalenter Ausdrücke wie* $(a + b) + c$ *und* $a + (b + c)$ *zu unterschiedlichen Abschätzungen für die Fehlerfortpflanzung!*

(3) *Die Faktoren in den Abschätzungen der Fehlerfortpflanzung wie beispielsweise (7.11) hängen von der Auswertungsstelle* x *ab.*

(4) *Für die Einträge* $\varepsilon_j^{(k)}$ *der Matrizen der relativen Fehler gilt die Abschätzung* $|\varepsilon_j^{(k)}| \leqslant \varepsilon_{\text{mach}}$*, zumindest dann, wenn die Zwischenergebnisse* $F^{(k)}(\widehat{x}^{(k)})$ *vor der Rundung komponentenweise im Bereich* $\mathbb{D}$ *liegen; siehe Lemma 5.6.*

Die Notation der Vorwärtsanalyse mit Hilfe von Jacobimatrizen wird schnell unübersichtlich. Einfacher ist es, zu einer grafischen Notation überzugehen, die auf Bauer, 1974 zurückgeht. Dies führen wir jetzt an der Aufgabe aus Beispiel 7.3 vor. Dazu stellen wir einen **Berechnungsgraphen** (englisch: *computational graph*) auf. Die Eingaben, Zwischenergebnisse und das Endergebnis[2] sind die Knoten dieses Graphen. Knoten, die Eingaben bzw. Ausgaben darstellen, sind farbig markiert. Die an einem Knoten eingehenden Kanten entsprechen einer unären (wie $\sqrt{a}$) oder binären (wie $a \cdot b$) Operation:



Die in jedem Knoten entstehenden relativen Rundungsfehler notieren wir in der Form ε_1 etc. An die Kanten schreiben wir die relativen partiellen Konditionszahlen der jeweiligen Operation. Diese geben die Faktoren an, mit denen die entstehenden Rundungsfehler jeweils weitergegeben werden. Mit ε_u usw. bezeichnen wir den relativen Fehler des jeweiligen Zwischenergebnisses.

Mit Hilfe eines solchen Graphen können wir folgende Arten von Fragen beantworten:

(1) Welchen Einfluss hat der Fehler in einem Knoten i auf einen später im Graph folgenden Knoten j?

[2] Im Falle einer vektorwertigen Ausgabe gibt es mehrere Ergebnisknoten.

(2) Wie hängt der Fehler in einem Knoten j von den Fehlern in allen früher im Graph vorkommenden Knoten i ab?

Frage (1) können wir beantworten, indem wir die Konditionszahlen entlang des Pfades von i nach j multiplizieren. Falls es mehrere solcher Kanten gibt, müssen wir die Ergebnisse aller möglichen Pfade von i nach j summieren. Im Beispiel trägt etwa der relative Fehler ε_a im Knoten a (der ausschließlich aus dem relativen Eingangsfehler ε_1 stammt) zum relativen Fehler ε_y im Ergebnisknoten y mit dem Faktor $\frac{a}{a+b} \cdot \frac{u}{u+c} = \frac{a}{a+b} \cdot \frac{a+b}{a+b+c} = \frac{a}{a+b+c}$ bei, also gilt

$$\varepsilon_y \doteq \frac{a}{a+b+c}\varepsilon_a + \cdots,$$

wobei $\cdots$ für die Fehlerbeiträge aller anderen Knoten steht, von denen ausgehend man ebenfalls auf mindestens einem Pfad zum Ergebnisknoten y gelangt. (**Quizfrage:** Wie groß ist der Einfluss von ε_c auf ε_y?)

Für die Antwort auf Frage (2) müssen wir die Fehlerbeiträge entlang aller Pfade, die von irgendwelchen Knoten i zum betrachteten Knoten j führen, aufsummieren. Dabei ist auch der im Knoten j selbst entstehende Rundungsfehler zu berücksichtigen sowie alle Rundungsfehler, die in den Vorgängerknoten i entstehen. Beispielsweise können wir aus dem Graphen ablesen, dass

$$\varepsilon_u \doteq \varepsilon_4 + \frac{a}{a+b}\,\varepsilon_1 + \frac{b}{a+b}\,\varepsilon_2$$

gilt. Für den Ergebnisknoten y gilt entsprechend

$$\begin{aligned}
\varepsilon_y &\doteq \varepsilon_5 + \frac{c}{u+c}\,\varepsilon_3 + \frac{u}{u+c}\,\varepsilon_u \\
&= \varepsilon_5 + \frac{c}{u+c}\,\varepsilon_3 + \frac{u}{u+c}\left[\varepsilon_4 + \frac{a}{a+b}\,\varepsilon_1 + \frac{b}{a+b}\,\varepsilon_2\right] \\
&= \varepsilon_5 + \frac{c}{a+b+c}\,\varepsilon_3 + \frac{a+b}{a+b+c}\left[\varepsilon_4 + \frac{a}{a+b}\,\varepsilon_1 + \frac{b}{a+b}\,\varepsilon_2\right] \\
&= \frac{a}{a+b+c}\,\varepsilon_a + \frac{b}{a+b+c}\,\varepsilon_b + \frac{c}{a+b+c}\,\varepsilon_c + \frac{a+b}{a+b+c}\,\varepsilon_4 + \varepsilon_5. \qquad (7.12)
\end{aligned}$$

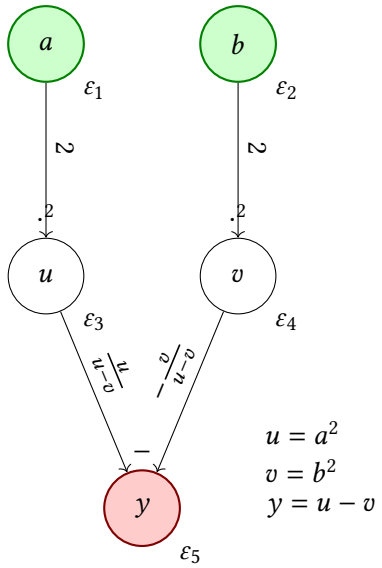
Diese Rechnung unterscheidet sich von unserem früheren Ergebnis

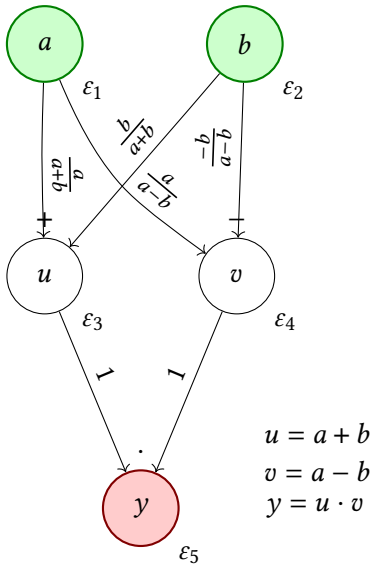
$$\varepsilon_y = \frac{\Delta y}{y} \doteq \frac{\Delta a + \Delta b + \Delta c}{a+b+c} + \frac{a+b}{a+b+c}\varepsilon_1^{(1)} + \varepsilon_1^{(2)} \qquad (7.11)$$

lediglich in der Notation, da per Definition $\varepsilon_a = \Delta a/a$ gilt etc.

Wir betrachten noch weitere Beispiele zur Vorwärtsanalyse.

Beispiel 7.5 (Vorwärtsanalyse für $a^2 - b^2$ in zwei Varianten, vgl. Freund, Hoppe, 2007, Kapitel 1.3, Beispiel 5, S.18)**.** *Wir werten den Ausdruck $a^2 - b^2 = (a+b)(a-b)$ auf die beiden offensichtlichen Weisen aus. Dazu gehören folgende Berechnungsgraphen:*





Im ersten (linken) Algorithmus erhalten wir

$$\begin{aligned}
\varepsilon_y &\doteq \varepsilon_5 + \frac{u}{u-v}\,\varepsilon_u - \frac{v}{u-v}\,\varepsilon_v \\
&= \varepsilon_5 + \frac{u}{u-v}\,\bigl[\varepsilon_3 + 2\,\varepsilon_1\bigr] - \frac{v}{u-v}\,\bigl[\varepsilon_4 + 2\,\varepsilon_2\bigr] \\
&= \varepsilon_5 + \frac{a^2}{a^2-b^2}\,\bigl[\varepsilon_3 + 2\,\varepsilon_1\bigr] - \frac{b^2}{a^2-b^2}\,\bigl[\varepsilon_4 + 2\,\varepsilon_2\bigr] \\
&= \frac{2\,a^2}{a^2-b^2}\,\varepsilon_a - \frac{2\,b^2}{a^2-b^2}\,\varepsilon_b + \frac{a^2}{a^2-b^2}\,\varepsilon_3 - \frac{b^2}{a^2-b^2}\,\varepsilon_4 + \varepsilon_5,
\end{aligned} \tag{7.13}$$

im zweiten (rechten) Algorithmus dagegen

$$\begin{aligned}
\varepsilon_y &\doteq \varepsilon_5 + \varepsilon_u + \varepsilon_v \\
&= \varepsilon_5 + \Bigl[\varepsilon_3 + \frac{a}{a+b}\varepsilon_1 + \frac{b}{a+b}\varepsilon_2\Bigr] + \Bigl[\varepsilon_4 + \frac{a}{a-b}\varepsilon_1 - \frac{b}{a-b}\varepsilon_2\Bigr] \\
&= \Bigl(\frac{a}{a+b} + \frac{a}{a-b}\Bigr)\,\varepsilon_1 + \Bigl(\frac{b}{a+b} - \frac{b}{a-b}\Bigr)\,\varepsilon_2 + \varepsilon_3 + \varepsilon_4 + \varepsilon_5 \\
&= \frac{2\,a^2}{a^2-b^2}\,\varepsilon_a - \frac{2\,b^2}{a^2-b^2}\,\varepsilon_b + \varepsilon_3 + \varepsilon_4 + \varepsilon_5.
\end{aligned} \tag{7.14}$$

*Die Terme mit ε_a und ε_b sind in beiden Verfahren gleich (**Quizfrage:** Warum muss das so sein?). Die verbleibenden Terme können wir im Fall von (7.13) abschätzen durch*

$$\left|\frac{a^2}{a^2-b^2}\,\varepsilon_3 - \frac{b^2}{a^2-b^2}\,\varepsilon_4 + \varepsilon_5\right| \leqslant \frac{a^2+b^2+|a^2-b^2|}{|a^2-b^2|}\,\varepsilon_{\text{mach}}, \tag{7.15}$$

wohingegen wir im Fall von (7.14)

$$|\varepsilon_3 + \varepsilon_4 + \varepsilon_5| \leqslant 3\,\varepsilon_{\text{mach}} \tag{7.16}$$

erhalten. Wir vergleichen jetzt die beiden oberen Schranken. Die zweite ist genau dann kleiner gleich der ersten, also

$$a^2 + b^2 + |a^2 - b^2| \geqslant 3\,|a^2 - b^2|,$$

wenn $\left|\frac{a}{b}\right| \in \left[\frac{1}{3}, 3\right]$ *gilt, wenn also die Daten* a *und* b *betragsmäßg relativ nahe beieinander liegen. In diesem Fall ist also die Auswertung von der Form* $(a+b)(a-b)$ *vorteilhaft.*

Beispiel 7.6 (Vorwärtsanalyse der p–q-Formel in zwei Varianten, vgl. Freund, Hoppe, 2007, Kapitel 1.4, Beispiel 1, S.23)**.** *Die quadratische Gleichung*

$$y^2 + 2\,p\,y - q = 0 \tag{7.17}$$

hat bekanntlich die beiden Lösungen

$$y_1 = -p + \sqrt{p^2+q} \quad \textit{und} \quad y_2 = -p - \sqrt{p^2+q}.$$

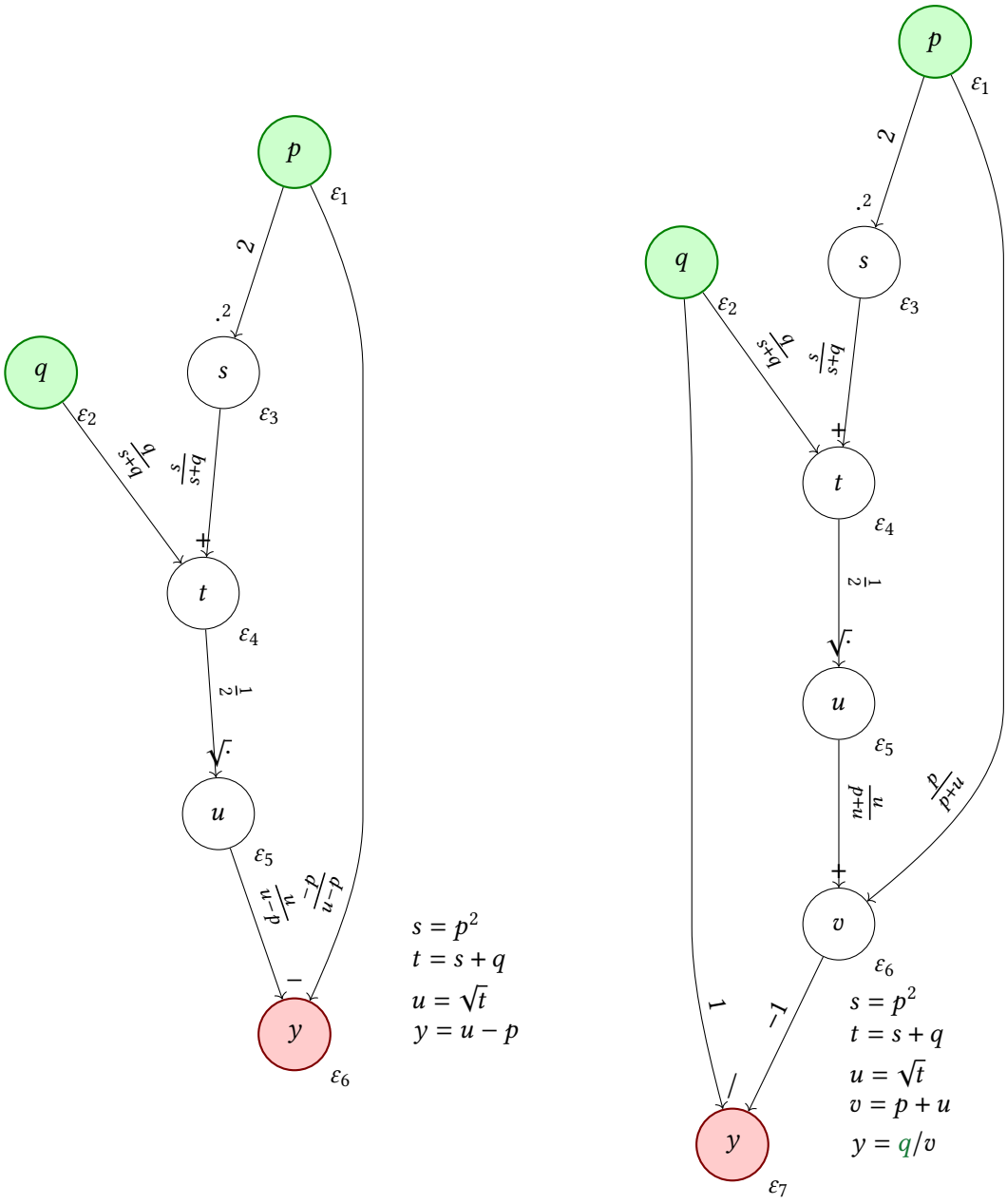
Nach dem Wurzelsatz von Vieta

$$y_1\,y_2 = -q$$

(elementares Nachrechnen) hat man aber auch die Möglichkeit, eine der beiden Wurzeln aus der Kenntnis der anderen zu berechnen, etwa $y_1 = -q/y_2$*. Was ist numerisch besser?*

Wir betrachten hier nur den Fall $p > 0$ *und* $q > 0$*. Mit diesen Daten ist die Wurzel* y_1 *numerisch die interessantere. (**Quizfrage:** Warum?)*

Wir diskutieren jetzt beide Varianten, $F(p,q) = -p + \sqrt{p^2+q}$ *zu berechnen:* ***Der rechte Berechnungsgraph war fehlerhaft und wurde korrigiert.***[RH]



*Anstatt wie bisher die komplette Fortpflanzung aller Fehler anzusehen, suchen wir im Berechnungsgraphen nach den Stellen, an denen möglicherweise (abhängig von den Eingangsdaten) große Faktoren stehen. In der ersten Variante sind alle Schritte gut konditioniert (**Quizfrage:** Klar?) bis möglicherweise auf den letzten Schritt. An den Kanten, die zum Ergebnisknoten führen, stehen die Faktoren $\frac{u}{u-p}$ und $\frac{-p}{u-p}$. Wir betrachten exemplarisch nur den ersten, der ausgeschrieben*

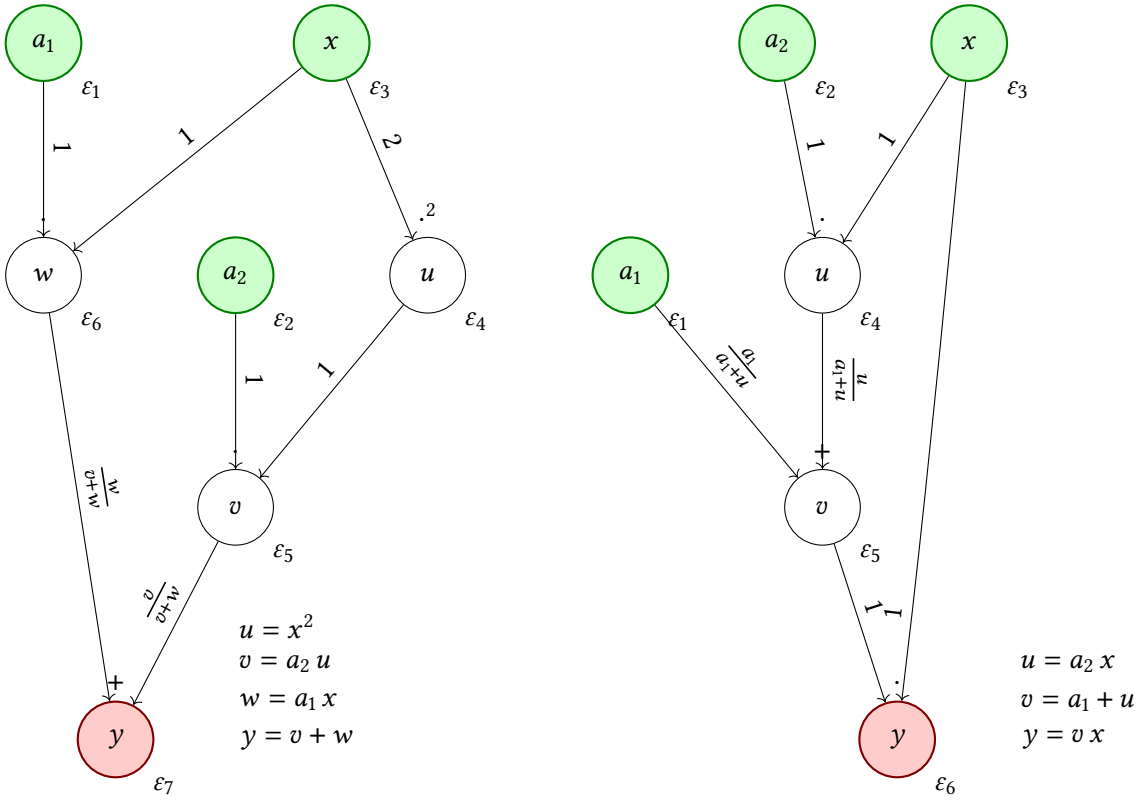
$$\frac{u}{u-p} = \frac{\sqrt{p^2+q}}{\sqrt{p^2+q}-p} = \frac{\sqrt{p^2+q}}{\sqrt{p^2+q}-p}\,\frac{\sqrt{p^2+q}+p}{\sqrt{p^2+q}+p} = \frac{p^2+q+p\sqrt{p^2+q}}{q} \geqslant \frac{2p^2}{q}$$

lautet. Die Abschätzung gilt wegen der oben getroffenen Annahme $p > 0$ und $q > 0$. Bei konstantem q und $p \to \infty$ geht also die relative partielle Konditionszahl $\frac{u}{u-p}$ auf der betrachteten Kante gegen ∞. Bereits

dieser eine Beitrag zum Gesamtfehler ε_y kann also für große Werte von p inakzeptabel groß werden. ***Quizfrage:*** *Wie verhält sich der andere Faktor $\frac{-p}{u-p}$?*

Betrachten wir nun die zweite Variante. In dieser liegen alle auftretenden Kantenfaktoren betragsmäßig im Intervall $[0,2]$, was kein Problem darstellt.

Beispiel 7.7 (Horner-Schema). *Das* ***Horner-Schema*** *ist ein Verfahren zur Auswertung eines durch seine Koeffizienten gegebenen Polynoms an einer gegebenen Stelle $x \in \mathbb{R}$. Für Polynome der Bauart $p(x) = a_1 x + a_2 x^2 = x(a_1 + a_2 x)$ gibt die rechte Darstellung die Auswertungsvorschrift nach dem Horner-Schema an. Wir vergleichen sie mit der «naiven» (linken) Darstellung.*



Aus diesen Berechnungsgraphen können wir folgende Darstellungen für den relativen Fehler ablesen: Für die naive Auswertung gilt

$$\begin{aligned}\varepsilon_y &\doteq \varepsilon_7 + \frac{a_1}{a_1 + a_2\, x}\big(\varepsilon_{a_1} + \varepsilon_x + \varepsilon_6\big) + \frac{a_2\, x}{a_1 + a_2\, x}\big(\varepsilon_{a_2} + 2\,\varepsilon_x + \varepsilon_4 + \varepsilon_5\big)\\ &= \varepsilon_{a_1} + \varepsilon_x + \varepsilon_6 + \varepsilon_7 + \frac{x}{a_1/a_2 + x}\big(-\varepsilon_{a_1} + \varepsilon_{a_2} + \varepsilon_x + \varepsilon_4 + \varepsilon_5 - \varepsilon_6\big),\end{aligned}$$

und für das Horner-Schema haben wir

$$\begin{aligned}\varepsilon_y &\doteq \varepsilon_5 + \varepsilon_6 + \varepsilon_x + \frac{a_1}{a_1 + a_2\, x}\,\varepsilon_{a_1} + \frac{a_2\, x}{a_1 + a_2\, x}\big(\varepsilon_{a_2} + \varepsilon_x + \varepsilon_4\big)\\ &= \varepsilon_{a_1} + \varepsilon_x + \varepsilon_5 + \varepsilon_6 + \frac{x}{a_1/a_2 + x}\big(-\varepsilon_{a_1} + \varepsilon_{a_2} + \varepsilon_x + \varepsilon_4\big).\end{aligned}$$

Im Vergleich fällt also das Horner-Schema bei der Vorwärtsanalyse günstiger aus, vor allem, wenn die Auswertungsstelle nahe der Nullstelle $-a_1/a_2$ liegt. Außerdem benötigt es eine Multiplikation weniger.

Der Vollständigkeit halber geben wir das Horner-Schema noch für allgemeine Polynome vom Grad n an. Es ist das gängige Verfahren zur Auswertung von Polynomen.

Algorithmus 7.8 (Horner-Schema für $p(x) = a_0 + a_1\,x + \cdots + a_n\,x^n$)**.**
Eingabe: *Koeffizienten $a_0, \ldots, a_n$ eines Polynoms $p(x) = a_0 + a_1\,x + \cdots + a_n\,x^n$*
Eingabe: *Auswertungsstelle x*
Ausgabe: *Funktionswert $y = p(x)$*
1: *Setze $y := a_n$*
2: ***for*** $k = n-1, n-2, \ldots, 0$ ***do***
3: *Setze $y := x\,y + a_k$*
4: ***end for***

Um die Stabilität eines konkreten Algorithmus zur Lösung einer Aufgabe $y = F(x)$ zu beurteilen, bietet sich ein Vergleich mit einem hypothetischen Modellalgorithmus an, und zwar mit

$$\overline{F} := \mathrm{rd} \circ F \circ \mathrm{rd}\,. \tag{7.18}$$

Das heißt, die Eingabe wird auf das verwendete Fließkommagitter gerundet, dann wird ohne Fehler die exakte Funktion F ausgewertet, und anschließend wird das Ergebnis wieder auf das Fließkommagitter gerundet. Der Modellalgorithmus (7.18) stellt damit den geringstmöglichen Eingriff in die exakte Funktionsauswertung dar, der unter Verwendung von Fließkommaarithmetik möglich ist.

Mit den uns bekannten Techniken können wir den relativen Fehler $\varepsilon_{\overline{y}}$ von $\overline{y} := \overline{F}(x)$ gegenüber dem exakten Wert $y = F(x)$ abschätzen. Es gilt folgendes Resultat:

$$\|\overline{F}(x) - F(x)\|_2 \mathrel{\dot{\leqslant}} \big(K(x)\|x\|_2 + \|y\|_2\big)\,\varepsilon_{\mathrm{mach}}, \tag{7.19a}$$

$$\left\| \left[\frac{\overline{F}_i(x) - F_i(x)}{F_i(x)} \right]_{i=1}^{m} \right\|_\infty \mathrel{\dot{\leqslant}} \big(k(x) + 1\big)\,\varepsilon_{\mathrm{mach}} \tag{7.19b}$$

mit den absoluten und relativen Konditionszahlen $K(x) = \|F'(x)\|_{2\to 2}$ und $k(x) = \|(k_{ij}(x))\|_{\infty\to\infty}$.
Quizfrage: Beweis?

Die Schranke auf der rechten Seite von (7.19a) können wir als den unvermeidbaren (absoluten) Fehler ansehen, der allein durch den Übergang zu Fließkommazahlen passiert und der wesentlich durch die Kondition, also eine Eigenschaft der Aufgabe, bestimmt ist. Dieser unvermeidbare Fehler dient als Referenz, mit der wir die Abschätzungen der Rundungsfehler, die wir für konkrete Algorithmen wie etwa im Beispiel 7.5 zeigen können, vergleichen. Entsprechend gibt die Schranke auf der rechten Seite von (7.19b) diesen unvermeidbaren Fehler in relativen Termen an.

Beachte: Die Abschätzung (7.19b) ergibt sich gerade aus der Vorwärtsanalyse, wenn man die in den Zwischenschritten auftretenden Rundungsfehler gleich Null setzt, also bis auf den Eingangsfehler und die Rundung des Ergebnisses keine weiteren Rundungsfehler zwischendurch berücksichtigt. In den beiden Algorithmen aus Beispiel 7.5 sind dazu beispielsweise die Zahlen ε_3 und ε_4 gleich Null zu setzen.

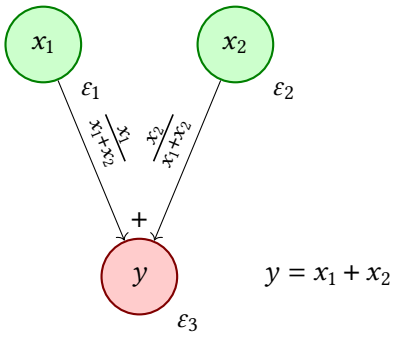
Definition 7.9 (Vorwärtsstabilität). *Ein mit Maschinenoperationen realisierter Algorithmus $\widehat{F}(x)$ zur Auswertung von $y = F(x)$ heißt **vorwärtsstabil** (englisch: forward stable) an der Stelle x, wenn gilt: Der relative Fehler ε_y der Ausgabe y liegt in derselben Größenordnung wie die Schranke in (7.19b):*

$$\|\varepsilon_y\|_\infty = \frac{\|\widehat{F}(x) - F(x)\|_\infty}{\|F(x)\|_\infty} \leqslant \big(C(x)\, k(x) + 1\big)\, \varepsilon_{\text{mach}} \tag{7.20}$$

mit einem «nicht zu großen» Faktor $C(x)$.

Beachte: Algorithmen mit vektorwertigem Ergebnis sind genau dann vorwärtsstabil an der Stelle x, wenn sie dort komponentenweise vorwärtsstabil sind.

Beispiel 7.10 (Vorwärtsstabilität der Maschinenoperationen $\oplus$, $\ominus$, $\odot$ und $\oslash$).
*Lemma 5.6 und Satz 6.1 zeigen, dass die Maschinenoperation $\oplus$ mit vorangegangener Rundung der Eingabe an allen Stellen $x = (x_1, x_2)^\mathsf{T}$ vorwärtsstabil sind, für die x_1, x_2 und $x_1 + x_2$ in $\mathbb{D}$ liegen. Dasselbe gilt für $\ominus$, $\odot$ und $\oslash$. Der Faktor $C(x)$ kann gleich 1 gewählt werden. (**Quizfrage:** Klar?) Für $\oplus$ beispielsweise haben wir:*



Damit ergibt sich $\varepsilon_y = \varepsilon_3 + \frac{x_1}{x_1+x_2}\,\varepsilon_1 + \frac{x_2}{x_1+x_2}\,\varepsilon_2$ und daher $|\varepsilon_y| \leqslant \varepsilon_{\text{mach}} + \frac{|x_1|+|x_2|}{|x_1+x_2|}\,\varepsilon_{\text{mach}} = (k(x)+1)\,\varepsilon_{\text{mach}}$.

Beispiel 7.11 (Vorwärtsstabilität für $a^2 - b^2$ in zwei Varianten, vgl. Beispiel 7.5).
Die Abschätzungen (7.15) und (7.16) zeigen, dass beide Algorithmen zur Bestimmung von $a^2 - b^2$ aus Beispiel 7.5 «überall» (sofern a, b, $a+b$, $a-b$ und $a^2 - b^2$ in $\mathbb{D}$ liegen) vorwärtsstabil sind, da die durch die Rundungsfehler «zwischendurch» entstehenden Terme mit ε_3 und ε_4 jeweils in derselben Größenordnung wie die restlichen Fehlerterme liegen.

Beispiel 7.12 (Vorwärtsstabilität des Innenprodukts).
Das auf die offensichtliche Art und Weise berechnete Innenprodukt $F(x, y) = x^\mathsf{T} y = \sum_{i=1}^n x_i\, y_i$ zweier Vektoren $x, y \in \mathbb{R}^n$ ist vorwärtsstabil, sofern die Eingaben und alle Zwischenergebnisse in $\mathbb{D}$ liegen.

Beweis. □

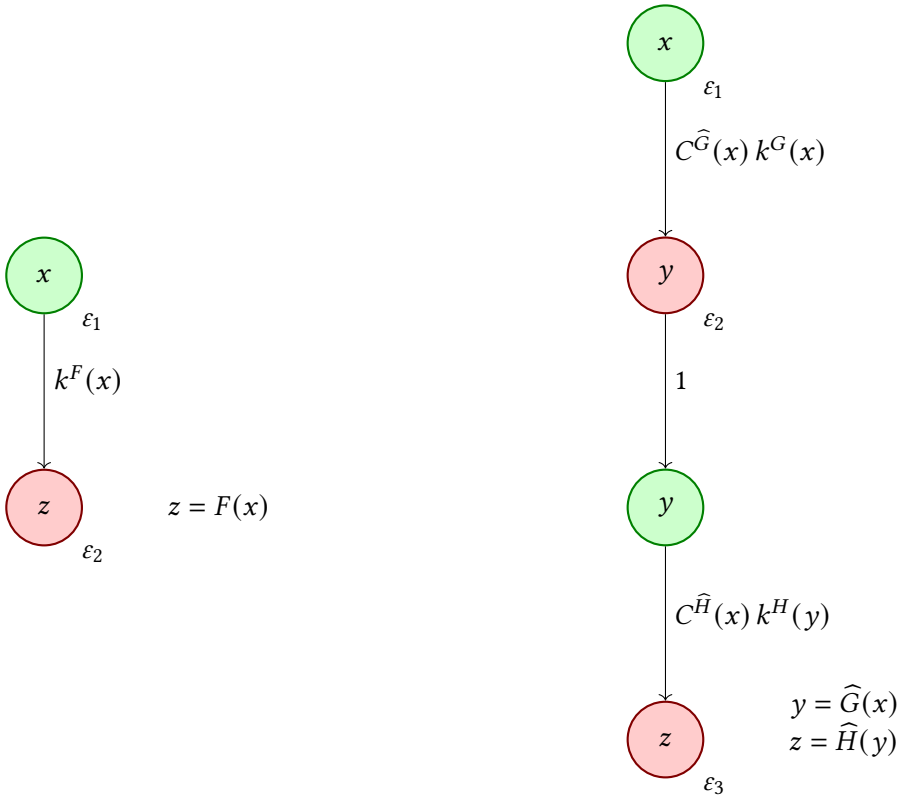
Bemerkung 7.13 (zur Vorwärtsstabilität)**.**

(1) Der Übergang von einem vorwärtsstabilen Algorithmus zu einem nicht vorwärtsstabilen Algorithmus ist fließend, da in (7.20) verlangt wird, dass der Faktor $C(x)$ «nicht zu groß» ist.

(2) Wie wir schon in Beispiel 7.11 gesehen haben, hängt es i. A. von den Eingabedaten ab, welche von zwei (oder mehr) Varianten eines Algorithmus die bessere Schranke für den Vorwärtsfehler liefert. Es bietet sich also an, beide Varianten zu implementieren und mittels einer Fallunterscheidung zur Laufzeit die passendere auszuwählen!

Die Vorwärtsstabilität eines Algorithmus kann also nur dadurch gefährdet sein, dass die bei den Zwischenergebnissen auftretenden Rundungsfehler auf ihrem weiteren Weg hin zum Endergebnis eine derart große Verstärkung erfahren, dass ihr Beitrag zum Gesamtfehler die Schranke auf der rechten Seite von (7.19b) um Größenordnungen überschreitet. Leider bedeutet das auch, dass die Hintereinanderausführung zweier für sich genommen vorwärtsstabiler Algorithmen nicht notwendig ebenfalls wieder vorwärtsstabil ist!

Um das einzusehen, betrachten wir eine Funktion $F = H \circ G$. Die folgenden Berechnungsgraphen stellen die Situationen dar, dass wir einerseits ein Verfahren für die komplette Funktion F betrachten und andererseits zwei Verfahren für die Teilabbildungen H und G:[3]



[3] Im Unterschied zu unseren bisherigen Berechnungsgraphen sind hier die Ein- und Ausgaben möglicherweise vektorwertig.

Das erste (hypothetische) Verfahren dient nur zur Festlegung der Referenz, vgl. (7.18) und (7.19b), wobei ein unvermeidbarer relativer Fehler der Größe

$$\begin{aligned} \varepsilon_z &\doteq \left(k_{ij}^F(x)\right) \varepsilon_1 + \varepsilon_2 \\ \|\varepsilon_z\|_\infty &\dot{\leqslant} \left(k^F(x) + 1\right) \varepsilon_{\text{mach}} \end{aligned} \tag{7.21}$$

produziert wird. Bei der Hintereinanderausführung der Algorithmen für die Teilabbildungen G und H passiert Folgendes. Das Zwischenergebnis y weist wegen der Vorwärtsstabilität des Verfahrens $\widehat{G}$ den relativen Fehler

$$\|\varepsilon_y\|_\infty \dot{\leqslant} \left[C^{\widehat{G}}(x)\, k^G(x) + 1\right] \varepsilon_{\text{mach}}$$

auf. Das Endergebnis nach Ausführung des zweiten Teilalgorithmus hat also den relativen Fehler

$$\begin{aligned} \|\varepsilon_z\|_\infty &\dot{\leqslant} C^{\widehat{H}}(y)\, k^H(y)\, \|\varepsilon_y\|_\infty + \|\varepsilon_3\|_\infty \\ &\dot{\leqslant} \left[C^{\widehat{H}}(y)\, k^H(y)\, \left(C^{\widehat{G}}(y)\, k^G(x) + 1\right) + 1\right] \varepsilon_{\text{mach}}. \end{aligned} \tag{7.22}$$

Der möglicherweise problematische Term ist in rot markiert. Der Faktor $\varepsilon_{\text{mach}}$ repräsentiert den bei der Bildung des Zwischenergebnisses y unvermeidbaren Rundungsfehler (im Berechnungsgraphen oben ε_2). Dieser wird dann mit der Konditionszahl $k^H(y)$ (und dem wegen der Vorwärtsstabilität gutartigen Faktor $C^{\widehat{H}}(y)$) durch die nachfolgende Teilabbildung verstärkt und liefert den Beitrag

$$C^{\widehat{H}}(y)\, k^H(y)\, \varepsilon_{\text{mach}}$$

zum relativen Gesamtfehler im Ergebnis z. Um die Vorwärtsstabilität der Hintereinanderausführung $\widehat{H} \circ \widehat{G}$ nachzuweisen, müssten wir insbesondere diesen Term nach oben gegen die rechte Seite in (7.21) (mal Faktor) abschätzen.

Wegen der Submultiplikativität der Matrixnormen, vgl. (3.27), gilt zwar

$$k^F(x) \leqslant k^H(y)\, k^G(x),$$

aber wir würden die umgekehrte Abschätzung benötigen, die nicht gilt.

Der Punkt ist, dass die Teilabbildung H eine wesentlich größere Konditionszahl besitzen kann als die Gesamtabbildung $F = H \circ G$! Da es hierbei um Eigenschaften der Abbildungen, also um die Aufgabe selbst geht, tritt dieses Problem unabhängig davon auf, welche konkreten Algorithmen man für die Teilaufgaben G und H verwenden würde!

Regel: Unterteile eine Aufgabe niemals so in Teilaufgaben, dass von diesen eine wesentlich schlechter konditioniert ist als die Gesamtaufgabe!

Beispiel 7.14. *Die in Vorlesungen zur linearen Algebra eingeübte Technik, die Eigenwerte einer (symmetrischen) Matrix über die Nullstellen ihres charakteristischen Polynoms zu berechnen, ist numerisch kein geeignetes Verfahren! Man kann zeigen, dass die Aufgabe, die Abbildung «Matrix ↦ Eigenwerte» gut konditioniert ist und ebenso die erste Teilabbildung «Matrix ↦ Koeffizienten des charakteristischen Polynoms». Die Berechnung der Nullstellen eines Polynoms aus seinen Koeffizienten ist jedoch keine gut konditionierte Aufgabe (**Quizfrage:** Beispiel?). Siehe dazu auch Bornemann, 2018, Kapitel 18.*

§ 7.2 Rückwärtsanalyse

In Abschnitt 7.1 haben wir die Qualität von Algorithmen mit Hilfe der Vorwärtsanalyse beurteilt. Ein anderer Zugang ist der der Rückwärtsanalyse. Wir betrachten wiederum

$F(x)$ die auszuwertende Funktion,

$\widehat{F}(x)$ die durch den implementierten Algorithmus realisierte Funktion.

Bei der **Rückwärtsanalyse** von $\widehat{F}$ stellen wir die Frage, ob wir nicht das numerische Ergebnis $\widehat{F}(x)$ interpretieren können als $F(\widehat{x})$, also als exaktes Ergebnis zu einer veränderten Eingabe $\widehat{x}$. Da ein solches $\widehat{x}$ i. A. nicht eindeutig ist, könnten wir die Frage nach einem $\widehat{x}$ mit minimaler Norm stellen, das also die Aufgabe

$$\begin{aligned} \text{Minimiere} \quad & \|\overline{x}\|_\infty, \quad \overline{x} \in \mathbb{R}^n \\ \text{unter der Bedingung} \quad & F(\overline{x}) = \widehat{F}(x) \end{aligned}$$

löst. Da die Bestimmung eines normminimalen $\widehat{x}$ allerdings aufwändig sein kann, begnügt man sich häufig mit einem $\overline{x}$ mit ausreichend kleiner Norm, das die Bedingung $F(\overline{x}) = \widehat{F}(x)$ erfüllt.

Definition 7.15 (Rückwärtsstabilität)**.** *Ein mit Maschinenoperationen realisierter Algorithmus $\widehat{F}(x)$ zur Auswertung von $y = F(x)$ heißt **rückwärtsstabil** (englisch:* backward stable*) an der Stelle x, wenn gilt: Es gibt ein $\overline{x}$ mit der Eigenschaft $\widehat{F}(x) = F(\overline{x})$, sodass gilt:*

$$\frac{\|\overline{x} - x\|_\infty}{\|x\|_\infty} \leqslant B(x)\, \varepsilon_{\text{mach}} \tag{7.23}$$

mit einem «nicht zu großen» Faktor $B(x)$.

Um es mit den Worten von Trefethen, Bau, 1997, S.104 zu sagen:

> A backward stable algorithm gives exactly the right answer to nearly the right question.

Satz 7.16 (Rückwärtsanalyse für die Maschinenoperationen $\oplus$, $\ominus$, $\odot$ und $\oslash$)**.** *Es seien $x, y \in \mathbb{F}$.*

(*i*) *Falls $x + y \in \mathbb{D}$ liegt, so gilt mit einer Zahl $\varepsilon \in \mathbb{R}$, $|\varepsilon| \leqslant \varepsilon_{\text{mach}}$:*

$$x \oplus y = x(1+\varepsilon) + y(1+\varepsilon).$$

(*ii*) *Falls $x - y \in \mathbb{D}$ liegt, so gilt mit einer Zahl $\varepsilon \in \mathbb{R}$, $|\varepsilon| \leqslant \varepsilon_{\text{mach}}$:*

$$x \ominus y = x(1+\varepsilon) - y(1+\varepsilon).$$

(*iii*) *Falls $x \cdot y \in \mathbb{D}$ liegt, so gilt mit einer Zahl $\varepsilon \in \mathbb{R}$, $|\varepsilon| \leqslant \varepsilon_{\text{mach}}$:*

$$x \odot y = x(1+\varepsilon) \cdot y.$$

(*iv*) *Falls $y \neq 0$ ist und $x/y \in \mathbb{D}$ liegt, so gilt mit einer Zahl $\varepsilon \in \mathbb{R}$, $|\varepsilon| \leqslant \varepsilon_{\text{mach}}$:*

$$x \oslash y = x(1+\varepsilon)/y.$$

Beweis. Der Beweis folgt sofort aus Satz 6.1. □

Folgerung 7.17 (Rückwärtsstabilität der Maschinenoperationen $\oplus$, $\ominus$, $\odot$ und $\oslash$)**.** *Die Maschinenoperationen $\oplus$, $\ominus$, $\odot$ und $\oslash$ sind also rückwärtsstabil an den in Satz 7.16 genannten Stellen. Der Faktor $B(x)$ kann gleich 1 gewählt werden.*

Beispiel 7.18 (Rückwärtsstabilität des Innenprodukts)**.**
Das auf die offensichtliche Art und Weise berechnete Innenprodukt $F(x, y) = x^\mathsf{T} y = \sum_{i=1}^n x_i\, y_i$ zweier Vektoren $x, y \in \mathbb{R}^n$ ist rückwärtsstabil, sofern die Eingaben und alle Zwischenergebnisse in $\mathbb{D}$ liegen.

Beweis. □

Beispiel 7.19 (Rückwärtsstabilität der Polynomauswertung)**.**
Die Auswertung eines Polynoms $p(x) = a_1\, x + a_2\, x^2 = x\,(a_1 + a_2\, x)$ nach der naiven Methode und nach dem Horner-Schema, vgl. Beispiel 7.7, sind beide rückwärtsstabil.

Beweis. □

Was können wir über den Vorwärtsfehler an der Stelle x eines rückwärtsstabilen Algorithmus sagen? Dazu sei $\overline{x}$ wie in Definition 7.15 gewählt. Wir schätzen ab: **Das hier komponentenweise schreiben**[RH] **Das hier gilt nur, wenn wir in der Definition der Vorwärtsstabilität mit $\dot{\leqslant}$ arbeiten**[GM]

$$\begin{aligned}
\frac{\|\widehat{F}(x) - F(x)\|_\infty}{\|F(x)\|_\infty} &= \frac{\|F(\overline{x}) - F(x)\|_\infty}{\|F(x)\|_\infty} \\
&\dot{\leqslant} k(x)\, \frac{\|\overline{x} - x\|_\infty}{\|x\|_\infty} \quad \text{nach Definition der relativen Konditionszahl } k(x) \\
&= B(x)\, k(x)\, \varepsilon_{\text{mach}} \quad \text{nach Definition 7.15 der Rückwärtsstabilität.} \qquad (7.24)
\end{aligned}$$

Damit erfüllt $\widehat{F}$ also auch die Definition 7.9 der Vorwärtsstabilität, und zwar mit der gleichen Konstante $C(x) = B(x)$! Wir haben damit bewiesen:

Satz 7.20 (Rückwärtsstabilität impliziert Vorwärtsstabilität)**.** *Es sei $\widehat{F}(x)$ ein mit Maschinenoperationen realisierter Algorithmus zur Auswertung von $y = F(x)$. Wenn der Algorithmus $\widehat{F}$ an der Stelle x rückwärtsstabil ist, dann ist er dort auch vorwärtsstabil. Kurz: Ein rückwärtsstabiles Verfahren ist auch vorwärtsstabil.*

Die Umkehrung von Satz 7.20 gilt aber nicht, wie folgendes Beispiel zeigt:

Beispiel 7.21 (Äußere Produkte sind vorwärtsstabil, aber nicht rückwärtsstabil)**.** *Wir betrachten die Aufgabe, das äußere Produkt zweier Vektoren $x, y \in \mathbb{R}^n$ zu berechnen, also*

$$F(x, y) = x\, y^\intercal.$$

Das Ergebnis ist eine Matrix, liegt also — entgegen unserem allgemeinen Setting — nicht im $\mathbb{R}^m$, sondern im $\mathbb{R}^{n\times n}$. Das macht aber keine Schwierigkeiten, da wir $x\, y^\intercal$ einfach vektorisieren (z. B. spaltenweise untereinander schreiben) und dann z. B. die ∞-Norm für Vektoren verwenden können, um Fehler zu messen.

Der offensichtliche Algorithmus, $\widehat{F}_{ij}(x, y) = \mathrm{rd}(x_i) \odot \mathrm{rd}(y_j)$ zu bestimmen, ist komponentenweise vorwärtsstabil nach Beispiel 7.10, also insgesamt vorwärtsstabil. Dieser Algorithmus ist allerdings nicht rückwärtsstabil, denn das Ergebnis $\big(\widehat{F}_{ij}(x)\big)$ wird i. A. keine Rang-1-Matrix sein, d. h., $\widehat{F}(x)$ lässt sich überhaupt nicht schreiben als äußeres Produkt von zwei Vektoren.

§ 7.3 Allgemeine Stabilitätsanalyse

Das allgemeinste Konzept zur Stabilitätsanalyse kombiniert die Vorwärts- und Rückwärtsanalyse.

Definition 7.22 (Stabilität)**.** *Ein mit Maschinenoperationen realisierter Algorithmus $\widehat{F}(x)$ zur Auswertung von $y = F(x)$ heißt **(numerisch) stabil** (englisch:* (numerically) stable*) an der Stelle x, wenn gilt: Es gibt ein $\overline{x}$, sodass gilt:*

$$\frac{\|\widehat{F}(x) - F(\overline{x})\|_\infty}{\|F(\overline{x})\|_\infty} \leqslant C(x)\, \varepsilon_{\text{mach}} \tag{7.25a}$$

$$\frac{\|\overline{x} - x\|_\infty}{\|x\|_\infty} \leqslant B(x)\, \varepsilon_{\text{mach}} \tag{7.25b}$$

mit «nicht zu großen» Faktoren $B(x)$ und $C(x)$.

Beachte: Der hypothetische Modellalgorithmus (7.18), also $\widehat{F} := \mathrm{rd} \circ F \circ \mathrm{rd}$ erfüllt diese Definition mit $B(x) = C(x) = 1$. (**Quizfrage:** Beweis?) Die Definition 7.22 kann also so gelesen werden, dass man vom Algorithmus $\widehat{F}$ qualitativ dieselbe Abschätzung verlangt, die für den Modellalgorithmus gilt.

Um es nochmal mit den Worten von Trefethen, Bau, 1997, S.104 zu sagen:

> A stable algorithm gives nearly the right answer to nearly the right question.

Wie man leicht sieht, gilt

$$\text{Rückwärtsstabilität} \quad \Rightarrow \quad \text{Stabilität}.$$

Quizfrage: Gilt auch Vorwärtsstabilität $\Rightarrow$ Stabilität?

Ende der Woche 5

Kapitel 4 Direkte Lösungsverfahren für lineare Gleichungssysteme

Das Lösen linearer Gleichungssysteme ist eine Grundaufgabe in der numerischen Mathematik.

§ 8 LR-Zerlegung und das Gausssche Eliminationsverfahren

Wir betrachten in diesem Abschnitt lineare Gleichungssysteme

$$A\,x = b$$

mit quadratischen Matrizen $A \in \mathbb{R}^{n\times n}$ und rechten Seiten $b \in \mathbb{R}^n$. Wenn A invertierbar (regulär, englisch: *non-singular*) ist, dann ist $A\,x = b$ eindeutig lösbar. Andernfalls ist das System entweder überhaupt nicht lösbar, oder es besitzt unendlich viele Lösungen. Bei numerischen Lösungsverfahren für lineare Gleichungssysteme geht es immer auch darum, festzustellen, ob die Matrix invertierbar ist bzw. numerisch nahe an der Nicht-Invertierbarkeit.

Wir beginnen mit einem einfachen Beispiel:

$$\left.\begin{array}{rcrcrcr} 6\,x_1 & + & 1\,x_2 & + & 1\,x_3 & = & 11 \\ 3\,x_1 & + & 3\,x_2 & + & 1\,x_3 & = & 5 \\ -6\,x_1 & - & 6\,x_2 & - & 3\,x_3 & = & -9 \end{array}\right\} \quad \Leftrightarrow \quad \begin{bmatrix} 6 & 1 & 1 \\ 3 & 3 & 1 \\ -6 & -6 & -3 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 11 \\ 5 \\ -9 \end{pmatrix}.$$

Beachte: Jede Zeile der Matrix steht für eine Gleichung, jede Spalte gehört zu einer der Variablen.

Folgende Operationen lassen die Lösungsmenge eines linearen Gleichungssystems unverändert:

(1) Vertauschen von Zeilen (Gleichungen),

(2) Multiplikation einer Zeile mit einer Zahl $\neq 0$,

(3) Addition einer Zeile zu einer anderen Zeile.

Diese Operationen wirken sich natürlich immer auch auf die rechte Seite aus.

Die oben genannten Möglichkeiten der Äquivalenztransformationen nutzt das Gaußsche Eliminationsverfahren, das Sie vermutlich kennen, aus, um die Gestalt des Systems zu vereinfachen. Dabei

wird zunächst während der sogenannten **Vorwärtselimination** (englisch: *forward elimination*) dafür gesorgt, dass die Anzahl der in den Gleichungen vorkommenden Variablen von Zeile zu Zeile um eins abnimmt. Die Matrix wird also auf **obere (rechte) Dreiecksgestalt** gebracht.

In einem ersten Schritt werden dazu die Einträge der ersten Spalte unterhalb der Diagonalen durch eine geeignete Linearkombination der betreffenden Zeile mit der ersten Zeile zu Null gemacht. Wir addieren also in unserem Beispiel zur Zeile 2 das $\frac{-3}{6}$-Fache der Zeile 1. Wir addieren weiterhin zur Zeile 3 das $\frac{6}{6}$-Fache der Zeile 1. Die erste Zeile selbst bleibt unverändert:

$$\begin{matrix}\frac{-3}{6}\\ \\ \frac{6}{6}\end{matrix}\!\!\left[\begin{array}{ccc|c} 6 & 1 & 1 & 11 \\ 3 & 3 & 1 & 5 \\ -6 & -6 & -3 & -9 \end{array}\right] \rightsquigarrow \left[\begin{array}{ccc|c} 6 & 1 & 1 & 11 \\ 0 & \frac{5}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & -5 & -2 & 2 \end{array}\right].$$

Im zweiten Schritt werden die Einträge der zweiten Spalte unterhalb der Diagonalen zu Null gemacht, indem geeignete Linearkombinationen mit der zweiten Zeile gebildet werden. In diesem Fall addieren wir das $\frac{5}{5/2} = 2$-Fache der Zeile 2 zur Zeile 3:

$$\begin{matrix}\\ \\ \frac{5}{5/2}\end{matrix}\!\!\left[\begin{array}{ccc|c} 6 & 1 & 1 & 11 \\ 0 & \frac{5}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & -5 & -2 & 2 \end{array}\right] \rightsquigarrow \left[\begin{array}{ccc|c} 6 & 1 & 1 & 11 \\ 0 & \frac{5}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -1 & 1 \end{array}\right].$$

Die Phase der Vorwärtselimination ist hier beendet. Das System lautet jetzt ausgeschrieben

$$\left.\begin{array}{rcrcrcl} 6\,x_1 & + & 1\,x_2 & + & 1\,x_3 & = & 11 \\ 0\,x_1 & + & \frac{5}{2}\,x_2 & + & \frac{1}{2}\,x_3 & = & -\frac{1}{2} \\ 0\,x_1 & + & 0\,x_2 & - & x_3 & = & 1 \end{array}\right\}.$$

Von hier aus können wir es mittels **Rückwärtssubstitution** (englisch: *back substitution*) leicht lösen:

$$\begin{aligned} x_3 &= \frac{1}{-1}\,(1) && = -1, \\ x_2 &= \frac{2}{5}\left(-\frac{1}{2} - \frac{1}{2}x_3\right) && = 0, \\ x_1 &= \frac{1}{6}\left(11 - 1\,x_2 - 1\,x_3\right) && = 2. \end{aligned}$$

Wir wollen das Vorgehen noch einmal rekapitulieren. Wir haben A durch Zeilenmanipulationen auf eine obere Dreiecksmatrix transformiert. Welche Zeilenmanipulationen waren dazu notwendig? Im ersten Schritt haben wir A (und die rechte Seite b) von links mit der Matrix

$$G^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & 0 \\ -\frac{a_{31}}{a_{11}} & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{3}{6} & 1 & 0 \\ -\frac{-6}{6} & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

multipliziert. Es entstand die Matrix

$$A^{(2)} := G^{(1)}A = \begin{bmatrix} 6 & 1 & 1 \\ 0 & \frac{5}{2} & \frac{1}{2} \\ 0 & -5 & -2 \end{bmatrix},$$

deren erste Zeile nach Konstruktion mit der ersten Zeile der Vorgängermatrix $A^{(1)} := A$ identisch ist. Die Matrix $A^{(2)}$ wurde dann von links mit

$$G^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{a_{32}^{(2)}}{a_{22}^{(2)}} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

multipliziert. Es entstand die Matrix

$$A^{(3)} := G^{(2)}A^{(1)} = \begin{bmatrix} 6 & 1 & 1 \\ 0 & \frac{5}{2} & \frac{1}{2} \\ 0 & 0 & -1 \end{bmatrix} =: R,$$

deren ersten zwei Zeilen mit den ersten zwei Zeilen der Vorgängermatrix $A^{(2)}$ identisch sind. Die Transformation auf die obere Dreiecksmatrix R war damit abgeschlossen.

Die obige Rechnung zeigt den Zusammenhang

$$G^{(2)}G^{(1)}A = R \quad \Leftrightarrow \quad A = \left(G^{(1)}\right)^{-1}\left(G^{(2)}\right)^{-1}R.$$

Matrizen, die sich wie $G^{(k)}$ nur darin von der Einheitsmatrix unterscheiden, dass in genau einer Spalte unterhalb der Diagonale Einträge ungleich Null stehen dürfen, heißen **Frobeniusmatrizen**. Genauer sprechen wir von einer *k*-**Frobeniusmatrix**, wenn dies in der k-ten Spalte der Fall ist. Frobeniusmatrizen sind immer invertierbar (**Quizfrage:** Warum?), und sie haben folgende für uns nützliche Eigenschaften:

Lemma 8.1 (Eigenschaften von Frobeniusmatrizen). *Es sei $G^{(k)} \in \mathbb{R}^{n\times n}$ eine k-Frobeniusmatrix, $k = 1, \dots, n$. Weiter sei $G'^{(k)} := G^{(k)} - \mathrm{Id}$.*[1] *Dann gilt:*

(*i*) $\left(G^{(k)}\right)^{-1} = \mathrm{Id} - G'^{(k)}$.

(*ii*) $G^{(1)}G^{(2)}\cdots G^{(n)} = \mathrm{Id} + \sum_{j=1}^{n} G'^{(j)}$, *ist also eine untere Dreiecksmatrix.*

(*iii*) $\left(G^{(1)}\right)^{-1}\left(G^{(2)}\right)^{-1}\cdots\left(G^{(n)}\right)^{-1} = \mathrm{Id} - \sum_{j=1}^{n} G'^{(j)}$, *ist also eine untere Dreiecksmatrix.*

Beweis. □

Quizfrage: Gelten die Aussagen (*ii*) und (*iii*) auch dann, wenn die Matrizen in anderer Reihenfolge stehen?

[1] $G'^{(k)}$ entspricht also $G^{(k)}$ mit «genullter» Diagonale.

Die Inverse einer Frobeniusmatrix lässt sich also leicht angeben, man negiert einfach die Vorzeichen der Einträge unterhalb der Diagonale (Aussage *(i)*). Produkte von Inversen (in der richtigen Reihenfolge!) werden einfach spaltenweise zusammengesetzt (Aussage *(iii)*).

In unserem Beispiel ist $G^{(1)}$ eine 1-Frobeniusmatrix und

$$\left(G^{(1)}\right)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix},$$

und $G^{(2)}$ ist eine 2-Frobeniusmatrix mit

$$\left(G^{(2)}\right)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}.$$

Wir definieren

$$L := \left(G^{(1)}\right)^{-1}\left(G^{(2)}\right)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -1 & -2 & 1 \end{bmatrix},$$

eine **untere (linke) Dreiecksmatrix**.

Wir haben damit eine Zerlegung der Form

$$A = LR$$

erhalten mit einer unteren Dreiecksmatrix L mit lauter Einsen auf der Diagonale und einer oberen Dreiecksmatrix R. Eine solche Zerlegung nennt man eine **LR-Zerlegung** (oder **LU-Zerlegung**, englisch: *LU decomposition*) der quadratischen Matrix A.

Frage: Existiert die LR-Zerlegung immer? Ist sie eindeutig?

Wir beantworten die zweite Frage zuerst.

Lemma 8.2 (Eindeutigkeit der LR-Zerlegung)**.** *Es sei $A \in \mathbb{R}^{n\times n}$ eine reguläre Matrix. Wenn A die beiden LR-Zerlegungen $A = L_1 R_1 = L_2 R_2$ besitzt, dann gilt $L_1 = L_2$ und $R_1 = R_2$.*

Beweis. □

Wie das Beispiel der regulären (und gut konditionierten[2]) Matrix

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

[2] $A^\mathsf{T}A = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ hat die Eigenwerte $\frac{3}{2} \pm \frac{1}{2}\sqrt{5}$, deren Wurzeln die Singulärwerte von A sind. Der größere durch den kleineren Singulärwert ist also die Wurzel aus $\frac{3+\sqrt{5}}{3-\sqrt{5}}$, was gerade etwa $\kappa(A) \approx 2.62$ ergibt.

zeigt, besitzen nicht alle regulären Matrizen eine LR-Zerlegung. **Quizfrage:** Was geht hier schief?

Wir können das Problem aber beheben, indem wir Zeilenvertauschungen in A zulassen. Solche Vertauschungen werden durch Permutationsmatrizen realisiert.

Definition 8.3 (Permutationsmatrix). *Eine Matrix $\Pi \in \mathbb{R}^{n\times n}$ heißt **Permutationsmatrix** (englisch: permutation matrix), wenn in jeder Zeile und jeder Spalte genau ein Eintrag gleich 1 ist und alle anderen Einträge gleich 0 sind. Eine Permutationsmatrix Π heißt **Vertauschungsmatrix** oder **Transpositionsmatrix** (englisch: transposition matrix), wenn es genau zwei Indizes $i \neq j$ gibt mit der Eigenschaft $\Pi_{ij} = 1$. Wir bezeichnen sie dann auch mit $\Pi^{(ij)}$.*

Aus Gründen einer vereinfachten Notation lassen wir auch den Fall $i = j$ zu und setzen $\Pi^{(ii)} = \mathrm{Id}$ für beliebiges $1 \leqslant i \leqslant n$.

Jede $n \times n$-Permutationsmatrix Π entspricht genau einer Permutation π der Zahlen $\{1, \dots, n\}$, also einer bijektiven Abbildung von $\{1, \dots, n\}$ auf sich selbst. Dabei gilt

$$\Pi = \begin{bmatrix} — & e_{\pi(1)}^{\mathsf{T}} & — \\ & \vdots & \\ — & e_{\pi(n)}^{\mathsf{T}} & — \end{bmatrix}$$

mit den Einheitsvektoren e_j des $\mathbb{R}^n$. Permutationsmatrizen sind orthogonale Matrizen, d. h., es gilt insbesondere $\Pi^{-1} = \Pi^{\mathsf{T}}$. Weil jede Permutation als Produkt von Transpositionen (Vertauschungen genau zweier Elemente) geschrieben werden kann, kann jede Permutationsmatrix als Produkt von Transpositionsmatrizen geschrieben werden. Transpositionsmatrizen sind ihre eigenen Inversen.

Beispiel 8.4 (Permutationsmatrizen). *Die Transpositionsmatrix $\Pi^{(24)}$ der Größe 5×5 ist*

$$\Pi^{(24)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Sie entspricht der Permutation $\pi(1) = 1$, $\pi(2) = 4$, $\pi(3) = 3$, $\pi(4) = 2$, $\pi(5) = 5$, kurz: $\pi = (14325)$. Die Permutationsmatrix

$$\Pi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

ist selbst keine Transpositionsmatrix, kann aber als Produkt der Transpositionsmatrizen

$$\Pi = \Pi^{(24)}\,\Pi^{(45)}$$

geschrieben werden. Die Inverse ist demnach $\Pi^{-1} = \Pi^{(45)}\,\Pi^{(24)}$.

Die Multiplikation einer Matrix A mit einer Transpositionsmatrix oder allgemeiner mit einer Permutationsmatrix Π von links (also ΠA) entspricht einer Permutation der Zeilen von A. **Quizfrage:** Was passiert, wenn man A von rechts mit Π multipliziert, also $A\Pi$ bildet?

Wir beleuchten zunächst wieder an einem Beispiel, dass Zeilenvertauschungen i. A. erforderlich sind, um die Existenz der LR-Zerlegung zu garantieren. Wir betrachten dazu das Gleichungssystem

$$\left[\begin{array}{ccc|c} 0 & 1 & 1 & 4 \\ 2 & 1 & 3 & 7 \\ 3 & 1 & 6 & 2 \end{array}\right].$$

Bevor wir die Einträge der ersten Spalte unterhalb der Diagonalen zu Null machen können, benötigen wir ein Element $\neq 0$ im Eintrag $(1,1)$. Dazu tauschen wir zwei Zeilen so, dass ein betragsgrößtes Element der ersten Spalte in der ersten Zeile steht. Im vorliegenden Fall tauschen wir also die Zeilen 1 und 3 durch Multiplikation von links mit der Transpositionsmatrix $\Pi^{(13)}$. Dann erzeugen wir wie gewohnt die Nullen unterhalb der Hauptdiagonale:

$$\left[\begin{array}{ccc|c} 3 & 1 & 6 & 2 \\ 2 & 1 & 3 & 7 \\ 0 & 1 & 1 & 4 \end{array}\right] \overset{\frac{-2}{3},\ \frac{0}{3}}{\rightsquigarrow} \left[\begin{array}{ccc|c} 3 & 1 & 6 & 2 \\ 0 & \frac{1}{3} & -1 & \frac{17}{3} \\ 0 & 1 & 1 & 4 \end{array}\right].$$

Im zweiten Schritt tauschen wir zunächst wieder die Zeilen von der zweiten Zeile an abwärts derart, dass ein betragsgrößtes Element der zweiten Spalte in der zweiten Zeile steht. (Das ist hier zwar jetzt nicht unbedingt erforderlich, da das $(2,2)$-Element nicht Null ist, erfolgt aber bereits im Hinblick auf die numerische Stabilität des Verfahrens.) Dazu benötigen wir die Transpositionsmatrix $\Pi^{(23)}$. Danach erzeugen wir die letzte Null unterhalb der Hauptdiagonale:

$$\left[\begin{array}{ccc|c} 3 & 1 & 6 & 2 \\ 0 & 1 & 1 & 4 \\ 0 & \frac{1}{3} & -1 & \frac{17}{3} \end{array}\right] \overset{\frac{-1}{3}}{\rightsquigarrow} \left[\begin{array}{ccc|c} 3 & 1 & 6 & 2 \\ 0 & 1 & 1 & 4 \\ 0 & 0 & -\frac{4}{3} & \frac{13}{3} \end{array}\right].$$

Die rot eingekreisten Elemente heißen **Pivotelemente** (englisch: *pivot elements, pivots*).[3]

Die obige Rechnung entspricht der Abfolge

$$G^{(2)}\,P^{(2)}\,\overbrace{G^{(1)}\,P^{(1)}\,\underbrace{A}_{=:A^{(1)}}}^{=:A^{(2)}} = R \tag{8.1}$$

mit Transpositionsmatrizen $P^{(k)}$ (die auch Einheitsmatrizen sein könnten) für den Zeilentausch und Frobeniusmatrizen $G^{(k)}$ für die Erzeugung der Nullen. In unserem Beispiel war

$$G^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{3} & 1 \end{bmatrix}, \quad P^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad G^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{2}{3} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad P^{(1)} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

[3] vom Französischen *pivot*: Dreh- und Angelpunkt

Die Transpositionsmatrix $P^{(r)} := \Pi^{(rs)}$ wurde jeweils so gewählt, dass dabei die r-te Zeile getauscht wurde mit einer späteren Zeile Nummer s, die die Bedingung

$$|a_{sr}^{(r)}| = \max\{|a_{ir}^{(r)}| \,\big|\, r \leqslant i \leqslant n\} \tag{8.2}$$

erfüllt. Diese Strategie heißt **Spaltenpivotsuche** (englisch: *column pivoting*). Der Name kommt daher, dass man in der aktuellen Spalte Nummer r ab dem Diagonalelement abwärts nach einem betragsgrößten Element sucht.

Um (8.1) als LR-Zerlegung von A mit Zeilentausch zu schreiben, müssen wir die abwechselnden Produkte von Transpositions- und Frobeniusmatrizen auflösen. Dabei hilft folgendes Resultat:

Lemma 8.5 (Produkt von Transpositions- und Frobeniusmatrizen)**.** *Es sei $G^{(k)} \in \mathbb{R}^{n\times n}$ eine k-Frobeniusmatrix und $G'^{(k)} := G^{(k)} - \mathrm{Id}$. Mit der Transpositionsmatrix $\Pi^{(rs)}$ gilt unter der Voraussetzung $1 \leqslant k < r, s \leqslant n$:*

$$\Pi^{(rs)} G^{(k)} = H^{(k)}\, \Pi^{(rs)}, \tag{8.3}$$

wobei

$$H^{(k)} := \mathrm{Id} + \Pi^{(rs)} G'^{(k)}$$

wieder eine Frobeniusmatrix ist.

Beachte: Das Lemma besagt, dass eine k-Frobeniusmatrix $G^{(k)}$, in der zwei Zeilen (r, s) unterhalb von k getauscht werden, als eine andere k-Frobeniusmatrix $H^{(k)}$ geschrieben werden kann, in der die zwei Spalten mit denselben Nummern (r, s) rechts von k getauscht werden.

Im Fall $r = s$ ist $\Pi^{(rs)} = \mathrm{Id}$, $H^{(k)} = G^{(k)}$ und die Aussage (8.3) klar. An Stelle eines formalen Beweises illustrieren wir die Aussage im Fall $n = 3$, $k = 1$, $r = 2$ und $s = 3$. Die Matrix auf der linken Seite von (8.3) ist

$$\underbrace{\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}}_{=\Pi^{(23)}}\underbrace{\begin{bmatrix}1&0&0\\g_2&1&0\\g_3&0&1\end{bmatrix}}_{=:G^{(k)}} = \begin{bmatrix}1&0&0\\g_3&0&1\\g_2&1&0\end{bmatrix}.$$

Die Matrix auf der rechten Seite von (8.3) ist tatsächlich identisch:

$$\left(\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} + \begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}\begin{bmatrix}0&0&0\\g_2&0&0\\g_3&0&0\end{bmatrix}\right)\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix} = \underbrace{\begin{bmatrix}1&0&0\\g_3&1&0\\g_2&0&1\end{bmatrix}}_{=:H^{(k)}}\underbrace{\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}}_{=\Pi^{(23)}} = \begin{bmatrix}1&0&0\\g_3&0&1\\g_2&1&0\end{bmatrix}.$$

Man sieht hier auch, wie die neue Frobeniusmatrix $H^{(k)}$ entsteht, nämlich durch den Austausch der Elemente (r, k) und (s, k) in der k-ten Spalte von $G^{(k)}$.

Das in

$$G^{(2)}\, P^{(2)}\, G^{(1)}\, P^{(1)} A = R \tag{8.1}$$

vorkommende Produkt $P^{(2)}G^{(1)}$ erfüllt die Vorausetzungen von Lemma 8.5, da $G^{(1)}$ eine 1-Frobeniusmatrix ist und $P^{(2)}$ entweder die Einheitsmatrix oder eine Transpositionsmatrix ist, die die zweite Zeile ($r = 2$) mit einer späteren Zeile ($s > r$) vertauscht. Daher können wir (8.1) auch schreiben in der Form

$$G^{(2)}\, H^{(1)}\, P^{(2)}\, P^{(1)}\, A = R. \tag{8.4}$$

Wir können diese Gleichung nun von links mit der unteren Dreiecksmatrix (beachte Lemma 8.1 (*iii*))

$$L := \left(H^{(1)}\right)^{-1}\left(G^{(2)}\right)^{-1}$$

multiplizieren und erhalten schließlich eine **LR-Zerlegung mit Pivotisierung**

$$PA = LR, \tag{8.5}$$

wobei die Permutationsmatrix

$$P := P^{(2)}\, P^{(1)}$$

durch das Produkt der Transpositionsmatrizen entsteht, die in den einzelnen Zerlegungsschritten bestimmt werden. Die Matrix L kann unter Ausnutzung von Aussage (*iii*) aus Lemma 8.1 spaltenweise von links nach rechts aufgebaut werden.

In der Praxis wird die LR-Zerlegung wie folgt implementiert:

(1) Die Matrix A wird nach und nach *in-place* überschrieben und nimmt spaltenweise die Einträge von L strikt unterhalb der Diagonale auf und zeilenweise die Einträge von R ab der Diagonale aufwärts. (**Quizfrage:** Warum ist das Überschreiben der Matrix A während der noch laufenden LR-Zerlegung möglich?)

(2) Die Vertauschungen werden in einem separaten Vektor $p \in \mathbb{N}^n$ gespeichert.

(3) Statt zunächst mit den Matrizen $G^{(k)}$ zu arbeiten, diese anschließend gemäß Lemma 8.5 umzuschreiben und dann zu invertieren, wird direkt die Matrix L Spalte für Spalte aufgebaut.

In unserem obigen Beispiel führt das zu folgenden Schritten. Zunächst initialisieren wir den Permutationsvektor als $p := (1, 2, \ldots, n)^\mathsf{T}$. Dann pivotisieren wir die erste Spalte, d. h., wir tauschen die Zeilen 1 und 3 und entsprechend auch die Einträge 1 und 3 im Vektor p:

$$\begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & 3 \\ 3 & 1 & 6 \end{bmatrix} \quad \rightsquigarrow \quad \begin{bmatrix} 3 & 1 & 6 \\ 2 & 1 & 3 \\ 0 & 1 & 1 \end{bmatrix}, \qquad p := \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}.$$

Wir erzeugen Nullen in der ersten Spalte (die nicht gespeichert werden) und speichern die (vorläufige) erste Spalte von L — das ist die erste Spalte von $(G^{(1)})^{-1}$ — strikt unterhalb der Diagonale an Stelle der entstehenden Nullen. Die erste Zeile von R liegt ebenfalls schon fest:

$$\begin{bmatrix} 3 & 1 & 6 \\ 2 & 1 & 3 \\ 1 & 1 & 1 \end{bmatrix} \quad \rightsquigarrow \quad \begin{bmatrix} 3 & 1 & 6 \\ \frac{2}{3} & \frac{1}{3} & -1 \\ 0 & 1 & 1 \end{bmatrix}, \qquad p := \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}.$$

Wir pivotisieren jetzt die zweite Spalte, d. h., tauschen die Zeilen 2 und 3 und entsprechend die Einträge 2 und 3 im Vektor p. Dabei werden auch die Einträge der bereits gespeicherten ersten Spalte von L mit getauscht:

$$\begin{bmatrix} 3 & 1 & 6 \\ \frac{2}{3} & \frac{1}{3} & -1 \\ 0 & 1 & 1 \end{bmatrix} \rightsquigarrow \begin{bmatrix} 3 & 1 & 6 \\ 0 & 1 & 1 \\ \frac{2}{3} & \frac{1}{3} & -1 \end{bmatrix}, \quad p := \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}.$$

Wir erzeugen jetzt die letzte fehlende Null in der zweiten Spalte (die wieder nicht gespeichert wird) und speichern die zweite Spalte von L strikt unterhalb der Diagonale an Stelle der entstehenden Null. Die zweite Zeile von R liegt ebenfalls schon fest:

$$\begin{bmatrix} 3 & 1 & 6 \\ 0 & 1 & 1 \\ \frac{2}{3} & \frac{1}{3} & -1 \end{bmatrix} \rightsquigarrow \begin{bmatrix} 3 & 1 & 6 \\ 0 & 1 & 1 \\ \frac{2}{3} & \frac{1}{3} & -\frac{4}{3} \end{bmatrix}, \quad p := \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}.$$

Damit ist die Faktorisierung beendet, und wir erhalten

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{2}{3} & \frac{1}{3} & 1 \end{bmatrix} \quad \text{und} \quad R = \begin{bmatrix} 3 & 1 & 6 \\ 0 & 1 & 1 \\ 0 & 0 & -\frac{4}{3} \end{bmatrix}.$$

Die Permutationsmatrix P ergibt sich aus dem Vektor p. Dessen Einträge geben an, in welcher Reihenfolge die Zeilen der $n \times n$-Einheitsmatrix in P erscheinen. Im Beispiel ist

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Wir führen noch die Probe durch:

$$PA = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & 3 \\ 3 & 1 & 6 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 6 \\ 0 & 1 & 1 \\ 2 & 1 & 3 \end{bmatrix},$$

$$LR = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{2}{3} & \frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & 6 \\ 0 & 1 & 1 \\ 0 & 0 & -\frac{4}{3} \end{bmatrix} = \begin{bmatrix} 3 & 1 & 6 \\ 0 & 1 & 1 \\ 2 & 1 & 3 \end{bmatrix}.$$

Mit Hilfe von NumPy und SciPy kann eine LR-Zerlegung von A wie folgt bestimmt werden:

```
import numpy as np
import scipy.linalg
A = np.array([[0,1,1],[2,1,3],[3,1,6]])
P, L, R = scipy.linalg.lu(A)
```

Beachte: Durch `scipy.linalg.lu` wird eine Zerlegung bestimmt, sodass $A = P\,L\,R$ gilt, also $P^{\mathsf{T}}A = L\,R$. Im Vergleich zu der von uns benutzten Konvention ist die Permutationsmatrix also gerade transponiert!

Mit den gerade beschriebenen Techniken kann man einen Satz über die Existenz von LR-Zerlegung mit Spaltenpivotisierung zeigen. Dazu muss man im Prinzip nur zeigen, dass es immer gelingt, in jeder Spalte von der Diagonale an abwärts mindestens ein Element zu finden, das nicht Null ist. Diese Zerlegung

ist nicht mehr eindeutig (vgl. Lemma 8.2), da es i. A. verschiedene mögliche Permutationsmatrizen P gibt, die zu nicht-verschwindenden Pivot-Elementen führen.

Satz 8.6 (Existenz einer LR-Zerlegung, vgl. Bornemann, 2018, Satz 7.11)**.** *Es sei $A \in \mathbb{R}^{n\times n}$ regulär. Wird die Permutationsmatrix $P \in \mathbb{R}^{n\times n}$ gemäß dem Prinzip der Spaltenpivotsuche gewählt, dann besitzt PA eine (eindeutige) LR-Zerlegung. Zu einem solchen P existiert also eine untere Dreiecksmatrix L mit lauter Einsen auf der Diagonale und eine obere Dreiecksmatrix R, sodass $PA = LR$ ist. Weiterhin gilt, dass die Einträge von L betragsweise $\leqslant 1$ sind.*

Quizfrage: Können Sie begründen, warum die Einträge von L betragsweise $\leqslant 1$ sind, also $|L| \leqslant 1$ gilt?

Beachte: Durch die Spaltenpivotsuche wird P nicht unbedingt eindeutig festgelegt, da durchaus mehrere Einträge der aktuellen Spalte gleichzeitig betragsmaximal sein können.

Da klar ist, dass wir selbst bei exakter Rechnung eine Spaltenpivotisierung für die LR-Zerlegung benötigen, werden wir das in Zukunft nicht immer betonen und einfach von LR-Zerlegungen von A sprechen.

Frage: Was sind die Vorteile, eine LR-Zerlegung einer Matrix A zu erstellen und zu speichern?

Wenn eine LR-Zerlegung von A vorliegt, können wir jedes lineare Gleichungssystem mit regulärer Koeffizientenmatrix A schnell lösen:

Algorithmus 8.7 (Lösung linearer Gleichungssysteme bei Vorliegen einer LR-Zerlegung mit Spaltenpivotisierung)**.**
Eingabe: *untere Dreiecksmatrix L mit lauter Einsen auf der Diagonale*
Eingabe: *obere Dreiecksmatrix R*
Eingabe: *Permutationsmatrix P, sodass gilt: $PA = LR$*
Eingabe: *Vektor der rechten Seite b*
Ausgabe: *Lösung des linearen Gleichungssystems $Ax = b$*
1. *Löse $Ly = Pb$ durch Vorwärtseinsetzen*
2. *Löse $Rx = y$ durch Rückwärtseinsetzen*

Quizfrage: Können Sie die Korrektheit von Algorithmus 8.7 begründen, also dass durch ihn $Ax = b$ gelöst wird?

Die Zwischengröße y berechnet sich dabei aus $c := Pb$ gemäß

$$y_i := c_i - \sum_{k=1}^{i-1} \ell_{ik}\, y_k, \quad i = 1, 2, \ldots, n. \tag{8.6}$$

Die anschließende Rückwärtssubstitution gelingt durch

$$x_i := \frac{1}{r_{ii}} \left(y_i - \sum_{k=i+1}^{n} r_{ik}\, x_k \right), \quad i = n, n-1, \ldots, 1. \tag{8.7}$$

Bei bereits vorliegender LR-Zerlegung ist der numerische Aufwand für die Lösung von $Ax = b$ durch Algorithmus 8.7 von der Größenordnung $2n^2$ Grundoperationen (+, −, · und /), genauer: $n^2 - n$ für die Lösung des Systems mit L und n^2 für die Lösung des Systems mit R. Man spricht dabei auch von **FLOPs** (englisch: *floating point operations*). **Quizfrage:** Ist die Ermittlung des Aufwands klar?

Quizfrage: Wie können wir mit Hilfe von Algorithmus 8.7 mehrere lineare Gleichungssysteme gleichzeitig lösen, die dieselbe Koeffizientenmatrix A besitzen? Wie groß ist dann der Aufwand?

Der Aufwand für die eigentliche Erzeugung der LR-Zerlegung wird durch folgendes Resultat bestimmt:

Satz 8.8 (Aufwand der LR-Zerlegung)**.** *Der numerische Aufwand für die Erzeugung der LR-Zerlegung einer regulären Matrix $A \in \mathbb{R}^{n\times n}$ mit Spaltenpivotsuche ist von der Größenordnung $\frac{2}{3}n^3 + O(n^2)$.*

Beweis. Die Manipulation der k-ten Spalte erfordert

- einen Aufwand von $n - k + 1$ Beträgen und Vergleichen für die Bestimmung des Pivotelements (**Quizfrage:** Warum?)
- einen Aufwand von $n - k + 1$ Multiplikationen und ebensovielen Additionen für das Bilden der Linearkombination der k-ten Zeile mit *einer* der darunter stehenden Zeilen (**Quizfrage:** Warum sind das nicht n Multiplikation und Additionen?)
- Es gibt $n-k$ solcher darunter stehenden Zeilen, also ist der Aufwand für die Linearkombinationen (die zur Erzeugung der Nullen unterhalb der Hauptdiagonale in Schritt k führen) insgesamt $2(n - k)(n - k + 1)$ FLOPs.

Die Summation über die Zeilennummern $k = 1, \ldots, n$ liefert einen Gesamtaufwand proportional zu

$$\sum_{k=1}^{n}(n - k + 1) = \frac{1}{2}n(n + 1) \in O(n^2)$$

FLOPs für die Spaltenpivotsuche und von

$$\sum_{k=1}^{n} 2(n - k)(n - k + 1) = \frac{2}{3}n(n^2 - 1) \in \frac{2}{3}n^2 + O(n^2)$$

FLOPs für die Linearkombinationen. Die Spaltenpivotsuche fällt dabei also nicht weiter ins Gewicht. □

Beachte: Die Lösung eines linearen Gleichungssystems $Ax = b$ mit Hilfe der Gauß-Elimination hat den Aufwand $\frac{2}{3}n^3 + O(n^2)$. Wenn man die dabei ohnehin notwendigenden Manipulationen von A in Form einer LR-Zerlegung speichert, dann kann man nachfolgende Gleichungssysteme mit derselben Matrix A in $2n^2$ FLOPs lösen. Es ist also sinnvoll und kein zusätzlicher Aufwand, bei der Lösung eines linearen Gleichungssystems mit Hilfe des Gauß-Verfahrens immer die LR-Zerlegung für eine eventuelle zukünftige Verwendung zu speichern.

Quizfrage: Kann man die LR-Zerlegung von A auch dazu verwenden, wenn man lineare Gleichungssysteme mit A^T lösen muss? Oder mit A^2?

Die Erstellung der LR-Zerlegung einer regulären Matrix A hat noch weitere Vorteile:

(1) Wir können mittels einer LR-Zerlegung die Determinante berechnen:

$$\det A = \det(P^\mathsf{T} LR) = (\det P)(\det L)(\det R) = \pm \det R,$$

wobei sich das Vorzeichen danach richtet, ob die durch P dargestellte Permutation das Produkt einer geraden oder einer ungeraden Anzahl von Transpositionen ist.

(2) Wir können mittels einer LR-Zerlegung die Inverse berechnen: Dazu lösen wir n lineare Gleichungssysteme $AX = B$ mit rechter Seite $B = \mathrm{Id}$. Die Lösung X ist dann die Inverse von A.

Bemerkung 8.9 (Totalpivotisierung)**.** *An Stelle der bisher verwendeten Spaltenpivotsuche kann auch eine* ***totale Pivotsuche*** *(englisch:* total pivoting*) durchgeführt werden. Dabei wird an Stelle von* (8.2) *im r-ten Schritt die Strategie*

$$|a_{st}^{(r)}| = \max\{|a_{ij}^{(r)}| \,\big|\, r \leqslant i, j \leqslant n\} \tag{8.8}$$

verfolgt. Es wird also bei der Suche nach dem betragsgrößten Pivotelement nicht nur nach unten geschaut, sondern auch nach rechts. Entsprechend werden nicht nur die Zeilen r und s getauscht, sondern zusätzlich auch die Spalten r und t. Das entspricht einer Umsortierung der Variablen im Gleichungssystem. Dieses Verfahren führt zu einer ***LR-Zerlegung mit Totalpivotisierung*** *der Form*

$$PAQ^\mathsf{T} = LR, \tag{8.9}$$

wobei Q ebenfalls eine Permutationsmatrix ist. Der Aufwand für die Suche nach dem Pivotelement ist jetzt ebenfalls von der Größenordnung $O(n^3)$, sodass das Gesamtverfahren für die LR-Zerlegung noch immer von der Größenordnung $O(n^3)$ FLOPs ist. ***Quizfrage:*** *Wieviele Vergleiche sind bei der totalen Pivotsuche sind genau erforderlich?*

Der Vollständigkeit halber geben wir auch für den Fall der LR-Zerlegung mit totaler Pivotisierung das Verfahren zur Lösung eines linearen Gleichungssystems $Ax = b$ analog zu Algorithmus 8.7 an:

Algorithmus 8.10 (Lösung linearer Gleichungssysteme bei Vorliegen einer LR-Zerlegung mit Totalpivotisierung)**.**
Eingabe: *untere Dreiecksmatrix L mit lauter Einsen auf der Diagonale*
Eingabe: *obere Dreiecksmatrix R*
Eingabe: *Permutationsmatrizen P und Q, sodass gilt: $PAQ^\mathsf{T} = LR$*
Eingabe: *Vektor der rechten Seite b*
Ausgabe: *Lösung des linearen Gleichungssystems $Ax = b$*
1: *Löse $Ly = Pb$ durch Vorwärtseinsetzen*
2: *Löse $Rx = y$ durch Rückwärtseinsetzen*
3: *Setze $x := Q^\mathsf{T} x$*

Ende der Woche 6

§ 9 Fehleranalyse bei der Lösung linearer Gleichungssysteme

Wir untersuchen jetzt die Stabilität der algorithmischen Komponenten bei der Lösung linearer Gleichungssysteme. Wir beginnen mit der Lösung von Systemen mit unteren und oberen Dreiecksmatrizen, $Lx = b$ bzw. $Rx = b$. Etwas allgemeiner als eigentlich erforderlich bestehen wir dabei nicht daraus, dass die Diagonale von L aus lauter Einsen besteht.[4] Wir betrachten folgende konkrete Implementierungen der Vorwärtssubstitution (8.6) bzw. Rückwärtssubstitution (8.7):

Algorithmus 9.1 (Lösung linearer Gleichungssysteme mit unterer Dreiecksmatrix durch Vorwärtssubstitution)**.**
Eingabe: *reguläre untere Dreiecksmatrix L (nicht notwendig mit Einsen auf der Diagonale)*
Eingabe: *Vektor der rechten Seite b*
Ausgabe: *Lösung des linearen Gleichungssystems $Ly = c$*

1: ***for*** $i = 1, 2, \dots, n$ ***do***
2: Setze $s_i := c_i$
3: ***for*** $j = i-1, i-2, \dots, 1$ ***do***
4: Setze $s_i := s_i - \ell_{ij}\, y_j$
5: ***end for***
6: Setze $y_i := \frac{s_i}{\ell_{ii}}$
7: ***end for***

Algorithmus 9.2 (Lösung linearer Gleichungssysteme mit oberer Dreiecksmatrix durch Rückwärtssubstitution, vgl. Higham, 2002, Algorithm 8.1)**.**
Eingabe: *reguläre obere Dreiecksmatrix R*
Eingabe: *Vektor der rechten Seite b*
Ausgabe: *Lösung des linearen Gleichungssystems $Rx = b$*

1: ***for*** $i = n, n-1, \dots, 1$ ***do***
2: Setze $s_i := b_i$
3: ***for*** $j = i+1, i+2, \dots, n$ ***do***
4: Setze $s_i := s_i - r_{ij}\, x_j$
5: ***end for***
6: Setze $x_i := \frac{s_i}{r_{ii}}$
7: ***end for***

Beachte: Beide Algorithmen sind so formuliert, dass die Summation in s_i der Beiträge «von der Diagonale aus nach außen» erfolgt. Diese Reihenfolge erweist sich als günstig für die Fehlerabschätzung in den Sätze 9.4 und 9.5.

Für die Fehlerabschätzung benötigen wir folgendes Lemma.

Lemma 9.3 (vgl. Higham, 2002, Lemma 3.1)**.** *Für $n \in \mathbb{N}$ seien $\delta_1, \dots, \delta_n$ Zahlen mit $|\delta_i| \leqslant \varepsilon_{\text{mach}}$ und*

[4] Das ist praktisch u. a. bereits im Vorgriff auf die Cholesky-Zerlegung in Abschnitt 10.2.

$\rho_1, \dots, \rho_n \in \{\pm 1\}$. *Falls* $n\,\varepsilon_{\text{mach}} \leqslant C < 1$ *gilt, dann ist*

$$\prod_{i=1}^{n} (1+\delta_i)^{\rho_i} = 1 + \theta,$$

und θ *erfüllt die Abschätzung*

$$|\theta| \leqslant \gamma_n := \frac{n\,\varepsilon_{\text{mach}}}{1 - n\,\varepsilon_{\text{mach}}} \leqslant \frac{n\,\varepsilon_{\text{mach}}}{1-C}.$$

Beweis. Wir führen einen Induktionsbeweis. Für $n = 1$ gilt im Fall von $\rho_1 = 1$

$$\theta = \delta_1$$

und daher $|\theta| \leqslant \varepsilon_{\text{mach}} < \frac{1\,\varepsilon_{\text{mach}}}{1 - 1\,\varepsilon_{\text{mach}}}$. Im Fall von $\rho_1 = -1$ gilt

$$\theta = \frac{1}{1+\delta_1} - 1 = -^{\text{GM}} \frac{\delta_1}{1+\delta_1},$$

und damit

$$|\theta| = \frac{|\delta_1|}{|1+\delta_1|} \leqslant \frac{1\,\varepsilon_{\text{mach}}}{1 - 1\,\varepsilon_{\text{mach}}}.$$

~~und wegen $|1+\delta_1| \geqslant 1$ gilt ebenfalls $|\theta| \leqslant \varepsilon_{\text{mach}} < \frac{1\,\varepsilon_{\text{mach}}}{1-1\,\varepsilon_{\text{mach}}}$~~ GM

Nun zum Induktionsschritt. Die Behauptung sei bereits bewiesen für gegebenes n. Im Fall von $\rho_{n+1} = 1$ gilt

$$\prod_{i=1}^{n+1} (1+\delta_i)^{\rho_i} = (1+\theta)(1+\delta_{n+1}) =: 1 + \theta',$$

also ist $\theta' = \theta(1+\delta_{n+1}) + \delta_{n+1}$. Wir schätzen ab:

$$\begin{aligned}
|\theta'| &\leqslant |\theta||1+\delta_{n+1}| + |\delta_{n+1}| \\
&\leqslant \frac{n\,\varepsilon_{\text{mach}}}{1 - n\,\varepsilon_{\text{mach}}} (1 + \varepsilon_{\text{mach}}) + \varepsilon_{\text{mach}} \quad \text{(nach Induktionsvoraussetzung)} \\
&= \frac{n\,\varepsilon_{\text{mach}} + n\,\varepsilon_{\text{mach}}^2 + \varepsilon_{\text{mach}} - n\,\varepsilon_{\text{mach}}^2}{1 - n\,\varepsilon_{\text{mach}}} \\
&= \frac{(n+1)\,\varepsilon_{\text{mach}}}{1 - n\,\varepsilon_{\text{mach}}} \\
&\leqslant \frac{(n+1)\,\varepsilon_{\text{mach}}}{1 - (n+1)\,\varepsilon_{\text{mach}}}.
\end{aligned}$$

Im Fall von $\rho_{n+1} = -1$ haben wir hingegen

$$\prod_{i=1}^{n+1} (1+\delta_i)^{\rho_i} = \frac{1+\theta}{1+\delta_{n+1}} =: 1 + \theta',$$

also ist

$$\theta' = \frac{1+\theta}{1+\delta_{n+1}} - 1 = \frac{\theta - \delta_{n+1}}{1+\delta_{n+1}}.$$

Wir schätzen ab: (Die folgende Gleichung wurde ausgetauscht.)[GM]

$$\begin{aligned}
|\theta'| &= \frac{|\theta - \delta_{n+1}|}{|1 + \delta_{n+1}|} \\
&\leqslant \frac{1}{|1 + \delta_{n+1}|} \left(|\theta| + |\delta_{n+1}|\right) \\
&\leqslant \frac{1}{1 - \varepsilon_{\text{mach}}} \left(\frac{n\varepsilon_{\text{mach}}}{1 - n\varepsilon_{\text{mach}}} + \varepsilon_{\text{mach}} \right) \\
&= \frac{1}{1 - \varepsilon_{\text{mach}}} \left(\frac{n\varepsilon_{\text{mach}} + \varepsilon_{\text{mach}} - n\varepsilon_{\text{mach}}^2}{1 - n\varepsilon_{\text{mach}}} \right) \\
&\leqslant \frac{(n+1)\varepsilon_{\text{mach}}}{(1 - \varepsilon_{\text{mach}})(1 - n\varepsilon_{\text{mach}})} \\
&= \frac{(n+1)\varepsilon_{\text{mach}}}{1 + n\varepsilon_{\text{mach}}^2 - \varepsilon_{\text{mach}} - n\varepsilon_{\text{mach}}^2} \\
&\leqslant \frac{(n+1)\varepsilon_{\text{mach}}}{1 - (n+1)\varepsilon_{\text{mach}}}
\end{aligned}$$

Damit ist in beiden Fällen der Induktionsschritt gezeigt. Die Aussage $\gamma_n \leqslant \frac{n\,\varepsilon_{\text{mach}}}{1-c}$ ist wegen $1 - n\,\varepsilon_{\text{mach}} \geqslant 1 - c$ klar. □

Wir können nun ein Resultat zur Rückwärtsstabilität von Algorithmus 9.1 zeigen, wobei nur die Matrix L gestört ist und die rechte Seite nicht.

Satz 9.4 (Rückwärtsanalyse von Algorithmus 9.1, vgl. Higham, 2002, Theorem 8.3)**.** *Es sei $L \in \mathbb{F}^{n\times n}$ eine reguläre untere Dreicksmatrix, $c \in \mathbb{F}^n$ und $n\,\varepsilon_{\text{mach}} \leqslant C < 1$. Weiter sei $\widehat{y}$ die von Algorithmus 9.1 in Maschinenoperationen berechnete Näherungslösung. Dann gibt es eine untere Dreicksmatrix ΔL, sodass gilt*

$$(L + \Delta L)\,\widehat{y} = c \quad \textit{mit} \quad |\Delta \ell_{ij}| \leqslant \begin{cases} \gamma_i\,|\ell_{ii}| & \textit{für } i = j, \\ \gamma_{i-j}\,|\ell_{ij}| & \textit{für } i > j \end{cases}$$

mit $\gamma_i := \frac{i\,\varepsilon_{\text{mach}}}{1 - i\,\varepsilon_{\text{mach}}} \leqslant \frac{i\,\varepsilon_{\text{mach}}}{1-C}$.

Quizfrage: Inwiefern ist dies ein Resultat über die Rückwärtsstabilität von Algorithmus 9.1?

Beweis. Wir beginnen mit $\widehat{y}_1 = \widehat{s}_1 \oslash \ell_{11} = \frac{c_1}{\ell_{11}}(1 + \delta_1)$. Dabei ist $\widehat{s}_1 = s_1 = c_1$ (ohne Fehler, da $c \in \mathbb{F}^n$) und $|\delta_1| \leqslant \varepsilon_{\text{mach}}$. Daraus folgt

$$\frac{1}{1 + \delta_1}\,\ell_{11}\,\widehat{y}_1 = (1 + \theta)\,\ell_{11}\,\widehat{y}_1 = c_1$$

mit einem $|\theta| \leqslant \gamma_1$. Es gilt also $\Delta\ell_{11} := \theta\,\ell_{11}$ mit $|\Delta\ell_{11}| = |\theta|\,|\ell_{11}| \leqslant \gamma_1\,|\ell_{11}|$.

An Stelle einer formalen Induktion zeigen wir noch die nächsten zwei Schritte, zunächst also $i = 2$. Dabei gilt

$$\begin{aligned}
\widehat{s}_2 &= c_2 \ominus (\ell_{21} \odot \widehat{y}_1) \\
&= c_2 \ominus (\ell_{21}\,\widehat{y}_1\,(1 + \varepsilon_1)) \\
&= c_2\,(1 + \varepsilon_2) - \ell_{21}\,\widehat{y}_1\,(1 + \varepsilon_1)\,(1 + \varepsilon_2)
\end{aligned}$$

und weiter

$$\widehat{y}_2 = \widehat{s}_2 \oslash \ell_{22} = \frac{\widehat{s}_2}{\ell_{22}} (1 + \delta_2).$$

Daraus folgt

$$\frac{1}{1+\delta_2} \ell_{22} \widehat{y}_2 = \widehat{s}_2 = c_2 (1+\varepsilon_2) - \ell_{21} \widehat{y}_1 (1+\varepsilon_1) (1+\varepsilon_2).$$

Sortieren der Terme ergibt

$$\frac{1}{1+\delta_2} \frac{1}{1+\varepsilon_2} \ell_{22} \widehat{y}_2 + \ell_{21} \widehat{y}_1 (1+\varepsilon_1) = c_2.$$

Wir setzen $1 + \theta := \frac{1}{1+\delta_2} \frac{1}{1+\varepsilon_2}$. Dann gilt $|\theta| \leqslant \gamma_2$ und $|\varepsilon_1| \leqslant \gamma_1$, und mit $\Delta\ell_{22} := \theta\, \ell_{22}$ und $\Delta\ell_{21} := \varepsilon_1\, \ell_{21}$ ist auch die zweite Zeile von $(L + \Delta L)\, \widehat{y} = c$ erfüllt.

Schließlich betrachten wir noch den Index $i = 3$. Dabei gilt

$$\begin{aligned}
\widehat{s}_3 &= (c_3 \ominus (\ell_{32} \odot \widehat{y}_2)) \ominus (\ell_{31} \odot \widehat{y}_1) \\
&= (c_3 \ominus (\ell_{32} \widehat{y}_2 (1+\varepsilon_3))) \ominus (\ell_{31} \odot \widehat{y}_1) \\
&= (c_3 (1+\varepsilon_4) - \ell_{32} \widehat{y}_2 (1+\varepsilon_3) (1+\varepsilon_4)) \ominus (\ell_{31} \widehat{y}_1 (1+\varepsilon_5)) \\
&= c_3 (1+\varepsilon_4) (1+\varepsilon_6) - \ell_{32} \widehat{y}_2 (1+\varepsilon_3) (1+\varepsilon_4) (1+\varepsilon_6) - \ell_{31} \widehat{y}_1 (1+\varepsilon_5) (1+\varepsilon_6)
\end{aligned}$$

und weiter

$$\widehat{y}_3 = \widehat{s}_3 \oslash \ell_{33} = \frac{\widehat{s}_3}{\ell_{33}} (1 + \delta_3).$$

Daraus folgt

$$\frac{1}{1+\delta_3} \ell_{33} \widehat{y}_3 = \widehat{s}_3 = c_3 (1+\varepsilon_4) (1+\varepsilon_6) - \ell_{32} \widehat{y}_2 (1+\varepsilon_3) (1+\varepsilon_4) (1+\varepsilon_6) - \ell_{31} \widehat{y}_1 (1+\varepsilon_5) (1+\varepsilon_6).$$

Sortieren der Terme ergibt

$$\frac{1}{1+\delta_3} \frac{1}{1+\varepsilon_4} \frac{1}{1+\varepsilon_6} \ell_{33} \widehat{y}_3 + \ell_{32} \widehat{y}_2 (1+\varepsilon_3) + \ell_{31} \widehat{y}_1 \frac{1+\varepsilon_5}{1+\varepsilon_4} = c_3.$$

Wir setzen $1 + \theta := \frac{1}{1+\delta_2} \frac{1}{1+\varepsilon_4} \frac{1}{1+\varepsilon_6}$ und $1 + \theta' := \frac{1+\varepsilon_5}{1+\varepsilon_4}$. Dann gilt $|\theta| \leqslant \gamma_3$ und $|\theta'| \leqslant \gamma_2$ sowie $|\varepsilon_3| \leqslant \gamma_1$, und mit $\Delta\ell_{33} := \theta\, \ell_{33}$, $\Delta\ell_{32} := \varepsilon_3\, \ell_{32}$ sowie $\Delta\ell_{31} := \theta'\, \ell_{31}$ ist auch die dritte Zeile von $(L + \Delta L)\, \widehat{y} = c$ erfüllt. □

Ein analoges Resultat gilt auch für Gleichungssysteme mit oberer Dreiecksmatrix und Algorithmus 9.2:

Satz 9.5 (Rückwärtsanalyse von Algorithmus 9.2, vgl. Higham, 2002, Theorem 8.3). *Es sei $R \in \mathbb{F}^{n\times n}$ eine reguläre obere Dreicksmatrix, $b \in \mathbb{F}^n$ und $n\, \varepsilon_{\text{mach}} \leqslant C < 1$. Weiter sei $\widehat{x}$ die von Algorithmus 9.2 in Maschinenoperationen berechnete Näherungslösung. Dann gibt es eine obere Dreicksmatrix ΔR, sodass gilt*

$$(R + \Delta R)\, \widehat{x} = b \quad \textit{mit} \quad |\Delta r_{ij}| \leqslant \begin{cases} \gamma_{n+1-i}\, |r_{ii}| & \textit{für } i = j, \\ \gamma_{j-i}\, |r_{ij}| & \textit{für } i < j \end{cases}$$

mit $\gamma_i := \frac{i\, \varepsilon_{\text{mach}}}{1 - i\, \varepsilon_{\text{mach}}} \leqslant \frac{i\, \varepsilon_{\text{mach}}}{1-C}$.

Die größte in den Abschätzungen von Satz 9.4 und Satz 9.5 vorkommende Konstante ist

$$\gamma_n = \frac{n\,\varepsilon_{\text{mach}}}{1 - n\,\varepsilon_{\text{mach}}} \leqslant \frac{n\,\varepsilon_{\text{mach}}}{1 - C}.$$

Daraus folgt sofort folgendes Resultat:

Folgerung 9.6. *Unter den Voraussetzungen von Satz 9.4 und Satz 9.5 gilt:*

$$(L + \Delta L)\,\widehat{y} = c \quad \textit{mit} \quad |\Delta L| \leqslant \gamma_n\,|L| \leqslant \frac{n}{1 - C}\,|L|\,\varepsilon_{\text{mach}}, \tag{9.1a}$$

$$(R + \Delta R)\,\widehat{x} = b \quad \textit{mit} \quad |\Delta R| \leqslant \gamma_n\,|R| \leqslant \frac{n}{1 - C}\,|R|\,\varepsilon_{\text{mach}}. \tag{9.1b}$$

Wenn

Wir verwenden hier und im Folgenden die Notation $|A|$ für den komponentenweisen Betrag einer Matrix A, also für diejenige Matrix, deren Einträge sich aus den Absolutbeträgen der Einträge von A ergeben.

§ 9.1 Fall ohne Pivotsuche

Es bleibt noch eine Fehlerabschätzung für die Erstellung einer LR-Zerlegung für eine Matrix A herzuleiten. Wir tun dies zuerst für den Fall, dass die LR-Zerlegung ohne Piovitisierung berechnet wird. Wir betrachten dazu konkret das folgende Verfahren:

Algorithmus 9.7 (Berechnung der LR-Zerlegung ohne Pivotisierung, vgl. Golub, van Loan, 1983, Algorithm 3.2.1)**.**
Eingabe: *reguläre Matrix A*
Ausgabe: *Faktoren L und R der LR-Zerlegung $A = LR$*

1: ***for*** $k = 1, 2, \ldots, n-1$ ***do***
2: *Setze* $I := k+1 : n = \{k+1, k+2, \ldots, n\}$
3: *Setze* $A(I,k) := A(I,k)/A(k,k)$
4: *Setze* $A(I,I) := A(I,I) - A(I,k)\,A(k,I)$
5: ***end for***
6: *Extrahiere L aus dem strikteren unteren Dreieck von A und ergänze Einsen auf der Diagonale*
7: *Extrahiere R aus dem oberen Dreieck von A*

Quizfrage: Ist klar, dass Algorithmus 9.7 das oben («In der Praxis wird die LR-Zerlegung wie folgt implementiert») anhand von Beispielen erklärte Verfahren der Transformation *in-place* (im Fall ohne Pivotisierung) implementiert?

Die Zeilen 6 und 7 erzeugen die Faktoren L und R der Zerlegung $A = LR$, ohne weitere Rundungsfehler hinzuzufügen. **Quizfrage:** Klar?

Satz 9.8 (Rückwärtsanalyse der LR-Zerlegung ohne Pivotisierung, vgl. Golub, van Loan, 1983, Theorem 3.3.1). *Es sei $A \in \mathbb{F}^{n\times n}$ eine reguläre Matrix. Weiter seien $\widehat{L}$ and $\widehat{R}$ die von Algorithmus 9.7 in Maschinenoperationen berechnete Näherungslösung der LR-Zerlegung von $A = LR$. Dabei nehmen wir an, dass in jeder Iteration das Pivotelement $\widehat{A}(k,k) \neq 0$ ist. Dann gibt es eine Matrix ΔA, sodass gilt:*

$$\widehat{L}\,\widehat{R} = A + \Delta A \quad \textit{mit} \quad |\Delta A| \dot{\leqslant} 3\,(n-1)\big(|A| + |\widehat{L}|\,|\widehat{R}|\big)\,\varepsilon_{\text{mach}}. \tag{9.2}$$

Beachte: $|\widehat{L}|\,|\widehat{R}|$ bezeichnet das Produkt der Matrizen $|\widehat{L}|$ und $|\widehat{R}|$, also $\big(|\widehat{L}|\,|\widehat{R}|\big)_{ij} = \sum_{k=1}^{n} |\widehat{\ell}_{ik}|\,|\widehat{r}_{kj}|$.

Beweis. Wir führen einen Induktionsbeweis über die Dimension n. Für $n = 1$ führt Algorithmus 9.7 keine Berechnungen durch und liefert $\widehat{A} = A$ zurück. Also ist $\Delta A = 0$ und (9.2) erfüllt.

Wir machen jetzt den Induktionsschritt von n auf $n+1$. Dazu benennen wir die Einträge der Matrix $A \in \mathbb{F}^{(n+1)\times(n+1)}$ wie folgt:

$$A = \left[\begin{array}{c|c} \alpha & w^{\mathsf{T}} \\ \hline v & B \end{array}\right]$$

mit $v, w \in \mathbb{F}^n$ und $\alpha \in \mathbb{F}$. Der Algorithmus 9.7 führt in der ersten Iteration ($k = 1$) folgende Schritte aus: Zunächst wird (siehe Zeile 3 in Algorithmus 9.7)

$$\widehat{z} = v \oslash \alpha$$

bestimmt und in den Einträgen $(2,1)$ bis $(n+1,1)$ der Matrix gespeichert (als erste Spalte von L). Gegenüber dem exakten Ergebnis $z = v/\alpha$ erhalten wir nach Satz 6.1 einen Fehlervektor

$$f := \widehat{z} - z \quad \text{mit} \quad |f| \leqslant \frac{|v|}{|\alpha|}\,\varepsilon_{\text{mach}}.$$

Anschließend werden die Linearkombinationen der Zeilen $2, \ldots, n+1$ mit der ersten Zeile bestimmt (Zeile 4). In Maschinenoperationen bedeutet das die Berechnung der Untermatrix

$$(\widehat{A}_1)_{ij} := b_{ij} \ominus (\widehat{z}_i \odot w_j), \quad i, j = 2, \ldots, n+1.$$

Gemäß Satz 6.1 gilt

$$(\widehat{A}_1)_{ij} = \big[b_{ij} - \widehat{z}_i\, w_j\,(1+\varepsilon_1)\big](1+\varepsilon_2)$$

mit relativen Fehlern $|\varepsilon_1| \leqslant \varepsilon_{\text{mach}}$ und $|\varepsilon_2| \leqslant \varepsilon_{\text{mach}}$, die natürlich vom Index (i,j) abhängen. Damit erhalten wir für den Fehler $\Delta A_1 := \widehat{A}_1 - A_1$ gegenüber der exakten Matrix $A_1 = B - \widehat{z}\,w^{\mathsf{T}}$ die Abschätzung

$$|\Delta A_1| \dot{\leqslant} \big(|B| + 2\,|\widehat{z}\,w^{\mathsf{T}}|\big)\,\varepsilon_{\text{mach}} \leqslant 2\,\big(|B| + |\widehat{z}\,w^{\mathsf{T}}|\big)\,\varepsilon_{\text{mach}}. \tag{9.3}$$

Am Ende der Iteration $k = 1$ sieht die Matrix jetzt wie folgt aus:

$$\widehat{A}^{(1)} = \left[\begin{array}{cc} \alpha & w^{\mathsf{T}} \\ \widehat{z} & \widehat{A}_1 \end{array}\right]$$

Die Berechnung der weiteren Schritte in Algorithmus 9.7 sind genau dieselben, als würden wir Algorithmus 9.7 neu starten und auf die Untermatrix $\widehat{A}_1 \in \mathbb{F}^{n\times n}$ anwenden. Aufgrund der Induktionsvoraussetzung gilt für die in Maschinenoperationen bestimmte LR-Zerlegung von $\widehat{A}_1$ aber

$$\widehat{L}_1\,\widehat{R}_1 = \widehat{A}_1 + \Delta\widehat{A}_1 \quad \text{mit} \quad |\Delta\widehat{A}_1| \dot{\leqslant} 3\,(n-1)\big(|\widehat{A}_1| + |\widehat{L}_1|\,|\widehat{R}_1|\big)\,\varepsilon_{\text{mach}}. \tag{9.4}$$

Die in Maschinenoperationen berechnete Gesamtzerlegung von A hat also schließlich die Darstellung

$$\widehat{L}\,\widehat{R} = \begin{bmatrix} 1 & 0 \\ \widehat{z} & \widehat{L}_1 \end{bmatrix} \begin{bmatrix} \alpha & w^\mathsf{T} \\ 0 & \widehat{R}_1 \end{bmatrix} = \begin{bmatrix} \alpha & w^\mathsf{T} \\ \alpha\,\widehat{z} & \widehat{z}\,w^\mathsf{T} + \widehat{L}_1\,\widehat{R}_1 \end{bmatrix}.$$

Für die Fehlermatrix $\Delta A := \widehat{A} - A = \widehat{L}\,\widehat{R} - A$ gilt also

$$\Delta A = \begin{bmatrix} 0 & 0 \\ \alpha\,\widehat{z} - \alpha\,z & \widehat{z}\,w^\mathsf{T} + \widehat{L}_1\,\widehat{R}_1 - B \end{bmatrix}$$

und wegen $f = \widehat{z} - z$, $\widehat{A}_1 = A_1 + \Delta A_1$ und $A_1 = B - \widehat{z}\,w^\mathsf{T}$ weiter auch

$$\Delta A = \begin{bmatrix} 0 & 0 \\ \alpha\,f & \Delta A_1 + \Delta\widehat{A}_1 \end{bmatrix}. \tag{9.5}$$

Wir schätzen noch den in (9.4) vorkommenden Term $|\widehat{A}_1|$ ab:

$$\begin{aligned} |\widehat{A}_1| &= |A_1 + \Delta A_1| \quad \text{nach Definition von } \delta A_1 \\ &\leqslant |A_1| + |\Delta A_1| \\ &\dot{\leqslant} |B - \widehat{z}\,w^\mathsf{T}| + 2\,\big(|B| + |\widehat{z}\,w^\mathsf{T}|\big)\,\varepsilon_{\text{mach}} \quad \text{nach Definition von } A_1 \text{ und (9.3)} \\ &\leqslant (1 + 2\,\varepsilon_{\text{mach}})\big(|B| + |\widehat{z}\,w^\mathsf{T}|\big) \end{aligned} \tag{9.6}$$

und erhalten damit

$$\begin{aligned} |\Delta A_1 + \Delta\widehat{A}_1| &\leqslant |\Delta A_1| + |\Delta\widehat{A}_1| \\ &\dot{\leqslant} 2\,\big(|B| + |\widehat{z}\,w^\mathsf{T}|\big)\,\varepsilon_{\text{mach}} + 3\,(n-1)\big(|\widehat{A}_1| + |\widehat{L}_1|\,|\widehat{R}_1|\big)\,\varepsilon_{\text{mach}} \quad \text{nach (9.3) und (9.4)} \\ &\dot{\leqslant} 2\,\big(|B| + |\widehat{z}\,w^\mathsf{T}|\big)\,\varepsilon_{\text{mach}} + 3\,(n-1)\big(|B| + |\widehat{z}\,w^\mathsf{T}| + |\widehat{L}_1|\,|\widehat{R}_1|\big)\,\varepsilon_{\text{mach}} \quad \text{nach (9.6)} \\ &\leqslant 3\,n\big(|B| + |\widehat{z}\,w^\mathsf{T}| + |\widehat{L}_1|\,|\widehat{R}_1|\big)\,\varepsilon_{\text{mach}}. \end{aligned}$$

Mit dieser Abschätzung und $|\alpha\,f| \leqslant |v|\,\varepsilon_{\text{mach}}$ erhalten wir schließlich aus (9.5):

$$\begin{aligned} |\Delta A| &\dot{\leqslant} \begin{bmatrix} 0 & 0 \\ |v| & 3\,n\big(|B| + |\widehat{z}\,w^\mathsf{T}| + |\widehat{L}_1|\,|\widehat{R}_1|\big) \end{bmatrix} \varepsilon_{\text{mach}} \\ &\leqslant 3\,n \left(\begin{bmatrix} |\alpha| & |w^\mathsf{T}| \\ |v| & |B| \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ |\widehat{z}| & |\widehat{L}_1| \end{bmatrix} \begin{bmatrix} |\alpha| & |w^\mathsf{T}| \\ 0 & |\widehat{R}_1| \end{bmatrix} \right) \varepsilon_{\text{mach}}, \end{aligned}$$

was zu beweisen war. □

Was bedeuten die Resultate der Sätze 9.4, 9.5 und 9.8 nun für die Lösung eines linearen Gleichungssystems $Ax = b$, bei dem zunächst die LR-Zerlegung von A (ohne Pivotisierung, Algorithmus 9.7) bestimmt wird und dann die Lösung x mittels Vorwärts- und Rückwärtssubstitution $Ly = b$ (Algorithmus 9.1) und $Rx = y$ (Algorithmus 9.2)?

Wir können folgende Abschätzung des Rückwärtsfehlers zeigen:

Satz 9.9 (Rückwärtsanalyse der Lösung eines linearen Gleichungssystems mittels LR-Zerlegung ohne Pivotisierung, vgl. Golub, van Loan, 1983, Theorem 3.3.2)**.** *Es gelten die Voraussetzungen von Satz 9.8 und* $n\,\varepsilon_{\text{mach}} \leqslant C < 1$. *Es werde die LR-Zerlegung von* $A \in \mathbb{F}^{n\times n}$ *ohne Pivotisierung mittels Algorithmus 9.7 bestimmt und anschließend das lineare Gleichungssystem* $Ax = b$ *mit* $b \in \mathbb{F}^n$ *mit Hilfe der Vorwärts- und Rückwärtssubstitution (Algorithmen 9.1 und 9.2) gelöst. Die erhaltene Näherungslösung sei* $\widehat{x}$. *Dann gilt: Es gibt eine Matrix* E, *sodass gilt:*

$$(A+E)\,\widehat{x} = b \quad \textit{mit} \quad |E| \dot{\leqslant} 3\,(n-1)\big(|A| + |\widehat{L}|\,|\widehat{R}|\big)\,\varepsilon_{\text{mach}} + 2\,\gamma_n\,|\widehat{L}|\,|\widehat{R}|. \tag{9.7}$$

Beweis. Es seien $\widehat{L}$ und $\widehat{R}$ die mittels Algorithmus 9.7 (ohne Pivotisierung) in Maschinenoperationen bestimmten Faktoren. Wir wissen nach Folgerung 9.6, dass die numerische Lösung $\widehat{x}$ die Bedingungen

$$\begin{aligned}(\widehat{L} + \Delta\widehat{L})\,\widehat{y} &= b,\\ (\widehat{R} + \Delta\widehat{R})\,\widehat{x} &= \widehat{y}\end{aligned}$$

erfüllt. Dabei gilt $|\Delta\widehat{L}| \leqslant \gamma_n\,|\widehat{L}\widehat{L}^{\text{GM}}|$ und $|\Delta\widehat{R}| \leqslant \gamma_n\,|\widehat{R}\widehat{R}^{\text{GM}}|$. Einsetzen ergibt

$$(\widehat{L} + \Delta\widehat{L})(\widehat{R} + \Delta\widehat{R})\,\widehat{x} = \big(\widehat{L}\,\widehat{R} + \widehat{L}\,\Delta\widehat{R} + \Delta\widehat{L}\,\widehat{R} + \Delta\widehat{L}\,\Delta\widehat{R}\big)\,\widehat{x} = b.$$

Gemäß Satz 9.8 gibt es eine Matrix ΔA, sodass $\widehat{L}\,\widehat{R} = A + \Delta A$ gilt und ΔA die Abschätzung (9.2) erfüllt. Wir haben also $(A+E)\,\widehat{x} = b$ mit der Matrix

$$E = \Delta A + \widehat{L}\,\Delta\widehat{R} + \Delta\widehat{L}\,\widehat{R} + \Delta\widehat{L}\,\Delta\widehat{R}.$$

Wir schätzen die Größe von E komponentenweise ab mit

$$\begin{aligned}|E| &\leqslant |\Delta A| + |\Delta\widehat{L}|\,|\widehat{R}| + |\widehat{L}|\,|\Delta\widehat{R}| + |\Delta\widehat{L}|\,|\Delta\widehat{R}|\\ &\dot{\leqslant} 3\,(n-1)\big(|A| + |\widehat{L}|\,|\widehat{R}|\big)\,\varepsilon_{\text{mach}} + \gamma_n\,|\widehat{L}|\,|\widehat{R}| + \gamma_n\,|\widehat{L}|\,|\widehat{R}|.\end{aligned}$$

Damit ist die Behauptung gezeigt. □

Bemerkung 9.10 (Bedeutung von Satz 9.8 und Satz 9.9)**.**

(*i*) *Der Satz 9.8 ist ein Resultat zur Rückwärtsanalyse der LR-Zerlegung ohne Pivotisierung mittels Algorithmus 9.7. Dabei ist die Matrix* A *die Eingabe des Verfahrens, und die Faktoren* L *und* R *sind die Ausgabe.*

Um die Rückwärtsstabilität beurteilen zu können, schreiben wir die Abschätzung (9.2) in die Form

$$|\Delta A| \dot{\leqslant} 3\,(n-1)\big(|A| + |\widehat{L}|\,|\widehat{R}|\big)\,\varepsilon_{\text{mach}} = 3\,(n-1)\left(\mathbf{1} + \frac{|\widehat{L}|\,|\widehat{R}|}{|A|}\right)|A|\,\varepsilon_{\text{mach}}$$

um. Dabei steht $\frac{|\widehat{L}|\,|\widehat{R}|}{|A|}$ *für den komponentenweisen Quotienten aus den Einträgen von* $|\widehat{L}|\,|\widehat{R}|$ *und* $|A|$, *und* $\mathbf{1}$ *ist die Matrix bestehend aus lauter Einsen. Wir müssen hier voraussetzen, dass* A *keine Nulleinträge hat. Der Faktor* $B(x)$ *in der Definition (7.23) der Rückwärtsstabilität entspricht hier also der Größe*

$$3\,(n-1)\left(\mathbf{1} + \frac{|\widehat{L}|\,|\widehat{R}|}{|A|}\right).$$

Die Größen $\widehat{L}$ und $\widehat{R}$ «dürfen» hier vorkommen, da sie ja aus der Eingabe A berechenbar sind.

Die Abschätzung liefert den Hinweis, dass der Fall kritisch ist, bei dem mindestens ein Eintrag von $|\widehat{L}|\,|\widehat{R}|$ sehr groß ist gegenüber dem entsprechenden Eintrag von $|A|$.

(*ii*) *Der Satz 9.9 betrifft die Rückwärtsanalyse der auf dieser Zerlegung basierenden Lösung eines linearen Gleichungssystems $Ax = b$. Dabei sind A und b die Eingaben, und die Lösung x ist die Ausgabe. Aus der Abschätzung (9.7) erhalten wir genau dieselbe Information über den kritischen Fall.*

Wir betrachten jetzt eine typische Situation für diesen kritischen Fall, dass ein Eintrag von $|\widehat{L}|\,|\widehat{R}|$ sehr groß ist gegenüber dem entsprechenden Eintrag von $|A|$.

Beispiel 9.11 ($|\widehat{L}||\widehat{R}|$ groß gegenüber $|A|$, vgl. Higham, 2002, Chapter 1.12.1)**.** *Wir betrachten folgende Matrix und ihre LR-Zerlegung ohne Pivotisierung in exakter Rechnung:*

$$A = \begin{bmatrix} \varepsilon & -1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{bmatrix} \varepsilon & -1 \\ 0 & 1+\frac{1}{\varepsilon} \end{bmatrix}.$$

Die Matrix A ist gut konditioniert[5]*.*

Wodurch entstehen jetzt die Probleme bei der Fließkommarechnung? Nehmen wir dazu an, dass ε und $\frac{1}{\varepsilon}$ im Fließkommagitter $\mathbb{F}$ liegen. Dann liegt $A \in \mathbb{F}^{2\times 2}$, und wir erhalten in Fließkommarechnung mit Hilfe von Algorithmus 9.7 die folgende LR-Zerlegung von A mit Faktoren

$$\widehat{L} = L = \begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \quad \text{und} \quad \widehat{R} = \begin{bmatrix} \varepsilon & -1 \\ 0 & 1 \oplus \frac{1}{\varepsilon} \end{bmatrix}.$$

Quizfrage: *Können Sie nachvollziehen, dass das tatsächlich die Faktoren sind, die wir in Fließkommarechnung erhalten?*

Nun sei $\varepsilon > 0$ so gewählt, dass zusätzlich zu den obigen Bedingungen $1 \oplus \frac{1}{\varepsilon} = \frac{1}{\varepsilon}$ gilt. ***Quizfrage:*** *Wann passiert das? Welche Auswirkungen hat nun dieser kleine relative Fehler in einem einzigen Eintrag der LR-Zerlegung bei der Lösung eines linearen Gleichungssystems $Ax = b$ mit Hilfe der Vorwärts- und Rückwärtssubstitution (Algorithmen 8.7, 9.1 und 9.2)?*

Wir betrachten dazu den Fall einer ausgewählten rechten Seite und eines Parameters ε, für den sich während der Rechnung möglichst wenig Rundungsfehler ergeben, sodass wir den Fehlereinfuss gut verfolgen können. Jegliche Rundungsfehler werden in rot markiert.

(*i*) *Wir betrachten dazu jetzt die rechte Seite $b = \binom{1}{1} \in \mathbb{F}^2$. Die exakte Lösung ergibt sich aus*

$$Ly = \begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \Rightarrow \quad y = \begin{pmatrix} 1 \\ 1-\frac{1}{\varepsilon} \end{pmatrix}$$

und weiter

$$Rx = \begin{bmatrix} \varepsilon & -1 \\ 0 & 1+\frac{1}{\varepsilon} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1-\frac{1}{\varepsilon} \end{pmatrix} \quad \Rightarrow \quad x = \frac{1}{\varepsilon+1} \begin{pmatrix} 2 \\ \varepsilon - 1 \end{pmatrix}.$$

[5] $\kappa(A) \leqslant \left(\frac{3+\sqrt{5}}{3-\sqrt{5}}\right)^{1/2} \approx 2.62$ für alle $\varepsilon \in [0,1]$

Aus der LR-Zerlegung in Fließkommarechnung erhalten wir dagegen

$$\widehat{L}\widehat{y} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{pmatrix} \widehat{y}_1 \\ \widehat{y}_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \widehat{y} = \begin{pmatrix} 1 \\ 1 \ominus (\frac{1}{\varepsilon} \odot 1) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \ominus \frac{1}{\varepsilon} \end{pmatrix}.$$

Wir betrachten weiter den Fall eines Parameters ε, sodass $1 \ominus \frac{1}{\varepsilon} = 1 - \frac{1}{\varepsilon}$ und $\varepsilon \ominus 1 = \varepsilon - 1$ gilt.
***Quizfrage:** Wann tritt dieser Fall auf? Dann haben wir weiter*

$$\widehat{R}\widehat{x} = \begin{bmatrix} \varepsilon & -1 \\ 0 & \frac{1}{\varepsilon} \end{bmatrix} \begin{pmatrix} \widehat{x}_1 \\ \widehat{x}_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 - \frac{1}{\varepsilon} \end{pmatrix}.$$

Daher gilt

$$\widehat{x}_2 = \mathrm{rd}\left(\frac{1 - \frac{1}{\varepsilon}}{\frac{1}{\varepsilon}} \right) = \mathrm{rd}(\varepsilon - 1) = \varepsilon - 1$$

und

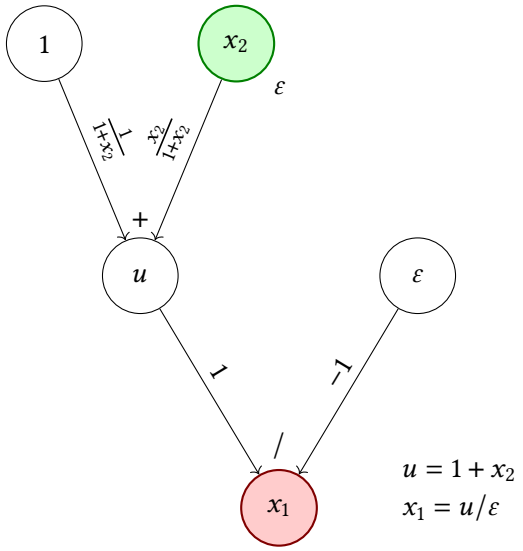
$$\widehat{x}_1 = \mathrm{rd}\left(\frac{\big(1 \oplus (\widehat{x}_2 \odot 1)\big)}{\varepsilon} \right) = \mathrm{rd}\left(\frac{\big(1 \oplus (\varepsilon - 1)\big)}{\varepsilon} \right) = 1,$$

also gilt

$$\widehat{x} = \begin{pmatrix} 1 \\ \varepsilon - 1 \end{pmatrix}.$$

Der relative Fehler von $\widehat{x}$ gegenüber x wird dominiert durch die erste Komponente, wo er etwa $1/2$ beträgt.

Wodurch kommt dieser große Fehler zustande? In unserem Beispiel entsteht der erste numerische Fehler erst bei der Rückwärtssubstitution (Lösen mit $\widehat{R}$), und zwar durch den geringfügig falschen $(2,2)$-Eintrag $\frac{1}{\varepsilon}$ in $\widehat{R}$. Der dadurch in verursachte relative Fehler in $\widehat{x}_2$ ist mit ε gering. Im nächsten Schritt wird er jedoch bei der Berechnung von $\widehat{x}_1$ wie folgt verstärkt:



Dabei ist die relative Konditionszahl mit $\frac{x_2}{1+x_2} = \frac{\varepsilon-1}{2\varepsilon} \approx -\frac{1}{2\varepsilon}$ sehr groß (Auslöschung bei einer echten Subtraktion!), verstärkt also den relativen Fehler in $\widehat{x}_2$ enorm zur beobachteten Größe von etwa $1/2$.

(*ii*) *Wir betrachten jetzt noch die alternative rechte Seite* $b = \begin{pmatrix} \varepsilon - 1 \\ 2 \end{pmatrix} \in \mathbb{F}^2$. *Die exakte Lösung ergibt sich aus*

$$Ly = \begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \varepsilon - 1 \\ 2 \end{pmatrix} \quad \Rightarrow \quad y = \begin{pmatrix} \varepsilon - 1 \\ 1 + \frac{1}{\varepsilon} \end{pmatrix}$$

und weiter

$$Rx = \begin{bmatrix} \varepsilon & -1 \\ 0 & 1 + \frac{1}{\varepsilon} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \varepsilon - 1 \\ 1 + \frac{1}{\varepsilon} \end{pmatrix} \quad \Rightarrow \quad x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Aus der LR-Zerlegung in Fließkommarechnung erhalten wir dagegen

$$\widehat{L}\widehat{y} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{pmatrix} \widehat{y}_1 \\ \widehat{y}_2 \end{pmatrix} = \begin{pmatrix} \varepsilon - 1 \\ 2 \end{pmatrix} \quad \Rightarrow \quad \widehat{y} = \begin{pmatrix} \varepsilon - 1 \\ 2 \ominus (1 - \frac{1}{\varepsilon}) \end{pmatrix} = \begin{pmatrix} \varepsilon - 1 \\ \mathrm{rd}(1 + \frac{1}{\varepsilon}) \end{pmatrix} = \begin{pmatrix} \varepsilon - 1 \\ \frac{1}{\varepsilon} \end{pmatrix}.$$

Dann haben wir weiter

$$\widehat{R}\widehat{x} = \begin{bmatrix} \varepsilon & -1 \\ 0 & \frac{1}{\varepsilon} \end{bmatrix} \begin{pmatrix} \widehat{x}_1 \\ \widehat{x}_2 \end{pmatrix} = \begin{pmatrix} \varepsilon - 1 \\ \frac{1}{\varepsilon} \end{pmatrix} \quad \Rightarrow \quad \widehat{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

In diesem Fall ist die numerische Lösung — trotz der Rundungsfehler im Faktor $\widehat{R}$ *und des Rundungsfehlers in* $\widehat{y}$ *— am Ende dennoch exakt.*

Die Fehlerschranke (9.7) *der Rückwärtsanalyse enthält den Term*

$$|\widehat{L}|\,|\widehat{R}| = \begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{bmatrix} \varepsilon & 1 \\ 0 & \frac{1}{\varepsilon} \end{bmatrix} = \begin{bmatrix} \varepsilon & 1 \\ 1 & \frac{2}{\varepsilon} \end{bmatrix},$$

wobei der rot markierte Eintrag sehr groß gegenüber dem entsprechenden Eintrag in

$$|A| = \begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix}$$

ist. Die relative Fehlerschranke des Rückwärtsfehlers ist also groß und damit auch die aus Satz 7.20 erhaltene relative Fehlerschranke des Vorwärtsfehlers $\frac{\widehat{x}-x}{x}$. *Ob tatsächlich ein Fehler in der Lösung dieser Größenordnung auftritt oder nicht, hängt — wie das Beispiel zeigt — von der konkreten rechten Seite ab!*

§ 9.2 Fall mit Spaltenpivotsuche

Welchen Vorteil bringt nun die Spaltenpivotisierung? Wir geben zunächst das Verfahren für die LR-Zerlegung mit Spaltenpivotisierung an, damit klar ist, welchen Algorithmus wir untersuchen.

Algorithmus 9.12 (Berechnung der LR-Zerlegung mit Spaltenpivotisierung, vgl. Golub, van Loan, 1983, Algorithm 3.4.1)**.**

Eingabe: *reguläre Matrix* A

Ausgabe: *Faktoren* L *und* R *sowie Permutationsmatrix* P *der LR-Zerlegung* $PA = LR$

1: *Setze* $p := (1, 2, \ldots, n)^\mathsf{T}$
2: ***for*** $k = 1, 2, \ldots, n-1$ ***do***
3: *Setze* $I := k+1 : n = \{k+1, k+2, \ldots, n\}$

4: *Bestimme einen Index* $s \in I \cup \{k\}$, *sodass gilt (vgl. (8.2))*

$$|a_{sk}| = \max\{|a_{ik}| \,\big|\, i \in I \cup \{k\}\},$$

5: *Tausche* $A(k,:)$ *mit* $A(s,:)$ *aus*
6: *Tausche* $p(k)$ *mit* $p(s)$ *aus*
7: *Setze* $A(I,k) \coloneqq A(I,k)/A(k,k)$ *// relevante Einträge der k-ten Spalte des Faktors L*
8: *Setze* $A(I,I) \coloneqq A(I,I) - A(I,k)\,A(k,I)$
9: ***end for***
10: *Extrahiere L aus dem strikteren unteren Dreieck von A und ergänze Einsen auf der Diagonale*
11: *Extrahiere R aus dem oberen Dreieck von A*
12: *Setze*

$$P \coloneqq \begin{bmatrix} — & e_{p(1)}^{\mathsf{T}} & — \\ & \vdots & \\ — & e_{p(n)}^{\mathsf{T}} & — \end{bmatrix}$$

Der Vorteil der Spaltenpivotisierung ist gar nicht so einfach zu charakterisieren. Es gilt aber folgendes Analogon zu Satz 9.8:

Satz 9.13 (Rückwärtsanalyse der LR-Zerlegung mit Spaltenpivotisierung, vgl. Golub, van Loan, 1983, Chapter 3.4.6)**.** *Es sei* $A \in \mathbb{F}^{n\times n}$ *eine reguläre Matrix. Weiter seien* $\widehat{L}$ *and* $\widehat{R}$ *die von Algorithmus 9.12 in Maschinenoperationen berechnete Näherungslösung der LR-Zerlegung von* $PA = LR$*. Dabei nehmen wir an, dass in jeder Iteration das Pivotelement* $\widehat{A}(k,k) \neq 0$ *ist. Dann gibt es eine Matrix* ΔA*, sodass gilt:*

$$\widehat{L}\,\widehat{R} = PA + \Delta A \quad \textit{mit} \quad |\Delta A| \dot{\leqslant} 3\,(n-1)\big(|PA| + |\widehat{L}|\,|\widehat{R}|\big)\,\varepsilon_{\text{mach}}. \tag{9.8}$$

Alle Vorkommen von der Permutationsmatrix P sind der Übersichtlichkeit halber rot hervorgehoben.

Beweis. Es ist lediglich der Satz 9.8 auf PA an Stelle von A anzuwenden. Die zusätzlichen Schritte in Algorithmus 9.12 gegenüber Algorithmus 9.7 haben keinen Einfluss auf die Abschätzung der Rundungsfehler. □

Quizfrage: Warum wurde in Satz 9.13 wie bereits in Satz 9.8 angenommen, dass die Pivotlemente $\widehat{A}(k,k) \neq 0$ sind? Folgt das nicht bereits aus der Annahme der Regularität von A und der Verwendung der Spaltenpivotsuche?

Mit diesem Resultat können wir nun folgendes Analogon zu Satz 9.9 zeigen:

Satz 9.14 (Rückwärtsanalyse der Lösung eines linearen Gleichungssystems mittels LR-Zerlegung mit Spaltenpivotisierung, vgl. Golub, van Loan, 1983, Chapter 3.4.6)**.** *Es gelten die Voraussetzungen von Satz 9.13 und* $n\,\varepsilon_{\text{mach}} \leqslant C < 1$*. Es werde die LR-Zerlegung von* $A \in \mathbb{F}^{n\times n}$ *mit Spaltenpivotisierung mittels Algorithmus 9.12 bestimmt und anschließend das lineare Gleichungssystem* $Ax = b$ *mit* $b \in \mathbb{F}^n$ *mit Hilfe der Vorwärts- und Rückwärtssubstitution (Algorithmen 9.1 und 9.2) in der Form* $Ly = Pb$ *und* $Rx = y$ *gelöst. Die erhaltene Näherungslösung sei* $\widehat{x}$*. Dann gilt: Es gibt eine Matrix E, sodass gilt:*

$$(A+E)\,\widehat{x} = b \quad \textit{mit} \quad |E| \dot{\leqslant} 3\,(n-1)\big(|A| + P^{\mathsf{T}}|\widehat{L}|\,|\widehat{R}|\big)\,\varepsilon_{\text{mach}} + 2\,\gamma_n\,P^{\mathsf{T}}|\widehat{L}|\,|\widehat{R}|. \tag{9.9}$$

Beweis. □

In der Praxis kann man durch Spaltenpivotisierung oft die Größe des Terms $\frac{P^T|\widehat{L}|\,|\widehat{R}|}{|A|}$ im Vergleich zum gleichen Term ohne P günstig beeinflussen. Damit reduziert sich dann die relative Fehlerschranke des Rückwärtsfehlers (9.9) gegenüber (9.7) und damit (Satz 7.20) auch die relative Fehlerschranke des Vorwärtsfehlers $\frac{\widehat{x}_i - x_i}{x_i}$. **Beachte:** Für $P \neq \mathrm{Id}$ sehen natürlich auch die Faktoren $\widehat{L}$ und $\widehat{R}$ anders aus, daher ist eine Vorhersage der Auswirkungen der Spaltenpivotisierung i. A. schwierig. Es gibt Beispiele, bei denen eine Spaltenpivotisierung tatsächlich keine Vorteile bringt. In der Praxis ist die Spaltenpivotisierung aber eine gute Wahl; siehe Übung.

§ 10 LR-ähnliche Zerlegungen für spezielle Matrizen

§ 10.1 Matrizen, für die keine Pivotsuche erforderlich ist

Es gibt Klassen von Matrizen, für die wir auch ohne Spaltenpivotsuche eine brauchbare Fehlerschranke (9.7) erhalten. Dazu zählen

(1) **total nichtnegative Matrizen** A. Das sind Matrizen, für die gilt, dass jede quadratische Untermatrix, die durch eine Anzahl von Zeilen und Spalten gewonnen wird, positive Determinante besitzt. Die LR-Faktorisierung dieser Matrizen erfüllt (ohne Pivotisierung) die Bedingungen $L \geqslant 0$ und $R \geqslant 0$, woraus $|L|\,|R| = |LR| = |A|$ folgt.

(2) **diagonal-dominante Matrizen** A. Das sind Matrizen, für die der Diagonaleintrag in jeder Zeile betragsmäßig mindestens so groß ist wie die Summe der Beträge der restlichen Elemente derselben Zeile:

$$|a_{ii}| \geqslant \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \quad \text{für alle } i = 1, \ldots, n.$$

Für reguläre, diagonal-dominante Matrizen A kann man zeigen (Higham, 2002, Theorem 9.9), dass $\||L|\,|R|\|_\infty \leqslant (2n-1)\,\|A\|_\infty$ gilt.

(3) Wenn die Matrix A zusätzlich zur Diagonaldominanz noch **tridiagonal** ist, also $a_{ij} = 0$ gilt für alle Indizes mit $|i - j| > 1$, dann kann man sogar $|L|\,|R| \leqslant 3\,|A|$ zeigen (Higham, 2002, Theorem 9.13).

§ 10.2 Cholesky-Zerlegung für symmetrische positiv definite Matrizen

Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt **positiv definit**, wenn für alle $x \neq 0$ gilt: $x^T A x > 0$. Die Cholesky-Zerlegung verallgemeinert den Begriff der Wurzel von positiven Zalen auf symmetrische, positiv definite Matrizen. Man kürzt die Eigenschaft einer Matrix, «symmetrisch und positiv definit» zu sein, häufig mit **s. p. d.** ab.

Definition 10.1 (Cholesky-Zerlegung). *Es sei $A \in \mathbb{R}^{n\times n}$ s. p. d.. Eine Zerlegung der Form $A = LL^\mathsf{T}$, wobei L eine untere Dreiecksmatrix mit positiver Diagonale ist, heißt eine* ***Cholesky-Zerlegung*** *von A.*

Satz 10.2 (Cholesky-Zerlegung, vgl. Bornemann, 2018, Kapitel 8.3). *Jede s. p. d. Matrix $A \in \mathbb{R}^{n\times n}$ besitzt eine eindeutige Cholesky-Zerlegung.*

Beweis. Wir führen einen konstruktiven Beweis per Induktion über die Dimension n. Für $n = 1$ ist $A \in \mathbb{R}$ eine positive Zahl und besitzt eine eindeutige positive Wurzel L. Für den Induktionsschritt sei die Behauptung bereits bewiesen für Dimension n. Wir betrachten den Ansatz

$$A' = \left[\begin{array}{c|c} A & a \\ \hline a^\mathsf{T} & \alpha \end{array}\right] = \left[\begin{array}{c|c} L & 0 \\ \hline \ell^\mathsf{T} & \lambda \end{array}\right] \left[\begin{array}{c|c} L^\mathsf{T} & \ell \\ \hline 0 & \lambda \end{array}\right] =: L'(L')^\mathsf{T},$$

wobei $A' \in \mathbb{R}^{(n+1)\times(n+1)}$ s. p. d. ist. Die Dimensionen von A und L sind $n \times n$, die Dimensionen von a und ℓ sind $n \times 1$, und α und λ sind Zahlen. Am Ansatz lesen wir folgende Bedingungen ab:

$$\begin{aligned} A &= LL^\mathsf{T} \\ a &= L\ell \\ \alpha &= \ell^\mathsf{T}\ell + \lambda^2. \end{aligned}$$

Quizfrage: Was ist mit der vierten Gleichung?

Der Faktor L' soll eine untere Dreiecksmatrix mit positiver Diagonale sein. Insbesondere ist daher L ebenfalls eine untere Dreiecksmatrix mit positiver Diagonale. Aufgrund der Induktionsvoraussetzung ist L aber durch die erste Gleichung $A = LL^\mathsf{T}$ bereits eindeutig bestimmt. Da L auch invertierbar ist (**Quizfrage:** Klar?), können wir die zweite Gleichung auflösen und erhalten $\ell = L^{-1}a$. Schließlich ergibt sich aus der letzten Gleichung $\lambda^2 = \alpha - \ell^\mathsf{T}\ell$. Wenn wir gleich noch zeigen können, dass dieser Ausdruck positiv ist, dann ist auch λ als die (positive) Wurzel eindeutig bestimmt. Wir wählen dazu $x := -L^{-\mathsf{T}}\ell$ und bilden mit dem Vektor $x' := \binom{x}{1}$ die quadratische Form:

$$\begin{aligned} \begin{pmatrix} x \\ 1 \end{pmatrix}^\mathsf{T} A' \begin{pmatrix} x \\ 1 \end{pmatrix} &= \begin{pmatrix} x \\ 1 \end{pmatrix}^\mathsf{T} \left[\begin{array}{c|c} LL^\mathsf{T} & L\ell \\ \hline \ell^\mathsf{T}L^\mathsf{T} & \alpha \end{array}\right] \begin{pmatrix} x \\ 1 \end{pmatrix} \\ &= \ell^\mathsf{T}L^{-1}LL^\mathsf{T}L^{-\mathsf{T}}\ell - 2\,\ell^\mathsf{T}L^{-1}L\ell + \alpha \\ &= \alpha - \ell^\mathsf{T}\ell. \end{aligned}$$

Da nach Voraussetzung A' positiv definit ist, ist die linke Seite positiv, also auch die rechte. Damit haben wir gezeigt, dass auch die s. p. d. Matrix A' eine Cholesky-Zerlegung besitzt, die auch eindeutig ist. □

Der Beweis von Satz 10.2 ist konstruktiv und kann direkt in ein numerisches Verfahren umgesetzt werden:

Algorithmus 10.3 (Cholesky-Zerlegung einer s. p. d. Matrix, vgl. Golub, van Loan, 1983, Algorithm 3.4.1).

Eingabe: *s. p. d. Matrix A*

Ausgabe: *Faktor L der Cholesky-Zerlegung $A = LL^\mathsf{T}$*

1: *Setze* $L \coloneqq \mathbf{0}$ *(Nullmatrix)*
2: ***for*** $k = 1, 2, \ldots, n$ ***do***
3: *Setze* $I \coloneqq 1 : k-1 = \{1, 2, \ldots, k-1\}$
4: *Löse* $L(I, I)\, \ell = A(I, k)$
5: *Setze* $L(k, I) \coloneqq \ell^\mathsf{T}$
6: *Setze* $L(k, k) \coloneqq \sqrt{A(k, k) - \ell^\mathsf{T}\ell}$
7: ***end for***

Beachte: In der ersten Iteration ($k = 1$) ist $I = \emptyset$. Das bedeutet, dass in den Zeilen 4 und 5 nichts getan wird und $\ell \in \mathbb{R}^{0\times 1}$ als leerer Vektor zu deuten ist mit $\ell^\mathsf{T}\ell = 0$ in Zeile 6.

Satz 10.4 (Aufwand der Cholesky-Zerlegung)**.** *Der numerische Aufwand für die Erzeugung der Cholesky-Zerlegung einer s. p. d. Matrix $A \in \mathbb{R}^{n\times n}$ ist von der Größenordnung $\frac{1}{3}n^3 + O(n^2)$.*

Beweis. Der wesentliche Aufwand in jeder Iteration ist die Lösung des Gleichungssystems in Zeile 4 mittels Vorwärtssubstitution (Algorithmus 9.1) mit dem Aufwand $(k-1)^2$. In Zeile 6 kommen $(k-1)$ Multiplikationen, $k-1$ Additionen und eine Wurzel hinzu, zusammen $2k-1$ FLOPs. Die Summation über $k = 1, 2, \ldots, n$ ergibt $\frac{1}{3}n^3 + O(n^2)$ FLOPs. □

Bemerkung 10.5 (Zur Cholesky-Zerlegung)**.**

(*i*) *Der Aufwand für die Berechnung der Cholesky-Zerlegung einer s. p. d. Matrix ist mit dem führenden Term $\frac{1}{3}n^3$ nur halb so groß wie der Aufwand für die LR-Zerlegung einer allgemeinen regulären Matrix.*

(*ii*) *Algorithmus 10.3 greift nur auf das obere Dreieck von A zu. Man kann das Verfahren also auch für Datenstrukturen implementieren, bei denen überhaupt nur die obere Dreiecksmatrix von A gespeichert wird, womit der Speicheraufwand i. W. halbiert wird.*

(*iii*) *An Stelle der Zerlegung $A = LL^\mathsf{T}$ wird manchmal auch die Zerlegung $A = LDL^\mathsf{T}$ verwendet, wobei L eine untere Dreiecksmatrix mit lauter Einsen auf der Diagonale und D eine positive Diagonalmatrix ist.*

(*iv*) *Es gibt eine alternative Definition der Cholesky-Zerlegung*

$$A = R^\mathsf{T}R,$$

bei der eine obere Dreiecksmatrix R mit positiver Diagonale verwendet wird.[6] *Es gilt der Zusammenhang $L = R^\mathsf{T}$. Natürlich gibt es hier auch wieder die Version $A = R^\mathsf{T}DR$.*

(*v*) *Die Existenz und Eindeutigkeit der Cholesky-Zerlegung (Satz 10.2) kann auch als Spezialfall der LR-Zerlegung hergeleitet werden.* ***Quizfrage:*** *Können Sie das?*

[6] Das ist zum Beispiel der Standard in MATLABs `chol` und `scipy.linalg.cholesky`.

Wenn die Cholesky-Zerlegung $A = LL^\mathsf{T}$ vorliegt, so können wir jedes lineare Gleichungssystem $Ax = b$ mit Hilfe der beiden Dreieckssysteme $Ly = b$ und $L^\mathsf{T}x = y$ mit dem Gesamtaufwand von n^2 FLOPs lösen.

Ende der Woche 7

Kapitel 5 Ausgleichsprobleme

§ 11 Einführung

Die (lineare) Ausgleichsrechnung befasst sich mit Lösungen überbestimmter linearer Gleichungssysteme $Ax = b$ mit $A \in \mathbb{R}^{m\times n}$ und $b \in \mathbb{R}^m$, wobei $m \geqslant n$ gilt. Solche Aufgaben entstehen vor allem bei der Bestimmung von Parametern x in (linearen) Modellen aufgrund von Messdaten. Dabei heißt A die **Designmatrix** oder **Modellmatrix** (englisch: *design matrix, model matrix*) und b der **Vektor der Messwerte** (englisch: *vector of measurements*).

Da man i. A. nicht alle Gleichungen in $Ax = b$ gleichzeitig erfüllen kann, betrachtet man Lösungen in einem ausgleichenden Sinne, die die **Fehlerquadratsumme** (englisch: *sum of squares*) minimieren, also die **Kleinste-Quadrate-Aufgabe** (englisch: *least-squares problem*)

$$\text{Minimiere} \quad \frac{1}{2}\|Ax - b\|_2^2, \quad x \in \mathbb{R}^n \tag{11.1}$$

lösen. Man spricht auch von **Regressionsanalyse**, **Parameterschätzung**, **Parameteridentifikation** oder **Modellkalibrierung** (englisch: *regression analysis, parameter estimation, parameter identification, model calibration*). Der Vektor

$$r := Ax - b \tag{11.2}$$

heißt der zu x gehörende **Residuenvektor**, kurz: das **Residuum** (englisch: *residual (vector)*).[1]

Beispiel 11.1 (Bremsweg). *Für den Bremsweg s eines Autos in Abhängigkeit der Geschwindigkeit v liegen folgende Messwerte vor:*

Geschwindigkeit v in $\mathrm{km\,h^{-1}}$	*10*	*20*	*25*	*30*	*40*	*50*
Bremsweg s in m	*4*	*10*	*14*	*20*	*30*	*42*

Der Zusammenhang wird durch eine quadratische Abhängigkeit der Form

$$s(v) = a\,v^2 + b\,v$$

mit noch unbekannten Modellparametern $(a, b) \in \mathbb{R}^2$ modelliert. Diese Parameter sollen so bestimmt werden, dass das Modell und die Messdaten möglichst gut übereinstimmen (im Sinne der kleinsten

[1] Fälschlicherweise wird in der Literatur leider oft auch die Norm des Residuums als «Residuum» bezeichnet.

Fehlerquadratsumme). Die Designmatrix und der Vektor der Messwerte (ohne Einheiten) sind in diesem Beispiel gegeben durch

$$A = \begin{bmatrix} 10^2 & 10 \\ 20^2 & 20 \\ 25^2 & 25 \\ 30^2 & 30 \\ 40^2 & 40 \\ 50^2 & 50 \end{bmatrix}, \quad b = \begin{pmatrix} 4 \\ 10 \\ 14 \\ 20 \\ 30 \\ 42 \end{pmatrix}$$

mit Dimensionen $n = 2$ and $m = 6$.

Satz 11.2 (Lösung von (11.1))**.** *Es sei $A \in \mathbb{R}^{m\times n}$ und $b \in \mathbb{R}^m$.*

(*i*) *Die Aufgabe* (11.1) *besitzt immer wenigstens eine (global optimale) Lösung.*

(*ii*) *Die optimalen Lösungen von* (11.1) *sind genau die Lösungen der* ***Normalengleichung(en)*** *(auch:* ***Normalgleichung(en)****, englisch:* normal equations*)*

$$A^\mathsf{T}Ax = A^\mathsf{T}b. \tag{11.3}$$

Beweis. Wir benennen die Zielfunktion in (11.1) mit

$$f(x) \coloneqq \frac{1}{2}\|Ax - b\|_2^2.$$

Wir verwenden die Singulärwertzerlegung $A = U\Sigma V^\mathsf{T}$, die die Gestalt (4.4b) hat. Wie in (4.6) nehmen wir an, dass

$$\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > 0 = \sigma_{r+1} = \cdots = \sigma_n$$

gilt, dass also $r = \operatorname{Rang}(A)$ ist für ein $0 \leqslant r \leqslant n$.

Zwei Punkte $x, y \in \mathbb{R}^n$, sodass $x - y \in \ker A$ liegt, besitzen denselben Funktionswert $f(x) = f(y)$. (**Quizfrage:** Klar?) Wir vergeben uns also nichts, wenn wir uns bei der Suche nach Minimierern auf $(\ker A)^\perp$ beschränken. Für solche $x \in (\ker A)^\perp$ folgt aus der Singulärwertzerlegung, dass $\|Ax\|_2 \geqslant \sigma_r\|x\|_2$ gilt. (**Quizfrage:** Klar?) Es gilt also für alle $x \in (\ker A)^\perp$:

$$f(x) = \frac{1}{2}\|Ax\|_2^2 - b^\mathsf{T}Ax + \frac{1}{2}\|b\|_2^2 \geqslant \sigma_r^2\|x\|_2^2 - \|A^\mathsf{T}b\|_2\|x\|_2 + \frac{1}{2}\|b\|_2^2.$$

Daher gilt insbesondere, dass $f(x) > f(0)$ wird für alle $x \in (\ker A)^\perp$ mit $\|x\|_2 > 2\,\frac{\|A^\mathsf{T}b\|_2}{\sigma_r^2}$. Daher können wir uns bei der Suche nach Minimierern auf die Menge

$$K \coloneqq (\ker A)^\perp \cap \left\{ x \in \mathbb{R}^n \,\middle|\, \|x\|_2 \leqslant \frac{\|A^\mathsf{T}b\|_2}{\sigma_r^2} \right\}$$

einschränken. Diese Menge ist aber kompakt, und die Funktion f ist stetig. Nach dem **Satz von Weierstraß** existiert also ein Minimierer von f über K, der gleichzeitig Minimierer von f über ganz $\mathbb{R}^n$ ist. (**Quizfrage:** Klar?) Das zeigt Aussage (*i*).

Aus der mehrdimensionalen Differenzialrechnung sollte bekannt sein, dass

$$\nabla f(x) = A^\mathsf{T}(Ax - b) \stackrel{!}{=} 0,$$

also gerade die Normalengleichung (11.3), eine notwendige Bedingung dafür ist, dass x ein (lokaler) Minimierer von f ist. Wir zeigen jetzt noch, dass diese Bedingung sogar hinreichend ist, und zwar für globale Optimalität. Es sei dazu $x \in \mathbb{R}^n$ ein Punkt, der die Normalengleichung (11.3) erfüllt, und $y \in \mathbb{R}^n$ ein beliebiger anderer Punkt. Dann gilt

$$\begin{aligned} f(y) - f(x) &= \frac{1}{2}\|Ay - b\|_2^2 - \frac{1}{2}\|Ax - b\|_2^2 \\ &= \frac{1}{2}\|A(y - x) + Ax - b\|_2^2 - \frac{1}{2}\|Ax - b\|_2^2 \\ &= \frac{1}{2}\|A(y - x)\|_2^2 + (y - x)^\mathsf{T} \underbrace{A^\mathsf{T}(Ax - b)}_{=0} \pm \frac{1}{2}\|Ax - b\|_2^2 \\ &= \frac{1}{2}\|A(y - x)\|_2^2 \\ &\geqslant 0. \end{aligned}$$

Das zeigt, dass jedes $x \in \mathbb{R}^n$, das die Normalengleichung (11.3) erfüllt, tatsächlich global optimal ist. Damit ist auch Aussage (ii) gezeigt. □

Folgerung 11.3 (Eindeutigkeit der Lösung der Kleinste-Quadrate-Aufgabe (11.1))**.** *Die folgenden Aussagen sind äquivalent:*

(i) *Die Lösung der Kleinste-Quadrate-Aufgabe* (11.1) *ist eindeutig.*

(ii) *Die zu A gehörige Abbildung ist injektiv, also* $\ker A = \{0\}$.

(iii) *A hat vollen Spaltenrang, also* $\operatorname{Rang}(A) = n$.

Beweis. □

Wir gehen im Folgenden, sofern nichts anderes gesagt wird, davon aus, dass die Bedingungen aus Folgerung 11.3 zutreffen, sodass die Normalengleichung (11.3) eine eindeutige Lösung hat. Die Matrix $A^\mathsf{T}A$ ist in diesem Fall positiv definit und nicht nur positiv semidefinit (und ohnehin symmetrisch). Man könnte also im Prinzip die Matrix $A^\mathsf{T}A$ formen, ihre Cholesky-Zerlegung bestimmen und dann die Normalengleichung durch die Lösung zweier Dreieckssysteme mit R bzw. R^T bestimmen.

Dieses Vorgehen hat aber gewisse Nachteile, die wir jetzt aufzeigen. Dazu sei $A = U\Sigma V^\mathsf{T}$ eine Singulärwertzerlegung von A. In Verallgemeinerung von (4.13) können wir

$$\kappa(A) := \frac{\sigma_1}{\sigma_{\min\{m,n\}}} \tag{11.4}$$

als die **Konditionszahl** einer nicht notwendig quadratischen Matrix A definieren.[2] Wegen $m \geqslant n$ gilt hier also $\kappa(A) = \frac{\sigma_1}{\sigma_n}$. Die Matrix $A^\mathsf{T}A$ hat dann die Darstellung

$$A^\mathsf{T}A = V\Sigma^2 V^\mathsf{T},$$

die gleichzeitig die Spektral- wie auch die Singulärwertzerlegung von $A^\mathsf{T}A$ ist[GM]. Die Konditionszahl von $A^\mathsf{T}A$ ist also gerade gleich $(\kappa(A))^2$!

Quizfrage: Wie groß ist der Aufwand für das Aufstellen von $A^\mathsf{T}A$, für die Cholesky-Zerlegung und für die Lösung der Normalengleichung mit Hilfe der Cholesky-Zerlegung?

Beispiel 11.4 (Auswirkungen der quadrierten Konditionszahl, vgl. das sogenannte Beispiel von Läuchli, Bornemann, 2018, S.70)**.** *In exakter Rechnung hat die Aufgabe mit*

$$A = \begin{bmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \end{bmatrix}, \quad b = \begin{pmatrix} 2 \\ \varepsilon \\ \varepsilon \end{pmatrix}$$

und $\varepsilon > 0$ die exakte Lösung $x = \binom{1}{1}$. Die Eigenwerte von

$$A^\mathsf{T}A = \begin{bmatrix} 1+\varepsilon^2 & 1 \\ 1 & 1+\varepsilon^2 \end{bmatrix}$$

sind $2+\varepsilon^2$ und ε^2. Die Kondition von A erfüllt demnach $\kappa(A) = \frac{\sqrt{2+\varepsilon^2}}{\varepsilon} \in O(\varepsilon^{-1})$, und $\kappa(A^\mathsf{T}A) = \frac{2}{\varepsilon^2} + 1 \in O(\varepsilon^{-2})$.

Bei Rechnung in double precision *etwa mit $\varepsilon = \mathrm{rd}(10^{-7})$ fallen bei der Berechnung von $A^\mathsf{T}A$ und von $A^\mathsf{T}b = \binom{2+\varepsilon^2}{2+\varepsilon^2}$ Rundungsfehler der relativen Größenordnung $\varepsilon_{\mathrm{mach}}$ an. (**Quizfrage:** Warum?) Diese lassen selbst bei exakter Weiterrechnung einen relativen Vorwärtsfehler in der 2-Norm der Lösung von bis zu $\kappa(A^\mathsf{T}A)\varepsilon_{\mathrm{mach}} \approx 1.4 \cdot 10^{14} \cdot 2.2 \cdot 10^{-16} \approx 4.4 \cdot 10^{-2}$ erwarten, siehe (3.18), (3.21). In der Tat ergibt sich die numerische Lösung mittels Cholesky-Zerlegung von $A^\mathsf{T}A$ und der anschließenden Vorwärts-/Rückwärtssubstitution*

```
import numpy as np
import scipy.linalg
e = 1e-7;
A = np.array([[1, 1], [e, 0], [0, e]])
b = np.array([2, e, e])
x = scipy.linalg.solve(A.T @ A, A.T @ b, assume_a = 'pos')  # select Cholesky solver
```

zu

$$\widehat{x} \approx \begin{pmatrix} 1.011235955056179 \\ 0.988764044943822 \end{pmatrix}.$$

Die numerische Lösung hat also einen relativen Fehler von etwa $1.1 \cdot 10^{-2}$.

[2] In der Tat ist das die Konditionszahl von A wie wir sie aus (4.13) für invertierbare, quadratische Matrizen kennen, wobei aber der Zielraum $\mathbb{R}^m$ auf den tatsächlichen Bildraum von A eingeschränkt wird.

§ 12 QR-Zerlegung

Es stellt sich die Frage, ob man die Normalengleichung (11.3) nicht auch direkter lösen kann, insbesondere ohne $A^\mathsf{T}A$ aufzustellen. Eine Möglichkeit dazu bietet die QR-Zerlegung von A. Wir nehmen dafür wieder an, dass $A \in \mathbb{R}^{m\times n}$ vollen Spaltenrang hat, insbesondere also $m \geqslant n$ gilt.

Das Ziel ist eine Zerlegung von A in der folgenden Form.

Definition 12.1 (QR-Zerlegung). *Es sei $A \in \mathbb{R}^{m\times n}$ mit $m \geqslant n$ und vollem Spaltenrang. Eine Zerlegung der Form*

$$A = QR, \tag{12.1}$$

*wobei $Q \in \mathbb{R}^{m\times n}$ orthonormale Spaltenvektoren hat und $R \in \mathbb{R}^{n\times n}$ eine obere Dreiecksmatrix ist, heißt eine **(reduzierte) QR-Zerlegung** (englisch: (reduced, economic, thin) QR decomposition) von A. Eine QR-Zerlegung heißt **normiert**, wenn die Diagonale von R positiv ist.*

Beachte: Die Spalten von Q bilden unter den Voraussetzungen von Definition 12.1 eine Orthonormalbasis von Bild A.

Satz 12.2 (Existenz und Eindeutigkeit der QR-Zerlegung, vgl. Bornemann, 2018, Satz 9.3). *Es sei $A \in \mathbb{R}^{m\times n}$ mit $m \geqslant n$ und vollem Spaltenrang. Dann besitzt A eine eindeutige normierte QR-Zerlegung. Die Faktoren Q und R ergeben sich aus*

$$\begin{aligned} A^\mathsf{T}A &= R^\mathsf{T}R \quad && \text{(Cholesky-Zerlegung von } A^\mathsf{T}A\text{)}, \\ R^\mathsf{T}Q^\mathsf{T} &= A^\mathsf{T} && \text{(Lösung von } n \text{ linearen Gleichungssystemen mit Koeffizientenmatrix } R^\mathsf{T}\text{)}. \end{aligned}$$

Beweis. Wir beweisen zunächst die Eindeutigkeit. Dazu nehmen wir an, dass $A = QR$ irgendeine normierte QR-Zerlegung von A ist. Dann folgt

$$A^\mathsf{T}A = R^\mathsf{T}Q^\mathsf{T}QR = R^\mathsf{T}R.$$

Die Matrix $A^\mathsf{T}A$ ist s. p. d.. Nach Satz 10.2 (Existenz und Eindeutigkeit der Cholesky-Zerlegung) muss R der eindeutige Cholesky-Faktor von $A^\mathsf{T}A$ sein. Aus $A = QR$ folgt durch Multiplikation mit R^{-1} notwendig

$$Q = AR^{-1},$$

also ist auch Q eindeutig festgelegt.

Kommen wir nun zur Existenz einer normierten QR-Zerlegung. Dazu definieren wir zunächst R als Cholesky-Faktor von $A^\mathsf{T}A = R^\mathsf{T}R$, was nach Satz 10.2 möglich ist. Per Definition besitzt R positive Diagonaleinträge. Weiter definieren wir $Q := AR^{-1}$. Wir müssen noch zeigen, dass die Spalten von Q orthonormal sind. Wir bilden dazu

$$Q^\mathsf{T}Q = R^{-\mathsf{T}}A^\mathsf{T}AR^{-1} = R^{-\mathsf{T}}R^\mathsf{T}RR^{-1} = \mathrm{Id}.$$

□

Wie oben bereits bemerkt, ist das Vorgehen wie im Beweis ungeeignet zur numerischen Bestimmung von R. Es kommt noch hinzu, dass die Orthonormalität der Spalten von Q bei der Definition gemäß $Q := AR^{-1}$ nicht explizit eingefordert wird, sondern sich indirekt ergibt. Es ist also aufgrund von Rundungsfehlern damit zu rechnen, dass die Spalten von Q «nicht besonders» orthonormal sind.

Wir besprechen daher nun drei Alternativen zur Bestimmung der QR-Zerlegung.

§ 12.1 Gram-Schmidt-Verfahren

Das **Gram-Schmidt-Verfahren** ist ein Verfahren, eine gebenene (endliche) Folge linear unabhängiger Vektoren $a_1, \dots, a_n \in \mathbb{R}^m$ so in eine Folge von Vektoren $q_1, \dots, q_n$ zu modifizieren, dass folgende Eigenschaften erfüllt sind:

(1) $\operatorname{span}\{q_1, \dots, q_k\} = \operatorname{span}\{a_1, \dots, a_k\}$ für alle $k = 1, \dots, n$.

(2) Die Vektoren ~~span~~$^{\text{GM}}\{q_1, \dots, q_n\}$ sind paarweise orthonormal.

Zu diesem Zweck definieren wir q_1 als normierte Version von a_1, definieren anschließend q_2 als a_2 ohne die Anteile von a_2, die bereits in dem Unterraum liegen, der von q_1 aufgespannt wird, und normieren anschließend wieder usw. Man spricht auch davon, dass man den Vektor a_k «gegen die vorhergehenden Vektoren orthonormalisiert».

Dieses Verfahren wenden wir nun auf die Spaltenvektoren von A an. Das führt zu Algorithmus 12.3.

Algorithmus 12.3 (Gram-Schmidt-Verfahren zur Berechnung der QR-Zerlegung).
Eingabe: *Matrix* $A \in \mathbb{R}^{m\times n}$ *mit* $\operatorname{Rang}(A) = n$
Ausgabe: *Faktoren* Q *und* R *der normierten QR-Zerlegung* $A = QR$

1: ***for*** $k = 1, 2, \dots, n$ ***do***
2: ***for*** $\ell = 1, 2, \dots, k-1$ ***do***
3: *Berechne* $r_{\ell k} := a_k^{\mathsf{T}} q_\ell$ *// Bestimmung der Orthogonalisierungskoeffizienten*
4: ***end for***
5: *Setze* $q_k := a_k$
6: ***for*** $\ell = 1, 2, \dots, k-1$ ***do***
7: *Setze* $q_k := q_k - r_{\ell k}\, q_\ell$ *// Orthogonalisierung von* a_k *gegen* q_ℓ
8: ***end for***
9: *Setze* $r_{kk} := \|q_k\|_2$
10: *Setze* $q_k := \frac{q_k}{r_{kk}}$ *// Normalisierung von* q_k
11: ***end for***

Quizfrage: In der Praxis kann Algorithmus 12.3 so implementieren, dass kein zusätzlicher Speicher für Q benötigt wird, sondern dass A *in place* überschrieben wird. Was ist dafür zu ändern?

Algorithmus 12.3 bestimmt zunächst die Koeffizienten $r_{k\ell}$ zur Orthogonalisierung von a_k gegen alle früheren Vektoren q_ℓ, bevor die Orthogonalisierung ausgeführt wird (Zeile 7). Stattdessen könnten wir

auch die Orthogonalisierung nach und nach ausführen und dabei die Berechnung des Koeffizienten $r_{k\ell}$ hinauszögern, bis die Orthogonalisierung von a_k gegen ein bestimmtes q_ℓ an der Reihe ist. In Algorithmus 12.3 führt das dazu, dass wir Zeilen 2 bis 8 ersetzen durch

2: Setze $q_k := a_k$
3: **for** $\ell = 1, 2, \dots, k-1$ **do**
4: Berechne $r_{\ell k} := q_k^\mathsf{T} q_\ell$ // Bestimmung der Orthogonalisierungskoeffizienten
5: Setze $q_k := q_k - r_{\ell k}\, q_\ell$ // sukzessive Orthogonalisierung von a_k gegen q_ℓ
6: **end for**

Das ist in exakter Rechnung äquivalent zu Algorithmus 12.3, aber numerisch von Vorteil. Diese Variante heißt **modifiziertes Gram-Schmidt-Verfahren**. Wir können es, wie in Algorithmus 12.4 gezeigt, so implementieren, dass die Anteile einer Spalte q_k, sobald diese fertig vorliegt, sofort aus allen späteren Vektoren a_ℓ herausgerechnet wird.

Algorithmus 12.4 (modifiziertes Gram-Schmidt-Verfahren zur Berechnung der QR-Zerlegung)**.**
Eingabe: *Matrix $A \in \mathbb{R}^{m\times n}$ mit* $\operatorname{Rang}(A) = n$
Ausgabe: *Faktoren Q und R der normierten QR-Zerlegung $A = QR$*
1: *Setze $Q := A$*
2: ***for*** $k = 1, 2, \dots, n$ ***do***
3: *Setze $r_{kk} := \|q_k\|_2$*
4: *Setze $q_k := \frac{q_k}{r_{kk}}$* // *Normalisierung von q_k*
5: ***for*** $\ell = k+1, k+2, \dots, n$ ***do***
6: *Berechne $r_{k\ell} := q_k^\mathsf{T} q_\ell$* // *Bestimmung der Orthogonalisierungskoeffizienten*
7: *Setze $q_\ell := q_\ell - r_{k\ell}\, q_k$* // *sukzessive Orthogonalisierung von q_ℓ gegen q_k*
8: ***end for***
9: ***end for***

Quizfrage: In welcher Reihenfolge werden in Algorithmus 12.3 bzw. in Algorithmus 12.4 die Einträge der Matrizen Q und R berechnet? **Quizfrage:** Woran könnte es liegen, dass die Variante Algorithmus 12.4 numerisch vorteilhafter ist als Algorithmus 12.3?

Quizfrage: Was passiert, wenn die Matrix A nicht vollen Spaltenrang hat?

Satz 12.5 (Aufwand des modifizierten Gram-Schmidt-Verfahrens)**.** *Der numerische Aufwand für die Erzeugung der QR-Zerlegung einer Matrix $A \in \mathbb{R}^{m\times n}$ mit $m \geqslant n$ durch Algorithmus 12.4 ist von der Größenordnung $2mn^2$.*

Beweis. □

§ 12.2 Givens-Rotationen

Das (modifizierte) Gram-Schmidt-Verfahren in exakter Rechnung bricht ab, falls A nicht vollen Spaltenrang besitzt. In numerischer Rechnung kann es auch bis zum Ende durchlaufen, wenn der betreffende

Vektor q_k nicht exakt der Nullvektor ist. Das Verfahren produziert dann eine Matrix Q, die stark von der Orthogonalität abweicht.

Das in diesem Abschnitt entwickelte Verfahren hat diese Nachteile nicht. Es stellt keine Anforderung an den Spaltenrang von A. Es erstellt für eine Matrix $A \in \mathbb{R}^{m\times n}$ mit $m \geqslant n$ außerdem eine **(volle) QR-Zerlegung** (englisch: *full QR decomposition*

$$A = QR = \left[Q_1 \mid Q_2\right]\left[\frac{R_1}{0}\right], \tag{12.2}$$

wobei $Q \in \mathbb{R}^{m\times m}$ orthonormale Spaltenvektoren hat und $R \in \mathbb{R}^{m\times n}$ eine obere Dreiecksmatrix ist. Die Teilmatrizen haben die Dimensionen $Q_1 \in \mathbb{R}^{m\times n}$, $Q_2 \in \mathbb{R}^{m\times(m-n)}$ und $R_1 \in \mathbb{R}^{n\times n}$.

Die Idee ist es, A durch Multiplikation mit einer Folge von Matrizen von links auf obere Dreiecksgestalt zu transformieren, indem in geeigneter Reihenfolge Nullen unterhalb der Diagonalen erzeugt werden. Im Unterschied zur LR-Zerlegung erfolgt dies aber nicht durch Frobeniusmatrizen, sondern mit Hilfe orthogonaler Rotationsmatrizen, sogenannter Givens-Rotationen.

Definition 12.6 (Givens-Rotation). *Eine quadratische Matrix der Form*[3]

$$\Omega^{(k,\ell)} := \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & c & & -s & & \leftarrow \ell \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \\ & & & s & & c & & \leftarrow k \\ & & & & & & & 1 \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{bmatrix}$$

$$\uparrow \ell \qquad \uparrow k$$

heißt ***Givens-Rotation****, wenn c und s die Bedingung $c^2 + s^2 = 1$ erfüllen. Es gibt daher einen eindeutigen Winkel $\theta \in [0, 2\pi)$, sodass $c = \cos(\theta)$ und $s = \sin(\theta)$ gilt.*

Eine Givens-Rotation ist eine orthogonale Matrix mit Determinante 1, repräsentiert also tatsächlich eine Rotation. (**Quizfrage:** Beweis?)

Frage: Wie wirkt eine Givens-Rotation?

Es sei dazu $x := \binom{a}{b} \in \mathbb{R}^2 \setminus \{0\}$ und $r := \|x\|_2 = \sqrt{a^2 + b^2}$. Setzen wir $c := \frac{a}{r}$ und $s := -\frac{b}{r}$, dann wirkt die Givens-Rotation

$$\Omega = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$$

[3] Nicht notierte Einträge sind Null.

wie folgt auf den Vektor x:

$$\Omega x = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}.$$

Quizfrage: Was ist der zu dieser Transformation gehörende Winkel θ?

Diese Beobachtung können wir nutzen, um in geeigneter Reihenfolge Nullen unterhalb der Diagonale von A zu erzeugen.

Beispiel 12.7 (Transformation einer Matrix auf obere Dreiecksgestalt durch Givens-Rotationen)**.** *Es sei $A \in \mathbb{R}^{4\times 2}$. Wir führen die folgenden Givens-Rotationen aus. Dabei bezeichnet # ein im aktuellen Schritt geändertes Element. Die blau markierten Einträge sind diejenigen, die in der nächsten Givens-Rotation adressiert werden.*

$$\begin{bmatrix} * & * \\ * & * \\ * & * \\ * & * \end{bmatrix} \xrightarrow{\Omega^{(3,4)}} \begin{bmatrix} * & * \\ * & * \\ \# & \# \\ 0 & \# \end{bmatrix} \xrightarrow{\Omega^{(2,3)}} \begin{bmatrix} * & * \\ \# & \# \\ 0 & \# \\ 0 & * \end{bmatrix} \xrightarrow{\Omega^{(1,2)}} \begin{bmatrix} \# & \# \\ 0 & \# \\ 0 & * \\ 0 & * \end{bmatrix}$$

$$\begin{bmatrix} * & * \\ 0 & * \\ 0 & * \\ 0 & * \end{bmatrix} \xrightarrow{\Omega'^{(3,4)}} \begin{bmatrix} * & * \\ 0 & * \\ 0 & \# \\ 0 & 0 \end{bmatrix} \xrightarrow{\Omega'^{(2,3)}} \begin{bmatrix} * & * \\ 0 & \# \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} * & * \\ 0 & * \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \xrightarrow{\Omega''^{(3,4)}} \begin{bmatrix} * & * \\ 0 & * \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Am Ende dieses Vorgangs haben wir die Darstellung

$$\Omega''^{(3,4)}\Omega'^{(2,3)}\Omega'^{(3,4)}\Omega^{(1,2)}\Omega^{(2,3)}\Omega^{(3,4)}A = R.$$

Die gesuchte orthogonale Transformationsmatrix ist also die Inverse:

$$Q = \left(\Omega^{(3,4)}\right)^{\mathsf{T}}\left(\Omega^{(2,3)}\right)^{\mathsf{T}}\left(\Omega^{(1,2)}\right)^{\mathsf{T}}\left(\Omega'^{(3,4)}\right)^{\mathsf{T}}\left(\Omega'^{(2,3)}\right)^{\mathsf{T}}\left(\Omega''^{(3,4)}\right)^{\mathsf{T}}.$$

Quizfrage: Warum erzeugt man die Nullen Spalte für Spalte von unten nach oben und nicht in einer anderen Reihenfolge? Zum Beispiel Spalte für Spalte von oben nach unten? Oder erst alle Nullen in der letzten Zeile, dann die Nullen in der Zeile darüber usw.? Warum bleiben einmal erzeugte Nullen erhalten?

Bemerkung 12.8 (QR-Zerlegung mit Givens-Rotationen)**.**

(*i*) *In der Praxis berechnet man Q möglichst nicht als Matrix, sondern speichert stattdessen die Faktoren c and s für die Teilmatrizen. Wenn man später das Matrix-Vektor-Produkt Qx oder $Q^{\mathsf{T}}y$ benötigt, dann kann es Schritt für Schritt berechnet werden.*

(*ii*) *Durch Givens-Rotationen erhält man i. A. keine positive Diagonale in R, also keine normierte QR-Zerlegung.*

(*iii*) *Givens-Rotationen sind besonders dann günstig, wenn die Matrix A bereits viele Null-Einträge besitzt.*

(*iv*) *Das beschriebene Verfahren berechnet eine volle QR-Zerlegung (12.2), die natürlich die Faktoren der reduzierten QR-Zerlegung als Teilmatrizen enthält.*

Wir betonen nochmals, dass die QR-Zerlegung mittels Givens-Rotationen auch dann möglich ist, wenn A nicht vollen Spaltenrang besitzt.

§ 12.3 Householder-Transformationen

Die Idee einer Householder-Transformation ist es, einen Vektor $a \in \mathbb{R}^m$ an einer Hyperebene mit geeignetem Normalenvektor v zu reflektieren, sodass a auf $\|a\|_2\, e_1$ oder auf $-\|a\|_2\, e_1$ abgebildet wird, wobei $e_1 = (1, 0, \ldots, 0)^\mathsf{T} \in \mathbb{R}^m$ der erste Einheitsvektor ist.

Frage: Wie sehen solche Reflexionsmatrizen aus?

Der Anteil eines beliebigen Vektors x in Richtung von v ist gegeben durch

$$\frac{v^\mathsf{T} x}{\|v\|_2^2}.$$

Bei der Reflexion soll dieser Anteil das Vorzeichen wechseln. Also hat die gesuchte Reflexion die Darstellung

$$Qx = x - 2\,\frac{v^\mathsf{T} x}{\|v\|_2^2}\, v, \quad \text{d. h. } Q = \mathrm{Id} - 2\frac{v\, v^\mathsf{T}}{\|v\|_2^2}. \tag{12.3}$$

Beachte: Die Länge von v ist dabei unerheblich.

Definition 12.9 (Householder-Transformation)**.** *Eine Abbildung bzw. eine Matrix der Form (12.3) heißt* ***Householder-Transformation*** *oder* ***Householder-Reflexion*** *(englisch:* Householder transformation, Householder reflection*).*

Beachte: Solche Matrizen sind orthogonal und symmetrisch, erfüllen also auch $Q^2 = \mathrm{Id}$. **Quizfrage:** Klar?

Wie muss v gewählt werden, damit Q einen gegebenen Vektor a auf $\|a\|_2\, e_1$ abbildet? Aus (12.3) erhalten wir die Bedingung

$$Qa = a - 2\,\frac{v^\mathsf{T} a}{\|v\|_2^2}\, v \overset{!}{=} \|a\|_2\, e_1.$$

Umstellen ergibt

$$2\,\frac{v^\mathsf{T} a}{\|v\|_2^2}\, v = a - \|a\|_2\, e_1,$$

also ist v (irgendein) Vielfaches von $a - \|a\|_2 e_1$, und wir setzen

$$v := a - \|a\|_2 e_1. \tag{12.4}$$

Möchte man v stattdessen auf $-\|a\|_2 e_1$ abbilden, so kann

$$v := a + \|a\|_2 e_1 \tag{12.5}$$

verwendet werden. Damit liegt die gesuchte Matrix Q in (12.3) fest. In der Praxis wählt man (12.4), wenn der erste Eintrag des Vektors a negativ ist, sonst (12.5). **Quizfrage:** Warum ist das günstig?

Im ersten Schritt der QR-Zerlegung mittels Householder-Reflexionen wendet man nun eine solche Matrix $Q^{(1)}$ auf A an, wobei als Vektor a die erste Spalte von A verwendet wird. In diesem Schritt ergibt sich also

$$\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \xrightarrow{Q^{(1)}} \begin{bmatrix} \# & \# & \# \\ 0 & \# & \# \\ 0 & \# & \# \\ 0 & \# & \# \end{bmatrix}.$$

Anschließend wird dasselbe Verfahren nochmal auf die rot markierte Untermatrix von $Q^{(1)}A$ angewendet. Die dabei verwendete Matrix $Q \in \mathbb{R}^{(m-1)\times(m-1)}$ wird derart zu $Q^{(2)}$ ergänzt, dass die erste Zeile von $Q^{(1)}A$ unverändert bleibt, also

$$Q^{(2)} = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & Q \end{array}\right].$$

Durch wiederholte Anwendung dieser Idee erhalten wir die Darstellung

$$Q^{(n)} \cdots Q^{(1)} A = R.$$

Die gesuchte Transformationsmatrix Q ist also die Inverse:

$$Q = Q^{(1)} \cdots Q^{(n)}.$$

Quizfrage: Warum werden hier die Faktoren nicht transponiert?

Bemerkung 12.10 (QR-Zerlegung mit Householder-Transformationen)**.**

(*i*) *In der Praxis berechnet man Q möglichst nicht als Matrix, sondern speichert stattdessen die Vektoren v aus den einzelnen Teilschritten. Wenn man später das Matrix-Vektor-Produkt Qx oder $Q^T y$ benötigt, dann kann es Schritt für Schritt berechnet werden.*

(*ii*) *Durch Householder-Transformationen erhält man i. A. keine positive Diagonale in R, also keine normierte QR-Zerlegung.*

(*iii*) *Das beschriebene Verfahren berechnet eine volle QR-Zerlegung (12.2), die natürlich die Faktoren der reduzierten QR-Zerlegung als Teilmatrizen enthält.*

(*iv*) *Householder-Transformationen werden bei der QR-Zerlegung durch MATLABs* `qr`*-Routine sowie in* `numpy.linalg.qr` *und* `scipy.linalg.qr` *verwendet.*

Wir betonen auch hier, dass die QR-Zerlegung mittels Householder-Transformationen auch dann möglich ist, wenn A nicht vollen Spaltenrang besitzt.

§ 12.4 Nutzung der QR-Zerlegung

Wir nehmen jetzt wieder an, dass A vollen Spaltenrang n besitzt. Die Optimalitätsbedingung (Normalengleichung) der Kleinste-Quadrate-Aufgabe (11.1)

$$A^\mathsf{T}Ax = A^\mathsf{T}b \tag{11.3}$$

lässt sich bei Vorliegen einer (reduzierten) QR-Zerlegung (12.1) von A einfach lösen, denn (11.3) ist äquivalent zu $R^\mathsf{T}Q^\mathsf{T}QRx = R^\mathsf{T}Q^\mathsf{T}b$, also auch zu

$$Rx = Q^\mathsf{T}b. \tag{12.6}$$

Beispiel 12.11 (Lösung von Beispiel 11.4 mittels QR-Zerlegung)**.** *Wir greifen nochmal das Beispiel 11.4 auf, lösen das Ausgleichsproblem dieses Mal aber mittels QR-Zerlegung (durch Householder-Transformationen):*

```
import numpy as np
import scipy.linalg
e = 1e-7;
A = np.array([[1, 1], [e, 0], [0, e]])
b = np.array([2, e, e])
Q, R = np.linalg.qr(A, mode = 'reduced')
# Q, R = scipy.linalg.qr(A, mode = 'economic')
x = scipy.linalg.solve_triangular(R, Q.T @ b)
```

Das ergibt eine numerische Lösung von

$$\widehat{x} \approx \begin{pmatrix} 1.0000000000000000 \\ 1.0000000000000004 \end{pmatrix}.$$

Die numerische Lösung hat also nur noch einen relativen Fehler von etwa $4.4 \cdot 10^{-16}$.

Wenn die rechte Seite b bereits zu dem Zeitpunkt bekannt ist, wenn man die QR-Zerlegung von A berechnet, so kann man sie an die Matrix A anhängen und die QR-Zerlegung von $\begin{bmatrix} A & b \end{bmatrix}$ bestimmen lassen. Dann bekommt man im Ergebnis $Q^\mathsf{T}b$ gleich mitgeliefert und benötigt die Matrix Q später überhaupt nicht mehr. Durch Verzicht auf die Rückgabe von Q bei der QR-Zerlegung kann nochmal numerischer Aufwand gespart werden.

```
import numpy as np
import scipy.linalg
e = 1e-7;
A = np.array([[1, 1], [e, 0], [0, e]])
b = np.array([2, e, e])
R = np.linalg.qr(np.hstack((A, b[:,None])), mode = 'r')  # return (full size) R only
QTb = R[:2,2]  # extract Q' * b
R = R[:2,:2]  # extract reduced R
x = scipy.linalg.solve_triangular(R, QTb)
```

Das ergibt eine numerische Lösung von

$$\widehat{x} \approx \begin{pmatrix} 0.9999999999999996 \\ 1.0000000000000004 \end{pmatrix}$$

mit einem relativen Fehler von etwa $6.3 \cdot 10^{-16}$.

§ 13 Nutzung der Singulärwertzerlegung

In diesem Abschnitt betrachten wir noch, wie wir alternativ zur QR-Zerlegung auch die Singulärwertzerlegung[4]

$$A = U\Sigma V^\mathsf{T}$$

der Matrix $A \in \mathbb{R}^{m\times n}$ nutzen können, um die Kleinste-Quadrate-Aufgabe (11.1) zu lösen. Dafür ist es nicht einmal erforderlich, dass A vollen Spaltenrang besitzt. Sollte dies nicht der Fall sein, so ist die Lösung nicht eindeutig (Folgerung 11.3). In dem Fall können wir aber nach der eindeutigen Lösung mit der kleinsten Norm fragen. Anders ausgedrückt fragen wir nach derjenigen Lösung, die in $(\ker A)^\perp$ liegt. **Quizfrage:** Warum ist die normminimale Lösung der Normalengleichung (11.3) gerade auch diejenige, die in $(\ker A)^\perp$ liegt?

Satz 13.1 (Lösung der Kleinste-Quadrate-Aufgabe durch Singulärwertzerlegung, vgl. Rannacher, 2017, Satz 4.13)**.** *Es sei $A \in \mathbb{R}^{m\times n}$ mit $m \geqslant n$, und es gelte $r = \operatorname{Rang}(A)$. Dann ist die eindeutig bestimmte Lösung der Kleinste-Quadrate-Aufgabe (11.1) mit minimaler Norm gegeben durch*

$$x = \sum_{i=1}^{r} \frac{u_i^\mathsf{T} b}{\sigma_i} v_i. \tag{13.1}$$

Das zugehörige Residuum hat die quadrierte Norm

$$\|r\|_2^2 = \|Ax - b\|_2^2 = \sum_{i=r+1}^{m} (u_i^\mathsf{T} b)^2. \tag{13.2}$$

Beweis. Für jedes $x \in \mathbb{R}^n$ gilt

$$\|Ax - b\|_2^2 = \|AVV^\mathsf{T}x - b\|_2^2 = \|U^\mathsf{T}AVV^\mathsf{T}x - U^\mathsf{T}b\|_2^2 = \|\Sigma V^\mathsf{T}x - U^\mathsf{T}b\|_2^2.$$

Wir setzen zur Abkürzung $z := V^\mathsf{T}x \in \mathbb{R}^n$. Dann lautet der Ausdruck in der Norm ausgeschrieben:

$$\left[\begin{array}{ccc} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ \hline 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{array}\right] \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix} - \begin{bmatrix} — & u_1^\mathsf{T} & — \\ & \vdots & \\ — & u_n^\mathsf{T} & — \\ — & u_{n+1}^\mathsf{T} & — \\ & \vdots & \\ — & u_m^\mathsf{T} & — \end{bmatrix} \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_m \end{pmatrix}.$$

[4] deren Berechnung übrigens auch Householder-Transformationen nutzt

Durch die Wahl $z_i = \frac{u_i^\intercal b}{\sigma_i}$ für $i = 1, \ldots r$ können wir die ersten r Komponenten zu Null machen, wodurch $\|\Sigma V^\intercal x - U^\intercal b\|_2^2$ minimiert wird. Sollte $r < n$ sein, so wählen wir $z_i = 0$ für die verbleibenden Indizes $i = r{+}1, \ldots, n$, um die Lösung mit minimaler Norm zu erhalten. (**Quizfrage:** Warum führt eine minimale Norm in z auch zu einer minimaler Norm in $x = Vz$?) Die Aussage (13.2) folgt unmittelbar. □

```
import numpy as np
import scipy.linalg
e = 1e-7;
A = np.array([[1, 1], [e, 0], [0, e]])
b = np.array([2, e, e])
U, Sigma, VT = np.linalg.svd(A, full_matrices = False)
x = np.dot(VT.T, np.dot(np.diag(1/Sigma), np.dot(U.T, b)))
```

Das ergibt eine numerische Lösung von

$$\widehat{x} \approx \begin{pmatrix} 0.9999999999999997 \\ 1.0000000000000002 \end{pmatrix}$$

mit einem relativen Fehler von etwa $4.0 \cdot 10^{-16}$.

Ende der Woche 8

Kapitel 6 Interpolation und Approximation

§ 14 Einführung

In diesem Kapitel geht es darum, «beliebige» Funktionen f durch einfach zu handhabende Funktionen g zu approximieren (anzunähern). Dafür gibt es mindestens drei Motivationen:

(1) Die Auswertung von g (zum Beispiel ein Polynom) kann numerisch wesentlich günstiger sein als die Auswertung der (komplizierten) Funktion f.

(2) Die Funktion g kann durch i. d. R. wenige Parameter repräsentiert werden, also mit wenig Speicherbedarf dargestellt werden.

(3) Von der Funktion f sind nur endlich viele Funktionswerte (etwa aus Messungen) bekannt, die etwa in einer Tabelle von Stützstellen x_i und zugehörigen Funktionswerten $f(x_i)$ aufgelistet sind. Wir wollen für diese Funktion f aber gerne auch sinnvolle Werte zwischen den Stützstellen angeben können, sie also zu einer überall definierten Funktion g fortsetzen.

Es gibt verschiedene Möglichkeiten, eine Approximation g zu einer gegebenen Funktion f festzulegen. Eine Strategie ist es, g aus einer Menge von zur Verfügung stehenden Funktionen (meist ein endlich-dimensionaler Vektorraum) so auszuwählen, dass der Abstand (Approximationsfehler) $\|f - g\|$ minimiert wird. Dabei ist $\|\cdot\|$ eine geeignete Norm auf dem Funktionenraum, dem f angehört, und g wird in einem endlich-dimensionalen Unterraum gesucht. Solche Approximationsaufgaben behandeln wir in Abschnitt 17.

Eine andere Strategie besteht darin, g in einer Menge zur Verfügung stehender Funktionen dadurch festzulegen, dass g mit f an gewissen Stützstellen übereinstimmt. Man spricht dann davon, dass die Funktion g die Funktion f an diesen Stützstellen interpoliert. Solche Aufgaben sind Gegenstand von Abschnitt 15.

Wir behandeln im gesamten Kapitel 6 nur Aufgaben für Funktionen, die auf $\mathbb{R}$ definiert sind, also nur von einer einzigen Variablen $x \in \mathbb{R}$ abhängen. Typische Funktionen g, die man für die Approximation

oder Interpolation verwendet, sind:

$$
\begin{aligned}
&a_0 + a_1 x + \cdots + a_n x^n && \text{(Polynome)},\\
&\frac{a_0 + a_1 x + \cdots + a_n x^n}{b_0 + b_1 x + \cdots + b_m x^m} && \text{(rationale Funktionen)},\\
&\frac{1}{2} a_0 + \sum_{k=1}^{n} \big(a_k \cos(kx) + b_k \sin(kx) \big) && \text{(trigonometrische Polynome)},\\
&\sum_{k=1}^{n} a_k \exp(b_k x) && \text{(Exponentialsummen)}.
\end{aligned}
$$

Die Parameter, die die Funktionen beschreiben, sind jeweils hervorgehoben.

§ 15 Polynominterpolation

Wir behandeln in diesem Abschnitt die **Interpolation** (englisch: *interpolation*) mit Hilfe von Polynomen. Dazu bezeichnen wir mit

$$P_n := \left\{ p(x) = \sum_{k=0}^{n} a_k x^k \;\middle|\; a_0, \ldots, a_k \in \mathbb{R} \right\} \tag{15.1}$$

den Vektorraum der **Polynome** (englisch: *polynomials*) vom Höchstgrad $n \in \mathbb{N}_0$. Die Zahlen $a_0, \ldots, a_k$ heißen die **Koeffizienten** (englisch: *coefficients*) des Polynoms p.

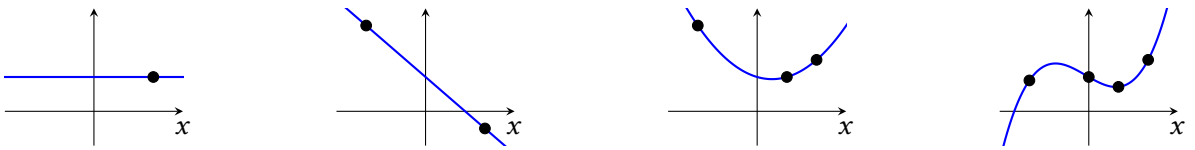
Definition 15.1 (Interpolationsaufgabe)**.** *Zu gegebenen **Stützstellen** (englisch:* supporting points, sampling points, interpolation points*) $x_0, \ldots, x_n \in \mathbb{R}$ und zugehörigen Funktionswerten (**Stützwerte**) $y_0, \ldots, y_n \in \mathbb{R}$ lautet die **Interpolationsaufgabe** (englisch:* interpolation problem*) wie folgt:*

> *Finde $p \in P_n$, sodass die **Interpolationsbedingungen** (englisch:* interpolation conditions*)*
> $p(x_i) = y_i$ *für alle $i = 0, 1, \ldots, n$ gelten.* (15.2)

Beachte: Wir beginnen mit der Nummerierung der Stützstellen beim Index 0, damit der Index n der letzten Stützstelle mit dem Maximalgrad des Polynoms übereinstimmt.

Für die Polynomgrade $n = 0, 1, 2, 3$ spricht man speziell von **konstanter**, **linearer** **quadratischer** bzw. **kubischer Interpolation**.

Zu einer Aussage über die eindeutige Lösbarkeit der Interpolationsaufgabe kommen wir gleich. Zunächst eine Illustration verschiedener Interpolationsaufgaben mit Polynomgraden $n = 0, 1, 2, 3$.



Quizfrage: Erwarten Sie, dass die Lösung der Interpolationsaufgabe (15.2) immer ein Polynom vom Grad n ist? Oder wann ist das ggf. nicht der Fall?

§ 15.1 Monombasis

Setzen wir die Stützstellen $x_0, \dots, x_n$ nacheinander in die Darstellung (15.1) ein, so erhalten wir mit

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{15.3}$$

eine Bestimmungsgleichung für die unbekannten Koeffizienten $a_0, \dots, a_n$.

Beispiel 15.2. *Für die Paare von Stützstellen und Stützwerten* $(-1, 5)$, $(0, 3)$, $(1, -7)$ *und* $(2, 0)$ *ergibt sich das System*

$$\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \end{bmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 3 \\ -7 \\ 0 \end{pmatrix}$$

mit der (eindeutigen) Lösung $\frac{1}{6}(18, -61, -24, 25)^\mathsf{T}$. *Das gesuchte Polynom ist also*

$$p(x) = 3 - \frac{61}{6}x - 4x^2 + \frac{25}{6}x^3.$$

Die Matrix in (15.3) heißt die **Vandermonde-Matrix** $V[x_0, \dots, x_n]$ zu den Stützstellen $x_0, \dots, x_n$. Sie hat folgende Eigenschaften:

Satz 15.3 (Eigenschaften der Vandermonde-Matrix)**.**

(*i*) $\det V[x_0, \dots, x_n] = \prod_{0 \leqslant i < j \leqslant n} (x_j - x_i)$.

(*ii*) $V[x_0, \dots, x_n]$ *ist genau dann regulär, wenn die Stützstellen paarweise verschieden sind.*

(*iii*) *In diesem Fall erfüllt die Konditionszahl die Abschätzung*

$$\kappa(V[x_0, \dots, x_n]) \geqslant \frac{2^{n-1}}{\sqrt{n+1}}. \tag{15.4}$$

Falls alle $|x_j| \geqslant 1$ *sind oder aber alle* $|x_j| \leqslant 1$, *dann gilt sogar*

$$\kappa(V[x_0, \dots, x_n]) \geqslant 2^{n-1}. \tag{15.5}$$

Beweis. Zu Aussage (*i*): Der Beweis verwendet Induktion über die Dimension der Matrix und folgt Beutelspacher, 2014, Kapitel 7.5. Für $n = 0$ ist die Indexmenge (i, j) in $\prod_{0 \leqslant i < j \leqslant n}$ leer. Definitionsgemäß ist daher das Produkt gleich 1, was mit $\det V[x_0] = \det(1) = 1$ übereinstimmt. Es sei nun die Behauptung für $n - 1$ bereits gezeigt. Wir multiplizieren die vorletzte Spalte mit x_0 und subtrahieren sie von der letzten. Dann multiplizieren wir die vorvorletzte Spalte mit x_0 und subtrahieren sie von der vorletzten usw. In Matrixform geschrieben bilden wir also das Produkt

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} 1 & -x_0 & & & \\ & 1 & \ddots & & \\ & & \ddots & -x_0 & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & x_1(x_1 - x_0) & \cdots x_1^{n-1}(x_1 - x_0) \\ \vdots & & & \vdots \\ 1 & x_n - x_0 & x_n(x_n - x_0) & \cdots x_n^{n-1}(x_n - x_0) \end{bmatrix}.$$

Die Determinante ändert sich durch diese Manipulation nicht. (**Quizfrage:** Warum?) Die Determinante der erhaltenen Produktmatrix kann mit Hilfe des Entwicklungssatzes nach der ersten Zeile entwickelt werden. Wir erhalten

$$\det\left(\begin{bmatrix} x_1 - x_0 & x_1(x_1 - x_0) & \cdots & x_1^{n-1}(x_1 - x_0) \\ \vdots & & & \vdots \\ x_n - x_0 & x_n(x_n - x_0) & \cdots & x_n^{n-1}(x_n - x_0) \end{bmatrix}\right).$$

Nun können wir aus der ersten Zeile den Faktor $x_1 - x_0$ ausklammern, aus der zweiten den Faktor $x_2 - x_0$ ausklammern usw. und erhalten

$$(x_1 - x_0)(x_2 - x_0) \cdots (x_n - x_0) \det\left(\begin{bmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & & & \vdots \\ 1 & x_n & \cdots & x_n^{n-1} \end{bmatrix}\right).$$

Die verbleibende Matrix ist dabei eine Vandermonde-Matrix der Dimension $(n-1) \times (n-1)$. Hier können wir die Induktionsannahme verwenden und erhalten schließlich

$$(x_1 - x_0)(x_2 - x_0) \cdots (x_n - x_0) \prod_{1 \leqslant i < j \leqslant n} (x_j - x_i) = \prod_{0 \leqslant i < j \leqslant n} (x_j - x_i),$$

was den Induktionsbeweis abschließt.

Aussage (*ii*) folgt sofort aus Aussage (*i*), weil die Determinante genau dann ungleich Null ist, wenn die Stützstellen paarweise verschieden sind.

Aussage (*iii*) wurde in Tyrtyshnikov, 1994, Theorem 4.1 bewiesen. □

Folgerung 15.4 (Eindeutige Lösbarkeit von Interpolationsaufgabe (15.2))**.** *Die Interpolationsaufgabe* (15.2) *ist genau dann eindeutig lösbar, wenn die Stützstellen* $x_0, \ldots, x_n$ *paarweise verschieden sind. (Auf die Stützwerte* $y_0, \ldots, y_n$ *kommt es nicht an.)*

Neben der schlechten Kondition der Matrix in (15.3) ist auch noch der Aufwand $\sim n^3$ als Nachteil unseres Ansatzes in diesem Abschnitt zu nennen. Dazu kommt, dass beim Hinzufügen einer weiteren

Stützstelle das Gleichungssystem (15.3) neu gelöst werden muss. Alle Nachteile kommen daher, dass offenbar die gewählte Darstellung

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

des gesuchten Polynoms nicht gut geeignet ist. Diese Darstellung kam daher, dass wir (ohne uns Gedanken darüber zu machen) als Basis des Raumes P_n die **Monome** $\{1, x, x^2, \ldots, x^n\}$ verwendet haben. Daher behandeln wird in den folgenden zwei Abschnitten zwar dieselbe Interpolationsaufgabe (15.2), jedoch mit anderen Basen für den Polynomraum P_n.

§ 15.2 Lagrange-Polynombasis

Zu gegebenen, paarweise verschiedenen Stützstellen $x_0, \ldots, x_n \in \mathbb{R}$ heißt das Polynom

$$L_i^{(n)}(x) := \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{15.6}$$

das i-te **Lagrange-Polynom** (englisch: *Lagrange polynomial*).

Satz 15.5 (Eigenschaften der Lagrange-Polynome)**.** *Es seien $x_0, \ldots, x_n$ gegeben. Dann gilt für die Lagrange-Polynome (15.6):*

(*i*)
$$L_i^{(n)}(x_j) = \begin{cases} 1, & \text{falls } i = j, \\ 0, & \text{falls } i \neq j. \end{cases} \tag{15.7}$$

(*ii*) *Die Lagrange-Polynome $L_0^{(n)}, \ldots, L_n^{(n)}$ bilden eine Basis von P_n.*

(*iii*) *Die Summe aller Lagrange-Polynome ist konstant gleich* 1*:*

$$\sum_{i=0}^{n} L_i^{(n)} \equiv 1. \tag{15.8}$$

Beweis. □

Aufgrund der Eigenschaft (15.7) ist die Interpolationsaufgabe (15.2) bei einer Darstellung in der Lagrange-Basis besonders einfach zu lösen, denn es gilt

$$p(x) = \sum_{i=0}^{n} y_i \, L_i^{(n)}(x). \tag{15.9}$$

Es muss also kein Gleichungssystem gelöst werden. Wir haben jedoch auch Nachteile. Der Aufwand für die Auswertung des Interpolationspolynoms (15.9), wenn man die $L_i^{(n)}$ in der Darstellung (15.6)

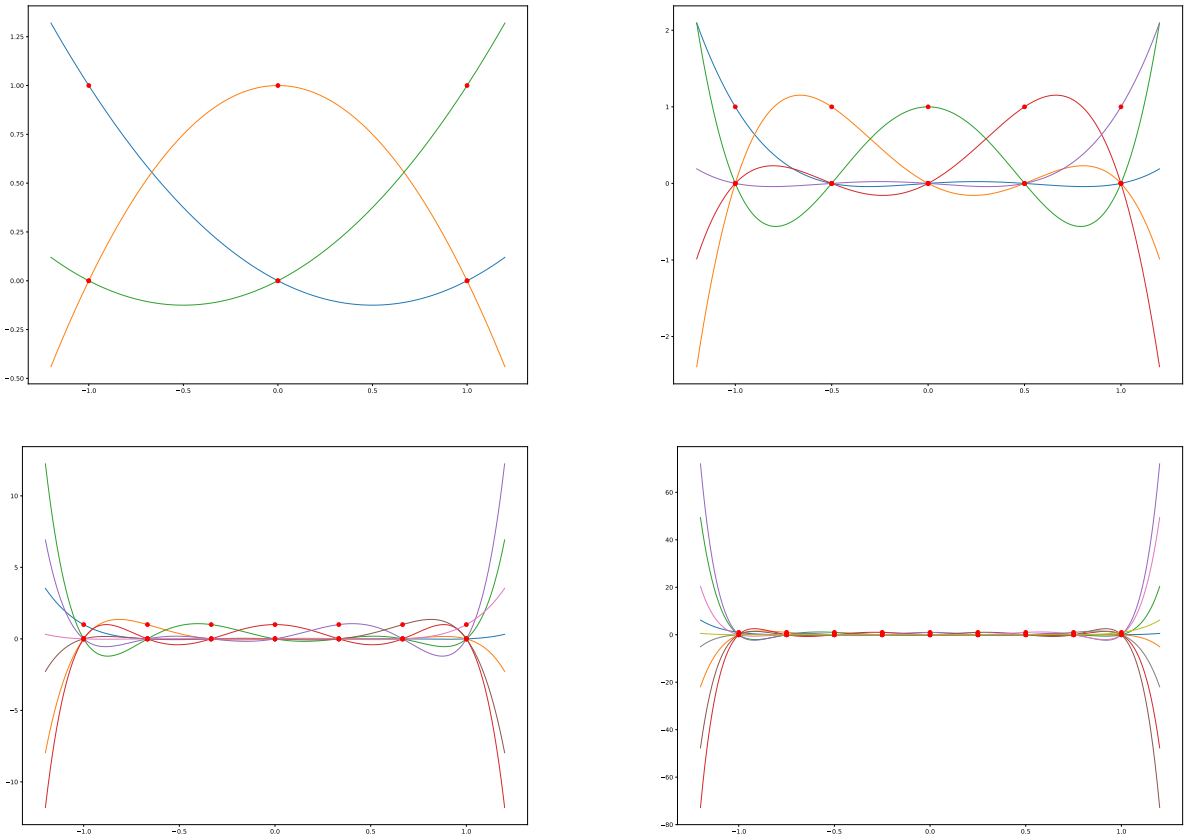


Abbildung 15.1: Lagrange-Polynome für verschiedene Anzahlen $n+1 \in \{3, 5, 7, 9\}$ von (äquidistant auf $[-1, 1]$ gewählten) Stützstellen $x_0, \ldots, x_n$.

belässt, ist von der Größenordnung n^2 FLOPs. (**Quizfrage:** Klar?) Multipliziert man stattdessen das Polynom aus und wertet es mit dem Horner-Schema aus (Algorithmus 7.8), so sinkt der Aufwand auf die Größenordnung n FLOPs. Es ergibt sich dann allerdings der Nachteil, dass sich beim Hinzufügen einer neuen Stützstelle alle Polynome $L_i^{(n)}$ ändern und das ausmultiplizierte Polynom neu berechnet werden muss.

Es gibt jedoch eine Möglichkeit, beide Probleme (Aufwand bei der Auswertung und bei der Aufdatierung) gleichzeitig zu adressieren. Dazu definieren wir das **Knotenpolynom**

$$\ell(x) := \prod_{j=0}^{n} (x - x_j) \tag{15.10}$$

und erhalten mit der Produktregel folgende Darstellung der Ableitung:

$$\ell'(x) = \sum_{k=0}^{n} \prod_{\substack{j=0 \\ j \neq k}}^{n} (x - x_j). \tag{15.11}$$

Insbesondere gilt

$$\ell'(x_i) = \sum_{k=0}^{n} \prod_{\substack{j=0 \\ j\neq k}}^{n} (x_i - x_j) = \prod_{\substack{j=0 \\ j\neq i}}^{n} (x_i - x_j), \tag{15.12}$$

denn alle Summanden bis auf denjenigen für $k = i$ sind gleich Null.

Wir erhalten also folgende alternative Darstellung des Lagrange-Polynoms $L_i^{(n)}$:

$$L_i^{(n)}(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} = \frac{\displaystyle\prod_{\substack{j=0 \\ j\neq i}}^{n} (x - x_j)}{\displaystyle\prod_{\substack{j=0 \\ j\neq i}}^{n} (x_i - x_j)} = \frac{\frac{\ell(x)}{x-x_i}}{\ell'(x_i)} = \frac{\ell(x)}{(x - x_i)\,\ell'(x_i)} \quad \text{wegen (15.12)} \tag{15.13}$$

für alle Punkte $x \neq x_i$ (wo $L_i^{(n)}$ aber bekanntermaßen gleich 1 ist).

Wir führen jetzt zur Abkürzung die **baryzentrischen Gewichte**

$$\omega_i := \frac{1}{\displaystyle\prod_{\substack{j=0 \\ j\neq i}}^{n} (x_i - x_j)} = \frac{1}{\ell'(x_i)}, \quad i = 0, \dots, n \tag{15.14}$$

ein. Gemäß (15.13) können wir dann

$$L_i^{(n)}(x) = \frac{\ell(x)\,\omega_i}{x - x_i} \tag{15.15}$$

schreiben, und da die Summe aller Lagrange-Polynome konstant gleich 1 ist (**Quizfrage:** Warum ist das so?), gilt weiter

$$L_i^{(n)}(x) = \frac{\ell(x)\,\omega_i}{x - x_i} = \frac{\ell(x)\,\frac{\omega_i}{x-x_i}}{\sum_{j=0}^{n} L_j^{(n)}(x)} = \frac{\frac{\omega_i}{x-x_i}}{\sum_{j=0}^{n} L_j^{(n)}(x)\,\frac{1}{\ell(x)}} = \frac{\frac{\omega_i}{x-x_i}}{\sum_{j=0}^{n} \frac{\omega_j}{x-x_j}}, \tag{15.16}$$

wobei die letzte Gleichheit aus (15.15) folgt. **Beachte:** Diese Darstellung ist verwendbar für alle Punkte x, die keine Stützstelle sind.

Damit können wir nun schließlich die Lösung der Interpolationsaufgabe (15.2) statt wie in (15.9) auch wie folgt darstellen:

$$p(x) = \frac{\displaystyle\sum_{i=0}^{n} y_i \frac{\omega_i}{x - x_i}}{\displaystyle\sum_{j=0}^{n} \frac{\omega_j}{x - x_j}}. \tag{15.17}$$

Diese Darstellung heißt die **baryzentrische Darstellung** des Lagrangeschen Interpolationspolynoms, und sie gilt für alle $x \notin \{x_0, \dots, x_n\}$. (Dort ist aber ja $p(x_i) = y_i$ ohnehin bekannt.)

Da die Gewichte ω_i, siehe (15.14), nur von den Stützstellen abhängen, können sie vorher berechnet werden. (Dafür benötigen wir Größenordnung n^2 FLOPs.) Die Auswertung von (15.17) an einem Punkt x erfordert dann nur noch Größenordnung n FLOPs. (**Quizfrage:** Wieviele genau?) Zudem können die Gewichte bei Hinzukommen einer weiteren Stützstelle (und übrigens auch beim Entfernen oder Verschieben) mit Größenordnung n FLOPs aufdatiert werden.

§ 15.3 Newton-Polynombasis

Die Lösung der Interpolationsaufgabe (15.2) kann auch sukzessive aus der Lösung mit weniger Punkten aufgebaut werden.

Definition 15.6 (Newtonsche Basispolynome)**.** *Zu gegebenen, paarweise verschiedenen Stützstellen $x_0, \dots, x_n \in \mathbb{R}$ heißt das Polynom*

$$N_i(x) := \prod_{j=0}^{i-1} (x - x_j) \tag{15.18}$$

*das i-te **Newtonsche Basispolynom** (englisch:* Newton basis polynomial*).*

Beachte: Insbesondere ist $N_0(x) \equiv 1$ (leeres Produkt).

Satz 15.7 (Eigenschaften der Newtonschen Basispolynome)**.** *Es seien $x_0, \dots, x_n$ gegeben. Dann gilt für die Newtonschen Basispolynome (15.18):*

(*i*)
$$N_i(x_j) = 0 \quad \text{für } 1 \leqslant j < i \leqslant n. \tag{15.19}$$

(*ii*) *Die $N_0, \dots, N_n$ bilden eine Basis von P_n.*

Beweis. □

Wir schreiben also nun das gesuchte Interpolationspolynom zur Aufgabe (15.2) in der Form

$$p(x) = \sum_{i=0}^{n} c_i \, N_i(x). \tag{15.20}$$

Setzen wir die Stützstellen $x_0, \dots, x_n$ nacheinander in die Darstellung (15.20) ein, so erhalten wir das lineare Gleichungssystem

$$\begin{bmatrix} N_0(x_0) & N_1(x_0) & \cdots & N_n(x_0) \\ N_0(x_1) & N_1(x_1) & \cdots & N_n(x_1) \\ \vdots & & & \vdots \\ N_0(x_n) & N_1(x_n) & \cdots & N_n(x_n) \end{bmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

Durch Ausnutzen der Eigenschaften von $N_i(x_j)$ können wir dies äquivalent schreiben als

$$\begin{bmatrix} 1 & & & & \\ 1 & x_1 - x_0 & & & \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & & \\ \vdots & \vdots & \vdots & \ddots & \\ 1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & \cdots & \prod_{j=0}^{n-1}(x_n - x_j) \end{bmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \tag{15.21}$$

wobei nicht bezeichnete Einträge in der Matrix für Nullen stehen. Dieses untere Dreieckssystem können wir sukzessive durch Vorwärtseinsetzen lösen. Die ersten Schritte dieses Vorgehens führen auf

$$\begin{aligned} c_0 &= y_0 \\ c_1 &= \frac{y_1 - c_0}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \\ c_2 &= \frac{y_2 - c_0 - (x_2 - x_0)\, c_1}{(x_2 - x_0)(x_2 - x_1)} = \frac{y_2 - y_0 - (x_2 - x_0)\, \frac{y_1 - y_0}{x_1 - x_0}}{(x_2 - x_0)(x_2 - x_1)} = \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0} \end{aligned}$$

usw. Das motiviert die folgende Definition.

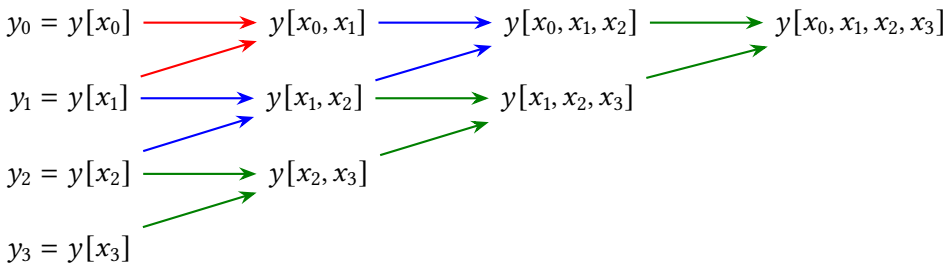
Definition 15.8 (Dividierte Differenzen). *Zu gegebenen paarweise verschiedenen Stützstellen $x_0, \dots, x_n$ und zugehörigen Stützwerten $y_0, \dots, y_n$ heißen die rekursiv definierten Größen*

$$y[x_i] := y_i, \quad i = 0, \dots, n \tag{15.22a}$$

$$y[x_i, \dots, x_{i+k}] := \frac{\overbrace{y[x_{i+1}, \dots, x_{i+k}]}^{\text{Index startet später}} - \overbrace{y[x_i, \dots, x_{i+k-1}]}^{\text{Index endet früher}}}{x_{i+k} - x_i}, \quad i \geqslant 0,\ i + k \leqslant n,\ k \geqslant 0 \tag{15.22b}$$

die ***(Newtonsche) dividierten Differenzen*** *(englisch:* (Newton) divided differences*).*

Man kann die dividierten Diffenzen in folgendem Schema anordnen:



Die gesuchten Koeffizienten stehen in der ersten Zeile. Man berechnet die Größen im Schema üblicherweise Diagonale für Diagonale, die jeweils in einer anderen Farbe kodiert sind.[1] Jede Diagonale entspricht der Hinzunahme eines weiteren Stützpunkts, hier: x_1, x_2 und x_3.

[1] Eine Berechnung Spalte für Spalte ist ebenfalls möglich.

Satz 15.9 (Lösung der Interpolationsaufgabe mit Hilfe dividierter Differenzen)**.** *Die eindeutige Lösung der Interpolationsaufgabe* (15.2) *in Newtonscher Darstellung* (15.20) *ist gegeben durch*

$$p(x) = \sum_{i=0}^{n} y[x_0, \dots, x_i]\, N_i(x). \tag{15.23}$$

Beweis. Wir können beispielsweise durch Induktion zeigen, dass die gesuchten Koeffizienten c_i des Interpolationspolynoms in der Newtonschen Darstellung (15.20) die Beziehung $c_i = y[x_0, \dots, x_i]$ erfüllen. Für einen vollständigen Beweis verweisen wir auf Rannacher, 2017, Satz 2.2. □

Die Auswertung eines Polynoms in der Form (15.20), also

$$c_0 + c_1\,(x - x_0) + c_2\,(x - x_0)(x - x_1) + c_3\,(x - x_0)(x - x_1)(x - x_2) + \cdots, \tag{15.24}$$

erfolgt ähnlich zum Horner-Schema:

Algorithmus 15.10 (Horner-Schema für (15.24))**.**
Eingabe: *Koeffizienten $c_0, \dots, c_n$ eines Polynoms der Form* (15.24)
Eingabe: *Auswertungsstelle x*
Ausgabe: *Funktionswert $y = p(x)$*
1: *Setze $y := c_n$*
2: ***for*** $k = n-1, n-2, \dots, 0$ ***do***
3: *Setze $y := (x - x_k)\, y + c_k$*
4: ***end for***

Die Berechnung der dividierten Differenzen erfordert $\sim n^2$ FLOPs. Das Hinzufügen einer neuen Stützstelle x_{n+1} zu $x_0, \dots, x_n$ erfordert die Berechnung einer neuen Diagonalzeile mit dem Aufwand $\sim n$ FLOPs. Die Auswertung mit Algorithmus 15.10 geht wiederum in $\sim n$ FLOPs.

§ 15.4 Approximationsfehler und Konvergenz bei der Interpolation

Frage: Um wieviel kann das Interpolationspolynom von der Funktion, die es interpoliert, abweichen?

Beachte: Das ist eine neue Art von Fehler, die hier auftritt. Wir können ihn als **Approximationsfehler** oder **Verfahrensfehler** bezeichnen. Er entsteht dadurch, dass wir eine gegebene Funktion f durch eine andere Funktion (hier ein Polynom) approximieren.

Der folgende Satz liefert eine Abschätzung des Approximationsfehlers. Auf welche Art und Weise (Abschnitte 15.1 bis 15.3) das Interpolationspolynom dabei gewonnen wird, ist völlig unerheblich.

In Erweiterung von $\langle a, b\rangle$, siehe (2.9), definieren wir $\langle a_0, a_1, \dots, a_n\rangle$ als das kleinste Intervall, das alle genannten Zahlen $a_0, a_1, \dots, a_n \in \mathbb{R}$ enthält.[2]

[2] Man könnte auch sagen: $\langle a_0, a_1, \dots, a_n\rangle$ ist die konvexe Hülle der Zahlen $a_0, a_1, \dots, a_n$.

Satz 15.11 (Approximationsfehler vgl. Rannacher, 2017, Satz 2.3). *Es sei $[a,b]$ ein Intervall und $x_0, x_1, \dots, x_n$ paarweise verschiedene Stützstellen in $[a,b]$. Weiter sei*[3] *$f \in C^{n+1}([a,b])$ und p das nach Folgerung 15.4 eindeutig definierte Interpolationspolynom zu den Stützstellen $x_0, \dots, x_n$. Dann existiert für jedes $x \in [a,b]$ ein $\xi \in \langle x_0, x_1, \dots, x_n, x\rangle \subseteq [a,b]$, sodass gilt:*

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\,\ell(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x - x_i) \tag{15.25}$$

und folglich

$$\|f - p\|_{C([a,b])} := \sup_{x\in[a,b]} |f(x) - p(x)| \leqslant \sup_{x\in[a,b]} \frac{|f^{(n+1)}(x)|}{(n+1)!} \sup_{x\in[a,b]} |\ell(x)|. \tag{15.26}$$

Hierbei ist $\ell(x) = \prod_{i=0}^{n}(x - x_i)$ wieder das Knotenpolynom aus (15.10). Die Norm $\|\cdot\|_{C([a,b])}$ heißt die **Supremumsnorm** oder **Norm der gleichmäßigen Konvergenz** auf $[a,b]$ (englisch: *supremum norm, norm of uniform convergence*). Statt dem Supremum dürften wir in (15.26) auch überall Maximum schreiben. (**Quizfrage:** Warum?)

Beweis. Für $x = x_i$ ist nichts zu zeigen. (**Quizfrage:** Klar?) Es sei also $x \in [a,b] \setminus \{x_0, \dots, x_n\}$ beliebig, aber fest gewählt. Wir setzen

$$c := \frac{f(x) - p(x)}{\ell(x)}.$$

(Der Nenner ist ungleich Null.) Dann besitzt die Funktion $F(t) := f(t) - p(t) - c\,\ell(t)$ mindestens die $n+2$ Nullstellen $x_0, \dots, x_n$ und das gewählte x im Intervall $\langle x_0, x_1, \dots, x_n, x\rangle$. Nach dem Satz von Rolle[4] besitzt also F' in $\langle x_0, x_1, \dots, x_n, x\rangle$ mindestens $n+1$ Nullstellen. Man argumentiert so weiter, bis man die Aussage erhält, dass die $(n+1)$-te Ableitung $F^{(n+1)}$ mindestens noch eine Nullstelle $\xi \in \langle x_0, x_1, \dots, x_n, x\rangle$ besitzt. Es gilt also

$$\begin{aligned} 0 = F^{(n+1)}(\xi) &= f^{(n+1)}(\xi) - p^{(n+1)}(\xi) - c\,\ell^{(n+1)}(\xi) \\ &= f^{(n+1)}(\xi) - 0 - c\,(n+1)!, \end{aligned}$$

d. h. $c = \frac{f^{(n+1)}(\xi)}{(n+1)!}$. (**Quizfrage:** Warum ist $p^{(n+1)}(\xi) = 0$?) (**Quizfrage:** Und warum ist $\ell^{(n+1)}(\xi) = (n+1)!$?) Wir erhalten also

$$f(x) - p(x) = c\,\ell(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\,\ell(x),$$

was zu zeigen war. □

Beachte: Der Auswertungspunkt x kann durchaus außerhalb des Intervalls $\langle x_0, x_1, \dots, x_n\rangle$ der Stützstellen liegen. Wenn man das Interpolationspolynom $p(x)$ an einem solchen Punkt auswertet, spricht man auch von **Extrapolation** (englisch: *extrapolation*).

[3] Wir hatten die Räume C^k in Definition 2.7 nur für offene Mengen definiert. Mit $C^{n+1}([a,b])$ ist der Vektoraum der Funktionen gemeint, die in $C^{n+1}((a,b))$ liegen und deren Ableitungen der Ordnungen $0, 1, \dots, n+1$ sich stetig auf $[a,b]$ fortsetzen lassen.

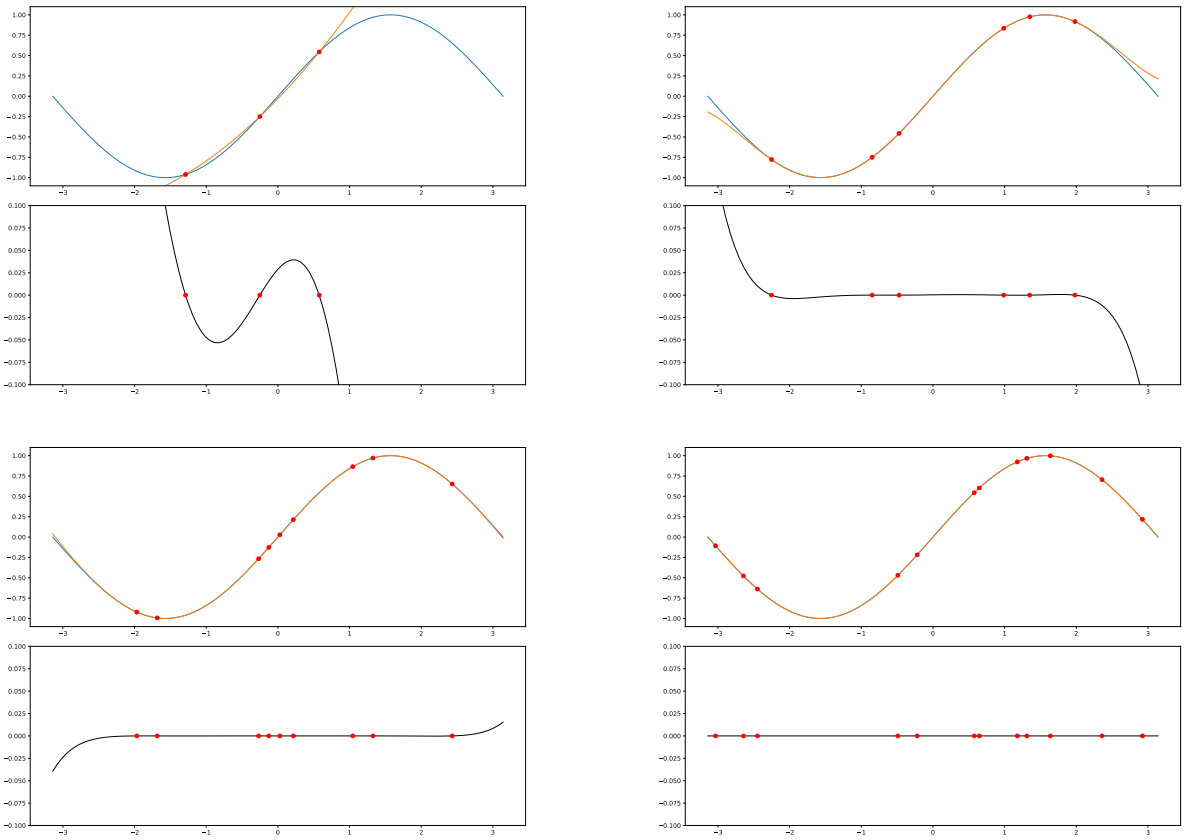
[4] Zwischen zwei Nullstellen einer differenzierbaren Funktion liegt (mindestens) eine Nullstelle ihrer Ableitung.

Beispiel 15.12 (Fehlerabschätzung). *Wir betrachten die Funktion $f(x) = \sin(x)$ auf irgendeinem Intervall $[a, b]$ mit irgendwelchen(!) Stützstellen $x_0, x_1, \ldots, x_n \in [a, b]$. Dann folgt aus (15.26) die Fehlerabschätzung*

$$\begin{aligned}\|f - p\|_{C([a,b])} = \sup_{x\in[a,b]} |f(x) - p(x)| &\leqslant \sup_{x\in[a,b]} \frac{|f^{(n+1)}(x)|}{(n+1)!} \sup_{x\in[a,b]} |\ell(x)| \\ &\leqslant \frac{1}{(n+1)!}(b-a)^n.\end{aligned}$$

*(**Quizfrage:** Wieso können wir $|f^{(n+1)}(x)| \leqslant 1$ abschätzen?) Für $n \to \infty$ konvergiert die rechte Seite gegen Null. Mit anderen Worten konvergiert jede Folge von Interpolationspolynomen mit beliebigen Stützstellen auf jedem festen Intervall $[a, b]$ gleichmäßig gegen die Funktion $f(x) = \sin(x)$.*

Die folgenden Abbildungen illustrieren den Sachverhalt aus Beispiel 15.12 auf dem Intervall $[a, b] = [-\pi, \pi]$ mit zufälligen Interpolationspunkten wachsender Anzahl $n + 1 \in \{3, 6, 9, 12\}$. Gezeigt ist jeweils die Funktion $f(x) = \sin(x)$ (blau), das Interpolationspolynom (orange) und darunter der Fehler $f(x) - p(x)$ (schwarz).



Die Situation aus Beispiel 15.12 ist deshalb bemerkenswert, weil die Ableitungen $f^{(n+1)}$ der Sinus-Funktion alle auf dem betrachteten Intervall beschränkt sind. Für viele andere Funktionen wachsen die Ableitungen für $n \to \infty$ dagegen stark an. In diesem Fall ist die gleichmäßige Konvergenz der

Interpolationspolynome auf kompakten Intervallen nicht garantiert, sondern hängt davon ab, wie die Stützstellen gewählt werden (siehe Satz 15.14).

Ein bekanntes Beispiel für die Nicht-Konvergenz ist das folgende.

Beispiel 15.13 (Runge). *Wir definieren die Funktion*

$$f(x) := \frac{1}{1+25x^2}$$

und wählen die äquidistanten Stützstellen

$$x_i := \frac{2i}{n} - 1, \quad i = 0, \dots, n$$

auf $[-1,1]$. *Dann gilt für die Folge* $(p^{(n)})$ *der durch diese Stützstellen festgelegten Interpolationspolynome*

$$\lim_{n\to\infty} \|f - p^{(n)}\|_{C([-1,1])} = \infty.$$

Dabei sind die Werte der Fehlerfunktion $f - p^{(n)}$ *an den Rändern des Intervalls* $[-1,1]$ *maßgeblich, siehe Abbildung 15.2. Diese Beobachtung ist als* ***Runge-Phänomen*** *bekannt.*

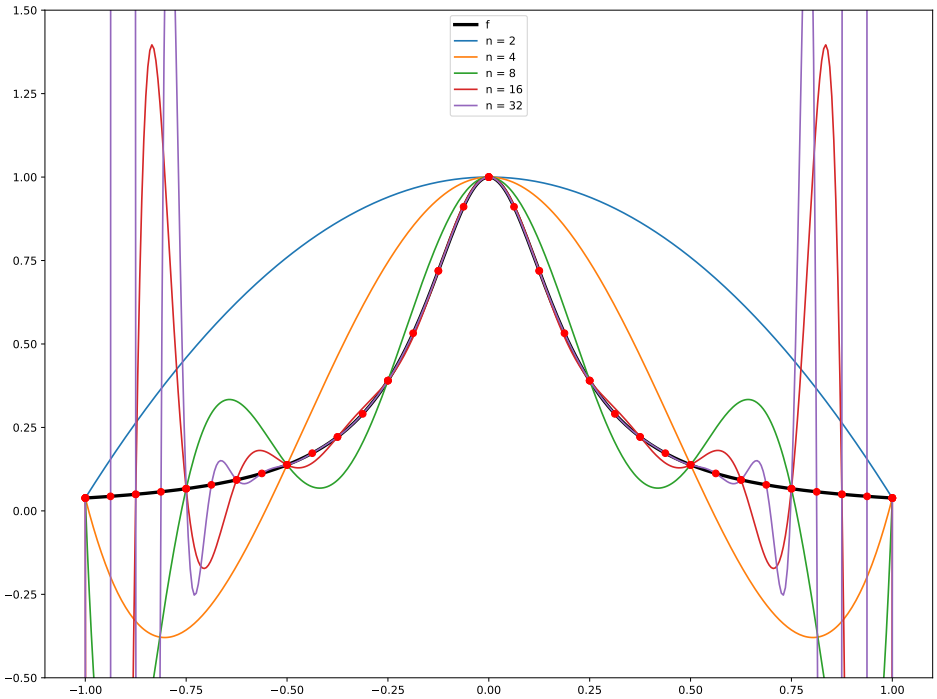


Abbildung 15.2: Illustration zu Beispiel 15.13. Funktion $f(x) = \frac{1}{1+25x^2}$ (schwarz) und ihre Interpolation durch Polynome mit verschiedenen Anzahlen äquidistanter Stützstellen auf $[-1,1]$.

Das Beispiel von Runge 15.13 ist kein Widerspruch zum berühmten Approximationssatz von Weierstraß. Dieser besagt, dass jede stetige Funktion auf einem kompakten Intervall $[a,b]$ dort gleichmäßig durch eine Folge von Polynomen approximiert werden kann. Diese Polynome können aber nicht als *beliebige* Folge von Interpolationspolynome gewählt werden. Stattdessen gilt:

Satz 15.14 (Gleichmäßige Konvergenz der Interpolationspolynome bei geeigneter Wahl der Stützstellen)**.** *Es sei $f \in C^0([a,b])$. Dann existiert eine Folge $\left(T^{(n)}\right)$ von Stützstellen — $T^{(n)}$ ist eine Teilmenge von $[a,b]$ der Mächtigkeit $n+1$ —, sodass die Folge der zugehörigen Interpolationspolynome gleichmäßig auf $[a,b]$ gegen f konvergiert.*

Beweis. Wir skizzieren hier nur den Beweis. Wir betrachten die Aufgabe der Bestapproximation von f auf $[a,b]$ durch ein Polynom gegebenen Höchstgrades $n \in \mathbb{N}_0$:

$$\text{Minimiere} \quad \|f-p\|_{C([a,b])} \coloneqq \sup_{x\in[a,b]} |f(x)-p(x)|, \quad p \in P_n. \tag{15.27}$$

Man kann zeigen, dass diese Aufgabe für jedes $n \in \mathbb{N}_0$ eine Lösung besitzt, die sogar eindeutig ist. Zudem gilt für das optimale Polynom $p^{(n)}$ der sogenannte **Alternantensatz** 17.2 (englisch: *equioscillation theorem*), der besagt, dass es (mindestens) $n+2$ Extremstellen $a \leqslant \xi_0 < \xi_1 < \cdots < \xi_{n+1} \leqslant b$ von $f-p^{(n)}$ gibt mit der Eigenschaft

$$f(\xi_i) - p^{(n)}(\xi_i) = s\,(-1)^i\,\|f-p^{(n)}\|_{C([a,b])}$$

mit $s=1$ oder $s=-1$. Das heißt, dass das für (15.27) optimale Polynom $p^{(n)}$ die Eigenschaft hat, dass die Fehlerfunktion $f-p^{(n)}$ abwechselnd und insgesamt mindestens $n+2$ Mal an der oberen bzw. unteren Grenze anschlägt. Nach dem Zwischenwertsatz für stetige Funktionen gibt es also zwischen zwei benachbarten ξ_i immer eine Zahl $\xi_i < x_i < \xi_{i+1}$, wo $f(x_i) - p^{(n)}(x_i) = 0$ ist, $i = 0,1,\ldots,n+1$. Diese x_i sind gerade die gesuchten Interpolationspunkte, die die Menge $T^{(n)}$ bilden.

Dass die Folge $\|f-p^{(n)}\|_{C([a,b])}$ gegen Null konvergiert — mit anderen Worten, dass die Folge $(p^{(n)})$ gleichmäßig gegen f konvergiert, folgt aus dem Approximationssatz von Weierstraß. Dieser besagt, dass es eine Folge $(g^{(n)})$ gibt mit $g^{(n)} \in P_n$, sodass $\|f-g^{(n)}\|_{C([a,b])} \to 0$ konvergiert. Wegen der Bestapproximationseigenschaft der Lösung $p^{(n)}$ von (15.27) gilt ja sogar

$$\|f-p^{(n)}\|_{C([a,b])} \leqslant \|f-g^{(n)}\|_{C([a,b])} \to 0$$

für $n \to \infty$. □

Der Nachteil an diesem Resultat ist natürlich, dass die «geeignete Folge» von Stützstellen von der Funktion f abhängt und man sie i. A. nicht kennt.

§ 15.5 Kondition der Interpolationsaufgabe

Wir betrachten noch die Frage der Kondition der Interpolationsaufgabe (15.2), genauer: das Änderungsverhalten des Interpolationspolynoms bei festen (paarweise verschiedenen) Stützstellen $x_0,\ldots,x_n$, aber veränderlichen Stützwerten $y_0,\ldots,y_n$, die wir zum Vektor $y \in \mathbb{R}^{n+1}$ zusammenfassen.

Wir betrachten dazu einmal das eindeutige Interpolationspolynom $p(\cdot\,; y)$ zu den nominalen Stützwerten y und einmal das Interpolationspolynom $p(\cdot\,; y+\Delta y)$ zu gestörten Stützwerten $y+\Delta y$. Dann gilt

wegen der linearen Abhängigkeit $y \mapsto p(\cdot; y)$ nach (15.9):

$$\begin{aligned} p(x; y + \Delta y) - p(x; y) &= \sum_{i=0}^{n} (y_i + \Delta y_i)\, L_i^{(n)}(x) - \sum_{i=0}^{n} y_i\, L_i^{(n)}(x) \\ &= \sum_{i=0}^{n} \Delta y_i\, L_i^{(n)}(x) \end{aligned}$$

bzw. in relativer Betrachtungsweise

$$\frac{p(x; y + \Delta y) - p(x; y)}{p(x; y)} = \sum_{i=0}^{n} \frac{y_i\, L_i^{(n)}(x)}{p(x; y)}\, \frac{\Delta y_i}{y_i}.$$

Die punktweise relative partielle Konditionszahl an der Stelle $x \in \mathbb{R}$

$$\frac{y_i\, L_i^{(n)}(x)}{p(x; y)} = \frac{y_i\, L_i^{(n)}(x)}{\sum_{j=0}^{n} y_j\, L_j^{(n)}(x)}$$

wird maßgeblich beeinflusst von dem Wert des i-ten Lagrange-Polynoms $L_i^{(n)}(x)$, der vor allem außerhalb des Intervalls $\langle x_0, x_1, \ldots, x_n \rangle$ der Stützstellen bereits für moderate Werte von n sehr groß sein kann, siehe Abbildung 15.1.

Dazu noch ein Beispiel.

Beispiel 15.15 (vgl. Rannacher, 2017, Beispiel 2.2)**.** *Wir betrachten das Interpolationspolynom zu den $n + 1 = 2m + 1$ äquidistanten Stützstellen auf $[-1, 1]$ und Stützwerten $y_i = 0$ mit Ausnahme des mittleren Wertes von $y_m = \varepsilon$ (bei $x_m = 0$). Das Interpolationspolynom hat die Darstellung*

$$p(x) = \varepsilon\, L_m^{(n)}(x) = \varepsilon \prod_{\substack{j=0 \\ j \neq m}}^{n} \frac{x - x_j}{0 - x_j}.$$

Einige dieser Polynome für verschiedene Werte von n (und mit $\varepsilon = 1$) werden in Abbildung 15.3 gezeigt.

Bemerkung 15.16 (Polynominterpolation)**.** *Die Interpolation mit Polynomen hat, wie wir bisher gesehen haben, gewisse Nachteile:*

*(1) Die Auswirkung der Änderung von Stützwerten ist nicht lokal. Ändern wir also auch nur einen der Stützwerte y_i, so ändert sich das Interpolationspolynom an allen Punkten außer an den Stützstellen $x_0, \ldots, x_n$. (**Quizfrage:** Wie sieht man das?)*

(2) Die Interpolationsaufgabe ist bereits für moderate Polynomgrade schlecht konditioniert.

Das ist der Grund, warum man bei einer großen Anzahl von Stützstellen bevorzugt mit stückweisen Polynomen interpoliert, die man glatt aneinanderstückelt. Solche Funktionen heißen **Splines**. Eine klassische Wahl sind **kubische Splines**, die man im Funktionenraum

$$\left\{ s \in C^2([a, b]) \,\middle|\, s|_{[x_i, x_{i+1}]} \in P_3,\ i = 0, 1, \ldots, n - 1 \right\}$$

sucht.

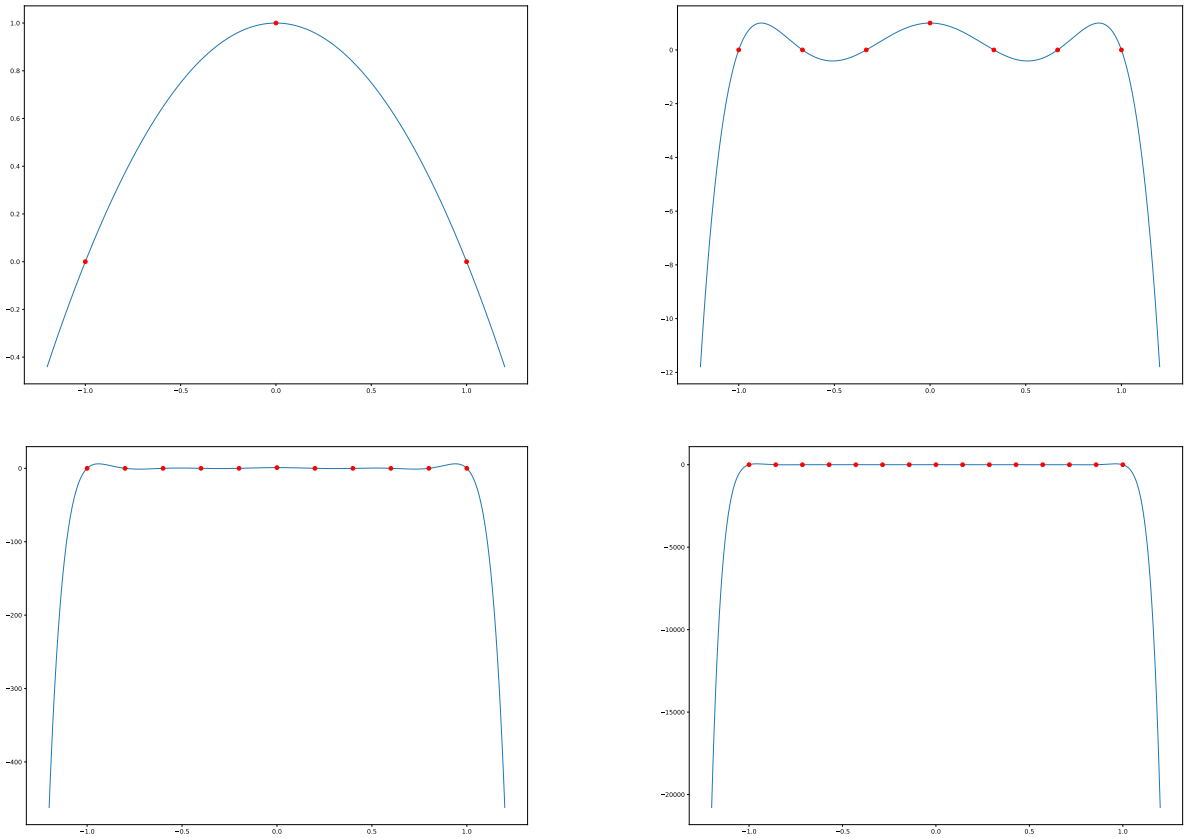


Abbildung 15.3: Illustration zu Beispiel 15.15 für verschiedene Anzahlen $n+1 \in \{3, 7, 11, 15\}$ von äquidistant auf $[-1, 1]$ gewählten Stützstellen $x_0, \ldots, x_n$.

Bemerkung 15.17 (Interpolationsaufgaben mit Polynomen)**.** *Wir haben uns hier nur mit Interpolationsaufgaben beschäftigt, bei denen die Funktionswerte $y_0, \ldots, y_n$ vorgegeben waren. Genauer spricht man dabei von* ***Lagrange-Interpolationsaufgaben*** *(englisch:* Lagrange interpolation*).*[5] *Allgemeiner kann man auch Aufgaben betrachten, bei denen man zusätzlich an einigen oder allen Stützstellen Werte der ersten und ggf. höherer Ableitungen des gesuchten Interpolationspolynoms vorgibt. Solche Aufgaben heißen* ***Hermite-Interpolationsaufgaben*** *(englisch:* Hermite interpolation*).*

§ 15.6 Anwendungen der Interpolation

Eine wichtige Anwendung der Polynominterpolation liegt in der Entwicklung von Formeln für die **numerische Differentiation** (englisch: *numerical differentiation*) durch **Finite-Differenzen-Approximation** (englisch: *finite difference approximation*) der Ableitung. Dabei geht es darum, die erste oder auch höhere Ableitungen einer Funktion f an einer Stelle x_0 auszuwerten, wobei aber nur Funktionswerte von f verwendet werden sollen. Diese sehr häufig verwendete Technik ist zum

[5] Dabei ist nicht gemeint, dass man notwendigerweise die Darstellung (15.9) des Interpolationspolynoms unter Verwendung der Lagrange-Polynome benutzt, sondern eben nur, dass man nur Funktionswerte interpoliert, keine Ableitungen.

Beispiel dann hilfreich, wenn Alternativen wie Ableiten durch symbolisches oder algorithmisches Differenzieren nicht möglich ist.

Häufig verwendet man dabei äquidistante Stützstellen, die wir hier mit $x_i = x_0 + i\,h$ bezeichnen. Dabei kann der Index i auch negative Werte annehmen, um Funktionsauswertungen «links» von x_0 mit einzubeziehen. Die Idee ist, die Funktion f durch ihr Interpolationspolynom $p \in P_n$ zu ersetzen und dann dessen Ableitung p' (stellvertretend für die Ableitung von f) am Punkt x_0 auszuwerten. Allgemeiner können wir statt des ersten Ableitung auch Ableitungen der Ordnung m verwenden, sofern $1 \leqslant m \leqslant n$ gilt. (**Quizfrage:** Warum darf m nicht größer als n sein?)

Wir geben die wichtigsten Beispiele an:

Stützstellen	n	m	Formel	Fehler	Vor.	Name
x_0, x_1	1	1	$f'(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h}$	$O(h)$	$f \in C^2$	Vorwärtsdifferenz
x_{-1}, x_0	1	1	$f'(x_0) \approx \frac{f(x_0)-f(x_0-h)}{h}$	$O(h)$	$f \in C^2$	Rückwärtsdifferenz
x_{-1}, x_0, x_1	2	1	$f'(x_0) \approx \frac{f(x_0+h)-f(x_0-h)}{2h}$	$O(h^2)$	$f \in C^3$	zentrale Diff.
x_{-1}, x_0, x_1	2	2	$f''(x_0) \approx \frac{f(x_0+h)-2f(x_0)+f(x_0-h)}{h^2}$	$O(h^2)$	$f \in C^4$	zentrale Diff. für 2. Abl.

Die Angabe des Fehlers ist beispielsweise im Falle der Vorwärtsdifferenz so zu verstehen, dass

$$\frac{f(x_0+h)-f(x_0)}{h} \in f'(x_0) + O(h)$$

gilt, also es existiert eine Konstante $C \geqslant 0$, sodass

$$\left|\frac{f(x_0+h)-f(x_0)}{h} - f'(x_0)\right| \leqslant C\,h$$

für alle $0 < h < \varepsilon$ gilt. Das kann man beispielsweise mit dem Satz von Taylor 2.10 zeigen unter der Voraussetzung, dass f eine C^2-Funktion auf einer offenen Obermenge von $\langle x_0\,, x_1\rangle = [x_0, x_0+h]$ ist. **Quizfrage:** Können Sie eine Vermutung darüber anstellen, wie der Zusammenhang zwischen der benötigten Glattheitsordnung C^k der Funktion f, der Fehlerordnung und der Ordnung m der Ableitung ist?

Ende der Woche 9

§ 16 Bernstein-Polynome und Bezier-Kurven

Bernstein-Polynome werden nicht zur Interpolation eingesetzt, sondern vorrangig zur Approximation und zur Beschreibung von Kurven und Flächen in der Computergrafik und Konstruktion (CAD). Mit ihnen kann man auch den Approximationssatz von Weierstraß konstruktiv beweisen. Bernstein-Polynome sind eine Familie reeller Polynome mit ganzzahligen Koeffizienten bzgl. der Monombasis.

Definition 16.1 (Bernstein-Polynome). *Es sei $n \in \mathbb{N}_0$. Die Polynome*[6]

$$B_i^{(n)}(t) := \binom{n}{i}(1-t)^{n-i}\,t^i, \quad i = 0, \dots, n \tag{16.1}$$

*heißen die **Bernstein-Polynome** (englisch:* Bernstein polynomials*) vom Grad n auf dem Interval $[0,1]$.*

Die Bernstein-Polynome für verschiedene Grade sind in Abbildung 16.1 gezeigt. (**Quizfrage:** Ist klar, dass die Bernstein-Polynome ganzzahlige Koeffizienten bzgl. der Monombasis haben?)

Man kann die Bernstein-Polynome auf beliebige Intervalle $[a,b]$ transformieren. Sie haben dann die Gestalt

$$B_i^{(n)}(t) := \frac{1}{(b-a)^n}\binom{n}{i}(b-t)^{n-i}\,(t-a)^i$$

und heißen auch **verallgemeinerte Bernstein-Polynome** (englisch: *generalized Bernstein polynomials*). Da sich die Eigenschaften aber nicht wesentlich verändern, betrachten wir nur die Polynome (16.1).

Satz 16.2 (Eigenschaften der Bernstein-Polynome). *Die Bernstein-Polynome (16.1) haben folgende Eigenschaften:*

(*i*) *Alle Bernstein-Polynome $B_i^{(n)}$, $i = 0, \dots, n$, sind Polynome vom Grad n.*

(*ii*) *Die Bernstein-Polynome haben auf $[0,1]$ Werte in $[0,1]$ und summieren sich zu eins:*

$$0 \leqslant B_i^{(n)}(t) \leqslant 1 \quad \text{für alle } t \in [0,1] \quad \text{und} \quad \sum_{i=0}^{n} B_i^{(n)}(t) \equiv 1 \quad \text{für alle } t \in \mathbb{R}. \tag{16.2}$$

*Man spricht auch davon, dass die Funktionen $B_0^{(n)}, \dots, B_n^{(n)}$ eine **Zerlegung der Eins(funktion)** (englisch:* partition of unity*) bilden.*

(*iii*) *Es gilt sogar*

$$0 < B_i^{(n)}(t) < 1 \quad \text{für alle } t \in (0,1). \tag{16.3}$$

(*iv*) *Die Bernstein-Polynom erfüllen folgende Symmetriebedingung:*

$$B_i^{(n)}(t) = B_{n-i}^{(n)}(1-t). \tag{16.4}$$

(*v*) *$B_i^{(n)}(t)$ hat $t = 0$ als i-fache Nullstellen und $t = 1$ als $(n-i)$-fache Nullstelle.*

(*vi*) *Für $n \geqslant 1$ hat $B_i^{(n)}(t)$ hat in $[0,1]$ genau eine Maximalstelle, und zwar bei i/n.*

(*vii*) *Die Polynome $\{B_0^{(n)}, \dots, B_n^{(n)}\}$ bilden eine Basis von P_n.*

[6] Dabei ist $\binom{n}{i} = \frac{n!}{i!\,(n-i)!}$ der Binomialkoeffizient («n über i»).

(*viii*) *Die Bernstein-Polynome haben eine rekursive Darstellung:*

$$B_i^{(n)}(t) = \begin{cases} (1-t)\,B_0^{(n-1)}(t) & \text{im Fall } i = 0, \\ t\,B_{i-1}^{(n-1)}(t) + (1-t)\,B_i^{(n-1)}(t) & \text{im Fall } 0 < i < n, \\ t\,B_{n-1}^{(n-1)}(t) & \text{im Fall } i = n. \end{cases} \tag{16.5}$$

(*ix*) *Die erste Ableitung der Bernstein-Polynome hat ebenfalls eine rekursive Darstellung:*

$$\frac{\mathrm{d}}{\mathrm{d}t} B_i^{(n)}(t) = \begin{cases} -n\,B_0^{(n-1)}(t) & \text{im Fall } i = 0, \\ n\,B_{i-1}^{(n-1)}(t) - n\,B_i^{(n-1)}(t) & \text{im Fall } 0 < i < n, \\ n\,B_{n-1}^{(n-1)}(t) & \text{im Fall } i = n. \end{cases} \tag{16.6}$$

Beweis. □

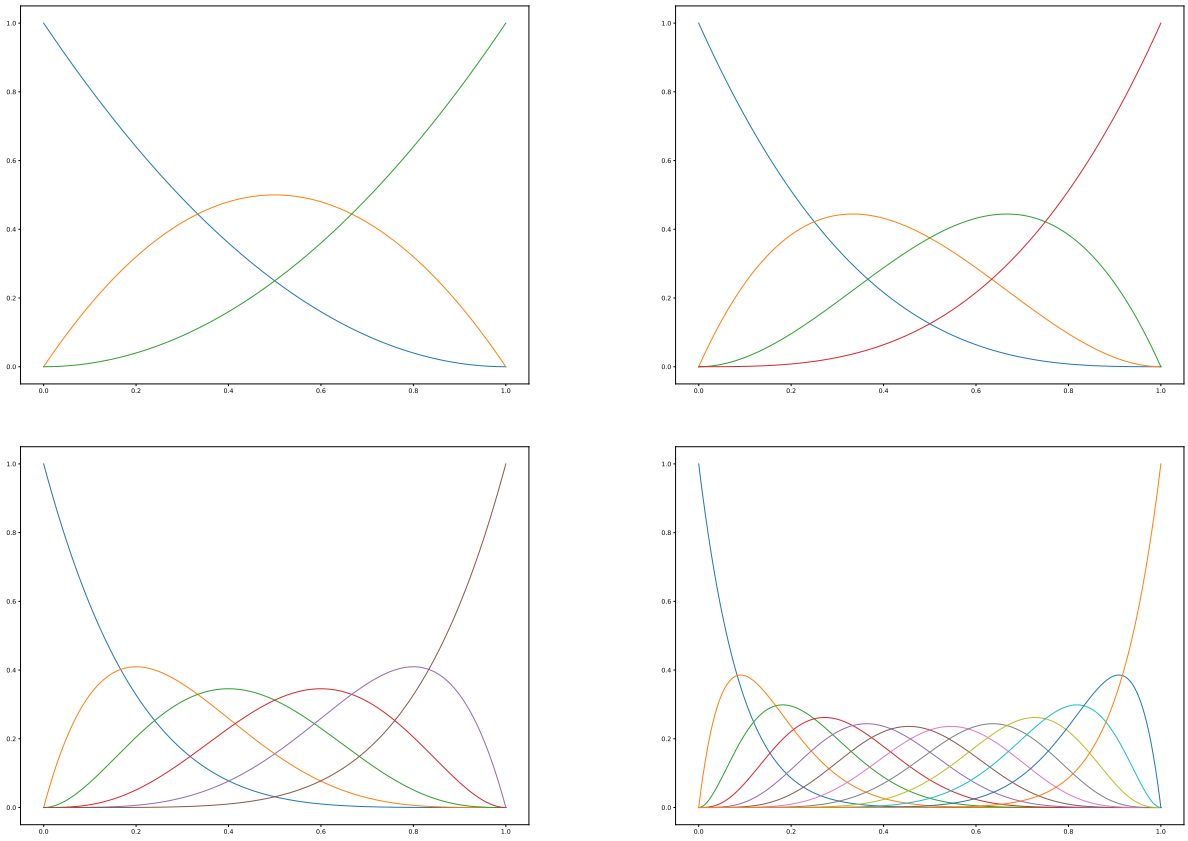


Abbildung 16.1: Illustration zu Bernstein-Polynome für verschiedene Grade $n \in \{2, 3, 5, 11\}$ auf $[0, 1]$.

Wir sehen jetzt, wie Bernstein-Polynome verwendet werden, um glatte Kurven in der Ebene oder im Raum zu definieren.

Definition 16.3 (Bézier-Kurve). *Für gegebene Punkte $b_0, \dots, b_n \in \mathbb{R}^d$ heißt das vektorwertige Polynom*

$$B(t) := \sum_{i=0}^{n} B_i^{(n)}(t)\, b_i, \quad t \in [0,1] \tag{16.7}$$

*die zugehörige **Bézier-Kurve** (englisch: Bézier curve). Die Punkte $b_0, \dots, b_n$ heißen die **Kontrollpunkte** (englisch: control points) der Bézier-Kurve.*

Quizfrage: Kommt es auf die Reihenfolge der Kontrollpunkte an?

Beispiel 16.4 (Bézier-Kurve). *Gegeben sind die Kontrollpunkte*

$$\begin{bmatrix} b_0 & b_1 & b_2 & b_3 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{3} & \frac{2}{3} & 1 \\ 0 & -1 & -1 & 1 \end{bmatrix}$$

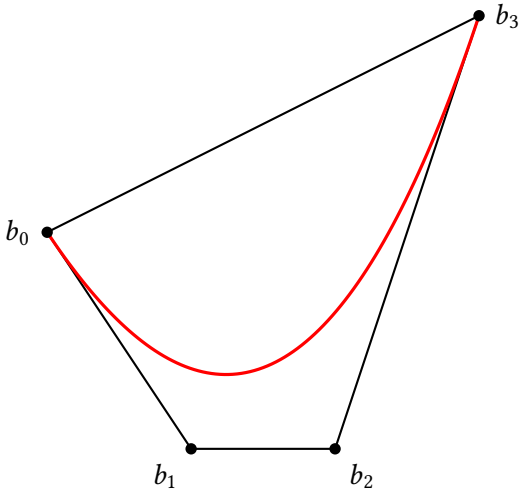
in $\mathbb{R}^2$. Die Bernstein-Polynome vom Grad 3 sind

$$\begin{aligned} B_0^{(3)}(t) &= 1 - 3t + 3t^2 - t^3, \\ B_1^{(3)}(t) &= 0 + 3t - 6t^2 + 3t^3, \\ B_2^{(3)}(t) &= 0 + 0t + 3t^2 - 3t^3, \\ B_3^{(3)}(t) &= 0 + 0t + 0t^2 + t^3. \end{aligned}$$

Die zu den Kontrollpunkten $b_0, \dots, b_3$ gehörige Bézier-Kurve ist daher gegeben durch

$$\begin{aligned} B(t) &= (1 - 3t + 3t^2 - t^3) \begin{pmatrix} 0 \\ 0 \end{pmatrix} + (3t - 6t^2 + 3t^3) \begin{pmatrix} \frac{1}{3} \\ -1 \end{pmatrix} + (3t^2 - 3t^3) \begin{pmatrix} \frac{2}{3} \\ -1 \end{pmatrix} + (t^3) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} t \\ t^3 + 3t^2 - 3t \end{pmatrix} \end{aligned}$$

für $t \in [0,1]$. Das folgende Bild zeigt diese Kurve (rot), die Kontrollpunkte sowie ihre konvexe Hülle.[7]



[7] Die konvexe Hülle einer Menge von Punkten in $\mathbb{R}^d$ ist die kleinste konvexe Menge, die alle diese Punkte enthält.

Das Beispiel illustriert folgende Sachverhalte, die auch allgemein gelten:

Satz 16.5 (Eigenschaften von Bézier-Kurven). *Es seien $b_0, \dots, b_n \in \mathbb{R}^d$ gegebene Kontrollpunkte, $n \in \mathbb{N}_0$, und B die dadurch erzeugte Bézier-Kurve (16.7). Dann gilt:*

(*i*) $B(0) = b_0$ *und* $B(1) = b_n$.

(*ii*) *Im Falle von* $n \geqslant 1$ *gilt für die Ableitung* $B'(0) = b_1 - b_0$ *und* $B'(1) = b_n - b_{n-1}$.

(*iii*) *Die Kurve $B(t)$ verläuft im Intervall $[0,1]$ innerhalb der konvexen Hülle von $b_0, \dots, b_n$. (Man bezeichnet diese konvexe Hülle der Kontrollpunkte als* ***Kontrollpolygon****.)*

Quizfrage: Wie sieht man die Aussagen (*ii*) und (*iii*) im Bild oben?

Für die Auswertung von Punkten $B(t)$ entlang der Kurve (etwa zum Zeichnen) ist die Verwendung der Darstellung (16.7) mitsamt der Definition (16.1) für $B_i^{(n)}$ numerisch ungünstig. Stattdessen verwendet man den **Algorithmus von de Casteljau**. Dieser nutzt die rekursive Darstellung (16.5) aus. Für $n \geqslant 1$ gilt nämlich:

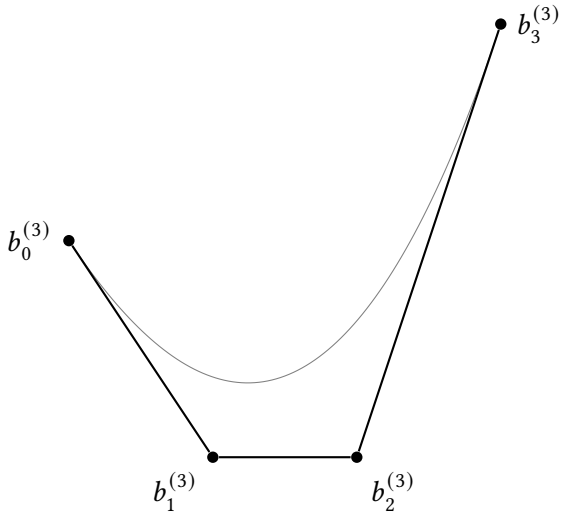
$$\begin{aligned}
B(t) &= \sum_{i=0}^{n} B_i^{(n)}(t)\, b_i \\
&= B_0^{(n)}(t)\, b_0 + \sum_{i=1}^{n-1} \left[B_i^{(n)}(t)\, b_i \right] + B_n^{(n)}(t)\, b_n \\
&= (1-t)\, B_0^{(n-1)}(t)\, b_0 + \sum_{i=1}^{n-1} \left[t\, B_{i-1}^{(n-1)}(t) + (1-t)\, B_i^{(n-1)}(t) \right] b_i + t\, B_{n-1}^{(n-1)}(t)\, b_n \quad \text{(wegen (16.5))} \\
&= \sum_{i=0}^{n-1} B_i^{(n-1)}(t) \underbrace{\left[(1-t)\, b_i + t\, b_{i+1} \right]}_{\substack{\text{neue Kontrollpunkte durch Konvexkombination der vorherigen}}} \quad \text{(durch Umsortieren).}
\end{aligned} \tag{16.8}$$

Diese Darstellung sieht wieder genauso aus wie die, mit der wir begonnen haben, jedoch mit dem Grad $n-1$ an Stelle von n und mit neuen Kontrollpunkten, die sich aus der Konvexkombination (anteiligen Mischung) von Paaren benachbarter Kontrollpunkte ergeben. Wir bezeichnen jetzt diese neuen Kontrollpunkte mit $b_0^{(n-1)}, \dots, b_{n-1}^{(n-1)}$ und die ursprünglichen mit $b_0^{(n)}, \dots, b_n^{(n)}$. Durch wiederholte Anwendung dieses Arguments erhalten wir schließlich eine «Kurve» mit dem Bernstein-Polynom $B_0^{(0)}$ (**Quizfrage:** Welche Funktion ist das?) und nur noch einem Kontrollpunkt $b_0^{(0)}$. Dieser ist der gesuchte Funktionswert $B(t)$.

Bevor wir den Algorithmus von de Casteljau angeben, wollen wir das Vorgehen noch grafisch anhand der Kurve aus Beispiel 16.4 veranschaulichen.

Beispiel 16.6 (Grafische Durchführung des Algorithmus von de Casteljau). *Wir betrachten wieder die durch die vier Kontrollpunkte $b_0, \dots, b_3$ (die wir jetzt mit $b_0^{(3)}, \dots, b_3^{(3)}$ bezeichnen) gegebene Bézier-*

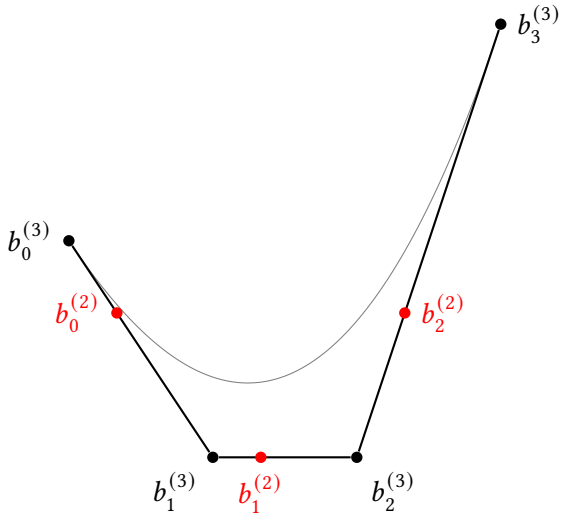
Kurve B aus Beispiel 16.4. Wir wollen diese an der Stelle $t = 1/3$ auswerten.[8]



Nach (16.8) müssen zunächst die drei neuen Kontrollpunkte

$$
\begin{aligned}
b_0^{(2)} &:= \frac{2}{3} b_0^{(3)} + \frac{1}{3} b_1^{(3)} \\
b_1^{(2)} &:= \frac{2}{3} b_1^{(3)} + \frac{1}{3} b_2^{(3)} \\
b_2^{(2)} &:= \frac{2}{3} b_2^{(3)} + \frac{1}{3} b_3^{(3)}
\end{aligned}
$$

konstruieren. Das heißt, dass wir die drei Strecken zwischen jeweils benachbarten Kontrollpunkten der Stufe 3 anteilig teilen und dort die drei neuen Kontrollpunkte der Stufe 2 einzeichnen:

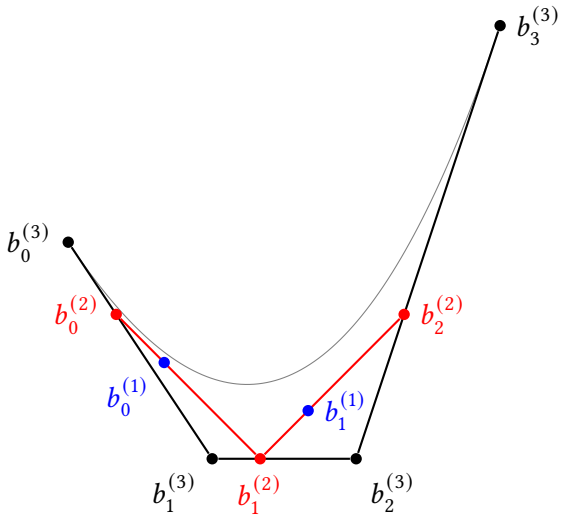


[8] Die Kurve ist zwar dünn in grau eingezeichnet, dies dient aber nur der Orientierung. Wir wollen ihre Funktionswerte, exemplarisch den für $t = 1/3$, ja gerade erst bestimmen.

Auf der nächste Stufe wiederholt sich das Ganze. Wir teilen also die Strecken zwischen den gerade hinzugefügten Kontrollpunkten der Stufe 2 *wiederum anteilig und fügen die zwei neuen Kontrollpunkte*

$$b_0^{(1)} := \frac{2}{3}b_0^{(2)} + \frac{1}{3}b_1^{(2)}$$
$$b_1^{(1)} := \frac{2}{3}b_1^{(3)} + \frac{1}{3}b_2^{(3)}$$

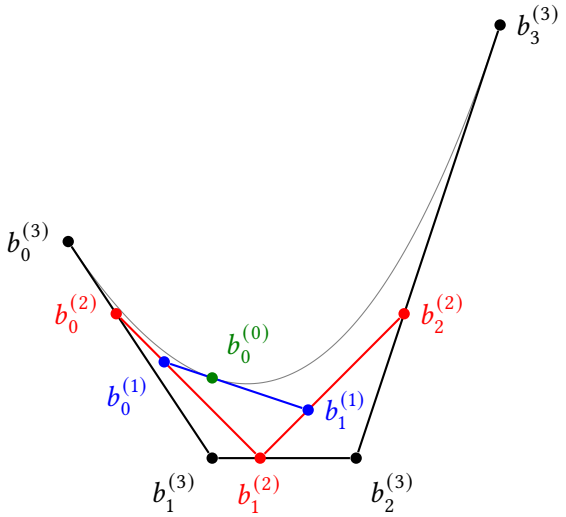
der Stufe 1 *hinzu:*



Schließlich wird auch die letzte Stecke wieder geteilt. Der neue Kontrollpunkt

$$b_0^{(0)} := \frac{2}{3}b_0^{(1)} + \frac{1}{3}b_1^{(1)}$$

der Stufe 0 *ist der gesuchte Punkt* $B(t)$ *auf der Kurve:*



Eine ansprechende interaktive Version dieser grafischen Konstruktion finden Sie zum Beispiel unter https://pomax.github.io/bezierinfo/#decasteljau.

Quizfrage: Wie kann man mit Hilfe der grafischen Konstruktion begründen, dass die Kurve $t \mapsto B(t)$ innerhalb der konvexen Hülle der Kontrollpunkte verläuft (Aussage *(iii)* aus Satz 16.5)?

Wir können das Verfahren nun formal angeben:

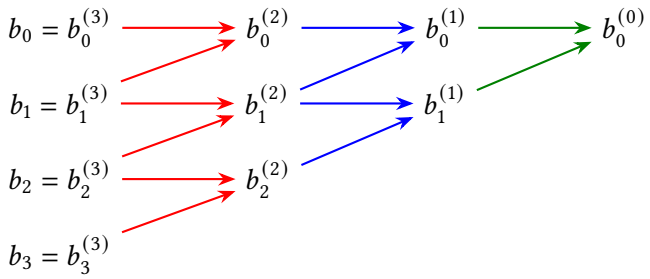
Algorithmus 16.7 (Algorithmus von de Casteljau zur Auswertung einer Bézier-Kurve).
Eingabe: *Kontrollpunkte* $b_0, \dots, b_n \in \mathbb{R}^d$ $(n \in \mathbb{N}_0)$ *einer Bézier-Kurve* B
Eingabe: *Auswertungsstelle* $t \in [0,1]$
Ausgabe: *Funktionswert* $y = B(t)$

1: ***for*** $i = 0, \dots, n$ ***do***
2: Setze $b_i^{(n)} := b_i$
3: ***end for***
4: ***for*** $\ell = n-1, \dots, 0$ ***do***
5: ***for*** $i = 0, \dots, \ell$ ***do***
6: Setze $b_i^{(\ell)} := (1-t)\, b_i^{(\ell+1)} + t\, b_{i+1}^{(\ell+1)}$
7: ***end for***
8: ***end for***
9: ***return*** $y := b_0^{(0)}$

Quizfrage: Wieviele FLOPs benötigt ungefähr eine Auswertung $B(t)$?

Quizfrage: Wie könnte man das Verfahren «vektorisieren», sodass die Schleifen über i jeweils wegfallen?

Die Struktur der Kontrollpunkte auf den verschiedenen Stufen und ihre Abhängigkeiten ähnelt dem Schema für die dividierten Differenzen, siehe nach Definition 15.8:



Unsere Variante von Algorithmus 16.7 berechnet die Kontrollpunkte Spalte für Spalte, wir könnten allerdings wie bei den dividierten Differenzen üblich auch auch Diagonale für Diagonale vorgehen.
Quizfrage: Wie kann das Hinzufügen einer Diagonale am unteren Ende grafisch gedeutet werden?

Quizfrage: Beim Vorgehen Diagonale für Diagonale erhalten wir als letzte Größe jeweils diejenige, die oben in der ersten Zeile steht, also $b_0^{(3)}$, dann $b_0^{(2)}$, anschließend $b_0^{(1)}$ und schließlich $b_0^{(0)}$. Wie lassen sich diese Größen interpretieren?

Bézier-Kurven haben viele vorteilhafte Eigenschaften:

(1) Auch für die Bestimmung der ersten und höherer Ableitungen lässt sich ein Verfahren ähnlich dem Algorithmus 16.7 von de Casteljau angeben.

(2) Bézier-Kurven sind affin invariant. Das heißt, dass das Bild einer Bézier-Kurve b unter einer affinen Abbildung $x \mapsto Ax + b$ wieder eine Bézier-Kurve ist, und zwar diejenige, die durch die affin transformierten Kontrollpunkte festgelegt wird.

(3) Bestimmte Operationen lassen sich für Bézier-Kurven effizient ausführen, darunter die Teilung einer Kurve in zwei Kurven und die (scheinbare) Erhöhung des Grades[9] einer Kurve.

(4) Bézier-Kurven lassen sich zu Bézier-Flächen verallgemeinern. In dieser Form haben sie große Bedeutung im *Computer Aided Design* (CAD).

§ 17 Approximation

Wie in der Einleitung zu Kapitel 6 erwähnt, geht es bei der Approximation einer gegebenen Funktion f darum, eine andere Funktion g aus einer Menge von zur Verfügung stehenden Funktionen so auszuwählen, dass Approximationsfehler $\|f - g\|$ in einer geeigneten Norm minimiert wird.

§ 17.1 Tschebyschow-Approximation

Eine häufig verwendete Approximationsaufgabe ergibt sich aus dem Wunsch, den *maximalen punktweisen* Approximationsfehler in einem gegebenen Intervall $[a, b]$ zu minimieren. Der Fehler wird also in der bereits bekannten Supremumsnorm gemessen:

$$\|f - p\|_{C([a,b])} := \sup_{x \in [a,b]} |f(x) - p(x)|.$$

Zur Approximation verwenden wir wieder Polynome mit gegebenem Höchstgrad n. Das führt auf die folgende Aufgabe: Zu einer gegebenen stetigen Funktion $f\colon [a, b] \to \mathbb{R}$ finde ein Polynom $p \in P_n$, das die Aufgabe

$$\text{Minimiere} \quad \|f - p\|_{C([a,b])}, \quad p \in P_n \tag{17.1}$$

löst. Diese Aufgabe heißt die **Tschebyschow-Approximationsaufgabe**[10], häufig ungenau in der Form **«Tschebyscheff»** transkribiert (englisch: *Chebyshev approximation problem*).

[9] Dadurch erhält man mehr Kontrollpunkte, also auch mehr Möglichkeiten, die Kurve z. B. in einem Grafikprogramm zu verändern.

[10] Чебышёв

Satz 17.1 (Existenz und Eindeutigkeit der Lösung der Tschebyschow-Approximationsaufgabe (17.1), vgl. Rannacher, 2017, Satz 2.8)**.** *Für jede stetige Funktion $f\colon [a,b] \to \mathbb{R}$ und jeden Polynomgrad $n \in \mathbb{N}_0$ besitzt die Tschebyschow-Approximationsaufgabe (17.1) eine Lösung.*

Beweis. Es ist ausreichend, einen Minimierer innerhalb der Menge

$$\left\{p \in P_n \,\middle|\, \|p\|_{C([a,b])} \leqslant 2\,\|f\|_{C([a,b])}\right\} \tag{17.2}$$

zu suchen, denn jedes $p \in P_n$ mit der Eigenschaft $\|p\|_{C([a,b])} > 2\,\|f\|_{C([a,b])}$ erfüllt

$$\begin{aligned}
\|f-p\|_{C([a,b])} &\geqslant \|p\|_{C([a,b])} - \|f\|_{C([a,b])} \quad \text{(umgedrehte Dreiecksungleichung)}\\
&> \|f\|_{C([a,b])} \quad \text{(Voraussetzung)}\\
&= \|f-0\|_{C([a,b])}\\
&\geqslant \inf_{p\in P_n} \|f-p\|_{C([a,b])}
\end{aligned}$$

und ist damit sicher kein Minimierer. Die Menge (17.2) ist aber als beschränkte abgeschlossene Menge (Kugel) in einem endlich-dimensionalen Raum kompakt. Da die Zielfunktion in (17.1) stetig ist (**Quizfrage:** Warum ist das so?), existiert nach dem **Satz von Weierstraß** ein Minimierer, also eine Lösung von (17.1). □

Minimierer von (17.1) haben folgende interessante charakterische Eigenschaft:

Satz 17.2 (**Alternantensatz** (englisch: *equioscillation theorem*), vgl. Rannacher, 2017, Satz 2.19)**.** *Folgende Bedingungen sind für eine Funktion $p \in P_n$ äquivalent:*

(*i*) *p ist eine Lösung von (17.1).*

(*ii*) *Es existieren $m \geqslant n+2$ Stellen $a \leqslant \xi_0 < \xi_1 < \cdots < \xi_m \leqslant b$ und $s \in \{\pm 1\}$, sodass für die Fehlerfunktion $f-p$ gilt:*

$$f(\xi_i) - p(\xi_i) = s\,(-1)^i\,\|f-p\|_{C([a,b])}. \tag{17.3}$$

*Die Menge der Extremstellen $\{\xi_0,\ldots,\xi_m\}$ wird **Alternante** der Funktion $f-p$ genannt.*

Wir verzichten hier auf einen Beweis. **Quizfrage:** Was bedeutet die Eigenschaft (17.3)?

Folgerung 17.3 (Eindeutigkeit der Lösung von (17.1), vgl. Rannacher, 2017, Korollar 2.2)**.** *Die Lösung der Tschebyschow-Approximationsaufgabe (17.1) ist eindeutig.*

Beweis. Nehmen wir an, p_1 und p_2 seien beides Lösungen von (17.1). Wir kürzen die Fehlerfunktionen mit $e_1 := f-p_1$ und $e_2 := f-p_2$ ab. Wir können annehmen, dass e_1 nicht die Nullfunktion ist, sonst wäre auch e_2 die Nullfunktion (**Quizfrage:** Warum?) und $p_1 = p_2 = f$ gezeigt. Es gilt also $\|e_1\|_{C([a,b])} = \|e_2\|_{C([a,b])} > 0$.

Es sei nun $\lambda \in (0,1)$ beliebig. Dann ist $\|\lambda\, e_2\|_{C([a,b])} < \|e_1\|_{C([a,b])}$. Daraus folgt, dass der Graph von $\lambda\, e_2$ den Graphen von e_1 mindestens $n+1$ Mal schneidet. (**Quizfrage:** Klar?) Die Funktion $e_\lambda := e_1 - \lambda\, e_2$ hat also mindestens $n+1$ verschiedene Nullstellen. Im Grenzübergang $\lambda \nearrow 1$ fallen einige davon möglicherweise zusammen, es bleiben jedoch $n+1$ Nullstellen (ihrer Vielfachheit entsprechend oft gezählt) von $e_1 - e_2 = p_2 - p_1$. Da jedoch $p_2 - p_1 \in P_n$ ist, muss dies die Nullfunktion sein. □

Für die Lösung der Tschebyschow-Optiimerungsaufgabe (17.1) gibt es Verfahren, die auf der Struktur der Lösung nach dem Alternantensatz 17.2 aufbauen, darunter der Algorithmus von Remez. Eine Alternative ist die Formulierung als eine lineare Optimierungsaufgabe (mit eigentlich unendlich vielen Nebenbedingungen, die man dann in der Regel diskretisiert), siehe Vorlesung *Grundlagen der Optimierung*.

Ein Beispiel für die Tschebyschow-Approximation wird in Abbildung 17.1 gezeigt.

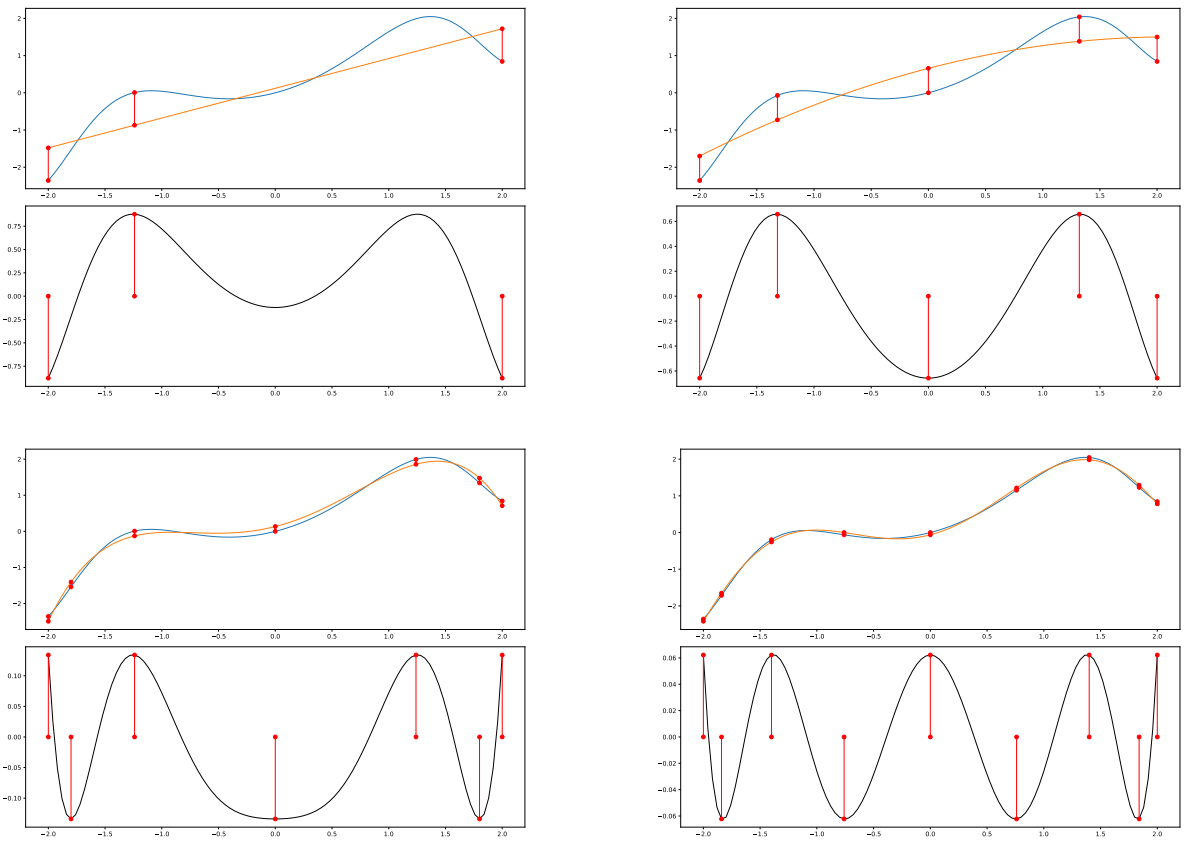


Abbildung 17.1: Illustration einer Tschebyschow-Approximation durch Polynome verschiedener Grade $n \in \{1, 3, 5, 7\}$ auf $[-2, 2]$.

§ 17.2 Anwendung der Tschebyschow-Approximation zur Optimierung der Stützstellen

Wir kommen noch einmal zurück auf die Darstellung (15.25) des Approximationsfehlers aus Satz 15.11

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\,\ell(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x - x_j) \tag{15.25}$$

bei der Polynominterpolation einer Funktion $f \in C^{n+1}([a,b])$.

Frage: Wie sind die Stützstellen $x_0, \dots, x_n \in [a,b]$ zu wählen, sodass das Knotenpolynom $\ell(x) = \prod_{j=0}^{n}(x - x_j)$ minimale Ausschläge hat? Wir betrachten also die Optimierungsaufgabe

$$\text{Minimiere} \quad \sup_{x\in[a,b]} \left|\prod_{j=0}^{n}(x - x_j)\right|, \quad x_0, \dots, x_n \in [a,b]. \tag{17.4}$$

Die Funktion $\prod_{j=0}^{n}(x - x_j)$ ist gleich $x^{n+1} - p$ für ein Polynom $p \in P_n$, das von den Stützstellen abhängt. Daher können wir die Aufgabe (17.4) auch so lesen, dass wir die Tschebyschow-Approximation (17.1) der Funktion $x \mapsto x^{n+1}$ auf $[a,b]$ suchen, also

$$\text{Minimiere} \quad \sup_{x\in[a,b]} |x^{n+1} - p(x)|, \quad p \in P_n. \tag{17.5}$$

Die Lösung der Aufgabe (17.5) kann explizit angegeben werden. Es reicht dabei aus, das Intervall $[a,b] = [-1,1]$ zu betrachten. Für andere Intervalle kann das Ergebnis nachher übertragen werden. Die Darstellung der Lösung in Satz 17.5 verwendet die folgenden Funktionen:

Definition 17.4 (Tschebyschow-Polynome)**.** *Es sei $n \in \mathbb{N}_0$. Das n-te **Tschebschow-Polynom** (erster Art) (englisch:* Chebyshev polynomial (of the first kind)*) ist definiert durch die Rekursion*

$$T_0(x) := 1, \tag{17.6a}$$
$$T_1(x) := x, \tag{17.6b}$$
$$T_{n+1}(x) := 2x\,T_n(x) - T_{n-1}(x). \tag{17.6c}$$

Mit Hilfe dieser Definition können wir leicht zeigen, dass T_n ein Polynom vom Grad n ist, dessen führender Koeffizient im Falle von $n \geqslant 1$ gleich 2^{n-1} ist. Eine andere Darstellung ist

$$T_n(x) = \cos\bigl(n \arccos(x)\bigr) \quad \text{für } x \in [-1,1]. \tag{17.7}$$

Die Nullstellen von T_n sind

$$x_j = \cos\left(\frac{\pi}{2}\,\frac{2j+1}{n}\right), \quad j = 0, \dots, n-1 \tag{17.8}$$

und heißen **Tschebyschow-Knoten** (englisch: *Chebyshev nodes*).

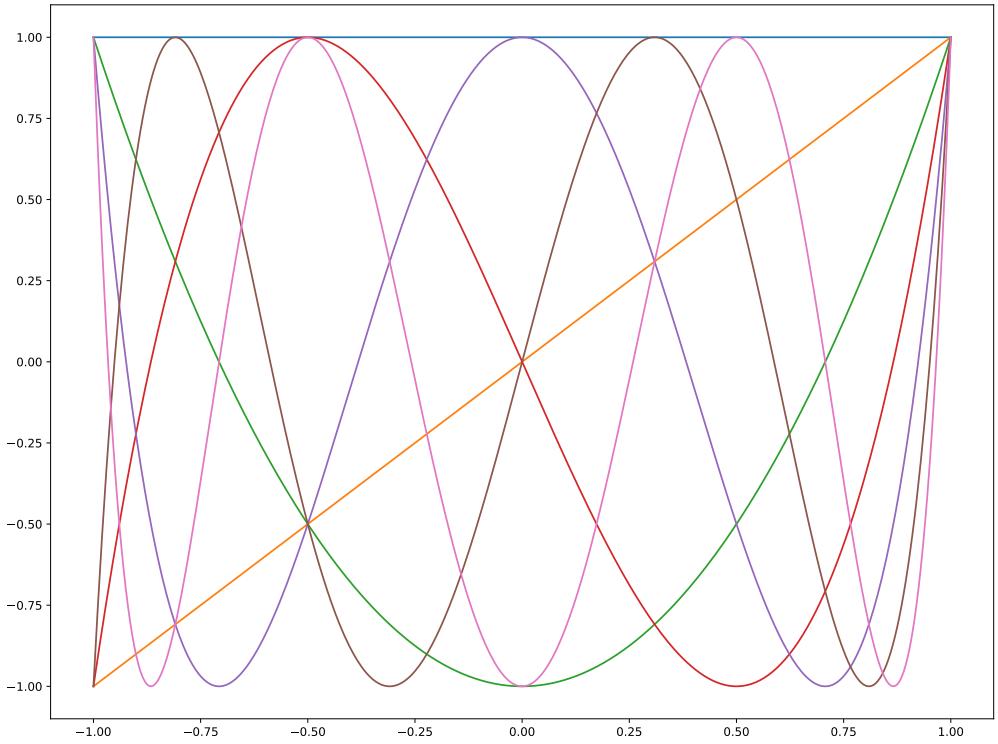


Abbildung 17.2: Illustration der Tschebyschow-Polynome verschiedener Grade $n \in \{0, 1, \dots, 6\}$ auf $[-1, 1]$.

Satz 17.5 (Lösung der Tschebyschow-Approximationsaufgabe (17.5) und optimale Stützstellen, vgl. Rannacher, 2017, Satz 2.20). *Es sei $n \in \mathbb{N}_0$. Die eindeutige Lösung der Aufgabe (17.5) auf $[a, b] = [-1, 1]$ ist gegeben durch*

$$p(x) := x^{n+1} - \frac{1}{2^n} T_{n+1}(x). \tag{17.9}$$

Die Tschebyschow-Knoten (Nullstellen von T_{n+1}) sind die optimale Stützstellen von (17.4).

Beweis. Wir wissen, dass T_{n+1} den führenden Term $2^n x^{n+1}$ besitzt, und kennen außerdem seine Nullstellen $x_0, \dots, x_n$, siehe (17.8). Daraus folgt, dass $\frac{1}{2^n} T_{n+1}$ die Darstellung

$$\frac{1}{2^n} T_{n+1}(x) = \prod_{j=0}^{n} (x - x_j) = \ell(x)$$

besitzen muss. Aus der Darstellung (17.7) folgt, dass T_{n+1} im Intervall $[-1, 1]$ genau $(n + 2)$ Mal seinen (positiven bzw. negativen) Extremwert annimmt, und zwar abwechselnd ± 1. Dasselbe gilt für $\frac{1}{2^n} T_{n+1}$ mit dem Extremwert $\pm \frac{1}{2^n}$. Mit anderen Worten: Diese $(n + 2)$ Extremstellen bilden eine Alternante der Funktion $x \mapsto x^{n+1} - \frac{1}{2^n} T_{n+1}(x)$, die ein Polynom in P_n ist. Der Alternantensatz 17.2 garantiert jetzt, dass $x \mapsto x^{n+1} - \frac{1}{2^n} T_{n+1}(x)$ die Aufgabe (17.5) löst.

Es bleibt noch zu zeigen, dass die Nullstellen x_j aus (17.8) die optimale Stützstellen von (17.4) sind. Wir

haben

$$\begin{aligned}\sup_{x\in[-1,1]}\left|\prod_{j=0}^{n}(x-x_j)\right| &= \frac{1}{2^n}\sup_{x\in[-1,1]}|T_{n+1}(x)|\\ &= \sup_{x\in[-1,1]}\left|x^{n+1}-\left[x^{n+1}-\frac{1}{2^n}T_{n+1}(x)\right]\right|, \quad \text{denn } x\mapsto x^{n+1}-\frac{1}{2^n}T_{n+1}(x) \text{ löst (17.5)}\\ &= \min_{p\in P_n}\sup_{x\in[-1,1]}|x^{n+1}-p(x)|.\end{aligned}$$

Wir wissen, dass $x^{n+1}-p(x)$ irgendein Polynom in P_{n+1} mit führendem Koeffizienten 1 ist. Die nachstehend verwendeten Polynome $\prod_{j=0}^{n}(x-y_j)$ mit Stützstellen $y_j\in[-1,1]$ sind ebenfalls Polynome in P_{n+1} mit führendem Koeffizienten 1 (aber möglicherweise nicht alle). Daher ist

$$\leqslant \min\left\{\sup_{x\in[-1,1]}\left|\prod_{j=0}^{n}(x-y_j)\right| \;\middle|\; y_j\in[-1,1],\ j=0,1,\dots,n\right\}.$$

Das zeigt, dass $\prod_{j=0}^{n}(x-x_j)$ tatsächlich optimale Stützstellen für (17.4) sind. □

Bemerkung 17.6 (Bedeutung der Tschebyschow-Knoten)**.** *Die Tschebyschow-Knoten sind diejenigen Stützstellen, die ohne Ansehen der zu interpolierenden Funktion f optimale (kleinstmögliche) Fehlerabschätzungen ermöglichen.*

Wir greifen nochmal die Runge-Funktion aus (15.13) auf und interpolieren sie durch Polynome wachsenden Grades, wobei wir als Stützstellen jetzt aber die jeweiligen Tschebyschow-Knoten verwenden. Dieses Mal können wir die gleichmäßige Konvergenz der Folge der Interpolationspolynome beobachten, siehe Abbildung 17.3.

Ende der Woche 10

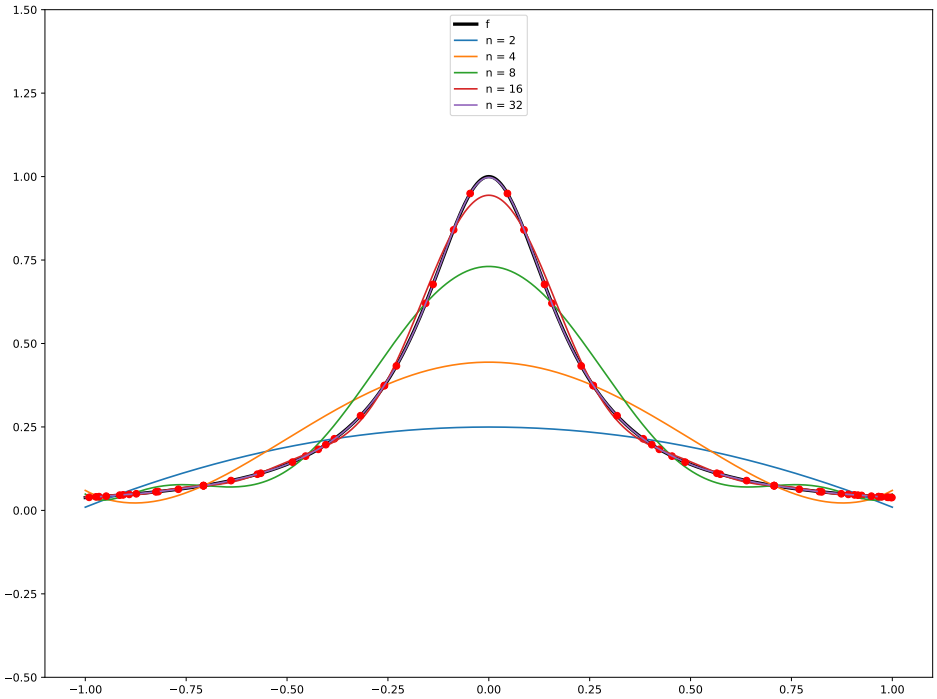


Abbildung 17.3: Interpolation der Runge-Funktion (schwarz) aus Beispiel 15.13 durch Polynome mit Tschebyschow-Knoten auf $[-1, 1]$ als Stützstellen.

Index

Literatur

Bauer, F. L. (1974). «Computational graphs and rounding error». *SIAM Journal on Numerical Analysis* 11.1, S. 87–96. DOI: 10.1137/0711010.

Beutelspacher, A. (2014). *Lineare Algebra. Eine Einführung in die Wissenschaft der Vektoren, Abbildungen und Matrizen*. 8. Aufl. Springer Fachmedien Wiesbaden. DOI: 10.1007/978-3-658-02413-0.

Bornemann, F. (2018). *Numerische lineare Algebra*. 2. Aufl. Springer Fachmedien Wiesbaden. DOI: 10.1007/978-3-658-24431-6.

Brin, S.; L. Page (1998). «The anatomy of a large-scale hypertextual Web search engine». *Computer Networks and ISDN Systems* 30.1-7, S. 107–117. DOI: 10.1016/s0169-7552(98)00110-x.

Freund, R. W.; R. W. Hoppe (2007). *Stoer/Bulirsch: Numerische Mathematik 1*. 10. Aufl. Springer Berlin Heidelberg. DOI: 10.1007/978-3-540-45390-1.

Golub, G.; C. van Loan (1983). *Matrix Computations*. Baltimore: Johns Hopkins University Press.

Heuser, H. (2002). *Lehrbuch der Analysis. Teil 2*. 12. Aufl. Stuttgart: B.G.Teubner. DOI: 10.1007/978-3-322-96826-5.

Higham, N. J. (2002). *Accuracy and Stability of Numerical Algorithms*. Society for Industrial und Applied Mathematics. DOI: 10.1137/1.9780898718027.

Higham, N. J.; M. R. Dennis; P. Glendinning; P. A. Martin; F. Santosa; J. Tanner, Hrsg. (2016). *The Princeton Companion to Applied Mathematics*. Princeton University Press. DOI: 10.1515/9781400874477.

Kahan, W. (1966). «Numerical linear algebra». *Canadian Mathematical Bulletin* 9.05, S. 757–801. DOI: 10.4153/cmb-1966-083-2.

Rannacher, R. (2017). *Numerik 0: Einführung in die Numerische Mathematik*. Heidelberg University Publishing. DOI: 10.17885/HEIUP.206.281.

Trefethen, L. N. (1992). «The definition of numerical analysis». *SIAM News*. URL: https://people.maths.ox.ac.uk/trefethen/publication/PDF/1992_55.pdf.

Trefethen, L. N.; D. Bau (1997). *Numerical Linear Algebra*. SIAM. DOI: 10.1137/1.9780898719574.

Tyrtyshnikov, E. E. (1994). «How bad are Hankel matrices?» *Numerische Mathematik* 67.2, S. 261–269. DOI: 10.1007/s002110050027.