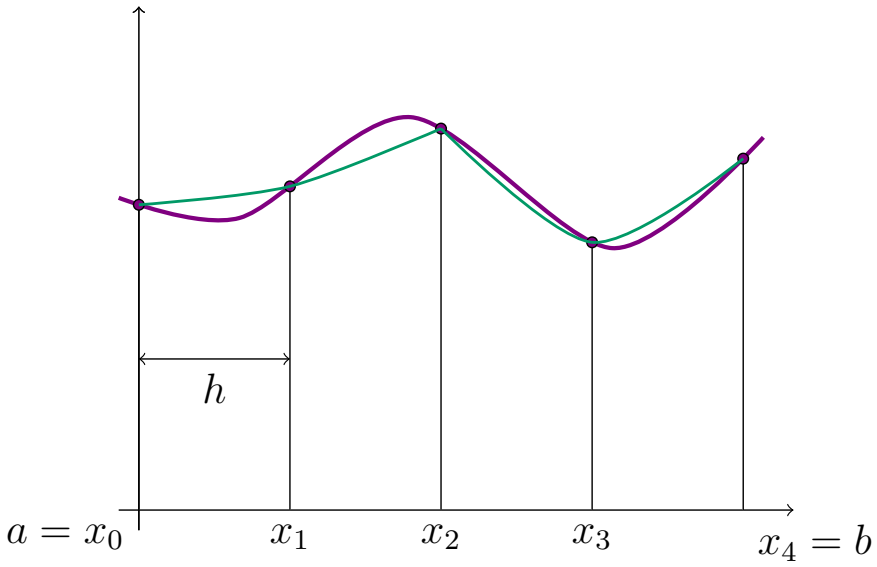
# COMPUTERNUMERIK

Gabriela SCHRANZ-KIRLINGER

unter der Mitarbeit von Laura Lotteraner




Institut für Analysis und Scientific Computing

Technische Universität Wien

Oktober 2017

# Inhaltsverzeichnis

# Kapitel 1

# Fehlerbetrachtungen

Da für komplexe mathematische Problemstellungen eine exakte Lösung meist gar nicht oder nur mit großem Aufwand gefunden werden kann, ist die Ermittlung einer Näherungslösung mit Hilfe eines numerischen Verfahrens oft die einzige Alternative. Oft ist die Kenntnis einer exakten Lösung ohnehin nicht notwendig, meist reicht eine entsprechend genaue numerische Näherung. In dieser Vorlesung werden die wichtigsten Konzepte und Eigenschaften solcher algorithmisch-numerischer Lösungsmethoden diskutiert und auf Schwachstellen aufmerksam gemacht.
Durch den Einsatz von numerischen Methoden ist die ermittelte Lösung im Allgemeinen mit **Fehlern** (Abweichungen von der exakten Lösung) behaftet. Die Fehler werden üblicherweise in vier Gruppen zusammengefasst,

**Modellfehler**, **Datenfehler**, **Verfahrensfehler** und **Rechenfehler**.

Eine numerische Methode besteht aus einem **Algorithmus**, dem Rechenverfahren zur Ermittlung der Näherungslösung, sowie einer zuverlässigen **Schätzung für den Fehler**, um zu überprüfen, ob die erzielte Genauigkeit für die gewünschte Anwendung ausreicht.

Was bedeutet aber die Aussage „*Die Genauigkeit soll für eine bestimmte Anwendung ausreichen.*“?

**Beispiel 1.0.1** (Landung einer Weltraumkapsel)**.** *Aus technischen Gründen kann eine Weltraumkapsel bei Rückkehr auf die Erde nicht landen, sondern nur „wassern“. Der beabsichtigte Punkt der Wasserung liegt im Golf von Mexiko. Die geringste Entfernung von diesem Punkt zur Küste ist a; daher ist die Genauigkeitsforderung bei der Berechnung des Landemanövers:* Toleranz $= a$.

*Falls darüber hinaus die Kapsel z.B. innerhalb einer Stunde nach der Wasserung geborgen werden muss, und die Bergungsschiffe innerhalb einer Stunde einen Weg der Länge b zurücklegen, gilt* Toleranz $= b$, *bzw.* Toleranz $= \min(a, b)$.

*Die geforderte Genauigkeit der Lösung hängt also stark von den äußeren Bedingungen des Modells ab.*

## 1.1 Modellfehler

Für keinen realen Vorgang gibt es ein mathematisches Modell, das die Realität voll erfasst. Im Allgemeinen gibt es dagegen eine ganze Schar verschieden feiner mathematischer Modelle, welche mehr oder weniger Aspekte des Vorganges berücksichtigen..

**Beispiel 1.1.1** (Wasserung einer Weltraumkapsel)**.** *Fortsetzung von Beispiel 1.0.1.*
*Betrachte erneut die Wasserung einer Weltraumkapsel. Es werden unterschiedlich feine Modelle verglichen.*

**Meistens:** *Am einfachsten (und ungenauesten) ist die Modellierung der Kapsel als* Massenpunkt*, man stellt sich also die ganze Masse in einem Punkt vereinigt vor.*

**Feiner:** *Genauer ist die Modellierung der Kapsel als dreidimensionaler* starrer Körper *– dies ermöglicht die Erfassung von Rotations und Schlingerbewegungen.*

**Noch feiner:** *Eine weitere Verfeinerung stellt die Modellierung der Kapsel als dreidimensionaler* elastischer Körper *dar, der unter Einfluß von Kräften verformbar oder sogar zerstörbar ist. Beim Wiedereintritt in die Erdatmosphäre ist die Kapsel (aufgrund ihrer hohen Geschwindigkeit) sehr starken Reibungskräften ausgesetzt und es sollte untersucht werden, ob sie diesen Belastungen standhält.*

**Noch feiner:** *Zusätzlich können auch* thermodynamischen Effekte *berücksichtigt werden. In der Erdatmosphäre werden Reibungskräfte teilweise in Wärme umgewandelt, es ergibt sich daher eine starke Hitzeentwicklung. Für die Raumkapsel sollte das Verglühen des Hitzeschildes vermieden werden.*

*Welche Feinheit das Modell der Raumkapsel aufweisen soll, hängt vom Anwendungszweck ab: bei der Berechnung der Landebahn (Ermittlung des Wasserungspunktes) genügt etwa i.A. die Modellierung als Massenpunkt. Dabei ist zu beachten:*

1. *Die* **Modellfehler** *sind in diesem Fall die Vernachlässigung der Ausdehnung und geometrischen Form der Kapsel, der Elastizitätseigenschaften, der thermodynamischen Effekte, ...*
   *Es muss untersucht werden, ob die Effekte dieser Modellfehler so klein sind, dass die Genauigkeitstoleranz (die Größen $a$ oder $b$ oder $\min(a,b)$) nicht gefährdet erscheint.*

2. *Der Massenpunkt bewegt sich unter dem Einfluß von Kräften (Gravitation, Luftwiderstand) durch die Luft. Die Bewegungsbahn kann durch das Lösen eines Systems gewöhnlicher Differentialgleichungen – eines Anfangswertproblems – ermittelt werden. Dabei müssen die Kräfte in jedem Raumpunkt (Kraftfelder) gegeben sein.*
   *Es wird also ein mathematisches Modell des Gravitationsfeldes und der Erdatmosphäre benötigt:*

   (a) *Nach dem Gesetz von Newton lässt sich die* **Gravitation** *durch*

$$K = \frac{Gm_1m_2}{r^2}$$

   *beschreiben, wobei*

$$\begin{array}{lcl} K & = & \text{Gravitationskraft (genauer: Betrag des Kraftvektors),} \\ G & = & \text{Gravitationskonstante,} \\ r & = & \text{Abstand,} \\ m_2 & = & \text{Masse der Kapsel,} \\ m_1 & = & \text{Masse der Erde, (oder ev. des Mondes,} \\ & & \text{der Sonne, von anderen Planeten,...).} \end{array}$$

*Die Gesamtgravitationskraft, die auf die Kapsel wirkt, ergibt sich durch Aufsummation (Integration) über alle Massenelemente.*
*Zur Vereinfachung werden die Gravitationswirkungen des Mondes, der Sonne, der anderen Planeten und Himmelskörper vernachlässigt und nur Gravitationswirkungen der Erde berücksichtigt.*
*Auch ungleichmäßige Massenverteilungen im Erdinneren bzw. die zeitabhängige Wasserverteilung auf der Erde (Ebbe-Flut) u.s.w. können vernachlässigt werden.*
*Nun muss wieder versucht werden, die Auswirkungen dieser Modellfehler bei der Modellierung des Gravitationsfeldes auf die Berechnung der Bahn der Kapsel einzuschätzen.*

(b) *Ähnliche Betrachtungen sind bezüglich der Modellierung der* **Erdatmosphäre** *zur Berechnung der Reibungskräfte (Luftwiderstand) notwendig, u.s.w.*

Oft können Auswirkungen von Modellfehler durch *Interpretation als Datenfehler* abgeschätzt werden. Abschätzung von Datenfehlereffekten sind für zahlreiche mathematische Problemstellungen bekannt.

Aber nicht alle Modellfehler lassen sich als Datenfehler interpretieren: z.B. dass die Raumkapsel als Massenpunkt (bzw. als starrer Körper) modelliert wurde, obwohl sie in Wirklichkeit ein elastisches Gebilde ist. Wollte man die Auswirkungen der Elastizität miterfassen, müsste man den Bereich der gewöhnlichen Differentialgleichungen verlassen. Die Elastizitätstheorie wird in der Physik durch partielle Differentialgleichungen dargestellt. Die Auswirkungen solcher Modellfehler müssten mit Konditionsabschätzungen partieller Differentialgleichungen untersucht werden – soferne man solche Abschätzungen überhaupt in der mathematischen Literatur findet. In der Praxis verzichtet man jedoch auf solche Untersuchungen; man geht davon aus, dass die Einflüsse von elastischen Verformungen auf die Bahn der Kapsel so gering sind, dass sie ohne jede praktische Relevanz sind.

## 1.2 Datenfehler

**Definition 1.2.1. Daten** *sind Größen, die schon vor der Rechnung bekannt (verfügbar) sind.*

In Beispiel 1.0.1/1.1.1 sind das z.B. die Masse der Raumkapsel, das Gravitationsfeld der Erde (Gravitationskraftvektor als Funktion des Ortes), die Dichte der Atmosphäre in Abhängigkeit von der Höhe (zur Zeit der Landung), ....
Daten können

- *Zahlen*, z.B. Masse der Raumkapsel,
- *Vektoren*, z.B. Lage (Ortsvektor) und Geschwindigkeit der Kapsel zu Beginn des Landemanövers,
- *Matrizen*,
- *Tensoren*,
- *Funktionen*, z.B. Dichte der Atmosphäre als Funktion $\rho : \mathbb{R}^1 \to \mathbb{R}^1$ der Höhe ($\rho(h)$), Gravitationsvektor als Funktion $\mathbb{R}^3 \to \mathbb{R}^3$ des Ortes bzw. als Funktion $\mathbb{R}^4 \to \mathbb{R}^3$ des Ortes und der Zeit (wegen der durch Ebbe und Flut bedingten zeitlich veränderlichen Massenverteilungen), etc.
- ...

sein, es kommen also beliebige mathematische Objekte als Daten in Betracht.

**Häufige Ursachen für Datenfehler** sind

- *Messfehler* (z.B. bei der Bestimmung der Masse der Raumkapsel – falls diese direkt durch Wägung ermittelt wurde oder, falls sie mithilfe ihrer Materialzusammensetzung berechnet wurde, durch messfehlerbehaftete spezifische Gewichte der Materialien).

- *Modellfehler,* die man als Datenfehler interpretiert (z.B. aufgrund von vereinfachten Modellannahmen ergibt sich verfälschtes Gravitationsfeld).

Wie die Lösung eines mathematischen Problems auf Datenänderungen reagiert, wird durch die **Kondition** des Problems beschrieben.

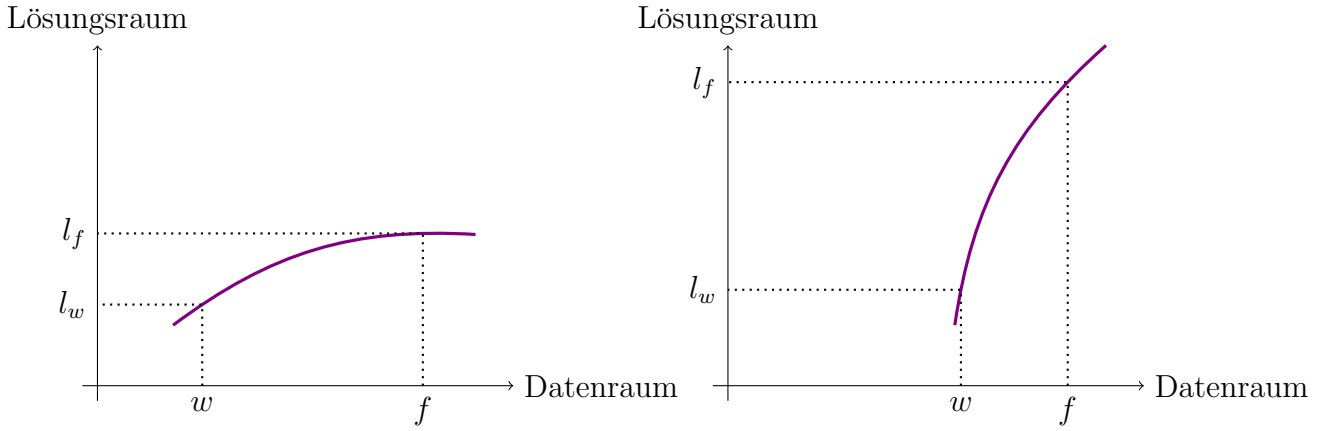
### 1.2.1 Kondition

**Definition 1.2.2. Kondition** *beschreibt die Abhängigkeit der Lösung von den Daten.*

*a) Ein Problem heißt* **gut konditioniert***, falls sich die Lösung nur wenig ändert, wenn die Daten des Problems verändert werden, z.B. etwa so stark wie die Daten selbst (vgl. Abb. 1.1a).*

*b) Ein Problem heißt* **schlecht konditioniert***, falls sich die Lösung bei geringer Änderung der Daten stark verändert, d.h. Datenfehler haben katastrophale Auswirkungen (vgl. Abb. 1.1b).*

Wissen über die Kondition eines Problems ist wichtig, um die Auswirkungen der stets unvermeidlichen Datenfehler auf die Genauigkeit der Lösung einschätzen zu können. Auch jene Modellfehler, die sich als Datenfehler interpretieren lassen, können dann mit Konditionsabschätzungen abgeschätzt werden.




(a) Gute Kondition (b) Schlechte Kondition

Abbildung 1.1: Abbildung vom Datenraum in den Lösungsraum: Jedem konkreten Datensatz (z.B. $w$ oder $f$) aus dem Datenraum entspricht eine konkrete Lösung ($l_w$ oder $l_f$) aus dem Lösungsraum

**Beispiel 1.2.3** (Lineares $2 \times 2$-Gleichungssystem)**.** *Es werden zwei lineare* $2 \times 2$*-Gleichungssysteme betrachtet.*

*a) Gegeben ist das System*

$$\begin{aligned} 1.253672417x_1 &+ 1.247798111x_2 &= 3.654199872 \\ -2.672344812x_1 &+ 2.695328007x_2 &= 2.479981003 \end{aligned}$$

*mit den Lösungen*

$$\begin{aligned} x_1 &= 1.006128817\ldots, \\ x_2 &= 1.917653108\ldots. \end{aligned}$$

*Das geringfügig verfälschte System*

$$\begin{aligned} 1.253672000x_1 &+ 1.247798000x_2 &= 3.654199000 \\ -2.672344000x_1 &+ 2.695328000x_2 &= 2.479981000 \end{aligned}$$

*hat die Lösungen*

$$\begin{aligned} x_1 &= \underline{1.006128}871\ldots, \\ x_2 &= \underline{1.917652}862\ldots, \end{aligned}$$

*die Verfälschung der Lösung ist also von derselben Größenordnung wie die Verfälschung der Daten.*

$\Longrightarrow$ *Das Problem ist gut konditioniert.*

*b) Gegeben ist das System*

$$\begin{aligned} 1.743681226x_1 &- 0.5287326143x_2 &= 2.666771987 \\ 4.359203065x_1 &- 1.321302803x_2 &= 6.667195145 \end{aligned}$$

*mit den Lösungen*

$$\begin{aligned} x_1 &= 1.682330907\ldots, \\ x_2 &= 0.5043710646\ldots. \end{aligned}$$

*Das erneut geringfügig verfälschte System*

$$\begin{aligned} 1.743681000x_1 &- 0.5287326000x_2 &= 2.666771000 \\ 4.359203000x_1 &- 1.321302000x_2 &= 6.667195000 \end{aligned}$$

*hat die Lösungen*

$$\begin{aligned} x_1 &= \underline{1.682}09869\ldots, \\ x_2 &= \underline{0.503}6052756\ldots, \end{aligned}$$

*die Verfälschung der Lösung ist also um den Faktor* 10000 *größer als die Verfälschung der Daten.*

$\Longrightarrow$ *Das Problem ist sehr schlecht konditioniert.*

Zwei verschiedene Probleme aus derselben Problemklasse (zwei lineare $2 \times 2$ Gleichungssysteme) können je nach Datensatz sehr verschieden konditioniert sein.
In der mathematischen Literatur finden sich zahlreiche Konditionsabschätzungen für verschiedene Problemklassen.

**Beispiel 1.2.4** (Anfangswertproblem einer gewöhnlichen Differentialgleichung)**.** *Das ungestörte Problem ist*

$$\begin{aligned} y'(t) &= f(t, y(t)), \\ y(0) &= y_0, \end{aligned}$$

*das Problem mit verfälschten Daten*

$$\begin{aligned} z'(t) &= f(t, z(t)) + \Delta(t, z(t)), \\ z(0) &= y_0 + \Delta z_0. \end{aligned}$$

*Wenn*

1. *$f$ lipschitzstetig mit Lipschitzkonstante $L$ ist, also*

$$\|f(t, y_1) - f(t, y_2)\| \leq L\|y_1 - y_2\|, \tag{1.1}$$

   *und*

2. *die Störungen die Abschätzungen*

$$\|\Delta(t, z(t))\| \leq \Delta \quad \textit{und} \quad \|\Delta z_0\| \leq \Delta_0$$

   *erfüllen,*

*dann gilt die Konditionsabschätzung*

$$\|y(t) - z(t)\| \leq e^{Lt} \cdot \Delta_0 + \frac{e^{Lt} - 1}{L} \cdot \Delta. \tag{1.2}$$

*Kritischer Fall: Für $L \gg 0$ liegt schlechte Kondition des Anfangswertproblems vor, falls die Konditionsabschätzung realistisch ist.*
*Betrachte z.B. $L = 100$ und $t = 1$, dann gilt bereits $e^{Lt} \approx 2.7 * 10^{43}$.*

Mit solchen bekannten Konditionsabschätzungen aus der mathematischen Literatur lässt sich die Kondition eines gegebenen Problems aus gewissen Klassen abschätzen. Kennt man z.B. für ein gegebenes Anfangswertproblem Schranken $\Delta_0$ und $\Delta$ für die Datenstörungen sowie $L$, erhält man mit 1.2 eine konkrete Abschätzung für $\|y(t) - z(t)\|$.
Für das Landemanöver der Raumkapsel etwa benötigt man Schranken $\Delta_0$ für die Messgenauigkeit des Anfangszustandes (Anfangslage und -geschwindigkeit) und eine Schranke $\Delta$ für die Ungenauigkeit der rechten Seite $f(t, y)$ der Differentialgleichung, also in diesem Fall die Modellfehler bei der Modellierung des Gravitationsfeldes und der Erdatmosphäre (Reibungskräfte).

Für einige konkrete Problemklassen werden wir später Konditionsabschätzungen kennenlernen.

## 1.3 Verfahrensfehler

Die meisten mathematischen Probleme sind nicht exakt lösbar!

**Beispiel 1.3.1** (Integration einer Funktion)**.** *Wir betrachten den*
**Hauptsatz der Differential und Integralrechnung:**
*Sei $f : [a, b] \to \mathbb{R}$ stetig und $F$ Stammfunktion von $f$. Dann gilt*

$$\int_a^b f(t)dt = F(b) - F(a) \qquad \forall a, b \in \mathbb{R}. \tag{1.3}$$

*Um ein Integral gemäß (1.3) berechnen zu können, muss die Funktion $f(t)$ erstens in einem gewissen Sinn (z.B. nach Riemann) integrierbar sein und zweitens eine Stammfunktion $F$ besitzen. Die meisten in den Anwendungen auftretenden Funktionen sind integrierbar und besitzen eine Stammfunktion, z.B. alle auf $[a, b]$ stetigen Funktionen.*
*Dennoch kann man in der Praxis sehr oft Integrale nicht gemäß (1.3) berechnen. Der entscheidende Punkt ist, dass sich sehr viele Funktionen nicht „geschlossen darstellen“ lassen. Man muss unterscheiden zwischen*

1. *der „Existenz von Funktionen im mathematischen Sinn“und*
2. *der Tatsache, dass sich gewisse Funktionen als „Formelausdruck“darstellen lassen, d.h. auf endliche Weise aus den 4 Grundrechenarten $+ \; - \; * \; \div$, aus $\sqrt[n]{}$ und aus den elementaren Funktionen* sin, cos, tan, exp, arcsin, arccos, arctan, ln, ... *aufgebaut sind.*

*Die Menge aller stetigen Funktionen auf dem Intervall $[0, 1]$ ist z.B. viel umfassender als die Menge der auf $[0, 1]$ stetigen, geschlossen darstellbaren Funktionen. Häufig liegt die Situation vor, dass der Integrand $f(t)$ zwar geschlossen darstellbar ist und eine Stammfunktion besitzt, diese aber keine geschlossene Darstellung hat. Beispiele dafür sind*

$$\frac{\exp(t)}{t}, \quad \frac{\sin(t)}{t} \quad \textit{und} \quad \frac{1}{\ln(t)} \qquad \textit{(ohne Beweis).}$$

*Für solche Integrale scheidet die Berechnung mittels (1.3) aus und man ist auf numerische Näherungsverfahren angewiesen. In anderen Fällen ist es einfach bequemer, das Integral über ein Näherungsverfahren zur berechnen, und zwar dann, wenn die Stammfunktion zwar geschlossen dargestellt werden kann, ihre Berechnung aber sehr mühsam ist!*

Eine analoge Situation ergibt sich für Differentialgleichungsprobleme (gewöhnliche und partielle Differentialgleichungen – Anfangs- Randwertaufgaben), Integralgleichungen, Integrodifferentialgleichungen, ... Fast immer können die Lösungen nicht geschlossen dargestellt werden und müssen mit numerischen Näherungsverfahren berechnet werden.

**Definition 1.3.2.** *Es gilt* **Verfahrensfehler** := *Numerische Näherung – exakte Lösung.*

**Beispiel 1.3.3** (Numerische Differentiation)**.** *Wenn $f$ geschlossen darstellbar ist, dann auch $f'$, das also mit Hilfe von Ableitungsregeln berechnet werden kann. Man könnte daher auf numerische Differentiation völlig verzichten. Oft ist aber numerische Differentiation bequemer.*
*Die* **Numerische Näherungsformel** *ist durch den Differenzenquotienten*

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

*gegeben, der Verfahrensfehler durch*

$$\underbrace{\frac{f(x+h) - f(x)}{h}}_{\textit{num. Näherungsausdruck}} \quad \underbrace{f'(x)}_{\textit{exakter Wert}} \; . \tag{1.4}$$

*Taylorreihenentwicklung liefert*

$$\begin{aligned}\frac{f(x+h)-f(x)}{h} - f'(x) = & \frac{1}{h}\left[\left(f(x)+h\cdot f'(x)+\frac{h^2}{2}\cdot f''(\theta)\right)-f(x)\right]-f'(x)\\ = & \frac{h}{2}f''(\theta) \qquad \theta\in(x,x+h),\end{aligned} \tag{1.5}$$

*also gilt die (a-priori)* **Verfahrensfehlerschranke**

$$\left|\frac{f(x+h)-f(x)}{h}-f'(x)\right| \le \frac{M_2}{2}h. \tag{1.6}$$

*Dabei ist $M_2$ eine Schranke für $f''$ in einer geeigneten Umgebung von $x$.*
*Die a-priori Schranke (1.6) wird in der Praxis nicht ausgewertet: Wenn man $f'(x)$ nicht kennt, sondern numerisch berechnet, ist $f''$ in einer Umgebung von $x$ und damit $M_2$ erst recht unbekannt. $\frac{M_2}{2}h$ kann also nicht berechnet und zur konkreten Abschätzung verwendet werden. Die Bedeutung von (1.6) liegt in der mathematischen Information:*

Konvergenzresultat: *Falls $f$ zweimal stetig differenzierbar ist, konvergiert der numerische Näherungsausdruck $\frac{f(x+h)-f(x)}{h}$ für $h\to 0$ gegen $f'(x)$.*

*Ein* **besseres Verfahren** *ist durch den zentralen Differenzenquotienten*

$$f'(x) \approx \frac{f(x+h)-f(x-h)}{2h} \tag{1.7}$$

*mit Verfahrensfehler*

$$\frac{f(x+h)-f(x-h)}{2h}-f'(x) \tag{1.8}$$

*gegeben. Wieder liefert die Taylorreihenentwicklung*

$$\begin{aligned}\frac{f(x+h)-f(x-h)}{2h}-f'(x) = & \frac{1}{2h}\Big[\Big(f(x)+hf'(x)+\frac{h^2}{2}f''(x)+\frac{h^3}{6}f'''(\theta_1)\Big)\\ & -\Big(f(x)-hf'(x)+\frac{h^2}{2}f''(x)-\frac{h^3}{6}f'''(\theta_2)\Big)\Big]-f'(x)\\ = & \frac{h^2}{12}f'''(\theta_1)+\frac{h^2}{12}f'''(\theta_2) \qquad \theta_1\in(x,x+h),\theta_2\in(x-h,x)\end{aligned} \tag{1.9}$$

*die (a-priori)* **Verfahrensfehlerschranke**

$$\left|\frac{f(x+h)-f(x-h)}{2h}-f'(x)\right| \le h^2\left(\frac{M_3}{12}+\frac{M_3}{12}\right)=h^2\frac{M_3}{6}. \tag{1.10}$$

Konvergenzuresultat: *Für $h\to 0$ geht der Verfahrensfehler wie $h^2$ gegen Null (d.h. bei Halbierung der Schrittweite geht die Fehlerschranke auf $\frac{1}{4}$ zurück).*

*Das zeigt die Überlegenheit des zentralen Differenzenquotienten für kleine $h$-Werte, wenn $f$ dreimal stetig differenzierbar ist.*

*Für hinreichend oft differenzierbares $f$ liefert eine längere Taylorreihenentwicklung in (1.9)*

$$\begin{aligned}
&\frac{f(x+h)-f(x-h)}{2h} - f'(x) \\
&\quad = \frac{1}{2h}\Big[\Big(f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{IV}(x) + \frac{h^5}{120}f^{V}(\theta_1)\Big) \\
&\qquad - \Big(f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{IV}(x) - \frac{h^5}{120}f^{V}(\theta_2)\Big)\Big] \\
&\qquad - f'(x) \\
&\quad = \frac{h^2}{6}f'''(x) + \frac{h^4}{240}f^{V}(\theta_1) + \frac{h^4}{240}f^{V}(\theta_2) \qquad (1.11)
\end{aligned}$$

*die* **Verfahrensfehlerdarstellung** *(asymptotische Entwicklung)*

$$\begin{aligned}
\frac{f(x+h)-f(x-h)}{2h} - f'(x) &= \frac{h^2}{6}f'''(x) + R \qquad (1.12)\\
\textit{mit} \quad |R| &\le h^4\frac{M_5}{120}.
\end{aligned}$$

**Zwei Anwendungen asymptotischer Entwicklungen:**

(i) Konkrete (a-posteriori) Fehlerschätzungen für den Verfahrensfehler

(ii) Konstruktion besserer Verfahren

Die numerische Differentiation ist ein einfaches und transparentes, aber durchaus typisches numerische Verfahren. Auch bei viel komplizierteren numerischen Algorithmen treten ähnliche Erscheinungen bzgl. des Verfahrensfehlers auf (*Konvergenz verschiedener Ordnungen*, *a-priori Schranken*, *a-posteriori Schätzungen*, *asymptotische Entwicklungen*, *Konstruktion genauerer Verfahren aufgrund von asymptotischen Entwicklungen des Verfahrensfehlers*, ...), diese sind nur komplizierter zu beweisen.
Der einzig untypische Punkt der Numerischen Differentiation ist, dass beim Grenzübergang $h \to 0$ (Fehler $\to 0$) keine Aufwandssteigerung des Rechenaufwandes erfolgt. Die Berechnung des Differenzenquotienten hat für jeden Wert von $h$ den selben Rechenaufwand. Das ist bei anderen Verfahren (Numerische Integration, Differentialgleichungsalgorithmen, ...) anders, da man den kleiner und kleiner werdenden Verfahrensfehler i.A. mit einer entsprechenden Rechenaufwandssteigerung erkaufen muss.

**Beispiel 1.3.4** (Numerische Integration mit der Trapezregel)**.** *Ein weiteres Beispiel für den Verfahrensfehler liefert die numerische Integration mit der* **Trapezregel** *(vgl. Abschnitt 5.2).*
*Als Näherung für $\int_a^b f(x)dx$ wird die Summe der Flächeninhalte der eingeschriebenen Trapeze herangezogen (vgl. Abb. 1.2).*

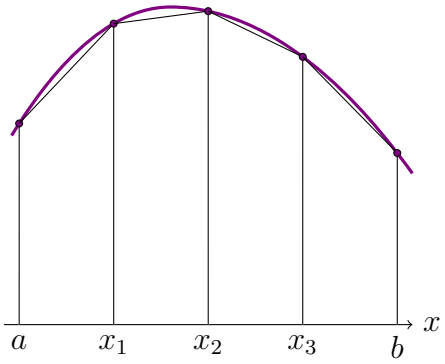



Abbildung 1.2: Trapezregel

*Die Fläche eines Trapezes ist durch* $\frac{h}{2}[f(x_i) + f(x_{i+1})]$ *gegeben (vgl. Abb. 1.3).*

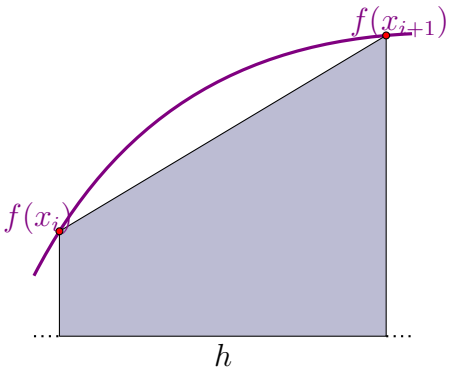



Abbildung 1.3: Fläche eines Trapezes

*Mithilfe des Näherungswertes*

$$T_h(f) = h\left[\frac{1}{2}f(a) + f(x_1) + \ldots + f(x_{n-1}) + \frac{1}{2}f(b)\right], \tag{1.13}$$

*wobei*

$$x_i = a + ih,\ i = 0, 1, \ldots, n; \quad \frac{b-a}{n} = h,\ (x_0 = a,\ x_n = b),$$

*kann der Verfahrensfehler*

$$T_h(f) - \int_a^b f(x)dx \tag{1.14}$$

*bestimmt werden. Betrachte zunächst den Fehler für ein Trapez (vgl. Abb. 1.4).*
*Es gilt (vgl. Satz 4.2.4 in Kapitel 4)*

$$f(x) - g(x) \ = \ \frac{f''(\theta(x))}{2}(x - x_i)(x - x_{i+1}),$$

*wobei* $g$ *die lineare Funktion ist, die* $f$ *im Intervall* $[x_i, x_{i+1}]$ *approximiert. Für die von* $x$ *abhängige Größe* $\theta(x)$ *gilt* $x_i \leq \theta(x) \leq x_{i+1}$*. Der Fehler für ein Trapez ist also*

$$|\int_{x_i}^{x_{i+1}} \frac{f''(\theta(x))}{2}(x - x_i)(x - x_{x+1})dx| \leq \ \frac{M_2}{2}\,|\int_{x_i}^{x_{i+1}} (x - x_i)(x - x_{i+1})dx|, \tag{1.15}$$

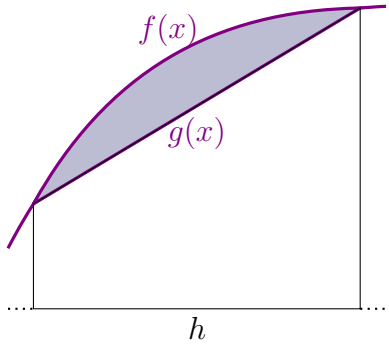



Abbildung 1.4: Fehler für ein Trapez

*wobei $M_2$ eine Schranke für $f''(x)$, $x \in [x_i, x_{i+1}]$ ist. Berechnung des Integrals liefert*

$$\begin{aligned}\int_{x_i}^{x_{i+1}}(x-x_i)(x-x_{i+1})dx &= \int_{-\frac{h}{2}}^{\frac{h}{2}}\Big(\xi+\tfrac{h}{2}\Big)\Big(\xi-\tfrac{h}{2}\Big)d\xi &= \int_{-\frac{h}{2}}^{\frac{h}{2}}\Big(\xi^2-\tfrac{h^2}{4}\Big)d\xi\\ &= \Big(\frac{\xi^3}{3}-\xi\frac{h^2}{4}\Big)\Big|_{\frac{-h}{2}}^{\frac{h}{2}} &= \Big(\frac{\frac{h^3}{8}}{3}-\frac{h^3}{8}\Big)-\Big(\frac{-\frac{h^3}{8}}{3}+\frac{h^3}{8}\Big)\\ &= -\frac{h^3}{6},\end{aligned}$$

*woraus*

$$\frac{M_2}{2}\,|\int_{x_i}^{x_{i+1}}(x-x_i)(x-x_{i+1})\,dx| \le \frac{M_2}{12}h^3$$

*folgt. Aufsummation über alle $n = \frac{b-a}{h}$ Intervalle ergibt schließlich*

$$|T_h(f) - \int_a^b f(x)dx| \le \frac{M_2}{12}\frac{b-a}{h}h^3 = \frac{M_2(b-a)}{12}h^2. \tag{1.16}$$

**Bemerkungen.**

1. Der Verfahrensfehler der Trapezregel konvergiert für $h \to 0$ wie $h^2$ gegen Null. Für kleiner werdendes $h$ hat man jedoch im Intervall mehr Gitterpunkte ($n = \frac{b-a}{h}$), also muss man mehr Funktionswerte $f(x_i)$ berechnen. Die wachsende Genauigkeit für $h \to 0$ wird also durch höheren Rechenaufwand erkauft.

2. Wie der Verfahrensfehler der numerischen Differentiation besitzt auch jener der Trapezregel eine asymptotische Entwicklung und zwar eine *Entwicklung nach geraden h-Potenzen* (wie der zentrale Differenzenquotient $D(h) = \frac{f(x+h)-f(x-h)}{2h}$). Siehe dazu auch Kapitel 5 über Numerische Quadratur (Euler-Maclaurinsche Summenformel, Romberg-Integration).

**Beispiel 1.3.5** (Numerische Integration mit der Trapezregel - Fortsetzung)**.** *Zur Illustration der Trapezregel folgt nun als konkretes Zahlenbeispiel*

$$\int_0^1 e^x dx \;=\; e-1 \;=\; 1.718281828\ldots$$

*Die Schrittweite $h = \frac{1}{4}$ liefert die Näherung*

$$\begin{aligned}T_h(f) &= h\Big[\frac{1}{2}f(0)+f(x_1)+f(x_2)+f(x_3)+\frac{1}{2}f(1)\Big]\\ &= \frac{1}{4}\Big[\frac{1}{2}e^0+e^{\frac{1}{4}}+e^{\frac{1}{2}}+e^{\frac{3}{4}}+\frac{1}{2}e^1\Big]\\ &= 1.727221905\ldots\end{aligned}$$

*mit dem Fehler* $T_h(f) - \int_0^1 e^x dx = 8.94\ldots \cdot 10^{-3}$.
*Die Schrittweite* $h = \frac{1}{8}$ *ergibt*

$$T_h(f) \;=\; 1.720518592\ldots$$

*mit dem Fehler* $T_h(f) - \int_0^1 e^x dx = 2.23\ldots \cdot 10^{-3}$. *Der Fehler reduziert sich also tatsächlich auf* $\frac{1}{4}$ *des Fehlers für* $h = \frac{1}{4}$.

## Zusammenfassung:

In den vergangenen Beispielen haben wir *a priori* Verfahrensfehlerschranken für verschiedene numerische Verfahren erhalten.

(a) Numerische Differentiation mit dem einseitigen Differenzenquotienten, vgl. (1.6):

$$\left| \frac{f(x+h) - f(x)}{h} - f'(x) \right| \;\leq\; \frac{M_2}{2} h$$

(b) Numerische Differentiation mit dem zentralen Differenzenquotienten, vgl. (1.10):

$$\left| \frac{f(x+h) - f(x-h)}{2h} - f'(x) \right| \;\leq\; \frac{M_3}{6} h^2$$

(c) Numerische Integration mit der Trapezregel, vgl. (1.16):

$$|T_h(f) - \int_a^b f(x)dx| \;\leq\; \frac{M_2(b-a)}{12} h^2$$

Außerdem wollen wir noch den Verfahrensfehler für das (explizite) **Eulerverfahren** (*Polygonzugmethode*) zur Lösung von Anfangswertproblemen

$$\begin{aligned} y'(t) &= f(t, y(t)) \\ y(0) &= y_0 \end{aligned}$$

betrachten.

(d) Numerische Lösung von Anfangswertproblemen mit dem expliziten Eulerverfahren:

$$\|\eta_\nu - y(t_\nu)\| \;\leq\; \frac{e^{Lt_\nu} - 1}{L} \frac{M_2}{2} h,$$

wobei

| | |
|---|---|
| $t \in [0, T]$ | ... Integrationsintervall |
| $y(t)$ | ... gesuchte (exakte) Lösung |
| $t_\nu = \nu h$ | |
| $\nu = 0, 1, \ldots, n;$ | ... Gitterpunkte |
| $h = \frac{T}{n}$ | ... Schrittweite |
| $\eta_\nu$ | ... numerische Näherung für $y(t_\nu)$ |
| $M_2$ | ... Schranke für $y''(t)$ für $t \in [0, T]$ |
| $L$ | ... Lipschitzkonstante von $f$ (vgl. (1.1)) |

**Bemerkungen.**

1. Die Abschätzungen (a)-(d) sind theoretische mathematische Aussagen, die für konkrete mathematische Probleme **nicht** ausgewertet werden (können).

   - In den Fällen (a) und (b) ist eine explizite Auswertung der Fehlerschranken in der Praxis gar nicht möglich, vgl. S. 11 nach (1.6).
   - Auch die rechte Seite in (d) lässt sich in konkreten Fällen nicht bestimmen, da bei unbekannter Lösung $y(t)$ eine Schranke $M_2$ für $y''$ meist nicht zur Verfügung steht.
   - Eine andere Situation liegt im Fall (c) vor: Die Differentiation ist i.A. viel einfacher als die Bestimmung von Stammfunktionen und auch in Fällen formelmanipulativ möglich, in denen sich die Stammfunktion nicht geschlossen darstellen lässt, man also zu numerischer Integration gezwungen ist. Es ist daher möglich, den Integranden $f$ zweimal zu differenzieren und $M_2$ und damit die a-priori Schranke als Ausdruck in $h$ zu ermitteln, das Integral selbst aber durch numerische Integration mit der Trapezregel näherungsweise zu berechnen. In der Praxis geschieht dies dennoch eher selten.

2. Die Bedeutung der a-priori Schranken liegt in der mathematischen Rechtfertigung der betrachteten Näherungsverfahren: die Konstruktion von numerischen Methoden beruht zunächst nur auf heuristischen Überlegungen, erst a-priori Schranken weisen die entsprechenden Verfahren als vernünftig aus.

3. Die in jeder Verfahrensfehlerschätzung auftretenden Faktoren $h^p$ (z.B. $h$ für $p = 1$) stellen die *Konvergenz* der Verfahren sicher (vgl. *Konvergenzresultat* auf S. 11). Für $h \to 0$ geht die Fehlerschranke (und damit auch der tatsächliche Verfahrensfehler) wie $h^p$ gegen Null.

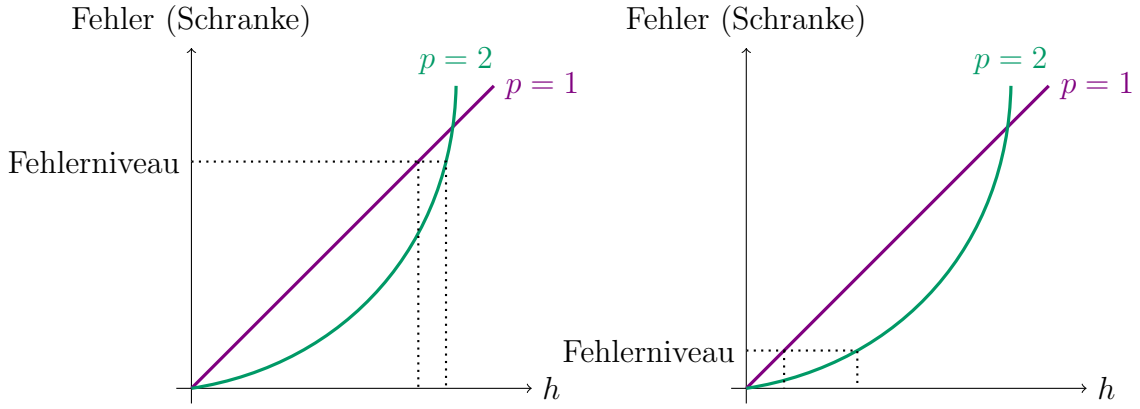
   **Definition 1.3.6.** *Den Exponenten $p$ bezeichnet man als die Ordnung des Verfahrens.*

   Die a-priori Schranke stellt sicher, dass man durch entsprechende Wahl der Schrittweite $h$ den Fehler beliebig klein machen kann und dass mit $h$ der Fehler umso rascher gegen Null geht, je höher die Ordnung des Verfahrens ist. Von den obigen Beispielen haben (a) und (d) Ordnung 1 und (b) und (c) Ordnung 2.

   Auch für die Software-Entwicklung sind Effizienzbetrachtungen mittels a priori Schranken wichtig, siehe auch folgendes Beispiel.

   **Beispiel 1.3.7.** *Wir vergleichen zwei Verfahren der Ordnungen 1 und 2 unter der Annahme, dass beim Verfahren 2. Ordnung jeder Schritt einen doppelt so hohen Aufwand hat wie beim Verfahren 1. Ordnung.*
   *So eine Situation wäre z.B. gegeben, wenn man für das Lösen von Anfangswertproblemen gewöhnlicher Differentialgleichungen das Eulerverfahren ($p = 1$) und die* Methode von Heun *($p = 2$) vergleicht, siehe auch Kapitel* 6.

   *Die Abbildungen 1.5a und 1.5b zeigen, dass sich das Verfahren der Ordnung* 1 *als effizienter erweist, wenn ein hohes Fehlerniveau ausreicht, da die Ersparnis bei der Anzahl der Schritte gegenüber dem höheren Aufwand nicht ins Gewicht fällt. (Bei dem Verfahren mit $p = 2$ hat man etwa* 80% *der Schritte im Vergleich mit $p = 1$, aber pro Schritt doppelt hohen Aufwand!*
   *Ist jedoch hohe Genauigkeit (niedriges Fehlernivau) gefordert, so ist das Verfahren der Ordnung* 2 *viel effizienter (etwa nur* 10% *der Schritte verglichen mit $p = 1$, aber pro Schritt nur der doppelte Aufwand!)*




(a) Hohes Fehlerniveau wird akzeptiert (b) Niedriges Fehlerniveau wird gefordert

Abbildung 1.5: Vergleich der Fehlerreduktion für $h \to 0$ von Verfahren 1. und 2. Ordnung

Beispiel 1.3.7 illustriert folgende Faustregel:
**Verfahren hoher Ordnung lohnen sich nur bei strengen Genauigkeitsforderungen!**

4. Wenn auch für sehr viele Näherungsverfahren a-priori Schranken bekannt sind, gibt es noch zahlreiche mathematische Problemklassen und zugehörige Näherungsverfahren, für die es bisher noch nicht gelungen ist, solche Schranken zu beweisen.
   Ein Beispiel liefert der Bereich der Anfangswertprobleme gewöhnlicher Differentialgleichungen: Für große Klassen nichtlinearer steifer Differentialgleichungen und numerischer Näherungsverfahren, von denen man aus der Praxis weiß, dass sie effizient funktionieren, fehlen entsprechende Konvergenzresultate.

   Für gewisse Klassen mathematischer Probleme sind nicht einmal effiziente numerische Näherungsverfahren bekannt. Ein Beispiel dafür liefert der Bereich der partiellen Differentialgleichungen: Für eine bestimmte Klasse von Anfangs-Randwertprobleme würde erst eine Gitterfeinheit, die aus Gründen der Rechenzeit nicht vertretbar ist, zu korrekten Lösungen führen. In der Praxis ergeben sich diese Probleme bei der Modellierung von Klopfvorgängen in den Zylindern von Verbrennungsmotoren.

5. Typischerweise steigt für kleiner werdende Schrittweiten $h$ der Aufwand der numerischen Näherungsverfahren, es liegt also i.A. eine Situation wie bei (c) und (d) vor.
   Die Fälle (a) und (b) sind extrem untypische Ausnahmen dieser Regel, da bei der Berechnung von Differenzenquotienten die Anzahl der Funktionsauswertungen unabhängig von $h$ ist, vgl. die Bemerkung auf S. 12). Das Verfahren zweiter Ordnung, (b), scheint gegenüber jenem erster Ordnung, (a), keinen Vorteil aufzuweisen, da man mit beiden Verfahren durch entsprechende Wahl der Schrittweite $h$ einen beliebig kleinen Verfahrensfehler erreichen kann und darüber hinaus bei beiden Verfahren 2 Funktionsauswertungen ($f(x+h)$ und $f(x)$ bzw $f(x+h)$ und $f(x-h)$) benötigt. Kalkuliert man jedoch Rundungsfehler ein, erweist sich (b) als überlegen: Für das gleiche Genauigkeitsniveau ist beim Verfahren 2. Ordnung eine größere Schrittweite ausreichend, welche weniger problematisch ist im Bezug auf Rundungsfehler, vgl. Abschnitt 1.4.

## 1.4 Rechen- bzw. Rundungsfehler

Oft ergeben sich Probleme bei der numerischen Berechnung für kleine h-Werte. Betrachte etwa den zentralen Differenzenquotienten der Funktion $f(x) = e^x$ für $h = 10^{-4}$, gegeben durch

$$\frac{f(x+h) - f(x-h)}{2h} = \frac{e^{1+10^{-4}} - e^{1-10^{-4}}}{2 \cdot 10^{-4}}.$$

Der kritische Punkt ist hier die Differenzbildung $e^{1+10^{-4}} - e^{1-10^{-4}}$. Auf einem 10-stelligen Taschenrechner ergibt sich etwa:

| | | | |
|---|---|---|---|
| $e^{1+10^{-4}}$ | = | 2.718553670 | $\ldots *)$ |
| $e^{1-10^{-4}}$ | = | 2.718010014 | $\ldots *)$ |
| $e^{1+10^{-4}} - e^{1-10^{-4}}$ | = | 0.000543656 | $\ldots *)$ |

$*)$ in der 10-stelligen Arithmetik verlorene Stellen

Aufgrund der verlorenen Stellen von $e^{1+10^{-4}}$ und $e^{1-10^{-4}}$ ist das Ergebnis der Substraktion nur mehr 6-stellig! Die letzten 4 Stellen, die man für eine 10-stellige Arithmetik benötigen würde, sind verloren! Dieses Phänomen heißt **Auslöschung** und tritt bei der Differenzbildung von annähernd gleichen Zahlen auf.

**Bemerkung.** Bei der Differenzbildung von zwei etwa gleich großen *exakten* Zahlen wird kein neuer Rundungsfehler generiert. Betrachte dafür die folgenden exakten Zahlen in fünfstelliger Arithmetik:

$$\begin{array}{r} 3.6725 \\ -3.6723 \\ \hline 0.0002 \end{array} \qquad (\square)$$

Nur wenn die beiden Zahlen selbst fehlerhaft sind (z.B. von 12 auf 5 Stellen gerundet), so wirkt sich dies auf das Ergebnis katastrophal aus. Nehmen wir etwa in obigen Beispiel an, dass die Zahlen nur eine Rundung der folgenden exakten Zahlen in 12-stelliger Arithmetik sind:

$$\begin{array}{r} 3.67253748913 \\ -3.67231866741 \\ \hline 0.00021882172 \end{array} \qquad (\diamond)$$

Man sieht, dass bei $(\square)$, wo vor der Differenzbildung gerundet wurde, nur noch eine (!) Stelle vorhanden ist , während $(\diamond)$ mehr Information enthält. Dieses Problem wird tragend, wenn ich 12-stelliger Arithmetik weiter gerechnet wird.

Die Differenzbildung etwa gleicher Zahlen ist also harmlos, wenn man exakte Größen (die in der gegebenen Arithmetik exakt dargestellt werden können) zur Verfügung hat. Hat man aber (z.B. durch Rundung) verfälschte Zahlen, ergibt sich aufgrund von Auslöschung ein erheblich Genauigkeitsverlust. Es wird zwar bei der Differenzbildung kein neuer Fehler generiert, aber die Fehlerfortpflanzung (bzgl. der vorhergehenden Rundung der beiden Operanden) ist i.A. katastrophal, wobei nicht das absolute, sondern das relative Fehlerniveau betroffen ist.

Um Rundungsfehler systematisch untersuchen und abschätzen zu können, benötigen wir zunächst einige Informationen über **Computerarithmetik**.

### 1.4.1 Computerarithmetik

Zunächst werden einige Grundbegriffe erklärt.

- Die **Basis** der Zahlendarstellung beim menschlichen Rechnen ist 10. Die Stelle, an der eine Ziffer steht, gibt daher an, mit welcher 10-er Potenz die Ziffer zu multiplizieren ist, [1)]. betrachte etwa

$$365.22 = 3 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0 + 2 \cdot 10^{-1} + 2 \cdot 10^{-2}.$$

  Im Computer werden i.A. andere Basisgrößen als 10 verwendet, meist 2, 8, 16:

  *Binärarithmetik* (Basis 2): Mögliche Ziffern sind 0, 1, betrachte etwa

$$\underset{\substack{\uparrow \\ \text{Binärpunkt}}}{11001.11} = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = \underset{\substack{\uparrow \\ \text{Dezimalpunkt}}}{25.75}.$$

  *Oktalarithmetik* (Basis 8): Mögliche Ziffern sind 0, 1, 2, 3, 4, 5, 6, 7.

  *Hexadezimalarithmetik* (Basis 16): Mögliche Ziffern sind 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, $a$, $b$, $c$, $d$, $e$, $f$ (die Buchstaben $a$ - $f$ stehen für die Zahlen 10 - 15).

- Die **Gleitkommadarstellung** bezeichnet die Schreibweise einer Zahl mithilfe von Exponenten der Basis, z.B.

$$365.22 = 3.6522 \cdot 10^2 = \underset{\substack{\uparrow \\ \textit{normalisierte} \text{ Gleitkommadarstellung}}}{0.36522 \cdot 10^3}.$$

- Bei der **normalisierten Gleitkommadarstellung** gilt die Konvention, die führende (nicht verschwindende) Stelle unmittelbar hinter den Dezimalpunkt (Binärpunkt, ...) zu schreiben.[2)] Da die Zahl 0 keine nichtverschwindenden Stellen hat, nimmt sie eine Sonderdarstellung ein.

- Mithilfe der normalisierten Gleitkommadarstellung kann man eine Zahl (bei bekannter Basis) durch **Mantisse** und **Exponent** angeben. Die Dezimalzahl

$$-0.0027653 = -0.27653 \cdot 10^{-2}$$

  wird z.B. codiert als

| Vorzeichen d. Mantisse | Mantisse | Vorzeichen d. Exponenten | Exponent |
|---|---|---|---|
| − | 27653 | − | 2 |

[1)] Diese sehr vorteilhafte und intelligente Art, Zahlen zu codieren, erscheint uns heute ganz selbstverständlich; trotzdem waren früher ungeeignetere Systeme zur Codierung von Zahlen in Verwendung.
Die Römer schrieben z.B. 365 in der Form CCCLXV. Das System der römischen Zahlen unterscheidet sich von unserem System nicht nur durch die Ziffernsymbolik, sondern ganz grundsätzlich. Um den Vorteil unseres Systems zu erkennen, überlege man etwa, wie kompliziert bei den Römern Multiplikationsalgorithmen zu formulieren wären.
Außerdem erscheint die Basis 10 für das menschliche Rechnen ein idealer Kompromiss. Binärarithmetik wäre für den Menschen ungeeignet, da die Zahlen lang und unübersichtlich wären, wenn auch das kleine Einmaleins bei der Binärarithmetik sehr einfach zu lernen wäre: $0 \times 0 = 0$, $0 \times 1 = 0$, $1 \times 0 = 0$, $1 \times 1 = 1$

[2)] Heutzutage versteht man unter Normalisierung oft auch die Konvention, den Dezimalpunkt unmittelbar hinter die führende Stelle zu schreiben, also z.B. $-2.7653 \cdot 10^{-3}$ statt $-0.27653 \cdot 10^{-2}$.

Die tatsächliche interne Zahlendarstellung ist bei verschiedenen Rechnern unterschiedlich realisiert. Häufig gelingt es, das Vorzeichen des Exponenten zu ersparen. Bei einer Binärarithmetik wäre es redundant, die führende Stelle, die 1 sein muss, zu codieren, was daher meist unterbleibt (*verstecktes Bit*).

Unter Verwendung obiger Begriffe kann nun eine Computerarithmetik definiert werden.
Eine **Computerarithmetik** (Maschinenarithmetik) ist durch die Parameter

| | | |
|---|---|---|
| $b$ | ... | Basis |
| $l$ | ... | Anzahl der Mantissenstellen |
| $e_1$ | ... | kleinster Exponent |
| $e_2$ | ... | größter Exponent |

eindeutig festgelegt.
Die Menge der **Maschinenzahlen** zu einer festen Computerarithmetik wird mit

$$\mathbb{M}(b, l, e_1, e_2)$$

bezeichnet. Dabei betrachtet man *normalisierte Gleitkommazahlen* zur Basis $b$ mit $l$ Mantissenstellen und einem Exponenten $E$ mit $e_1 \leq E \leq e_2$. Auf jedem Computer stehen *nur endlich viele Maschinenzahlen* zur Verfügung.

**Beispiel 1.4.1** (Maschinenarithmetik)**.** *Die gegebene Maschinenarithmetik sei* $\mathbb{M}(2,\ 3,\ -3,\ +3)$*, die zugehörigen Mantissen und Exponenten sind in folgender Tabelle aufgelistet.*

| *verfügbare Mantissen* | | | *verfügbare Exponenten* |
|---|---|---|---|
| 100 | = | $1 \cdot \frac{1}{2}$ | $-3, -2, -1, 0, 1, 2, 3$ |
| 101 | = | $1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{8}$ | |
| 110 | = | $1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4}$ | |
| 111 | = | $1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + 1 \cdot \frac{1}{8}$ | |

*Das Paar* $(-101\,, -2)$ *kann z.B. als die Dezimalzahl*

$$\underbrace{-0.625}_{-\frac{5}{8}} \cdot \underbrace{2^{-2}}_{\frac{1}{4}} \;=\; -\frac{5}{32} \;=\; -0.15625$$

*angeschrieben werden. Zur Darstellung aller positven Maschinezahlen dieses Beispiels siehe Abb. 1.6.*

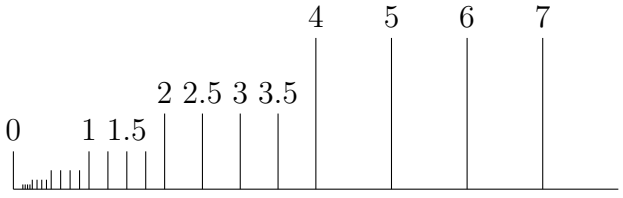



Abbildung 1.6: Positive Maschinenzahlen

*Die größtmögliche Maschinenzahl dieser Arithmetik ist*

$$(+111\,, +3), \quad \textit{das entspricht der Dezimalzahl} \quad \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8}\right) \cdot 2^3 \;=\; 7.$$

*Die kleinstmögliche positive Maschinenzahl dieser Arithmetik ist*

$$(+100\,, -3), \quad \textit{das entspricht der Dezimalzahl} \quad \frac{1}{2} \cdot 2^{-3} \;=\; 2^{-4}.$$

*Für eine Vergrößerung der Umgebung der Null siehe Abb. 1.7.*

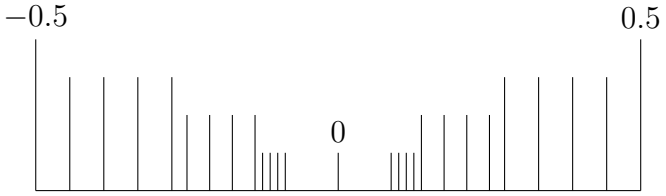



Abbildung 1.7: Umgebung der Null

*In unmittelbarer Umgebung der Null ergibt sich ein „Loch“, weil nur normalisierte Zahlen zugelassen sind. Das Ausfüllen dieses Lochs wäre möglich, wenn subnormale Zahlen zugelassen würden, also Mantissen, deren führende Stelle* 0 *sein darf. Das Paar* $(+001, -3)$ *entspricht z.B* $2^{-3} \cdot 2^{-3} = 2^{-6}$*, ist also schon viel näher bei* 0 *als die kleinstmögliche normalisierte Zahl. Heutzutage gibt es in vielen Standardarithmetiken (z.B. im IEEE-Standard) subnormale Zahlen.*

**Übliche Computerarithmetiken:**

- **Typische Taschenrechnerarithmetik:** $\mathbb{M}(10, 10, -98, +100)$, [3] also Basis 10, Mantissenlänge 10 und Exponentialbereich von $-98$ bis $+100$.
- **Typische IBM Anlagen:** $\mathbb{M}(16, 6, -64, +63)$, also Basis 16, Mantissenlänge 6 und Exponentialbereich von $-64$ bis $+63$.
- **IEEE Standard:** $\mathbb{M}(2, 24, -125, +128)$, also Basis 2, Mantissenlänge 24 und Exponentialbereich von $-125$ bis $+128$ (sehr oft auf PC's verwendet).

Um im Folgenden Rundungsfehleranalysen durchzuführen, muss noch klar sein, ob vom Rechner *gerundet* oder *abgeschnitten* wird.

- **Runden einer Zahl:** Es wird die nächstgelegene Maschinenzahl herangezogen. Liegt eine Zahl genau zwischen zwei Maschinenzahlen, so lautet die übliche Konvention, dass zur betragsgrößeren Zahl gerundet wird. Oft wird in diesem Fall aber zur nächstgelegenen geraden Maschinenzahl gerundet (*round to even*).
- **Abschneiden:** Es wird die betragsmäßig nächstkleinere Maschinenzahl herangezogen.

| | | |
|---|---|---|
| Betrachte als Beispiel $\mathbb{M}(10, 8 \ldots)$. | Runden: | $987.562438 \to 987.56244$ |
| | Abschneiden: | $987.562438 \to 987.56243$ |

Die Definition von **Rundungs-** bzw. **Abschneidefehler** bei einem einzigen Rundungs- (Abschneide-) vorgang zeigt folgende Tabelle.

| exakte Zahl | gerundete (abgeschn.) Zahl aus $\mathbb{M}$ | *absoluter Fehler* | *relativer Fehler* |
|---|---|---|---|
| $x$ | $\tilde{x}$ | $\tilde{x} - x$ | $\frac{\tilde{x}-x}{x}$ |

**Satz 1.4.2.** *Es gilt*

*a) Der* maximale absolute Abschneidefehler *ist der Abstand von zwei benachbarten Maschinenzahlen in der Umgebung von x (vgl. Abb. 1.8).*

[3] Tatsächlich werden am Taschenrechner 10-stellige Dezimalzahlen typischerweise so dargestellt, dass – im Gegensatz zu normalisierten Zahlen – die führende Stelle vor dem Komma steht, der 2-stellige Exponent läuft von $-99$ bis $+99$. Da aber $\mathbb{M}$ immer das Symbol für normalisierte Maschinenzahlen ist – mit führender Stelle hinter dem Komma – muss in der Schreibweise $\mathbb{M}(10, 10, -98, +100)$ zum Ausgleich der Exponent von $-98$ bis $+100$ laufen.

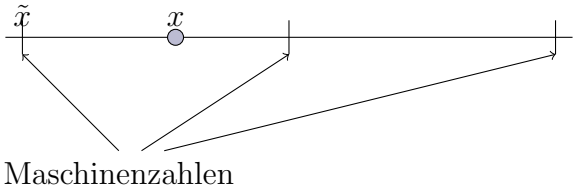



Abbildung 1.8: Abschneidefehler

*b) Der* maximale absolute Rundungsfehler *ist der halbe Abstand von zwei benachbarten Maschinenzahlen in der Umgebung von x (vgl. Abb. 1.9).*

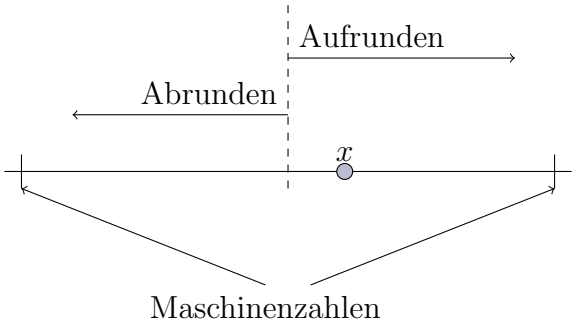



Abbildung 1.9: Rundungsfehler

**Beweis.** Siehe Abbildungen 1.8 und 1.9. □

Der Abstand von zwei benachbarten normalisierten Maschinenzahlen ist durch

$$\begin{array}{rccccc} & (d_1 \cdot b^{-1} & + \cdots + & d_l \cdot b^{-l}) \cdot b^e & & \\ - & (d_1 \cdot b^{-1} & + \cdots + & (d_l - 1) \cdot b^{-l}) \cdot b^e & = & b^{-l} \cdot b^e \end{array}$$

gegeben.
Es folgt für $x = (d_1 \cdot b^{-1} + \ldots) \cdot b^e$ für den absoluten Abschneidefehler

$$|\tilde{x} - x| \leq b^{-l} \cdot b^e$$

sowie für den absoluten Rundungsfehler

$$|\tilde{x} - x| \leq \frac{1}{2} b^{-l} \cdot b^e.$$

**Schranken für relative Fehler.** Da stets $|x| = |d_1 b^{-1} + d_2 b^{-2} + \ldots| \cdot b^e$ gilt, ist $1 \cdot b^{-1} \cdot b^e$ eine untere Schranke für $|x|$ und damit

$$\left| \frac{\tilde{x} - x}{x} \right| = \frac{|\tilde{x} - x|}{|x|} \leq \frac{\text{Schranke f. abs. Fehler}}{1 \cdot b^{-1} \cdot b^e}.$$

Es folgt für $x = (d_1 \cdot b^{-1} + \ldots) \cdot b^e$ für den relativen Abschneidefehler

$$\left| \frac{\tilde{x} - x}{x} \right| < \frac{b^{-l} \cdot b^e}{b^{-1} \cdot b^e} = b \cdot b^{-l} \tag{1.17}$$

sowie für den relativen Rundungsfehler

$$\left|\frac{\tilde{x}-x}{x}\right| \;\leq\; \frac{1}{2}\cdot b\cdot b^{-l}. \tag{1.18}$$

Eine triviale, aber wichtige Identität für den relativen Fehler $\epsilon$ ist

$$\tilde{x} \;=\; x(1+\epsilon) \tag{1.19}$$

Das folgt unmittelbar aus der Definition von $\epsilon := \frac{\tilde{x}-x}{x} \Rightarrow \tilde{x}-x = \epsilon x \Rightarrow \tilde{x} = x(1+\epsilon)$.

**Zusammenfassung:** Wenn $x$ eine exakte Zahl ist und $\tilde{x}$ die durch Abschneiden oder Runden in die Arithmetik $\mathbb{M}(b,l,\ldots)$ entstehende Größe ist, so gilt

$$\boxed{\begin{array}{rrll} & \tilde{x} & = & x(1+\epsilon) \\ \text{mit} & |\epsilon| & < & b\cdot b^{-l} \qquad \text{(Abschneiden)} \\ \text{bzw.} & |\epsilon| & \leq & \frac{1}{2}\cdot b\cdot b^{-l}. \qquad \text{(Runden)} \end{array}} \tag{1.20}$$

Die Identität (1.20) ermöglicht nun in bequemer Weise Rundungsfehlerabschätzungen. Der entscheidende Gedanke ist, dass bei jedem Rundungsvorgang eines Algorithmus (1.20) eingesetzt wird.

Das folgende Beispiel demonstriert dies für den zentralen Differenzenquotienten.

**Beispiel 1.4.3** (Auslöschung)**.** *Berechnung des zentralen Differenzenquotienten*

$$D(h=10^{-4}) \;=\; \frac{e^{1+10^{-4}} - e^{1-10^{-4}}}{2\cdot 10^{-4}} \tag{1.21}$$

*auf einem Taschenrechner mit der Arithmetik* $\mathbb{M}(10,\,10,\,-98,\,+100)$ *und Abschneiden, d.h. der relative Abschneidefehler* $\epsilon$ *bei einem Abschneidevorgang erfüllt wegen (1.20) die Relation*

$$|\epsilon| \;<\; 10\cdot 10^{-10} = 10^{-9}. \tag{1.22}$$

*Die Größen* $1+10^{-4} = 1.0001$ *und* $1-10^{-4} = 0.9999$ *sind in unserer Arithmetik exakt dargestellt. Ebenso der Nenner* $2\cdot 10^{-4}$ *von (1.21). Anstelle der Größen*

$$e^{1+10^{-4}} \;=\; \exp(1+10^{-4}) \quad \textit{und} \quad e^{1-10^{-4}} \;=\; \exp(1-10^{-4})$$

*entstehen im Rechner Näherungsausdrücke*

$$\widetilde{\exp}(1+10^{-4}) \quad \textit{und} \quad \widetilde{\exp}(1-10^{-4}),$$

*die aufgrund von (1.20) den folgenden Relationen genügen:*

$$\begin{array}{rcl} \widetilde{\exp}(1+10^{-4}) & = & \exp(1+10^{-4})\cdot(1+\epsilon_1), \\ \widetilde{\exp}(1-10^{-4}) & = & \exp(1-10^{-4})\cdot(1+\epsilon_2), \end{array} \tag{1.23}$$

*wobei wegen (1.22) gilt*

$$|\epsilon_1| \;<\; 10^{-9} \;\textit{und}\; |\epsilon_2| \;<\; 10^{-9}.$$

*Dabei wird angenommen, dass die Standardfunktion* exp *am Rechner von so guter Qualität ist, dass bei der Auswertung der Exponentialfunktion maximal ein elementarer Rundungsfehler wie bei einer Abschneideoperation gemacht wird, d.h. dass stets gilt:*

$$\widetilde{\exp}(\textit{Maschinenzahl}) = \exp(\textit{Maschinenzahl})(1+\epsilon) \quad \textit{mit} \quad |\epsilon| < 10^{-9}.$$

*Bei der Berechnung des Zählers von (1.21) wird die Differenz der beiden verfälschten Größen* $\widetilde{\exp}(1+10^{-4})$ *und* $\widetilde{\exp}(1-10^{-4})$ *exakt gebildet, d.h. bei dieser Differenzbildung wird kein neuer Abschneidefehler generiert.*[4] *Im Taschenrechner entsteht also als Zähler des Quotienten (1.21) die Größe*

$$\widetilde{\exp}(1+10^{-4}) - \widetilde{\exp}(1-10^{-4}) = \exp(1+10^{-4})\cdot(1+\epsilon_1) - \exp(1-10^{-4})\cdot(1+\epsilon_2). \tag{1.24}$$

*Bei der Division wird allerdings wieder ein Abschneidefehler gemacht*[5]*, es entsteht also im Taschenrechner der rechenfehlerbehaftete Quotient*

$$\tilde{D}(h=10^{-4}) = \frac{e^{1+10^{-4}}(1+\epsilon_1) - e^{1-10^{-4}}(1+\epsilon_2)}{2\cdot 10^{-4}}\cdot(1+\epsilon_3) \tag{1.25}$$

*wobei* $\epsilon_3$ *der relative Abschneidefehler der Division ist und und natürlich auch der Beziehung* $|\epsilon_3| < 10^{-9}$ *genügt. Bei der nun folgenden Umformung von* $\tilde{D}(h)$ *werden Produkte* $\epsilon_i \cdot \epsilon_k$ *vernachlässigt, da die* $\epsilon_i$ *in der Größenordnung* $10^{-9}$ *sind und diese Produkte daher in der Größenordnung* $10^{-18}$*.*

$$\begin{aligned}\tilde{D}(h=10^{-4}) &= \frac{e^{1+10^{-4}}(1+\epsilon_1)(1+\epsilon_3) - e^{1-10^{-4}}(1+\epsilon_2)(1+\epsilon_3)}{2\cdot 10^{-4}} \approx \\ &\approx \underbrace{\frac{e^{1+10^{-4}} - e^{1-10^{-4}}}{2\cdot 10^{-4}}}_{D(h=10^{-4})\ \textit{exakt}} + \underbrace{\frac{e^{1+10^{-4}}(\epsilon_1+\epsilon_3) - e^{1-10^{-4}}(\epsilon_2+\epsilon_3)}{2\cdot 10^{-4}}}_{\textit{Abschneidefehler}}\end{aligned} \tag{1.26}$$

*woraus sich die Fehlerabschätzung*[6]

$$|\tilde{D}(h=10^{-4}) - D(h=10^{-4})| < \frac{\exp(1+10^{-4})\cdot 4\cdot 10^{-9}}{2\cdot 10^{-4}} < 2.72\cdot 2\cdot 10^{-5} \tag{1.27}$$

*ergibt. Ganz analog hätte man für* $h = 10^{-9}$ *erhalten:*

$$|\tilde{D}(h=10^{-9}) - D(h=10^{-9})| < \frac{\exp(1+10^{-9})\cdot 4\cdot 10^{-9}}{2\cdot 10^{-9}} < 2.72\cdot 2 \tag{1.28}$$

*was noch deutlicher zeigt, wie die Fehlerschranke für den Abschneidefehler für* $h \to 0$ *explodiert.*

*Betrachtet man den Verfahrensfehler (konvergiert wie* $h^2$ *gegen Null) und den Rechenfehler gemeinsam, ergibt sich etwa das Bild aus Abb. 1.10.*

Diese Technik der Rundungsfehleranalyse (bei jedem Rundungs - oder Abschneideprozess einen $(1+\epsilon)$-Faktor anzuhängen) kann im Prinzip auf verschiedenste Algorithmen angewendet werden. Für umfangreiche Algorithmen mit vielen Rechenoperationen wird sie jedoch schnell unübersichtlich und

[4] Vgl. Seite 18 (□): Bei der Differenz von zwei Zahlen aus demselben Exponentialbereich wird kein neuer Runde- oder Abschneidefehler generiert.

[5] Nur wenn zufällig die letzte Stelle des Zählers gerade wäre, wäre die Division durch 2 exakt.

[6] Eine schärfere Abschätzung wäre möglich, wenn man $e^{1+10^{-4}}\epsilon_3 - e^{1-10^{-4}}\epsilon_3$ nicht durch $2e^{1+10^{-4}}\epsilon_3$ abschätzen würde.

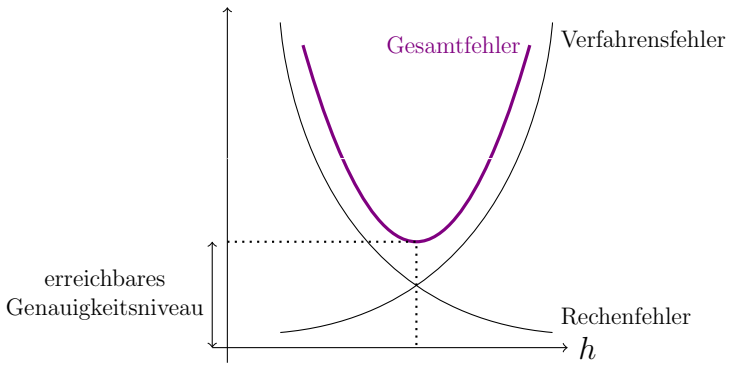



Abbildung 1.10: Gesamtfehler bei numerischer Differentiation

führt nur in Ausnahmefällen zum Erfolg.
Außerdem sind die sich ergebenden Schranken für umfangreiche Algorithmen in der Praxis meist zu pessimistisch. Die garantierte Schranke, die für alle denkbaren Datenkonstellationen des Algorithmus stimmen muss, muss – um diese Sicherheit zu bekommen – immer vom ungünstigsten Fall ausgehen, z.B. dem, dass sich alle einzelnen Rundungs- oder Abschneidefehler aufaddieren (akkumulieren), während in den meisten Fällen eine gewisse Kompensation und Auslöschung der einzelnen Rundungsfehler eintreten wird. In der Praxis bevorzugt man bei umfangreichen Algorithmen daher andere Rundungsfehleranalysetechniken.

### 1.4.2 Rundungsfehleranalysetechniken

In diesem Abschnitt soll ein kurzer Überblick über verschiedene Rundungsfehleranalysetechniken gegeben werden.

- **Statistische Schätzungen** der Rundungsfehler
- **A-posteriori (Ab)schätzungen,** die zwar nicht den Charakter einer mathematischen Aussage für alle denkbaren Anwendungsfälle des untersuchten Algorithmus haben, aber dafür in konkreten Fällen unter Einbeziehung des rundungsfehlerbehaften Resultates zu einer a-posteriori Schätzung des Rundungfehlers führen.
- **Experimentielle Rundungsfehlerschätzungen:** Durch künstliche (mit Zufallsgenerator erzeugte) Störungen bei den einzelnen Rechenschritten eines Algorithmus wird die Rundungsfehlersensitivität des Algorithmus bezüglich der gegebenen Problemdaten experimentiell erprobt.
- **(1+$\epsilon$)-Technik:** Die oben (Beispiel 1.4.3) beschriebene (1+$\epsilon$)-Technik wird i.A. nur zur Analyse von kleineren Programmstücken eingesetzt. Beispielsweise soll innerhalb eines Algorithmus $\sqrt{x+1}-\sqrt{x}$ berechnet werden, wobei $x$ ein Zwischenergebnis ist, das sich aufgrund von früheren Manipulationen ergibt. Wegen

$$\sqrt{x+1}-\sqrt{x}=\frac{(\sqrt{x+1}-\sqrt{x})(\sqrt{x+1}+\sqrt{x})}{\sqrt{x+1}+\sqrt{x}}=\frac{1}{\sqrt{x+1}+\sqrt{x}}$$

hat man die Wahl, ob man direkt $\sqrt{x+1}-\sqrt{x}$ oder statt dessen $\frac{1}{\sqrt{x+1}+\sqrt{x}}$ berechnet. Im Allgemeinen ist der zweite Weg wegen der zusätzlichen Division ungünstiger, für $x \gg 0$ ist er jedoch wegen der Auslöschung bei $\sqrt{x+1}-\sqrt{x}$ weit überlegen.

# Kapitel 2

# Numerische Lösung linearer Gleichungssysteme

## 2.1 Grundlagen aus der linearen Algebra (Wiederholung)

**Definition 2.1.1.** *Eine $m \times n$ -* **Matrix** $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \in \mathbb{R}^{m \times n}$ *hat $m$ Zeilen, $n$ Spalten.*

**Definition 2.1.2.** *Ein* **Vektor** $\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$ *ist ein $n$-Tupel reeller Zahlen. Meistens werden Spaltenvektoren (d.h. $n \times 1$ - Matrizen) betrachtet, Zeilenvektoren sind $1 \times n$ - Matrizen.*

**Definition 2.1.3.** *$A\vec{x}$ ist die algebraische Schreibweise einer* **linearen Abbildung** $\mathbb{R}^n \to \mathbb{R}^m$*:*

$$\underbrace{\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}}_{\vec{x}} = \underbrace{\begin{pmatrix} a_{11}x_1 & + & \cdots & + & a_{1n}x_n \\ \vdots & & & & \vdots \\ a_{m1}x_1 & + & \cdots & + & a_{mn}x_n \end{pmatrix}}_{Vektor \in \mathbb{R}^m} \tag{2.1}$$

$A\vec{x}$ ist tatsächlich eine lineare Abbildung, denn die Linearitätsaxiome sind erfüllt:

$$\begin{aligned} A(\vec{x}_1 + \vec{x}_2) &= A\vec{x}_1 + A\vec{x}_2 \qquad \forall \vec{x}_1, \vec{x}_2 \in \mathbb{R}^n \\ A(\lambda\vec{x}) &= \lambda A\vec{x} \qquad \lambda \in \mathbb{R} \end{aligned} \tag{2.2}$$

Bedeutung der Spalten von A: Wir betrachten die Standardbasis des $\mathbb{R}^n$ (*kanonische Basis*):

$$\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \vec{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \vec{e}_i = \begin{pmatrix} \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix} \leftarrow i, \ldots, \vec{e}_n = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 1 \end{pmatrix}, \tag{2.3}$$

dann ist die i-te Spalte ($i = 1(1)n$) von $A$ das Bild von $\vec{e}_i$, wegen

$$A\vec{e}_i = \begin{pmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1n} \\ \vdots & & & & \vdots \\ a_{m1} & \cdots & a_{mi} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} a_{1i} \\ \vdots \\ a_{mi} \end{pmatrix} \tag{2.4}$$

Umkehrung von (2.17):

**Satz 2.1.4.** *Jede lineare Abb. $\phi : \mathbb{R}^n \to \mathbb{R}^m$ lässt sich als $\phi(\vec{x}) = A\vec{x}$ schreiben, wobei die Spalten von $A$ die Bilder $\phi(\vec{e}_i)$ der Basisvektoren $\vec{e}_i$, $i = 1, 2, ..., n$ sind.*

**Beweis.**

$$\vec{x} \text{ beliebig} \in \mathbb{R}^n \qquad \vec{x} = \sum_{i=1}^{n} x_i \vec{e}_i$$

$$\phi(\vec{x}) = \phi(\sum_{i=1}^{n} x_i \vec{e}_i) = \sum_{i=1}^{n} x_i \phi(\vec{e}_i) = \sum_{i=1}^{n} x_i \begin{pmatrix} a_{1i} \\ \vdots \\ a_{mi} \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11}x_1 & + & \cdots & + & a_{1n}x_n \\ \vdots & & & & \vdots \\ a_{m1}x_1 & + & \cdots & + & a_{mn}x_n \end{pmatrix} = Ax \tag{2.5}$$

□

**Definition 2.1.5.** *Die transponierte Matrix zu*

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \in \mathbb{R}^{m\times n} \quad \textit{ist} \quad A^\top = \begin{pmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & & \vdots \\ a_{1n} & \cdots & a_{mn} \end{pmatrix} \in \mathbb{R}^{n\times m}.$$

Ein Spaltenvektor $\vec{x} \in \mathbb{R}^n$ kann durch $\vec{x}^\top = (x_1, x_2, ..., x_n)$ als Zeilenvektor geschrieben werden, d.h. aus der einspaltigen Matrix $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ wird durch Transposition die einzeilige Matrix $(x_1, \ldots, x_n)$.

**Bemerkung.** Die Bedeutung der linearen Algebra liegt vor allem auch in den Anwendungen: Lösung linearer Gleichungssysteme, Verständnis der Struktur linearer Abbildungen, ...
Beispielsweise sind die folgenden wichtigen Begriffe *linear abhängig* und *linear unabhängig* auf die Lösungstheorie linearer Gleichungssysteme zugeschnitten. Geometrisch bedeutet die Lösung eines linearen $3 \times 3$ Gleichungssystemes $A\vec{x} = \vec{b}$, wobei $A \in \mathbb{R}^{3\times 3}$, $\vec{b} \in \mathbb{R}^3$, den Vektor

$$\vec{b} = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \end{pmatrix} x_1 + \begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \end{pmatrix} x_2 + \begin{pmatrix} a_{13} \\ a_{23} \\ a_{33} \end{pmatrix} x_3$$

bezüglich der Basis

$$\begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \end{pmatrix}, \begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \end{pmatrix}, \begin{pmatrix} a_{13} \\ a_{23} \\ a_{33} \end{pmatrix}$$

also den Spaltenvektoren von $A$ darzustellen.
Man erkennt sofort einen mögliche Entartungsfall: Wenn die drei Vektoren $\vec{a}_1, \vec{a}_2, \vec{a}_3$ in einer Ebene liegen, bilden sie keine Basis und $\vec{b}$ lässt sich i.A. nicht so darstellen, (außer wenn auch $\vec{b}$ in der Ebene liegt, wobei in diesem Fall die eindeutige Darstellung von $\vec{b}$, also die eindeutige Lösbarkeit des Gleichungssystems verloren geht). In diesem Entartungsfall ist jeder der Vektoren $\vec{a}_i$ eine Linearkombination der anderen beiden. Durch diese Überlegungen wird man in ganz natürlicher Weise zu den nachfolgenden Begriffsbildungen geführt:

**Definition 2.1.6.** *1. $n$ Vektoren $\vec{a}_1, \ldots, \vec{a}_n \in \mathbb{R}^m$ heißen* **linear abhängig** *(l.a.), falls es $n$ Konstanten $c_1, \ldots, c_n \in \mathbb{R}$ gibt, die nicht alle gleichzeitig 0 sind, sodass gilt:*

$$c_1\vec{a}_1 + \cdots + c_n\vec{a}_n = \vec{0}. \tag{2.6}$$

*2. Falls aus (2.6) notwendig $c_1 = c_2 = \cdots = c_n = 0$ folgt, so heißen die $n$ Vektoren* **linear unabhängig** *(l.u.).*

Wir betrachten jetzt $n$ linear unabhängige Vektoren $\vec{a}_1, \ldots, \vec{a}_n \in \mathbb{R}^m$ und die Menge aller Linearkombinationen dieser Vektoren:

$$\vec{a} = c_1\vec{a}_1 + \cdots + c_n\vec{a}_n \qquad c_1, ..., c_n \in \mathbb{R} \tag{2.7}$$

Diese Vektoren $\vec{a}$ bilden einen *linearen Unterraum* des $\mathbb{R}^m$ (denn mit $\vec{a} = c_1\vec{a}_1 + \cdots + c_n\vec{a}_n$ und mit $\bar{\vec{a}} = \bar{c}_1\vec{a}_1 + \cdots + \bar{c}_n\vec{a}_n$ liegt auch $\vec{a} + \bar{\vec{a}} = (c_1 + \bar{c}_1)\vec{a}_1 + \cdots + (c_n + \bar{c}_n)\vec{a}_n$ in dieser Menge der Linearkombinationen und ebenso $\lambda\vec{a} = (\lambda c_1)\vec{a}_1 + \cdots + (\lambda c_n)\vec{a}_n$ für $\lambda \in \mathbb{R}$). Im Spezialfall $n = m$ fällt dieser Unterraum mit ganz $\mathbb{R}^m$ zusammen, $n > m$ ist nicht möglich, da dann die Vektoren nicht l.u. sein können (im $\mathbb{R}^2$ kann es z.B. nicht drei l.u. Vektoren geben, da drei Vektoren in einer Ebene stets l.a. sind). Jeder Vektor $\vec{a}$ aus diesen Unterraum ist durch die **Koordinaten** $c_1, \ldots, c_n \in \mathbb{R}$ *eindeutig* charakterisiert, denn indirekt angenommen

$$\begin{aligned} \vec{a} &= c_1\vec{a}_1 + \cdots + c_n\vec{a}_n \quad \text{und} \\ \vec{a} &= d_1\vec{a}_1 + \cdots + d_n\vec{a}_n \end{aligned} \tag{2.8}$$

$$\text{folgt } 0 = \vec{a} - \vec{a} = (c_1 - d_1)\vec{a}_1 + \cdots + (c_n - d_n)\vec{a}_n \tag{2.9}$$

und wegen der linearen Unabhänigkeit der Vektoren $\vec{a}_1, \ldots, \vec{a}_n$ also $c_1 = d_1, \ldots, c_n = d_n$.

Der Unterraum ist $n$-**dimensional** (bringt zum Ausdruck, dass jeder Vektor aus diesem Unterraum durch $n$ Koordinaten eindeutig festgelegt ist, in den Fällen $n = 1$, 2, 3 deckt sich diese Sprechweise mit der üblichen anschaulichen Bedeutung des Begriffes *Dimension* ) bedeutet dass $n$ l.u. Vektoren $\vec{a}_1, \ldots, \vec{a}_n$ eine **Basis** des Unterraums bilden bzw. den Unterraum aufspannen.

**Definition 2.1.7.** *Der* **Bildraum** *der Matrix $A \in \mathbb{R}^{m\times n}$ (bzw. der durch $A$ definierten linearen Abbildung) ist durch*

$$Bild(A) := \{\vec{y} \in \mathbb{R}^m : \vec{y} = A\vec{x}, \vec{x} \in \mathbb{R}^n\} \tag{2.10}$$

*definiert. Er ist ein linearer Unterraum des $R^m$.*

**Definition 2.1.8.** *Der* **Rang** *einer Matrix $A \in \mathbb{R}^{m\times n}$ (bzw. der durch $A$ definierten linearen Abbildung) ist durch*

$$r \;=\; Rang(A) \;=\; \dim(Bild(A)) \tag{2.11}$$

*definiert.*

Alternative Möglichkeiten, den Rang zu definieren:

i) *Spaltenrang*: Die maximale Anzahl $r$ linear unabhängiger Spalten von $A$ heißt der Spaltenrang von $A$. Es gilt *Spaltenrang* $=$ $\dim(Bild(A))$, da

$$A\vec{x} = x_1\vec{a}_1 + \cdots + x_n\vec{a}_n,$$

wobei $a_i\, i = 1, \ldots, n$ die Spalten von $A$ bezeichnen.

ii) *Zeilenrang*: Die maximale Anzahl $r$ linear unabhängiger Zeilenvektoren heißt der Zeilenrang von $A$. Es gilt *Zeilenrang* $=$ $\dim(Bild(A))$.

**Bemerkung.** Die drei Möglichkeiten, den Rang zu definieren, als Zeilenrang, oder als Spaltenrang oder als $\dim(Bild(A))$ sind äquivalent.

**Definition 2.1.9.** *Der* **Kern** *der Matrix $A$ (bzw. der durch $A$ definierten linearen Abbildung) ist durch*

$$Kern(A) \;:=\; \{\vec{x} :\, A\vec{x} = \vec{0},\, \vec{x} \in \mathbb{R}^n\} \tag{2.12}$$

*definiert.*

Der Kern ist ein linearer Unterraum des $\mathbb{R}^n$. Seine Dimension ist wichtig im Zusammenhang mit der Lösungstheorie von linearen Gleichungsystemen $A\vec{x} = \vec{b}$. (Vgl. später! Abschnitt 2.3)

## 2.2 Vektor- und Matrixnormen

**Definition 2.2.1.** *Sei $V$ ein beliebiger Vektorraum über $K$ $(= \mathbb{C}, \mathbb{R})$. Eine Abbildung $\|\cdot\| : V \to \mathbb{R}_{\geq 0}$ heißt* **Norm** *, falls folgende drei Eigenschaften erfüllt sind:*

1. *Dreiecksungleichung bzw. Subadditivität: $\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\| \quad \forall \vec{x}, \vec{y} \in V$*
2. *absolute Homogenität: $\|\alpha\vec{x}\| = |\alpha|\, \|\vec{x}\| \quad \forall \vec{x} \in V, \forall \alpha \in K$*
3. *Definitheit: $\|\vec{x}\| = 0 \Leftrightarrow \vec{x} = \vec{0}$*

Auf den Räumen $\mathbb{R}^n$ bzw. $\mathbb{C}^n$ sind verschiedene **Vektornormen** definiert:

- Die bekannteste Vektornorm ist die **euklidische Norm**

$$\|\vec{x}\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \ldots + |x_n|^2} = \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2}, \tag{2.13}$$

die anschaulich die „Länge eines Vektors“ beschreibt.

- Die euklidische Norm ist ein Spezialfall ($p = 2$) der **p-Norm**

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

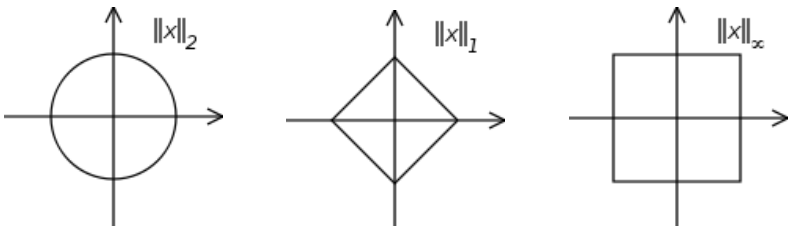
  die für $1 \leq p < \infty$, $p \in \mathbb{N}$ definiert ist.

- Ein weiterer Spezialfall ($p = 1$) ist die **Summennorm**

$$\|\vec{x}\|_1 = \sum_{i=1}^{n} |x_i|.$$

- Durch die Grenzwertbildung $p \to \infty$ erhält man schließlich die **Maximumsnorm**

$$\|\vec{x}\|_\infty = \max_{i \in \{1,2,\ldots,n\}} |x_i|.$$




(a) Euklidische Norm ($p = 2$) (b) Summennorm ($p = 1$) (c) Maximumsnorm ($p = \infty$)

Abbildung 2.1: Einheitskreis in verschiedenen $p$-Normen

Abbildung 2.1 zeigt die Form der Einheitskreise ($\|\vec{x}\| = 1$) im $\mathbb{R}^2$ bezüglich verschiedener $p$-Normen.

Es ist leicht zu sehen, dass alle oben erwähnten Normen absolute Homogenität und Definitheit erfüllen. Die Dreiecksungleichung für die allgemeine $p$-Norm ist etwas schwieriger zu zeigen.

**Beispiel 2.2.2.** *Sei $V = \mathbb{R}^3$ und $\vec{x} = (1, -2, 3)^T \in V$ gegeben. Dann gilt*

$$\begin{array}{rcccr} \|\vec{x}\|_2 & = & \sqrt{1^2 + 2^2 + 3^2} & = & \sqrt{14}, \\ \|\vec{x}\|_1 & = & 1 + 2 + 3 & = & 6, \\ \|\vec{x}\|_\infty & = & \max\{1, 2, 3\} & = & 3. \end{array}$$

**Satz 2.2.3.** *Alle Vektornormen auf $K^n$ sind zueinander äquivalent, das heißt für je zwei Normen $\|\cdot\|_a$ und $\|\cdot\|_b$ gibt es zwei Konstanten $c_1, c_2 \in \mathbb{R}_{\geq 0}$, sodass*

$$c_1 \|\vec{x}\|_b \leq \|\vec{x}\|_a \leq c_2 \|\vec{x}\|_b$$

*für alle $\vec{x} \in K^n$ gilt.*

Jede Vektornorm kann also nach oben und unten durch jede andere Vektornorm abgeschätzt werden.

Auch auf Vektorräumen von Matrizen kann man Normen definieren.

**Definition 2.2.4.** *Sei $V = K^{m\times n}$ ein Vektorraum über $K$ $(= \mathbb{C}, \mathbb{R})$. Eine Abbildung $\|\cdot\| : V \to \mathbb{R}_{\geq 0}$ heißt* **Matrixnorm** *, falls die Eigenschaften $1. - 3.$ aus Definition 2.2.1 für alle $A, B \in V$ erfüllt sind. Oft wird zusätzlich noch eine vierte Eigenschaft,*

*4. Submultiplikativität: $\|AB\| \leq \|A\| \cdot \|B\| \quad \forall A \in K^{m\times n}, B \in K^{n\times l}$,*

*gefordert.*

**Bemerkung.** Meist wird bei der Definition einer Matrixnorm eine Vektornorm zugrunde gelegt.

**Definition 2.2.5.** *Eine Matrixnorm $\|\cdot\|$ heißt* **verträglich mit der Vektornorm** *$\|\cdot\|_V$, falls*

$$\|A\vec{x}\|_V \leq \|A\| \, \|\vec{x}\|_V \quad \forall \vec{x} \in K^n.$$

Es gibt drei gängige Möglichkeiten, Matrixnormen zu definieren.

### 1. Matrix als Vektor interpretieren

Eine Matrix $A \in K^{m\times n}$ kann als langer Vektor $A \in K^{mn}$ angesehen werden. Die Matrixnorm kann nun direkt aus einer Vektornorm übernommen werden. Es ergibt sich also die Matrix-$p$-Norm

$$\|A\|_p = \left( \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^p \right)^{1/p}.$$

Für den Fall $p = 2$ erhält man die **Frobeniusnorm**

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}.$$

Diese ist submultiplikativ und mit der euklidischen Norm $\|\cdot\|_2$ verträglich. Die **Maximumsnorm**

$$\|A\|_M = \max_{i,j} |a_{ij}|$$

ist nicht submultiplikativ. Abhilfe schafft die **Gesamtnorm**

$$\|A\|_G = \sqrt{mn} \max_{i,j} |a_{ij}|,$$

die mit dem Faktor $\sqrt{mn}$ (dem geometrischen Mittel der beiden Raumdimensionen) gewichtet wird. Die Gesamtnorm ist submultiplikativ und für quadratische Matrizen mit allen Vektor-$p$-Normen (inklusive der Maximumsnorm) verträglich.

**Beispiel 2.2.6.** *Sei* $A = \begin{pmatrix} 0.5 & -2 \\ 1.5 & 1 \end{pmatrix}$ *gegeben. Dann gilt*

$$\begin{array}{rcrcr} \|A\|_F & = & \sqrt{0.5^2 + 2^2 + 1.5^2 + 1^2} & = & \sqrt{7.5}, \\ \|A\|_M & = & \max\{0.5, 2, 1.5, 1\} & = & 2, \\ \|A\|_G & = & \sqrt{2 \cdot 2} \max\{0.5, 2, 1.5, 1\} & = & 4. \end{array}$$

**Beispiel 2.2.7.** *Sei* $B = \begin{pmatrix} 2 & 2 \\ -2 & 0 \end{pmatrix}$ *gegeben. Dann gilt*

$$\begin{array}{rcrcr} \|B\|_F & = & \sqrt{2^2+2^2+2^2} & = & \sqrt{12}, \\ \|B\|_M & = & \max\{2,2,2,0\} & = & 2, \\ \|B\|_G & = & \sqrt{2\cdot 2}\max\{2,2,2,0\} & = & 4. \end{array}$$

**Beispiel 2.2.8.** *Anhand der Matrizen A und B aus Beispiel 2.2.6 und 2.2.7 wird deutlich, dass die Maximumsnorm nicht submultiplikativ ist, die Gesamtnorm jedoch schon. Es gilt*

$$AB = \begin{pmatrix} 5 & 1 \\ 1 & 3 \end{pmatrix}$$

*und damit*

$$\begin{array}{rcrcr} \|AB\|_M & = & \max\{5,1,1,3\} & = & 5, \\ \|AB\|_G & = & \sqrt{2\cdot 2}\max\{5,1,1,3\} & = & 10. \end{array}$$

*Für das Produkt der Einzelnormen erhält man*

$$\begin{array}{rcrcrcr} \|A\|_M\|B\|_M & = & 2\cdot 2 & = & 4 & < & 5, \\ \|A\|_G\,\|B\|_G & = & 4\cdot 4 & = & 16 & \geq & 10. \end{array}$$

**Beispiel 2.2.9.** *Sei nun die komplexe Matrix* $C = \begin{pmatrix} 1-i & -3i \\ 2i & 0 \end{pmatrix}$ *gegeben. Dann gilt*

$$\begin{array}{rcrcr} \|C\|_F & = & \sqrt{\sqrt{2}^2+3^2+2^2} & = & \sqrt{15}, \\ \|C\|_M & = & \max\{\sqrt{2},3,2,0\} & = & 3, \\ \|C\|_G & = & \sqrt{2\cdot 2}\max\{\sqrt{2},3,2,0\} & = & 6. \end{array}$$

### 2. Matrixnorm durch Vektornorm induzieren

**Definition 2.2.10.** *Die Matrixnorm* $\|.\|$ *heißt* **Operatornorm** *zur Vektornorm* $\|.\|_V$*, falls gilt*

$$\|A\| = \max_{\vec{x}\neq\vec{0}} \frac{\|A\vec{x}\|_V}{\|\vec{x}\|_V} = \max_{\|\vec{x}\|_V=1} \|A\vec{x}\|.$$

Eine Matrixnorm, die als Operatornorm von $\|.\|_V$ abgeleitet ist, heißt von $\|.\|_V$ **induziert**. Oft werden solche Normen auch als **natürliche Matrixnormen** bezeichnet.
Anschaulich entspricht eine induzierte Norm dem größtmöglichen Streckungsfaktor nach Anwendung der Matrix auf einen Vektor der Norm 1.
Es ist leicht zu sehen, dass induzierte Normen tatsächlich alle Normeigenschaften aus Definition 2.2.4 erfüllen. Außerdem sind sie submultiplikativ und mit der Vektornorm, von der sie abgeleitet sind, verträglich.
Als induzierte Norm zur Maximumsnorm ergibt sich die **Zeilensummennorm**

$$\|A\|_\infty = \max_{\|\vec{x}\|_\infty=1} \|A\vec{x}\|_\infty = \max_{i=1,2,\ldots,m} \sum_{j=1}^{n} |a_{ij}|. \tag{2.14}$$

**Beweis** von (2.14): Die Gleichheit wird durch zwei Ungleichungen gezeigt. Einerseits gilt

$$\|A\vec{x}\|_\infty = \max_{i=1,2,\ldots,m} \left| \sum_{j=1}^{n} a_{ij}x_j \right| \leq \max_{i=1,2,\ldots,m} \sum_{j=1}^{n} |a_{ij}|\,|x_j| \leq \left( \max_{i=1,2,\ldots,m} \sum_{j=1}^{n} |a_{ij}| \right) \|\vec{x}\|_\infty$$

und damit

$$\|A\|_\infty = \max_{\vec{x}\neq\vec{0}} \frac{\|A\vec{x}\|_\infty}{\|\vec{x}\|_\infty} \leq \max_{i=1,2,\ldots,m} \sum_{j=1}^{n} |a_{ij}|.$$

Für die umgekehrte Ungleichung hält man zunächst $k \in \{1, 2, \ldots, m\}$ fest und definiert $\vec{x} = (x_j) \in K^n$ durch

$$x_j = \begin{cases} |a_{kj}|/a_{kj}, & a_{kj} \neq 0 \\ 1, & \text{sonst} \end{cases} \qquad (j = 1, \ldots, n).$$

Dieses $\vec{x}$ erfüllt $\|\vec{x}\|_\infty = 1$ und damit

$$\|A\|_\infty = \|A\|_\infty \|\vec{x}\|_\infty \geq \|A\vec{x}\|_\infty \geq \left| \sum_{j=1}^{n} a_{kj} x_j \right| = \sum_{j=1}^{n} |a_{kj}|.$$

Da $k \in \{1, 2, \ldots, m\}$ beliebig gewählt war, folgt die Behauptung. □

Verwendet man stattdessen die Summennorm, erhält man die **Spaltensummennorm**

$$\|A\|_1 = \max_{\|\vec{x}\|_1=1} \|A\vec{x}\|_1 = \max_{j=1,2,\ldots,n} \sum_{i=1}^{m} |a_{ij}|. \tag{2.15}$$

**Beweis** von (2.15): analog zu (2.14), für die zweite Abschätzung wähle $k$-ten Einheitsvektor. □

Die euklidische Norm schließlich liefert die **Spektralnorm**

$$\|A\|_2 = \max_{\|\vec{x}\|_2=1} \|A\vec{x}\|_2 = r_\sigma(A^H A)^{1/2}, \tag{2.16}$$

wobei $A^H$ im reellen Fall die transponierte, im komplexen Fall die adjungierte Matrix von $A$ ist und $r_\sigma$ den Spektralradius der positiv semidefiniten Matrix $A^H A$ bezeichnet:

**Definition 2.2.11.** *Sei $B \in K^{n\times n}$, so heißt $\sigma(B) = \{\lambda \in C : \lambda$ ist Eigenwert von $B\}$ das* **Spektrum** *von $B$. Der Wert $r_\sigma(B) = \max_{\lambda\in\sigma(B)} |\lambda|$ ist der* **Spektralradius** *von $B$.*

Für eine symmetrische (bzw. im komplexen Fall hermitesche) quadratische Matrix $A$, also $A^H = A$, gilt

$$\|A\|_2 = r_\sigma(A^T A)^{1/2} = r_\sigma(A^2)^{1/2} = (r_\sigma(A)^2)^{1/2} = r_\sigma(A).$$

Der Beweis von (2.16) erfordert detaillierteres Wissen über lineare Algebra und wird an dieser Stelle ausgelassen.

**Beispiel 2.2.12.** *Sei die Matrix $A$ aus Beispiel 2.2.6 gegeben. Es gilt*

$$\begin{array}{rcrcr} \|A\|_\infty & = & \max\{0.5+2, 1.5+1\} & = & 2.5, \\ \|A\|_1 & = & \max\{0.5+1.5, 2+1\} & = & 3. \end{array}$$

*Um die Spektralnorm zu berechnen, müssen die Eigenwerte der Matrix $A^H A$ bestimmt werden. Matrizenmultiplikation führt auf*

$$A^H A = \begin{pmatrix} 0.5 & 1.5 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 0.5 & -2 \\ 1.5 & 1 \end{pmatrix} = \begin{pmatrix} 2.5 & 0.5 \\ 0.5 & 5 \end{pmatrix}.$$

*Die charakteristische Gleichung ist also $(2.5-\lambda)(5-\lambda) - 0.25 = 0$ mit den Lösungen $\lambda_1 \approx 5.096$ und $\lambda_2 \approx 2.404$. Es folgt*

$$\|A\|_2 = \sqrt{\max\{\lambda_1, \lambda_2\}} \approx 2.257.$$

**Beispiel 2.2.13.** *Sei die Matrix $B$ aus Beispiel 2.2.7 gegeben. Es gilt*

$$\begin{array}{rcccl} \|B\|_\infty & = & \max\{2+2, 2+0\} & = & 4, \\ \|B\|_1 & = & \max\{2+2, 2+0\} & = & 4. \end{array}$$

*Um die Spektralnorm zu berechnen, müssen wie zuvor die Eigenwerte der Matrix $B^H B$ bestimmt werden. Matrizenmultiplikation führt auf*

$$B^H B = \begin{pmatrix} 2 & -2 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} 2 & 2 \\ -2 & 0 \end{pmatrix} = \begin{pmatrix} 8 & 4 \\ 4 & 4 \end{pmatrix}.$$

*Die charakteristische Gleichung ist also $(8-\lambda)(4-\lambda)-16=0$ mit den Lösungen $\lambda_1 = 6+2\sqrt{5}$ und $\lambda_2 = 6-2\sqrt{5}$. Es folgt*

$$\|B\|_2 = \sqrt{\max\{\lambda_1, \lambda_2\}} = \sqrt{6+2\sqrt{5}} \approx 10.47.$$

**Beispiel 2.2.14.** *Sei nun die komplexe Matrix $C$ aus Beispiel 2.2.9 gegeben. Es gilt*

$$\begin{array}{rcccl} \|C\|_\infty & = & \max\{\sqrt{2}+3, 2+0\} & = & \sqrt{2}+3 \approx 4.41, \\ \|C\|_1 & = & \max\{\sqrt{2}+2, 3+0\} & = & \sqrt{2}+2 \approx 3.41. \end{array}$$

*Um die Spektralnorm zu berechnen, müssen wie zuvor die Eigenwerte der Matrix $C^H C$ bestimmt werden. Matrizenmultiplikation führt auf*

$$C^H C = \begin{pmatrix} 1+i & -2i \\ 3i & 0 \end{pmatrix} \begin{pmatrix} 1-i & -3i \\ 2i & 0 \end{pmatrix} = \begin{pmatrix} 6 & 3-3i \\ 3+3i & 9 \end{pmatrix}.$$

*Die charakteristische Gleichung ist also $(6-\lambda)(9-\lambda)-18=0$ mit den Lösungen $\lambda_1 = 12$ und $\lambda_2 = 3$. Es folgt*

$$\|C\|_2 = \sqrt{\max\{\lambda_1, \lambda_2\}} = \sqrt{12} \approx 3.46.$$

### 3. Singulärwerte

Eine dritte Möglichkeit, Matrixnormen zu definieren, liefern die Singulärwerte, also die Eigenwerte der Matrix $A^H A$. Im Folgenden werden die so erzeugten Normen allerdings nicht benötigt, weshalb hier auch nicht näher darauf eingegangen wird.

Wie bereits für Vektornormen, gilt auch für Matrixnormen die folgende Äquivalenzaussage.

**Satz 2.2.15.** *Alle Matrixnormen auf $K^{m\times n}$ sind zueinander äquivalent, das heißt für je zwei Normen $\|\cdot\|_a$ und $\|\cdot\|_b$ gibt es zwei Konstanten $c_1, c_2 \in \mathbb{R}_{\geq 0}$, sodass*

$$c_1\|A\|_b \leq \|A\|_a \leq c_2\|A\|_b$$

*für alle $A \in K^{m\times n}$ gilt.*

Jede Matrixnorm kann also nach oben und unten durch jede andere Matrixnorm abgeschätzt werden.

## 2.3 Lösungstheorie für lineare Gleichungssysteme

**Definition 2.3.1.** *In einem* **linearen Gleichungssystem** *(LGS) sind Daten*

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \in \mathbb{R}^{m\times n} \quad \textit{und} \quad \vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{R}^m$$

*($m, n$ beliebig aus $\mathbb{N}$) gegeben. Gesucht ist ein Vektor $\vec{x} = (x_1, x_2, \dots, x_m)^T \in \mathbb{R}^m$, sodass*

$$A\vec{x} \;=\; \vec{b}, \tag{2.17}$$

*also*

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_3 \\ \vdots & & & & & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & = & b_m \end{array} \tag{2.18}$$

*gilt. Die Matrix $A = (a_{ij})$ heißt* **Koeffizientenmatrix** *des linearen Gleichungssystems. Der Spaltenvektor $\vec{b}$ wird als* **Inhomogenität** *oder* **rechte Seite** *bezeichnet. Das System heißt* **homogen**, *falls $\vec{b} = \vec{0}$.*

Lineare Gleichungssysteme treten in sehr vielen Anwendungen auf. Manchmal führt ein einfacher mathematischer Modellierungsprozess unmittelbar zu einem linearen Gleichungssystem. Sehr häufig liegt aber auch eine etwas andere Situation vor: Zunächst führt der mathematische Modellbildungsprozess auf ein anders geartetes mathematisches Problem, und erst ein allfälliger numerischer Algorithmus zur Lösung dieses Problems resultiert schließlich in einem (oder mehreren) linearen Gleichungssystem(en).

**Zwei Lösungsbegriffe:**

(i) Lösung $\vec{x} \in \mathbb{R}^n$ gesucht, sodass das Gleichungssystem $A\vec{x} = \vec{b}$ erfüllt ist.

(ii) Lösung im **Ausgleichssinn** : Falls kein $\vec{x}$ existiert, das $A\vec{x} = \vec{b}$ im Sinne von (i) löst, sind die Gleichungen *widersprüchlich*. Man will nun $\vec{x}$ so bestimmen, dass $\vec{x}$ „in das Gleichungssystem so gut wie mglich hineinpasst“. D.h. $\vec{x}$ ist dann dadurch festgelegt, dass die euklidische Norm

$$\|A\vec{x} - \vec{b}\|_2 \quad \dots \quad \min! \tag{2.19}$$

minimiert wird (vgl. Abschnitt 2.2).

Eine wesentliche Rolle für die Lösbarkeit linearer Gleichungssysteme spielt die **erweiterte Matrix:**

$$(A|\vec{b}) = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{array}\right)$$

**Satz 2.3.2.** *Ein lineares inhomogenes Gleichungssystem $A\vec{x} = \vec{b}$ ist genau dann lösbar, wenn*

$$Rang(A) = Rang(A|\vec{b}).$$

Für den Spezialfall $Rang(A) = m$ hat das Gleichungssystem immer eine Lösung. (Da die Spalten von $A$ und von $(A|\vec{b})$ Elemente von $\mathbb{R}^m$ sind, gilt $m = Rang(A) \leq Rang((A|\vec{b})) \leq m$.)

Für $m = n = Rang(A)$ ist das lineare Gleichungssystem $A\vec{x} = \vec{b}$ für beliebige rechte Seiten $\vec{b} \in \mathbb{R}^m = \mathbb{R}^n$ sogar *eindeutig lösbar*:
Die l.u. Spalten $\vec{a}_1, \vec{a}_2, ..., \vec{a}_n$ von $A$ bilden eine Basis des $\mathbb{R}^n$, d.h. jeder Vektor $\vec{b} \in \mathbb{R}^n$ lässt sich eindeutig schreiben als Linearkombination $\vec{b} = x_1\vec{a}_1 + \cdots + x_n\vec{a}_n$. Die eindeutig durch $\vec{b}$ festgelegten Gewichte sind somit offenbar die eindeutige Lösung von $A\vec{x} = \vec{b}$.

Für ein lösbares LGS stellt sich die Frage nach der Dimension des Lösungsraums.

**Satz 2.3.3.** *Sei $\vec{x}_0$ eine Lösung von (2.17) und $\vec{x} \in Kern(A)$, so ist auch $\vec{x_0} + \vec{x}$ eine Lösung.*

**Beweis.**

$$\left.\begin{array}{rcl} A\vec{x}_0 & = & \vec{b} \\ A\vec{x} & = & \vec{0} \end{array}\right\} \quad \Rightarrow \quad A(\vec{x}_0 + \vec{x}) = \vec{b} \tag{2.20}$$

□

Die Menge aller $\vec{x}_0 + \vec{x}$ mit einer speziellen Lösung $\vec{x}_0$ von (2.17) und $\vec{x} \in Kern(A)$ liegt also im Lösungsraum.
Umgekehrt lässt sich jede Lösung von (2.17) schreiben als $\vec{x}_0 + \vec{x}$, wobei $\vec{x}_0$ die oben betrachtete spezielle Lösung ist und $\vec{x} \in Kern(A)$ beliebig, denn aus $A\vec{x}_1 = \vec{b}$, $A\vec{x}_0 = \vec{b}$ folgt sofort: $A(\vec{x}_1 - \vec{x}_0) = \vec{0}$, also $\vec{x}_1 - \vec{x}_0 \in Kern(A)$, also $\vec{x}_1 = \vec{x}_0 + \vec{x}$ mit $\vec{x} \in Kern(A)$.
Wenn man also zu einer speziellen Lösung $\vec{x}_0$ alle Elemente des Kerns addiert, erhält man genau alle Lösungen von (2.17), d.h. es gilt:

$$\dim(Kern(A)) \quad = \quad \dim(\text{Lösungsraum}). \tag{2.21}$$

**Satz 2.3.4.** *Für $A \in \mathbb{R}^{m\times n}$ gilt*

$$\dim(Kern(A)) \quad = \quad n - Rang(A) \tag{2.22}$$

*bzw. dual dazu*

$$\dim(Kern(A^\top)) = m - Rang(A). \tag{2.23}$$

## 2.4 Konditionsabschätzungen

Modellfehler, Datenfehler und Rechenfehler beim Einlesevorgang der Koeffizienten verfälschen ein lineares Gleichungssystem. Statt des „wahren“Systems

$$A\vec{x} \; = \; \vec{b} \tag{2.24}$$

hat man i.A. das verfälschte System

$$\tilde{A}\vec{\tilde{x}} \; = \; \vec{\tilde{b}} \tag{2.25}$$

im Rechner. Es müssen daher die Auswirkungen der Datenstörungen auf das Ergebnis, also der *absolute Fehler*

$$\|\vec{\tilde{x}} - \vec{x}\|$$

oder der *relative Fehler*

$$\frac{\|\vec{\tilde{x}} - \vec{x}\|}{\|\vec{x}\|} \quad \text{bzw.} \quad \frac{\|\vec{\tilde{x}} - \vec{x}\|}{\|\vec{\tilde{x}}\|}$$

abgeschätzt werden.

### 2.4.1 Konditionsabschätzungen bezüglich Störungen von $\vec{b}$

$$\begin{aligned} \text{Ungestörtes Problem:} \quad & A\vec{x} = \vec{b}, \qquad (A \ldots \text{regulär}) \\ \text{Gestörtes Problem:} \quad & A\vec{\tilde{x}} = \vec{\tilde{b}} = \vec{b} + \Delta\vec{b} \end{aligned}$$

$$\Rightarrow\ A(\vec{\tilde{x}} - \vec{x}) = \Delta\vec{b} \quad \Rightarrow\ \Delta\vec{x} := (\vec{\tilde{x}} - \vec{x}) = A^{-1}\Delta\vec{b} \quad \Rightarrow$$

$$\boxed{\|\Delta\vec{x}\| \ = \ \|\vec{\tilde{x}} - \vec{x}\| \ \leq \ \|A^{-1}\|\ \|\Delta\vec{b}\|} \tag{2.26}$$

**absolute Konditionsabschätzung**

Um eine relative Konditionsabschätzung zu erhalten, werden alle absoluten Fehlergrößen durch relative Fehlergrößen ersetzt, z.B. $\|\Delta\vec{b}\|$ durch $\frac{\|\Delta\vec{b}\|}{\|\vec{b}\|}$ oder $\|\Delta\vec{x}\|$ durch $\frac{\|\Delta\vec{x}\|}{\|\vec{x}\|}$:

$$\|\Delta\vec{x}\| \ \leq \ \|A^{-1}\|\ \|\Delta\vec{b}\| \quad \text{Vgl. (2.26)} \quad \Rightarrow$$

$$\boxed{\frac{\|\Delta\vec{x}\|}{\|\vec{x}\|} \ \leq \ \frac{\|A\|\ \|A^{-1}\|\ \|\Delta\vec{b}\|}{\|A\|\ \|\vec{x}\|} \ \leq \ \|A\|\ \|A^{-1}\|\frac{\|\Delta\vec{b}\|}{\|\vec{b}\|}} \tag{2.27}$$

**relative Konditionsabschätzung**

Dabei wurde benützt: $\|A\|\ \|\vec{x}\| \geq \|A\vec{x}\| = \|\vec{b}\|$.
Diese Abschätzung ist also vom Typ

$$\|\ \text{relative Störung des Ergebnisses } \vec{x}\ \| \ \leq\ \text{Faktor} \cdot \|\ \text{relative Störung von } \vec{b}\ \|.$$

**Definition 2.4.1.** *Der Faktor*

$$\kappa(A) \ = \ \|A\|\ \|A^{-1}\| \tag{2.28}$$

*wird als* **Konditionszahl** *bezeichnet.*

Man kann zeigen, dass die Abschätzungen (2.26) und (2.27) scharf sind in folgendem Sinn: Zu jedem $A$ gibt es ein $\vec{b}$ und ein $\Delta\vec{b}$, sodass das Gleichheitszeichen angenommen wird.

### 2.4.2 Konditionsabschätzungen bezüglich Störungen von $A$

Ungestörtes Problem: $A\vec{x} = \vec{b}$, $(A \ldots \text{regulär})$
Gestörtes Problem: $\tilde{A}\vec{\tilde{x}} = \vec{b} = (A + \Delta A)\vec{\tilde{x}} = \vec{b}$ $(\tilde{A} = A + \Delta A \ldots \text{regulär})$

$$
\begin{array}{lllll}
\Longrightarrow & (A + \Delta A)(\vec{x} + \Delta\vec{x}) & = \tilde{A}\vec{\tilde{x}} & = \vec{b} \\
\Leftrightarrow & A\vec{x} + \Delta A\vec{x} + A\Delta\vec{x} + \Delta A\Delta\vec{x} & & = \vec{b} \\
\Leftrightarrow & A\Delta\vec{x} & = \Delta A(\vec{x} + \Delta\vec{x}) & = -\Delta A\vec{\tilde{x}} \\
\Leftrightarrow & \Delta\vec{x} & & = -A^{-1}\Delta A\vec{\tilde{x}} \\
\Longrightarrow & & &
\end{array}
$$

$$
\boxed{\|\Delta\vec{x}\| \leq \|A^{-1}\| \, \|\vec{\tilde{x}}\| \, \|\Delta A\|} \tag{2.29}
$$

**absolute Konditionsabschätzung** bezüglich Störungen von A

$$
\boxed{\frac{\|\Delta\vec{x}\|}{\|\vec{\tilde{x}}\|} \leq \|A\| \, \|A^{-1}\| \frac{\|\Delta A\|}{\|A\|}} \tag{2.30}
$$

**relative Konditionsabschätzung** bezüglich Störungen von $A$

Wieder tritt als Faktor die Konditionszahl $\kappa(A)$ auf. Beim relativen Fehler $\frac{\|\Delta\vec{x}\|}{\|\vec{\tilde{x}}\|}$ ist allerdings nicht $\|\vec{x}\|$, sondern $\|\vec{\tilde{x}}\|$ der Vergleichswert. Für eine relative Abschätzung bzgl. $\frac{\|\Delta\vec{x}\|}{\|\vec{x}\|}$ kann man folgendermaßen vorgehen:

$$
\begin{array}{lll}
(A + \Delta A)(\vec{x} + \Delta\vec{x}) & = & \vec{b} \\
\Leftrightarrow \quad A\vec{x} + \Delta A\vec{x} + \underbrace{A\Delta\vec{x} + \Delta A\Delta\vec{x}}_{\tilde{A}\Delta\vec{x}} & = & \vec{b} \\
\Leftrightarrow \quad \tilde{A}\Delta\vec{x} & = & -\Delta A\vec{x} \\
\Leftrightarrow \quad \Delta\vec{x} & = & -\tilde{A}^{-1}\Delta A\vec{x} \\
\Longrightarrow & &
\end{array}
$$

$$
\boxed{\frac{\|\Delta\vec{x}\|}{\|\vec{x}\|} \leq \|A\| \, \|\tilde{A}^{-1}\| \frac{\|\Delta A\|}{\|A\|}} \tag{2.31}
$$

Die Konditionszahl ist jetzt $\|A\| \, \|\tilde{A}^{-1}\|$, hier ist also die Regularität von $\tilde{A}$ wesentlich. Man kann eine relative Konditionsabschätzung ganz ohne gestörte Größen folgender Form herleiten:

$$
\boxed{\frac{\|\Delta\vec{x}\|}{\|\vec{x}\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\Delta A\|}{\|A\|}} \frac{\|\Delta A\|}{\|A\|}}
$$

(unter der Voraussetzung, dass $\|\Delta A\|$ so klein ist, dass $\kappa(A)\frac{\|\Delta A\|}{\|A\|} < 1$ gilt).

### 2.4.3 Konditionsabschätzungen bezüglich Störungen von $A$ und $\vec{b}$

Ungestörtes Problem: $A\vec{x} = \vec{b}$, $(A \ldots \text{regulär})$
Gestörtes Problem: $\tilde{A}\vec{\tilde{x}} = \vec{\tilde{b}} = (A + \Delta A)\vec{\tilde{x}} = \vec{b} + \Delta\vec{b}$ $(\tilde{A} = A + \Delta A \ldots \text{regulär})$

In diesem Fall lässt sich (unter der Voraussetzung, dass $\|\Delta A\|$ so klein ist, dass $\kappa(A)\frac{\|\Delta A\|}{\|A\|} < 1$ gilt) die **relative Konditionsabschätzung**

$$\boxed{\frac{\|\Delta\vec{x}\|}{\|\vec{x}\|} \leq \frac{\kappa(A)}{1-\kappa(A)\frac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta A\|}{\|A\|}+\frac{\|\Delta\vec{b}\|}{\|\vec{b}\|}\right)}$$

herleiten.

## 2.5 Gaußelimination

Im Folgenden sei $A \in \mathbb{R}^{n\times n}$, $\vec{b} \in \mathbb{R}^n$ und $\mathrm{Rang}(A) = n$, dh.

$$A\vec{x} = \vec{b} \tag{2.32}$$

ist (bis auf Rundungsfehler) exakt und eindeutig lösbar.
Es tritt **kein Verfahrensfehler** auf, jedoch möglicherweise Datenfehler, aufgrund von Modellierung oder Messfehlern, und Rundungsfehler wegen der Computerarithmetik.
Die Lösung erfolgt mit einem *Eliminationsverfahren*, meist mit der sogenannten **Gaußelimination**.

**Beschreibung:**

1. Besonders leicht zu lösen sind sogenannte **gestaffelten Systeme** der Form

$$\begin{array}{rcrcccrcrcl} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1,n-1}x_{n-1} & + & a_{1,n}x_n & = & b_1 \\ & & a_{22}x_2 & + & \cdots & + & a_{2,n-1}x_{n-1} & + & a_{2,n}x_n & = & b_2 \\ & & & & \ddots & & & & \vdots & & \\ & & & & & & a_{n-1,n-1}x_{n-1} & + & a_{n-1,n}x_n & = & b_{n-1} \\ & & & & & & & & a_{nn}x_n & = & b_n. \end{array} \tag{2.33}$$

   Es folgt

$$x_n = \frac{b_n}{a_{nn}}, \quad x_{n-1} = \frac{b_{n-1} - a_{n-1,n}\frac{b_n}{a_{nn}}}{a_{n-1,n-1}}, \quad \dots \tag{2.34}$$

2. Die Grundidee der Gaußelimination ist, ein allgemeines System vom Typ (2.32) in die Form (2.33) bringen, um anschließend die Lösung gemäß (2.34) zu berechnen.

   Ausgangspunkt ist das *volle* System

$$\begin{array}{rcrcccrcl} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & & & & & \vdots & & \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & b_n \end{array} \tag{2.35}$$

   Multiplikation der ersten Gleichung mit $\frac{a_{21}}{a_{11}}$ und anschließende Subtraktion von der zweiten Gleichung lässt in der zweiten Zeile folgendes entstehen:

$$\begin{aligned} \underbrace{(a_{21} - \frac{a_{21}}{a_{11}}a_{11})}_{0}x_1 + \underbrace{(a_{22} - \frac{a_{21}}{a_{11}}a_{12})}_{a'_{22}}x_2 + \cdots + \\ + \underbrace{(a_{2n} - \frac{a_{21}}{a_{11}}a_{1n})}_{a'_{2n}}x_n \;=\; \underbrace{b_2 - \frac{a_{21}}{a_{11}}b_1}_{b'_2} \end{aligned} \tag{2.36}$$

Multiplikation der ersten Zeile mit $\frac{a_{31}}{a_{11}}$ und anschließende Subtraktion von der dritten Gleichung lässt in der dritten Zeile folgendes entstehen:

$$\begin{aligned} \underbrace{(a_{31} - \frac{a_{31}}{a_{11}} a_{11})}_{0} x_1 + \underbrace{(a_{32} - \frac{a_{31}}{a_{11}} a_{12})}_{a'_{32}} x_2 + \cdots + \\ + \underbrace{(a_{3n} - \frac{a_{31}}{a_{11}} a_{1n})}_{a'_{3n}} \cdot x_n \;=\; \underbrace{b_3 - \frac{a_{31}}{a_{11}} b_1}_{b'_3} \end{aligned} \tag{2.37}$$

u.s.w. Durch diese Manipulationen wird (2.35) in ein äquivalentes System (d.h. ein System mit derselben Lösung) von folgender Gestalt umgeformt:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ & & a'_{22}x_2 & + & \cdots & + & a'_{2n}x_n & = & b'_2 \\ & & \vdots & & & & & & \vdots \\ & & a'_{n2}x_2 & + & \cdots & + & a'_{nn}x_n & = & b'_n \end{array} \tag{2.38}$$

Wendet man nun die dieselbe Vorgangsweise auf das $(n-1) \times (n-1)$-Teilsystem

$$\begin{array}{ccccccc} a'_{22}x_2 & + & \cdots & + & a'_{2n}x_n & = & b'_2 \\ \vdots & & & & & & \vdots \\ a'_{n2}x_2 & + & \cdots & + & a'_{nn}x_n & = & b'_n \end{array}$$

von (2.38) an, erhält man

$$\begin{array}{ccccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ & & a'_{22}x_2 & + & a'_{23}x_3 & + & \cdots & + & a'_{2n}x_n & = & b'_2 \\ & & & & a''_{33}x_3 & + & \cdots & + & a''_{3n}x_n & = & b''_3 \\ & & & & \vdots & & & & \vdots & & \\ & & & & a''_{n3}x_3 & & \cdots & & a''_{nn}x_n & = & b''_n \end{array} \tag{2.39}$$

u.s.w. Es entstehen durch diese Vorgangsweise nach und nach Matrizen spezieller Form (siehe Abb. 2.2)

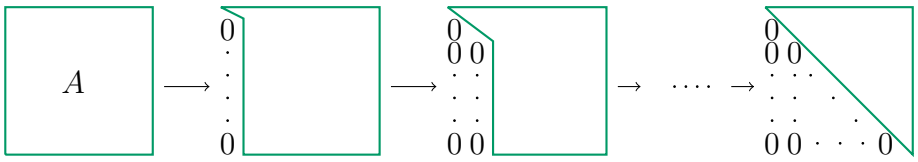



Abbildung 2.2: Reduktion auf Dreiecksgestalt

Zum Schluss erhält man ein gestaffeltes System, das äquivalent zum ursprünglichen System ist, und dessen Lösung gemäß (2.34) berechnet werden kann.

Die eben beschriebene Vorgangsweise ist ein systematischer Algorithmus und kann leicht programmiert werden.

### 2.5.1 Spaltenpivotisierung

**Beispiel 2.5.1.** *Führt dieser Algorithmus immer zum Ziel? Jein. Für*

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 2 & 1 & 3 \end{pmatrix}$$

*gilt* $a_{11} = 0$*; die Multiplikatoren* $\frac{a_{21}}{a_{11}}$ *und* $\frac{a_{31}}{a_{11}}$*, mit denen die erste Zeile multipliziert wird, bevor sie von der zweiten bzw. dritten Zeile abgezogen wird, können also nicht gebildet werden.*
*Es muss also, bevor der Eliminationsalgorithmus anläuft, die erste Zeile mit der zweiten oder dritten Zeile vertauscht werden.*

Genau dieselbe Situation tritt ein, wenn später während des Eliminationsvorganges in der Hauptdiagonale eine Null entsteht, sodass der entsprechende Multiplikator wegen Division durch 0 nicht gebildet werden kann. Dann muss die $i$-te Zeile mit einer weiter unten stehenden Zeile ($j$-te Zeile mit $j > i$) vertauscht werden, bei der ein nicht verschwindendes Element in der $i$-ten Spalte steht.
Nach diesem Vertauschungsprozess kann der Eliminationsvorgang fortgesetzt werden.

Ein echter Zusammenbruch würde nur entstehen, wenn in der $i$-ten Spalte auch unterhalb der Hauptdiagonalen lauter Nullen stehen: man kann dann die $i$-te Zeile mit der Null in der Hauptdiagonalen nicht wegtauschen.

**Satz 2.5.2.** *Ein Zusammenbruch des Systems tritt nur ein, wenn* $A$ *singulär ist, also* $Rang(A) < n$ *bzw.* $\det A = 0$ *gilt.*

**Numerische Problematik.** Bei rundungsfehlerfreiem, exaktem Rechnen müssen nur dann Zeilen vertauscht werden, wenn eine Null in der Hauptdiagonalen enstanden ist. Beim Rechnen in Maschinenarithmetik kann so eine Null jedoch aufgrund von Rundungsfehlern verfälscht werden, sodass statt der Null eine betragsmäßig sehr kleine Zahl in der Hauptdiagonalen steht.
Mit dieser nicht verschwindenden Größe könnte man den Eliminationsvorgang im Prinzip fortsetzen, würde dann jedoch offenbar etwas völlig Unsinniges rechnen. Vom Standpunkt der Numerik aus sollte man daher nicht nur Nullen, sondern auch betragsmäßig kleine Elemente wegtauschen. In diesem Sinn hat sich folgende Strategie bewährt:

**Definition 2.5.3.** *Bei der* **Spaltenpivotsuche** *mit Zeilentausch wird vor jedem Eliminationsschritt das betragsgrößte Element der* $i$*-ten Spalte bzgl. der Elemente in und unterhalb der Hauptdiagonalen gesucht. Steht dieses Element in der* $j$*-ten Zeile, wird die* $i$*-te mit der* $j$*-ten Zeile vertauscht. Erst anschließend wird der Eliminationsvorgang fortgesetzt.*

Diese Vorgangsweise lässt sich auch noch anders begründen: Im Allgemeinen ist es für Rundungsfehler ungünstig, wenn die Koeffizienten von $A$ von stark unterschiedlicher Größenordnung sind. Für die Matrix

$$\begin{pmatrix} 0.730 & \boxed{0.274} & 0.683 \\ 0.730 & \boxed{21.6} & 0.432 \\ 0.246 & 0.611 & 0.0723 \end{pmatrix}$$

etwa geht bei der Differenzbildung $21.6 - 0.274 = 21.3$ (auf 3 Stellen gerundet) die Information der beiden hinteren Stellen von $0.2\overline{|74}$ verloren.
Dass eine Matrix $A$ Koeffizienten unterschiedlicher Größenordnung besitzt, kann man im Allgemeinen nicht ändern, man wird jedoch vermeiden, dass während des Eliminationsvorgangs die Größenordnung der Koeffizienten von $A$ noch zusätzlich auseinandergezogen wird. Dies wird durch Spaltenpivotsuche mit Zeilentausch bewerkstelligt.

**Behauptung 2.5.4.** *Spaltenpivotsuche mit anschließendem Zeilentausch stellt sicher, dass alle Faktoren* $\frac{a_{ik}}{a_{kk}}$ *betragsmäßig* $\leq 1$ *sind.*

Außerdem können ohne diese Strategie auch harmlos erscheinende Multiplikatoren der Größenordnung 10 ungünstige Effekte haben, wenn sie im Laufe der Gaußelimination miteinander multipliziert werden (z.B. können 5 aufeinanderfolgende 10-er einen Effekt der Gößenordnung $10^5$ erzielen). In der Praxis, insbesondere für Systeme größerer Dimension, wo sich sehr viele harmlose Multiplikatoren zusammenmultiplizieren können, ist die Pivotstrategie daher unerlässlich.

**Behauptung 2.5.5.** *Nur Gaußeliminationsalgorithmen mit entsprechenden Pivotstrategien sind numerisch stabile Algorithmen.*

Oft wird die Spaltenpivotsuche mit anschließendem Zeilentausch mit der sogenannten **Skalierung** kombiniert. Sie wird anhand von Beispiel 2.5.6 erklärt:

**Beispiel 2.5.6.** *Gegeben ist das System*

$$\begin{aligned} 0.005x_1 &+ x_2 = 0.5 \\ x_1 &+ x_2 = 1. \end{aligned} \tag{2.40}$$

1. *Die exakte Lösung ist*

$$x_1 = \frac{500}{995} = 0.50251\ldots,\ x_2 = \frac{495}{995} = 0.49748\ldots \tag{2.41}$$

2. *Rechnung in* $\mathbb{M}(10, 2, \ldots)$ ohne Spaltenpivotsuche*: exakte Rechnung liefert*

$$\begin{aligned} 0.005x_1 + x_2 &= 0.5 \\ \underbrace{(1 - 200)}_{-199} x_2 &= \underbrace{(1 - 200 \cdot 0.5)}_{-99} \end{aligned}$$

   *und Rundung auf* 2 *Stellen ergibt*

$$\begin{aligned} 0.005\tilde{x_1} + \tilde{x}_2 &= 0.5 \\ -200\tilde{x}_2 &= -99 \end{aligned}$$

$$\begin{aligned} \Rightarrow\ \tilde{x}_2 &= 0.495 \quad \leftarrow \textit{ exaktes Divisionsergebnis} \\ &= 0.50 \quad \leftarrow \textit{ auf 2 Stellen gerundet} \\ \tilde{x}_1 &= 0. \end{aligned}$$

3. *Rechnung in* $\mathbb{M}(10, 2, \ldots)$*, aber* mit Spaltenpivotsuche*: vor der Elimination werden die Zeilen vertauscht*

$$\begin{aligned} x_1 + x_2 &= 1 \\ 0.005x_1 + x_2 &= 0.5 \end{aligned}$$

   *Exakte Rechnung liefert*

$$\begin{aligned} x_1 + x_2 &= 1 \\ \underbrace{(1 - 0.005)}_{0.995} x_2 &= \underbrace{(0.5 - 0.005)}_{0.495} \end{aligned}$$

*und Rundung auf 2 Stellen ergibt*

$$\begin{aligned} \tilde{x}_1 + \tilde{x}_2 &= 1 \\ \tilde{x}_2 &= 0.5 \end{aligned}$$

$$\Rightarrow \qquad \tilde{x}_2 = 0.5, \qquad \tilde{x}_1 = 0.5$$

*Rundet man das exakte Ergebnis (vgl. (2.41)) auf zwei Stellen, so erhält man ebenso* $\tilde{x}_1 = 0.5$, $\tilde{x}_2 = 0.5$.

*Man erkennt also tatsächlich die Überlegenheit der Variante mit Spaltenpivotsuche.*

*Diese vorteilhafte Wirkung der Spaltenpivotsuche kann aber sehr leicht zerstört werden, wenn man die erste Zeile von (2.40) mit einer hinreichend großen Zahl, etwa* 400 *multipliziert:*

$$\begin{aligned} 2x_1 + 400x_2 &= 200 \\ x_1 + x_2 &= 1 \end{aligned} \tag{2.42}$$

*Wenn durch irgendeinen Zufall das System statt in der Form (2.40) in der (dazu äquivalenten) Form (2.42) gegeben wäre, würde trotz Spaltenpivotsuche wegen* $2 > 1$ *kein Zeilentausch erfolgen und daher die Elimination rundungsfehlermässig in der unverteilhaften Variante ablaufen.*

*Abhilfe schafft die* **Zeilenskalierung***: Vor Beginn des Eliminationsprozesses wird jede Zeile mit einem Faktor* $d_i$ *multipliziert, sodass die maximalen Elemente aller Zeilen dieselbe Größenordnung haben,*

$$z.B. \quad d_i = \frac{1}{\max_{1\le j\le n} |a_{ij}|} \quad oder \quad d_i = \frac{1}{\sum_{j=1}^{n} |a_{ij}|}.$$

*Die so skalierte Version von (2.42) ist dann*

$$\begin{aligned} 0.005x_1 + x_2 &= 0.5 \\ 0.5x_1 + 0.5x_2 &= 0.5 \end{aligned} \tag{2.43}$$

*Lösung von (2.43) in zweistelliger Arithmetik liefert wieder* $\tilde{x}_1 = 0.5$, $\tilde{x}_2 = 0.5$.

## 2.6 LU-Zerlegung und der Crout-Algorithmus

Will man ein Gleichungssystem der Form $A\vec{x} = \vec{b}$ mit der regulären quadratischen Matrix $A$ für verschiedene Inhomogenitäten $\vec{b}$ lösen, ohne die inverse Matrix $A^{-1}$ zu berechnen, so macht es Sinn, die erfolgten Umformungen in einer Matrix zu speichern. Dies geschieht mit Hilfe der $LU$-Zerlegung, die eine übliche Implementierung des Gauß-Verfahrens ist.

Vorerst wird gefordert, dass das Gauß-Verfahren ohne Zeilenvertauschungen möglich ist.

**Definition 2.6.1.** *Die untere (englisch* lower*) Dreiecksmatrix*

$$L = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ l_{21} & 1 & \ddots & & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ l_{n1} & \cdots & \cdots & l_{n,n-1} & 1 \end{pmatrix} \in K^{n\times n}$$

*und die obere (englisch* upper*) Dreiecksmatrix*

$$U = \begin{pmatrix} u_{11} & \cdots & \cdots & \cdots & u_{1n} \\ 0 & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & u_{nn} \end{pmatrix} \in K^{n\times n}$$

*heißen LU-***Zerlegung** *der regulären Matrix $A \in K^{n\times n}$ $(K = \mathbb{C}, \mathbb{R})$, wenn*

$$A = LU$$

*gilt.*

Die Matrix $L$ dient dabei dem Speichern der Umformungsschritte des Gauß-Verfahrens, während $U$ die resultierende obere Dreiecksform hat.
Mithilfe der $LU$-Zerlegung lässt sich das Gleichungssystem $LU\vec{x} = A\vec{x} = \vec{b}$ umschreiben in die beiden Systeme

$$L\vec{y} = \vec{b} \quad \text{und} \quad U\vec{x} = \vec{y}.$$

Die Lösung erfolgt in zwei Schritten:

1. Mittels Vorwärtselimination wird $L\vec{y} = \vec{b}$ gelöst:

$$y_1 = b_1, \quad y_i = b_i - \sum_{j=1}^{i-1} l_{ij} y_j,\ i = 2, \ldots, n$$

   Dies ergibt sich direkt aus der unteren Dreiecksgestalt der Matrix $L$. Die Gleichungen haben nämlich die Form

$$y_1 = b_1, \quad l_{21}y_1 + y_2 = b_2, \quad \ldots, \quad \sum_{j=1}^{n-1} l_{nj} y_j + y_n = b_n.$$

2. Mittels Rückwärtselimination wird $U\vec{x} = \vec{y}$ gelöst:

$$x_n = \frac{y_n}{u_{nn}}, \quad x_i = \frac{1}{u_{ii}} \left( y_i - \sum_{j=i+1}^{n} u_{ij} x_j \right),\ i = n-1, \ldots, 1$$

   Erneut ergibt sich dies aus der oberen Dreiecksgestalt der Matrix $U$.

**Beispiel 2.6.2.** *Gegeben sind $LU = A \in \mathbb{R}^{n\times n}$ regulär sowie $\vec{b} \in \mathbb{R}^3$, gesucht ist die Lösung $\vec{x}$ von $A\vec{x} = \vec{b}$, wobei*

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 3 & 2 & 1 \\ 0 & 2 & 3 \\ 0 & 0 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} 3 & 2 & 1 \\ 6 & 6 & 5 \\ 9 & 10 & 10 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

.

1. *Im ersten Schritt erfolgt die Vorwärtselimination. Es ergibt sich*

$$\begin{array}{lll} y_1 = & b_1 & = 1, \\ y_2 = & b_2 - l_{21}y_1 & = 2 - 2 \cdot 1 = 0, \\ y_3 = & b_3 - l_{31}y_1 - l_{32}y_2 & = 3 - 3 \cdot 1 - 2 \cdot 0 = 0. \end{array}$$

2. *Die Rückwärtselimination im zweiten Schritt führt auf*

$$\begin{array}{lll} x_3 = y_3/u_{33} & = 0/1 & = 0, \\ x_2 = (y_2 - u_{23}x_3)/u_{22} & = (0 - 3 \cdot 0)/2 & = 0, \\ x_1 = (y_1 - u_{12}x_2 - u_{13}x_3)/u_{11} & = (1 - 2 \cdot 0 - 1 \cdot 0)/3 & = 0.\dot{3}. \end{array}$$

Um die *LU*-Zerlegung einer regulären Matrix zu bestimmen, wird im Allgemeinen der **Crout-Algorithmus** verwendet. Der Crout-Algorithmus ist im Prinzip nur eine Anwendung des Gauß-Verfahrens mit Speicherung der durchgeführten Schritte.

**Behauptung 2.6.3.** *Sei $A \in K^{n\times n}$ eine reguläre Matrix und sei das Gauß-Verfahren ohne Zeilenvertauschung durchführbar. Dann liefert derselbe Algorithmus eine LU-Zerlegung von A. Die Zerlegung ist eindeutig.*

**Beweis.** Da für die Elimination keine Zeilenvertauschungen notwendig sind, gilt $a_{11} \neq 0$. Das Element $a_{i1}$ kann durch die Subtraktion von $l_{i1} := (a_{i1}/a_{11})$-mal der ersten Zeile von der zweiten Zeile eliminiert werden. Die ganze erste Spalte ab der zweiten Zeile wird also durch die Multiplikation mit

$$L_1 := \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -l_{n1} & 0 & \cdots & 1 \end{pmatrix}$$

eliminiert, man erhält

$$L_1 A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a'_{n2} & \cdots & a'_{nn} \end{pmatrix},$$

wobei $a'_{ij} = a_{ij} - l_{i1}a_{1j}$.
Weil keine Zeilenvertauschungen notwendig sind, gilt auch $a'_{22} \neq 0$. Wie im ersten Schritt kann die ganze zweite Spalte ab der dritten Zeile durch Multiplikation mit

$$L_2 := \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & -l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & -l_{n2} & 0 & \cdots & 1 \end{pmatrix}$$

eliminiert werden, wobei $l_{i2} := a'_{i2}/a'_{22}$.

Nach $n-1$ Schritten erhält man

$$L_{n-1}L_{n-2}\dots L_2L_1A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} \\ 0 & 0 & a''_{33} & \cdots & a''_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n-1)} \end{pmatrix} =: U.$$

Die Matrizen $L_j$ sind jeweils Einheitsmatrizen, die zusätzlich in der $j$-ten Spalte unter der Diagonale die Werte $l_{ij} = a_{ij}^{(j-1)}/a_{jj}^{(j-1)}$, $i = j+1,\dots,n$ haben, $a_{ij}^{(j-1)}$ bezeichnet die Einträge der $j$-ten Zeile der Matrix $L_{j-1}L_{j-2}\dots L_2L_1A$.

Durch Matrixmultiplikation ist leicht zu sehen, dass die Produktmatrix $L_{n-1}L_{n-2}\dots L_2L_1$ und ihre Inverse ebenfalls untere Dreiecksmatrizen mit 1-Einträgen in der Diagonale sind. Mit der Definition

$$L := (L_{n-1}L_{n-2}\dots L_2L_1)^{-1}$$

gilt $LU = A$.

Die Eindeutigkeit der Zerlegung ist ebenso elementar zu zeigen, für das Verständnis des Algorithmus allerdings nicht notwendig. □

Für die Implementierung der $LU$-Zerlegung startet man mit den Matrizen $L = I_n$ (Einheitsmatrix der Dimension $n$) und $U = A$. In einer Schleife werden die Spalten $j = 1,\dots,n$ durchlaufen. In jedem Schritt werden zuerst die Einträge von $L$ in dieser Spalte berechnet, also

$$l_{ij} = \left(a_{ij} - \sum_{k=0}^{j-1} l_{ik}u_{kj}\right)/u_{jj}, \quad i = j+1,\dots,n-1,$$

dann jene von $U$, also

$$u_{ij} = a_{ij} - \sum_{k=0}^{i-1} l_{ik}u_{kj}, \quad i = 1,\dots,j.$$

Algorithmus 2.6.4 zeigt eine mögliche Implementierung des Crout-Algorithmus.

**Algorithmus 2.6.4** (CROUT).

```
Crout(A)

L := I
U := A

for Spalten j = 1,...,n
      for Zeilen i = j+1,...,n
            l_ij := u_ij/u_jj
            u_ij := 0
            for Spalten k = j+1,...,n
                  u_ik := u_ik - l_ij u_jk
```

```
            end
        end
end
```

**Beispiel 2.6.5.** *Gegeben ist die Matrix* $A = \begin{pmatrix} 1 & 1 & -2 \\ 1 & 3 & -1 \\ 1 & 5 & 1 \end{pmatrix}$, *gesucht ist ihre LU-Zerlegung.*

*Der Algorithmus beginnt mit der Initialisierung*

$$L = I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad U = A = \begin{pmatrix} 1 & 1 & -2 \\ 1 & 3 & -1 \\ 1 & 5 & 1 \end{pmatrix}.$$

*Es wird nun die Schleife über die Spalten durchlaufen. Jeder Durchlauf entspricht der Elimination der entsprechenden Spalte unterhalb der Diagonale in A.*

$j = 1$*: In jeder Zeile unterhalb der Diagonale wird zunächst der Eintrag in L berechnet, dann jener in U. Anschließend werden die Werte in U in der entsprechenden Zeile aktualisiert (vgl.* $a'_{ij}$ *im Beweis von Behauptung 2.6.3). Die neuen Matrizen nach diesem Durchlauf sind*

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 0 & 4 & 3 \end{pmatrix}.$$

$j = 2$*: Analoges Vorgehen führt im zweiten Schritt auf die Matrizen*

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

$j = 3$*: Im dritten Schritt ist nichts mehr zu tun, da in der letzten Spalte keine Werte eliminiert werden müssen und die Matrizen L und U bereits die gewünschte Gestalt besitzen.*

*Durch Matrizenmultiplikation sieht man leicht, dass* $LU = A$ *gilt.*

## LU-Zerlegung mit Pivotisierung

Bisher wurde in diesem Abschnitt angenommen, dass das Gauß-Verfahren ohne Zeilenvertauschungen und damit ohne Pivotisierung durchgeführt werden kann. Da dies im Allgemeinen aber nicht möglich ist, wird die *LU*-Zerlegung meist mit einer Spaltenpivotisierung implementiert.

Falls $A \in K^{n\times n}$ eine reguläre Matrix ist, auf die das Gauß-Verfahren *nicht* ohne Zeilenvertauschungen durchgeführt werden kann, so existiert eine *LU*-Zerlegung nur in der Form

$$P_\pi A = LU,$$

wobei die Matrix $P_\pi$ eine Permutationsmatrix ist.

**Definition 2.6.6.** *Eine bijektive Abbildung $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ heißt Permutation. Mit $\pi_{ij}$ wird die Vertauschung der Elemente $i$ und $j$ bezeichnet. Die Matrix*

$$P_\pi = \left(e_{\pi(1)}, e_{\pi(2)}, \ldots, e_{\pi(n)}\right)$$

*mit den permutierten Einheitsvektoren als Spalten heißt die der Permutation $\pi$ zugeordnete* **Permutationsmatrix**.

Multiplikation mit $P_\pi$ von links permutiert die Zeilen einer Matrix, Multiplikation von rechts die Spalten. $P_\pi A$ ist daher die ursprüngliche Matrix $A$ mit permutierten Zeilen.

Auch die $LU$-Zerlegung mit Zeilenvertauschungen ergibt sich direkt durch die Anwendung des Gauß-Verfahrens.

**Behauptung 2.6.7.** *Sei $A \in K^{n\times n}$ eine reguläre Matrix. Dann liefert der Gauß-Algorithmus eine $LU$-Zerlegung von $P_\pi A$ mit einer Permutationsmatrix $P_\pi$.*

**Beweis.** Es wird das Gauß-Verfahren mit Spaltenpivotsuche und Zeilenvertauschungen durchgeführt. Es unterscheidet sich nur dadurch von jenem im Beweis von Behauptung 2.6.3, dass vor jeder Multiplikation mit einer Matrix $L_i$ die Multiplikation mit einer Permutationsmatrix $P_{\pi_{i+1,j(i)}}$ (im Folgenden der Einfachheit halber mit $P_i$ bezeichnet) eingeschoben wird. Dadurch wird das betragsgrößte Element der Spalte in die Diagonale verschoben.

Nach $n-1$ Schritten erhält man daher

$$L_{n-1}P_{n-1}L_{n-2}P_{n-2}\ldots L_2P_2L_1P_1A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} \\ 0 & 0 & a''_{33} & \cdots & a''_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n-1)} \end{pmatrix} =: U.$$

Durch Nachrechnen ist leicht zu sehen, dass $P_i^{-1} = P_i$, also $P_iP_i = I$. Durch Einschieben der Identität gilt

$$L_{n-1}P_{n-1}L_{n-2}\underbrace{P_{n-1}P_{n-1}}_{=I}P_{n-2}L_{n-3}\underbrace{P_{n-2}P_{n-1}P_{n-1}P_{n-2}}_{=I}P_{n-3}L_{n-4}\ldots A = U$$

und durch Umklammern

$$L_{n-1}\underbrace{(P_{n-1}L_{n-2}P_{n-1})}_{:=L'_{n-2}}\underbrace{(P_{n-1}P_{n-2}L_{n-3}P_{n-2}P_{n-1})}_{:=L'_{n-3}}\ldots\underbrace{(P_{n-1}\ldots P_2L_1P_2\ldots P_{n-1})}_{:=L'_1}(P_{n-1}\ldots P_1)\,A = U,$$

also

$$L_{n-1}L'_{n-2}\ldots L'_1\,(P_{n-1}\ldots P_1)\,A = U.$$

Die Matrizen $L'_i$ haben wieder die gewünschte Form (nachrechnen). Mit den Definition

$$L := \left(L_{n-1}L'_{n-2}\ldots L'_1\right)^{-1} \quad \text{und} \quad P_\pi := P_{n-1}\ldots P_1$$

erhält man schließlich $P_\pi A = LU$. □

Eine Implementierung des $LU$-Zerlegung mit Zeilenvertauschungen zeigt Algorithmus 2.6.8.

**Algorithmus 2.6.8** (Crout mit Zeilenvertauschungen).

```
CroutPivot(A)

L := I
U := A

for Spalten j = 1,...,n
      j_piv := j
      for Zeilen i = j+1,...,n
            if |u_ij^(j-1)| > |u_{j_piv j}^(j-1)|
                  j_piv := i
            end
            erstelle P_j, vertausche Zeilen j, j_piv
      end
      for Zeilen i = j+1,...,n
            l_ij := u_ij / u_jj
            u_ij := 0
            for Spalten k = j+1,...,n
                  u_ik := u_ik - l_ij u_jk
            end
      end
end
P_π := P_{n-1} ... P_1
```

**Beispiel 2.6.9.** *Es soll erneut die Matrix $A$ aus Beispiel 2.6.5 zerlegt werden, diesmal mit Pivotisierung. Da die Einträge der ersten Spalte alle gleich sind, ist der Ablauf bis zum ersten Schleifendurchlauf gleich wie zuvor. Erst im zweiten Durchgang tritt eine Pivotisierung auf.*

*$j = 2$: Da $u_{32} = 4 > 2 = u_{22}$, werden die Matrizen*

$$P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \textit{und} \quad U = \begin{pmatrix} 1 & 1 & -2 \\ 0 & 4 & 3 \\ 0 & 2 & 1 \end{pmatrix}$$

*erstellt bzw. aktualisiert. Das weitere Vorgehen ist wie zuvor und man erhält*

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0.5 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 1 & -2 \\ 0 & 4 & 3 \\ 0 & 0 & -0.5 \end{pmatrix}, \quad P_\pi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

*Durch Nachrechnen überzeugt man sich leicht von $P_\pi A = LU$.*

## 2.7 Rundungsfehler bei der Gaußelimination

**Satz 2.7.1** (v. Wilkinson). *Gegeben sei $A\vec{x} = \vec{b}$ ($A \in \mathbb{R}^{n\times n}$, regulär) mit der exakten (d.h. rundungsfehlerfreien) Lösung $\vec{x}$. Weiters sei $\tilde{\vec{x}}$ die rundungsfehlerbehaftete Lösung, die durch Lösung von $A\vec{x} = \vec{b}$ mittels Gaußelimination mit Spaltenpivotsuche in einer bestimmten Maschinenarithmetik zustande kommt. Dann lässt sich $\tilde{\vec{x}}$ interpretieren als die exakte, rundungsfehlerfreie Lösung eines gestörten Systems*

$$(A + \Delta A)\tilde{\vec{x}} \;=\; \vec{b} \tag{2.44}$$

*mit*

$$\frac{\|\Delta A\|}{\|A\|} \;\leq\; 1.01\,(n^3 + 3n^2)\,\gamma\, eps. \tag{2.45}$$

Dabei ist

$$\gamma \;:=\; \frac{1}{\|A\|}\max_{i,j,k}|a_{ij}^{(k)}|, \tag{2.46}$$

wobei die $a_{ij}^{(k)}$ die während dem Eliminationsvorgang auftretenden Elemente

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \quad \equiv \quad \begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \cdots & a_{1n}^{(0)} \\ a_{21}^{(0)} & a_{21}^{(0)} & \cdots & a_{2n}^{(0)} \\ \vdots & & \vdots & \\ a_{n1}^{(0)} & a_{n2}^{(0)} & \cdots & a_{nn}^{(0)} \end{pmatrix} \rightarrow \begin{matrix}\text{erster El.}\\ \text{Schritt}\end{matrix} \rightarrow$$

$$\begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \cdots & a_{12}^{(0)} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix} \overset{\text{2.Schritt}}{\rightarrow} \begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} & \cdots & & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & \cdots & & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & & \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix} \rightarrow \text{u.s.w}$$

sind (vgl. auch (2.35) bis (2.39), wo die $a_{ij}^{(1)}$ und $a_{ij}^{(2)}$ mit $a'_{ij}$ bzw. $a''_{ij}$ bezeichnet werden).

Weiters ist *eps* aus (2.45) definiert als

$$eps \;:=\; \begin{cases} \text{Basis}^{-(\text{Mantissenlänge}-1)} & \ldots \text{ Abschneidearithmetik,} \\ \frac{1}{2}\cdot\text{Basis}^{-(\text{Mantissenlänge}-1)} & \ldots \text{ Rundearithmetik.} \end{cases} \tag{2.47}$$

(vgl. 1.20).

**Definition 2.7.2.** *Die Idee hinter dem Satz v. Wilkinson, Effekte von Rundungsfehlern als Datenfehlereffekte zu interpretieren ($\tilde{\vec{x}}$ soll das exakte Ergebnis des Systems mit der gestörten Matrix $A+\Delta A$ sein), wird als* **Rückwärtsanalyse** *bezeichnet.*

Die Konditionsabschätzung (2.30) ermöglicht nun sofort die *Rundungsfehlerabschätzung*

$$\frac{\|\tilde{\vec{x}} - \vec{x}\|}{\|\tilde{\vec{x}}\|} \leq \underbrace{\|A\|\,\|A^{-1}\|}_{\kappa(A)}\frac{\|\Delta A\|}{\|A\|} \;\leq \kappa(A)\,1.01\,(n^3 + 3n^2)\,\gamma\, eps. \tag{2.48}$$

Das ist eine Schranke vom Typ

$$\text{Konditionszahl} \cdot \text{Faktor} \cdot \text{Anzahl der Rechenoperationen} \cdot \text{Maschinengenauigkeit } \text{`}eps\text{'}, \tag{2.49}$$

wobei der Faktor $\gamma$ in den Fällen, in denen während der Elimination keine sehr großen Elemente $a_{ij}^{(k)}$ auftreten, als *moderat* bezeichnet werden kann. Die Anzahl der Rechenoperationen bei der Gaußelimination ist $\sim \frac{n^3}{3}$. Es folgt

**Satz 2.7.3.** *Die Gaußelimination mit Spaltenpivotsuche ist ein* **numerisch stabiler** *Algorithmus.*

**Definition 2.7.4. Numerische Stabilität** *ist gegeben, wenn eine Rundungsfehlerschranke vom Typ (2.49) mit moderatem Faktor existiert.*

**Bemerkung.** Dass in solch eine Schranke Maschinengenauigkeit *eps* und Anzahl der Rechenoperationen einfließen, folgt in natürlicher Weise. Dass zusätzlich nur ein moderater Faktor sowie die Konditionszahl des Problems aufscheinen, besagt, dass die Rundungsfehlersensitivität nicht schlechter liegt als die Datenfehlersensitivität, dass sich also Rundungsfehler (und daraus resultierende Datenstörungen) im Lauf des Algorithmus nicht schlimmer auswirken als Störungen des ursprünglichen Systems. Offensichtlich ist dies das Beste, das man für einem Algorithmus erhoffen darf.

Die Abschätzung (2.48) ist eine mathematische Aussage, die numerische Stabilität des Gaußalgorithmus mit Spaltenpivotsuche für alle (regulären) linearen Gleichungssysteme sicherstellt. Sie wird jedoch kaum herangezogen, um das Rundungsfehlerniveau in konkreten Fällen abzuschätzen.

Der Grund dafür ist, dass sie in den meisten konkreten Fällen zu pessimistisch ist: In der Realität ergibt sich bei der Lösung eines Systems mittels Gauß-Elimination meistens eine gewisse Kompensation der Rundungsfehler (wenn z.B. bei einem Rechenschritt aufgerundet wird und beim nächsten Rechenschritt wieder abgerundet, heben sich die entsprechenden Rundungsfehler teilweise weg). Die Schranke, die durch eine reine Betragsabschätzung zustande kommt, muss allerdings auch für den unwahrscheinlichen Fall (der theoretisch wirklich eintreten kann) gelten, dass es keine Kompensationen gibt, sondern alle einzelnen Rundungsfehler in dieselbe Richtung liegen und eine völlige Akkumulation dieser Fehler passiert.
Die folgende Behauptung liefert eine Möglichkeit zur realistischeren Fehlerabschätzung.

**Behauptung 2.7.5.** *Liegt in einem konkretem Fall die rundungsfehlerbehaftete Lösung $\tilde{x}$ vor, kann a-posteriori eine realistischere Abschätzung des Rundungsfehlerniveaus gefunden werden, z.B. aufgrund von folgender Überlegung:*

$$\begin{aligned} \tilde{x} - x &= A^{-1}A(\tilde{x} - x) \quad \Rightarrow \\ \frac{\|\tilde{x} - x\|}{\|\tilde{x}\|} &\leq \|A^{-1}\| \frac{\|A\tilde{x} - \overbrace{Ax}^{b}\|}{\|\tilde{x}\|} = \\ &= \underbrace{\|A\|\|A^{-1}\|}_{\kappa(A)} \underbrace{\frac{\|A\tilde{x} - b\|}{\|A\|\|\tilde{x}\|}}_{*)}. \end{aligned} \tag{2.50}$$

*Dabei ist $*$) ein berechenbarer Ausdruck, wenn $\tilde{x}$ gegeben. Der Zähler, das Residuum $A\tilde{x} - b$, muss in partieller doppelter Genauigkeit berechnet werden!*

**Geometrische Interpretation von Gleichungssystemen.** Wir betrachten jetzt wieder quadratische, $n \times n$-Gleichungssysteme (**??**). Diese sind genau dann eindeutig lösbar, wenn die Matrix $A$ vollen Rang besitzt, also $\text{Rang}(A) = n$ gilt. In diesem Fall spricht man von einem **regulären Gleichungssystem** .
Jede einzelne Gleichung von

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

stellt eine Ebene im $\mathbb{R}^3$ dar, mit Normalvektor $(a_{i1}, a_{i2}, a_{i3})$, $i = 1, 2, 3$. Jene Punkte $(x_1, x_2, x_3)^T \in \mathbb{R}^3$, welche zwei Gleichungen erfüllen, bilden den Schnitt der entsprechenden Ebenen und liegen auf einer Geraden $g$. Jene Punkte $(x_1, x_2, x_3)^T \in \mathbb{R}^3$, welche alle drei Gleichungen erfüllen, also auf $g$ und in der durch die dritte Gleichung dargestellten Ebene liegen, sind der im regulären Fall eindeutige Durchstoßpunkt von $g$ mit der dritten Ebene.

Im nicht regulären Fall gibt es folgende **Entartungsmöglichkeiten:**

- Rangabfall um 1, d.h.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

  hat nur 2 linear unabhängige Zeilen (Rang von $A = 3-1 = 2$): Der Normalvektor $(a_{31}, a_{32}, a_{33})$ der dritten Ebene ist eine Linearkombination der ersten beiden Normalvektoren: d.h. alle drei Normalvektoren liegen in einer Ebene. Zwei Möglichkeiten:

  a) Alle drei Ebenen schneiden sich in einer Geraden: Statt eindeutiger Lösung liegt eindimensionale Lösungsschar vor: (Dimension der Lösungschar: $n - \text{Rang}(A) = 3 - 2 = 1$)

  b) Die Schnittgeraden von jeweils zwei Ebenen sind parallel zueinander. Es gibt keine Lösung des Gleichungssystems

- Rangabfall um 2, d.h. $A$ hat den Rang $3 - 2 = 1$. Der zweite und dritte Normalvektor sind beide Vielfache des ersten Normalvektors. Alle drei Ebenen sind parallel. Zwei Möglichkeiten:

  a) Alle drei Ebenen fallen zusammen: Lösungsschar ist die Menge aller Punkte dieser Ebene. (Dimension der Lösungsschar: $n - \text{Rang}(A) = 3 - 1 = 2$).

  b) Die parallelen Ebenen fallen nicht zusammen: es gibt keine Lösungen des Systems.

In der reinen Mathemaitk ist der Begriff des Ranges einer Matrix ein scharfer Begriff. Die Zeilen einer Matrix sind entweder linear unabhängig (die Matrix hat vollen Rang) oder sie sind linear abhängig, (d.h. es gibt eine nichttriviale, verschwindende Linearkombination, die Matrix hat einen Rangabfall). In der Numerik bleibt diese Schärfe jedoch nicht bestehen: Werden die Koeffizienten einer Matrix in eine Computer eingelesen, müssen sie entsprechend der Maschinenarithmetik gerundet werden, die Koeffizienten werden also – abhängig vom verwendeten Computer – verändert. Lineare (Un-) Abhängigkeiten können dadurch verändert werden.

**Beispiel 2.7.6.**

$$A = \begin{pmatrix} \frac{7}{16} & \frac{3}{5} & \frac{3}{2} \\ \frac{7}{32} & \frac{4}{5} & 2 \\ \frac{7}{8} & \frac{6}{5} & 3 \end{pmatrix} = \begin{pmatrix} 0.4375 & 0.6 & 1.5 \\ 0.21875 & 0.8 & 2 \\ 0.875 & 1.2 & 3 \end{pmatrix}$$

*Es gibt die nichttriviale Linearkombination*

$$2 \cdot \left(\frac{7}{16}, \frac{3}{5}, \frac{3}{2}\right)^\top + 0 \cdot \left(\frac{7}{32}, \frac{4}{5}, 2\right)^\top + (-1) \cdot \left(\frac{7}{8}, \frac{6}{5}, 3\right)^\top = (0,0,0)^\top$$

*d.h. $A$ hat nicht vollen Rang. Nach Rundung von $A$ in $\mathbb{M}(10,3,\ldots)$ ergibt sich die reguläre Matrix*

$$\tilde{A} = \begin{pmatrix} 0.438 & 0.6 & 1.5 \\ 0.219 & 0.8 & 2 \\ 0.875 & 1.2 & 3 \end{pmatrix}$$

*Bei Rundung von $A$ in $\mathbb{M}(10,4,\ldots)$ hingegen erhält man*

$$\tilde{A} = \begin{pmatrix} 0.4375 & 0.6 & 1.5 \\ 0.2188 & 0.8 & 2 \\ 0.875 & 1.2 & 3 \end{pmatrix}$$

*und es gilt dieselbe nichttriviale Linearkombination wie bei $A$ selbst, d.h. $\tilde{A}$ hat in diesem Fall auch keinen vollen Rang.*

Aus Beispiel 2.7.6 ergibt sich die folgende Definition.

**Definition 2.7.7.** *Eine in eine Arithmetik gerundete Matrix $\tilde{A}$ heißt* **numerisch regulär***, wenn in der Menge aller Matrizen $A$, die nach Rundung in die entsprechende Arithmetik $\tilde{A}$ ergeben, keine singuläre Matrix ist.*

Nur im Fall einer numerisch regulären Matrix $\tilde{A}$ ist sichergestellt, dass auch die „wahre“ Matrix $A$ regulär ist und das „wahre“ Gleichungssystem eine eindeutige Lösung besitzt. Nur in diesem Fall ist es sinnvoll, das Gleichungssystem am Computer zu lösen. Ist $\tilde{A}$ hingegen nicht numerisch regulär, könnte das ursprüngliche Gleichungssystem eine singuläre Gleichungsmatrix besitzen und daher nicht lösbar sein. Ein numerischer Lösungsversuch würde in diesem Fall zu „falschen“ Lösungen führen und wäre daher sinnlos.

## 2.8 Lineares Ausgleichsproblem

Zunächst einige wichtige Begriffe, die für Ausgleichsprobleme bezüglich der $\|\cdot\|_2$-Norm (siehe (2.13) auf Seite 29) wesentlich sind:

**Definition 2.8.1.** *Das* **Skalarprodukt** *zweier Vektoren $\vec{x} \in \mathbb{R}^n$ und $\vec{y} \in \mathbb{R}^n$ ist definiert durch*

$$\vec{x} \cdot \vec{y} \;:=\; \sum_{i=1}^{n} x_i y_i \tag{2.51}$$

*bzw.*

$$\vec{x} \cdot \vec{y} = \vec{x}^\top \vec{y} = (x_1, \ldots, x_n) \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \sum_{i=1}^{n} x_i y_i. \tag{2.52}$$

**Definition 2.8.2.** *Zwei Vektoren $\vec{x} \in \mathbb{R}^n$, $\vec{y} \in \mathbb{R}^n$ heißen* **orthogonal** *(Schreibweise $\vec{x} \perp \vec{y}$), falls gilt:*

$$\vec{x} \cdot \vec{y} \;=\; 0 \tag{2.53}$$

**Beispiel 2.8.3.** *Gegeben sind*

$$\vec{x} = \begin{pmatrix} r_x \cos\varphi \\ r_x \sin\varphi \end{pmatrix} \in \mathbb{R}^2, \quad \vec{y} = \begin{pmatrix} r_y(-\sin\varphi) \\ r_y \cos\varphi \end{pmatrix} \in \mathbb{R}^2.$$

*Aus*

$$\vec{x} \cdot \vec{y} \;=\; r_x\, r_y\, \cos\varphi\,(-\sin\varphi) + r_x\, r_y\, \sin\varphi \cos\varphi \;=\; 0$$

*folgt die Orthogonalität der beiden Vektoren.*

**Definition 2.8.4.** *Sei $S$ ein linearer Unterraum des $\mathbb{R}^n$. Die Menge $T$ aller Vektoren $\vec{t} \in \mathbb{R}^n$, die orthogonal zu sämtlichen Vektoren $\vec{s} \in S \subset \mathbb{R}^n$ sind, heißt das* **orthogonale Komplement** *zu $S$.*

**Satz 2.8.5.** *Besitzt $S$ die Dimension $k_S$, so hat $T$ die Dimension $k_T = n - k_S$, d.h. es gilt*

$$n \;=\; k_S + k_T. \tag{2.54}$$

*Jedes $\vec{x} \in \mathbb{R}^n$ lässt sich in eindeutige Weise schreiben als*

$$\vec{x} = \vec{s} + \vec{t}, \quad \vec{s} \in S,\ \vec{t} \in T \tag{2.55}$$

*wobei $\vec{s}$ die* **Projektion** *von $\vec{x}$ auf $S$ und $\vec{t}$ die Projektion von $\vec{x}$ auf $T$ heißt! Man schreibt auch:*

$$\mathbb{R}^n \;=\; S \oplus T \tag{2.56}$$

*d.h. $\mathbb{R}^n$ ist die* **direkte Summe** *der orthogonalen Teilräume $S$ und $T$.*

**Beweisskizze**: Man dreht die Standardbasis (2.3) so, dass möglichst viele Basisvektoren in $S$ liegen. Diese bilden dann eine Orthonormalbasis von $S$, die anderen gedrehten Einheitsvektoren bilden eine Orthonormalbasis von $T$. Man erhält dann sofort die Projektionen $\vec{s}$ und $\vec{t}$ eines beliebigen Vektors $\vec{x} \in \mathbb{R}^n$ (vgl. (2.55)). Zunächst stellen wir den Vektor $\vec{x}$ in der gedrehten Basis dar:

$$\vec{x} \;=\; \sum_{i=1}^{n} \bar{x}_i \vec{\bar{e}_i}$$

wobei $\vec{\bar{e}_i}$ die verdrehten Einheitsvektoren sein sollen; sei die Basis von $S$ durch $\{\vec{\bar{e}_1}, \ldots, \vec{\bar{e}_{k_S}}\}$ gegeben und die Basis von $T$ durch $\{\vec{e_{k_S+1}}, \ldots, \vec{\bar{e}_n}\}$, so haben wir

$$\vec{s} = \sum_{i=1}^{k_S} \bar{x}_i \vec{\bar{e}_i}, \quad \vec{t} = \sum_{i=k_S+1}^{n} \bar{x}_i \vec{\bar{e}_i}.$$

□

**Definition 2.8.6.** *Das lineare Gleichungssystem $A\vec{x} = \vec{b}$, $A \in \mathbb{R}^{m\times n}$, $\vec{b} \in \mathbb{R}^m$, $\vec{x} \in \mathbb{R}^n$ sei widersprüchlich (d.h. $m > Rang(A)$, $\vec{b} \notin Bild(A)$). Das* **lineare Ausgleichsproblem** *besteht darin, jenes $\vec{x}$ (jene $\vec{x}$) suchen, das (die) die Gleichung so gut wie möglich löst (lösen), d.h. die Forderung lautet*

$$\|A\vec{x} - \vec{b}\|_2 \;=\; \min. \tag{2.57}$$

Es stellt sich die Frage nach Existenz und Eindeutigkeit von Lösungen sowie nach der Dimension von Lösungsscharen beim Ausgleichsproblem.

Zunächst beweisen wir folgenden Satz.

**Satz 2.8.7.** *Für $A \in R^{m\times n}$ gilt*

*a)*

$$\mathbb{R}^m = Bild(A) \oplus Kern(A^\top) \tag{2.58}$$

*b)*

$$\mathbb{R}^n = Bild(A^\top) \oplus Kern(A) \tag{2.59}$$

**Beweis.**
a) $O$ sei das orthogonale Komplement von $Kern(A^\top)$. Wir zeigen zunächst: $Bild(A) \subset O$. Es gilt für

$$\vec{y} \in Bild(A) \subset \mathbb{R}^m, \quad \vec{z} \in Kern(A^\top) \subset \mathbb{R}^m \ ^{1)}$$
$$\vec{y} \cdot \vec{z} = \vec{y}^\top \vec{z} = (A\vec{x})^\top \vec{z} = \vec{x}^\top A^\top \vec{z} = \vec{0}$$

$\Rightarrow \ \vec{y} \perp \vec{z}$ d.h. $Bild(A) \subset O$. Aus (2.23) folgt: $m = \dim(Kern(A^\top)) + Rang(A) = \dim(Kern(A^\top)) + \dim(Bild(A))$ d.h. durch die beiden linearen Teilräume $Kern(A^\top)$ und $Bild(A)$ wird der $\mathbb{R}^m$ aufgespannt und es gilt $Bild(A) = O \Rightarrow$ (2.58).

b) Man betrachtet statt der durch $A$ repräsentierten Abbildung, die durch $A^\top$ repräsentierte Abbildung von $\mathbb{R}^m$ in den $\mathbb{R}^n$, dann geht (2.58) in (2.59) über. □

**Satz 2.8.8.** *Das lineare Ausgleichsproblem (2.57) ist immer lösbar.*

**Beweis:** (vgl. Abb. 2.3)

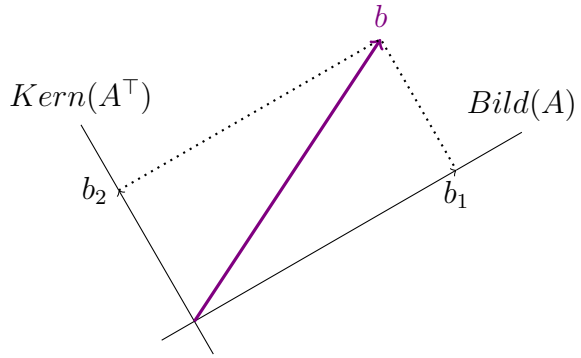



Abbildung 2.3: Zerlegung von $\vec{b}$ gemäß (2.58) und (2.55)

$$A\vec{x} - \vec{b} = \underbrace{A\vec{x} - \vec{b}_1}_{\in Bild(A)} - \underbrace{\vec{b}_2}_{\in Kern(A^\top)}$$

[1)] $\vec{z} \in \mathbb{R}^m$ gilt, da $A^\top$ eine lineare Abbildung vom $\mathbb{R}^m$ in den $\mathbb{R}^n$ darstellt und die Elemente des Kerns stets im Urbildraum liegen. (Vgl. Def. vom Kern S. 29)

Wegen $A\vec{x} - \vec{b}_1 \perp \vec{b}_2$ lässt sich der Satz von Pythagoras anwenden:

$$\|A\vec{x} - \vec{b}\|_2^2 \;=\; \|A\vec{x} - \vec{b}_1\|_2^2 + \|\vec{b}_2\|_2^2$$
$$\uparrow$$
$$\text{unabhängig von } \vec{x} \in \mathbb{R}^n$$

$\Rightarrow$ $\|A\vec{x} - \vec{b}\|_2$ ist genau dann minimal, wenn $\|A\vec{x} - \vec{b}_1\|_2$ minimal ist. $b_1 \in Bild(A) \;\Rightarrow\; A\vec{x} = \vec{b}_1$ ist lösbar; d.h. $\|A\vec{x} - \vec{b}_1\|_2$ kann zu Null gemacht werden $\Rightarrow$ alle $\vec{x}$ die das (stets lösbare) Gleichungssystem $A\vec{x} = \vec{b}_1$ lösen, sind auch Lösungen des Ausgleichsproblems. □

Aus dem Beweis folgt auch: Das lineare Ausgleichsproblem (2.57) ist genau dann *eindeutig* lösbar, wenn das lösbare Gleichungssystem $A\vec{x} = \vec{b}_1$ eindeutig lösbar ist, d.h. wenn $Rang(A) = n$ ist.Im Fall $Rang(A) < n$ bestimmt wieder $n - Rang(A)$ die Dimension der Lösungsschar von $A\vec{x} = \vec{b}_1$, d.h. die Dimension der Lösungsschar des linearen Ausgleichsproblems.

**Definition 2.8.9.** *Ist $\vec{x}$ Lösung des Auslgeichsproblems, löst also $A\vec{x} = \vec{b}_1$, so nennt man*

$$A\vec{x} - \vec{b} = \vec{b}_1 - \vec{b} = -\vec{b}_2 \in Kern(A^\top)$$

*das* **Residuum**.

**Satz 2.8.10.** *Es gelten die* **Gaußschen Normalgleichungen**

$$A^\top A\vec{x} = A^\top \vec{b}. \tag{2.60}$$

**Beweis.** $A\vec{x} - \vec{b} \in Kern(A^\top) \Longrightarrow A^\top(A\vec{x} - \vec{b}) = \vec{0} \in \mathbb{R}^n \Longleftrightarrow A^\top A\vec{x} = A^\top\vec{b}.$ □

Man beachte: $A^\top A \in \mathbb{R}^{n\times n},\; A^\top\vec{b} \in \mathbb{R}^n$;

Im Falle $Rang(A) = n$ gilt $Rang(A^\top A) = n$, d.h. (2.60) ist ein reguläres lineares Gleichungssystem mit quadratischer Matrix; die eindeutig bestimmte Lösung von (2.60) liefert dann die eindeutig bestimmte Lösung des linearen Ausgleichsproblems.

**Behauptung 2.8.11.** *Im Fall $Rang(A) < n$ ist das lineare Ausgleichsproblem $\|A\vec{x} - \vec{b}\|_2 = \min$ (nur) mit Zusatzbedingung $\|\vec{x}\|_2 = \min$ eindeutig lösbar.*

**Beweis:** $\vec{x}_0$ sei spezielle Lösung von $\|A\vec{x} - \vec{b}\|_2 = \min$, also von $A\vec{x} = \vec{b}_1$
$\vec{x}_0 = \vec{x}_{0,1} + \vec{x}_{0,2}$ mit $\vec{x}_{0,1} \in Kern(A^\top)$ und $\vec{x}_{0,2} \in Kern(A)$ d.h. $\vec{x}_{0,1} \perp \vec{x}_{0,2}$. Es gilt

$$\vec{x}_0 + \vec{x}_k \;=\; \underbrace{\vec{x}_{0,1}}_{\in Bild(A^\top)} + \underbrace{\vec{x}_{0,2} + \vec{x}_k}_{\in Kern(A)}$$

wobei $\vec{x}_k$ ein bel. Element aus $Kern(A)$ bezeichnet. Weiters gilt offenbar $\vec{x}_{0,1} \perp (\vec{x}_{0,2} + \vec{x}_k)$ und mit Pythagoras

$$\|\vec{x}_0 + \vec{x}_k\|_2^2 \;=\; \|\vec{x}_{0,1}\|_2^2 + \|\vec{x}_{0,2} + \vec{x}_k\|_2^2\,;$$

$\Rightarrow$ $\|\vec{x}_0 + \vec{x}_k\|_2$ genau minimal für $\vec{x}_k = -\vec{x}_{0,2}$ also für $\vec{x}_{0,1} = \vec{x}_0 - \vec{x}_{0,2}$ □

Es ist klar, dass in der Praxis beim Ausgleichsproblem meist gilt: $\|\vec{b}_1\|_2 \gg \|\vec{b}_2\|_2$

Denn: Hätte man keine fehlerbehafteten Messgeräte, so würde man bei redundanten Messungen den Spezialfall $\vec{b} = \vec{b}_1 \in Bild(A), \vec{b}_2 = \vec{0}$ erhalten, d.h. ein überbestimmtes ($m > n$) aber nicht widersprüchliches ($\vec{b} \in Bild(A)$) Gleichungssystem (vgl. etwa Seite 59). Durch kleine Messfehler fällt man mit $\vec{b}$ nur wenig aus $Bild(A)$ heraus.
In vielen Fällen ist das Lösen des Ausgleichsproblems mithilfe der Normalgleichungen wegen starker Rundungsfehlersensitivität numerisch bedenklich. Das Ausgleichsproblem kann dann besser mit dem sogenannten *QR-Algorithmus* behandelt werden, einem Eliminationsalgorithmus, der auf orthogonalen Eliminationsmatrizen aufgebaut ist.

**Definition 2.8.12.** *Das lineare Ausgleichsproblem*

$$\begin{array}{lll} \|A\vec{x} - \vec{b}\|_2 = \min & & Rang(A) = n \\ \|A\vec{x} - \vec{b}\|_2 = \min & \|\vec{x}\|_2 = \min & Rang(A) < n \end{array}$$

*ist stets eindeutig lösbar. Man kann daher zu gegebener Matrix A jene Abbildung betrachten, die jeder rechten Seite $\vec{b}$ die eindeutige Lösung $\vec{x}$ des Ausgleichsproblems zuordnet: $\vec{b} \to \vec{x}$. Man kann zeigen, dass diese Abbildung linear ist, also durch eine Matrix repräsentiert wird. Diese wird* **Pseudoinverse** *(Verallgemeinerte Inverse) genannt und mit $A^+$ bezeichnet, d.h.*

$$\vec{x} = A^+\vec{b}; \tag{2.61}$$

**Sonderfälle:**

- $A \in \mathbb{R}^{n\times n}$, $Rang(A) = n$ : $\|A\vec{x} - \vec{b}\|_2$ wird zu Null und liefert die Lösung dieses regulären Falls: $A^+ = A^{-1}$;
- $A \in \mathbb{R}^{m\times n}$, $Rang(A) = n$ : eindeutige Lösung des Ausgleichsproblems durch die Gaußschen Normalgleichungen. (2.60) $\Rightarrow A^+ = (A^\top A)^{-1}A^\top$

Die weiteren Fälle sind:

- $A\vec{x} = \vec{b}$ ist lösbar (d.h. $\|A\vec{x} - \vec{b}\|_2 = 0$) aber nicht eindeutig (wegen $Rang(A) < n$); dann stellt $\vec{x} = A^+\vec{b}$ jenes Element der Lösungsschar dar, für das gilt $\|\vec{x}\|_2 = \min$
- $A\vec{x} = \vec{b}$ ist widersprüchlich mit $Rang(A) < n$ und $\vec{x} = A^+\vec{b}$ ist die eindeutig bestimmte Lösung von $\|A\vec{x} - \vec{b}\|_2 = \min$ und $\|\vec{x}\|_2 = \min$

In den letzten beiden Fällen kann man $A^+$ mit Hilfe der sogenanten *Singulärwertzerlegung* darstellen.

**Beispiel 2.8.13** (Interpolation)**.** *Sei die Gestalt*

$$f(x) = c_0 + c_1x + c_2x^2$$

*einer Funktion f gegeben (Polynom zweiten Grades), die Parameter $c_0, c_1, c_2$ (Koeffizienten) jedoch unbekannt. Aus drei Messungen von $f(x)$ können sie durch ein lineares Gleichungssystem berechnet werden. $f(x_0)$, $f(x_1)$ und $f(x_2)$ seien die Messwerte bzgl. $x_0$, $x_1$ und $x_2$. Es ergibt sich*

$$\begin{array}{lll} x_0 & \ldots & c_0 + c_1x_0 + c_2x_0^2 = f(x_0), \\ x_1 & \ldots & c_0 + c_1x_1 + c_2x_1^2 = f(x_1), \\ x_2 & \ldots & c_0 + c_1x_2 + c_2x_2^2 = f(x_2), \end{array}$$

*d.h. ein lineares Gleichungssystem der Form*

$$\underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{pmatrix}}_{\substack{\uparrow \\ \text{Gleichungsmatrix} \\ \text{(Vandermondematrix)}}} \cdot \underbrace{\begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix}}_{\substack{\uparrow \\ \text{Unbekannten-} \\ \text{Vektor}}} = \underbrace{\begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2). \end{pmatrix}}_{\substack{\uparrow \\ \text{Vektor d.} \\ \text{rechten} \\ \text{Seite}}}$$

*Konkretes Zahlenbeispiel:*

$$\begin{array}{rcccccccl} x_0 & = & 0 & \dots & f(x_0) & = & f(0) & = & 2, \\ x_1 & = & \frac{1}{2} & \dots & f(x_1) & = & f(\frac{1}{2}) & = & \frac{5}{2}, \\ x_2 & = & 1 & \dots & f(x_2) & = & f(1) & = & 0 \end{array}$$

*liefert das Gleichungssystem:*

$$\begin{array}{rcccccl} c_0 & & & & & = & 2, \\ c_0 & + & \frac{1}{2}c_1 & + & \frac{1}{4}c_2 & = & \frac{5}{2}, \\ c_0 & + & c_1 & + & c_2 & = & 0. \end{array}$$

$c_0 = 2$ *in* 2. *und* 3. *Gleichung einsetzen:*

$$\begin{array}{rcccccl} \frac{1}{2}c_1 & + & \frac{1}{4}c_2 & = & \frac{5}{2} - 2 & = & \frac{1}{2} \\ c_1 & + & c_2 & = & 0 - 2 & = & -2 \end{array}$$

*erste Gleichung mit* 2 *multipliziert und von der* 2. *Gleichung abgezogen:*

$$\begin{array}{rcccl} \frac{1}{2}c_1 & + & \frac{1}{4}c_2 & = & \frac{1}{2} \\ & & \frac{1}{2}c_2 & = & -3 \end{array}$$

$\Rightarrow c_2 = -6,\ c_1 = 4$ *und somit ist die gesuchte Funktion*

$$f(x) = 2 + 4x - 6x^2. \tag{2.62}$$

*Man kann* $f(x)$ *nun an beliebigen* $x$*-Werte auswerten (Interpolation)! (vgl. Abb. 2.4)*

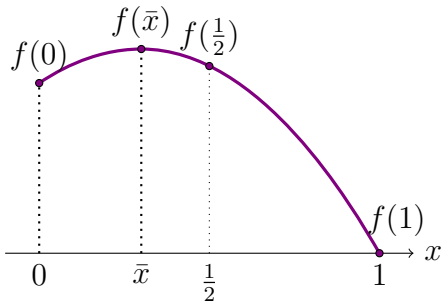



Abbildung 2.4: $x$ ... „Zwischenstelle“; $f(x)$ kann berechnet werden.

**Bemerkung.** Für diese Art der Interpolation – ein Interpolationspolynom an die Interpolationsdaten $(x_0, f(x_0))$, $(x_1, f(x_1))$, $\ldots$, $(x_n, f(x_n))$ (bei unserem Beispiel $n = 2$) anzupassen und das Polynom an einer Stelle $x \neq x_0, x_1, \ldots, x_n$ auszuwerten – gibt es effizientere numerische Algorithmen (vgl. Kap. 4); der Weg, zuerst die Koeffizienten über ein lineares Gleichungssystem zu bestimmen, und dann das so erhaltene Interpolationspolynom an der Stelle $x$ auszuwerten, ist zwar prinzipiell möglich, aber für wirkliche Berechnungen ein Umweg. Für theoretische Fragestellungen sind jedoch die hinter einem linearen Interpolationsprozess stehenden linearen Gleichungssysteme oft wichtig.

**Beispiel 2.8.14** (Redundante Daten bei Interpolation)**.** *Sei bei Beispiel 2.8.13 eine zusätzliche Messung für* $x_3 = \frac{1}{4}$,

$$f(x_3) = f\left(\frac{1}{4}\right) = \frac{21}{8}, \tag{2.63}$$

*gegeben. Es ergibt sich das überbestimmte lineare Gleichungssystem*

$$\begin{array}{lrcrcrcl} x_0 = 0: & c_0 & & & & & = & 2, \\ x_1 = \frac{1}{2}: & c_0 & + & \frac{1}{2}c_1 & + & \frac{1}{4}c_2 & = & \frac{5}{2}, \\ x_2 = 1: & c_0 & + & c_1 & + & c_2 & = & 0, \\ x_3 = \frac{1}{4}: & c_0 & + & \frac{1}{4}c_1 & + & \frac{1}{16}c_2 & = & \frac{21}{8} \end{array}$$

*(4 Gleichungen für 3 Unbekannte) bzw. in Matrixschreibweise*

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{4} \\ 1 & 1 & 1 \\ 1 & \frac{1}{4} & \frac{1}{16} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 2 \\ \frac{5}{2} \\ 0 \\ \frac{21}{8} \end{pmatrix}$$

*mit einer Rechtecksmatrix* $A \in \mathbb{R}^{4\times 3}$*, Lösungsvektor* $\vec{x} = (c_0, c_1, c_2)^\top \in \mathbb{R}^3$ *und rechter Seite* $\vec{b} = (2, \frac{5}{2}, 0, \frac{21}{8})^\top \in \mathbb{R}^4$*. Lässt man eine beliebige Gleichung weg, ergeben die restlichen 3 Gleichungen stets* $c_0 = 2, c_1 = 4, c_2 = -6$*. Das überbestimmte Gleichungssystem ist eben* **redundant**.

In der Praxis wird dieser Fall (Anzahl der Gleichungen > Anzahl Unbekannten) dennoch oft betrachtet, da mit Messfehlern (beim Aufstellen des Datensatzes $(x_0, f(x_0)), (x_1, f(x_1)), \ldots$) zu rechnen ist. (Vgl. Abb. 2.5)

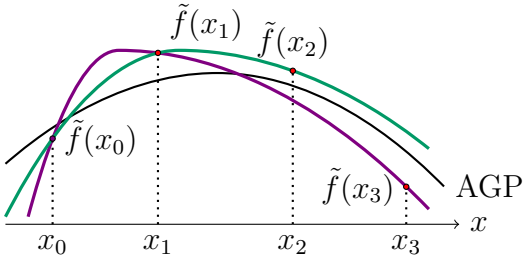



Abbildung 2.5: Messfehlerbehaftete Daten

Es liegt dann im Allgemeinen kein redundantes, sondern ein widersprüchliches, überbestimmtes Gleichungssystem vor. Man kann etwa durch einen Datensatz von 4 messfehlerbehafteten Datenpunkten i.A. kein quadratisches Polynom (nur drei Koeffizienten!) festlegen.

Man kann jedoch versuchen, die Information, die in allen 4 (messfehlerbehafteten) Messungen steckt, aus den Daten zu holen und hoffen, dass sich die Messfehler „herausmitteln“. Dieser Ansatz führt zur Bestimmung des **Ausgleichspolynoms** (vgl. Abb. 2.6).

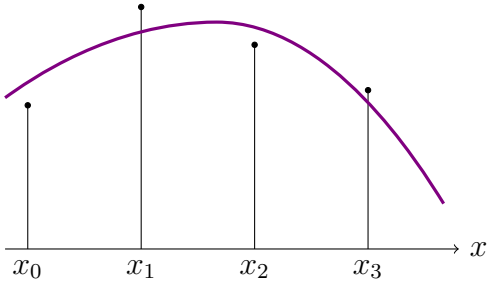



Abbildung 2.6: Ausgleichspolynom

Im obigen Beispiel soll $f(x) = c_0 + c_1 x + c_2 x^2$ optimal an Daten $(x_0, \tilde{f}(x_0)),\ \ldots,\ (x_3, \tilde{f}(x_3))$ angepasst werden, z.B. durch die Forderung, dass

$$\sum_{i=0}^{3} (c_0 + c_1 x_i + c_2 x_i^2 - \tilde{f}(x_i))^2$$

minimal wird.

**Etwas allgemeiner:** Datensatz: $(x_0, \tilde{f}(x_0))$, $(x_1, \tilde{f}(x_1))$, $\ldots$, $(x_m, \tilde{f}(x_m))$ (vgl. Abb. 2.7)

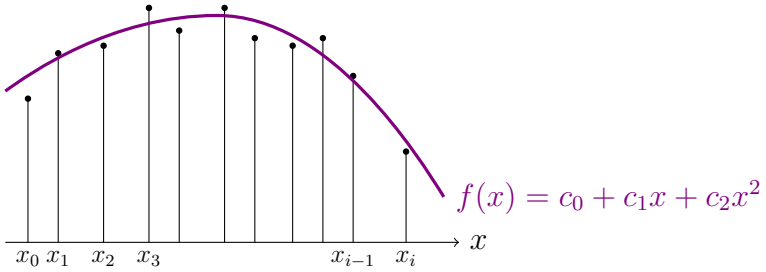



Abbildung 2.7: Ausgleichspolynom zu allgemeinem Datensatz

quadratisches Polynom so festlegen (d.h. die Koeffizienten so bestimmen), dass

$$E(c_0, c_1, c_2) \quad := \quad \sum_{i=0}^{m} (c_0 + c_1 x_i + c_2 x_i^2 - \tilde{f}(x_i))^2 \tag{2.64}$$

minimal wird. Notwendige Bedingungen für das Minimum sind

$$\frac{\partial E(c_0, c_1, c_2)}{\partial c_0} = 0, \quad \frac{\partial E(c_0, c_1, c_2)}{\partial c_1} = 0, \quad \frac{\partial E(c_0, c_1, c_2)}{\partial c_2} = 0\ , \tag{2.65}$$

wobei

$$\begin{aligned}
\frac{\partial E(c_0, c_1, c_2)}{\partial c_0} &= \sum_{i=0}^{m} 2(c_0 + c_1 x_i + c_2 x_i^2 - \tilde{f}(x_i)) \\
\frac{\partial E(c_0, c_1, c_2)}{\partial c_1} &= \sum_{i=0}^{m} 2(c_0 + c_1 x_i + c_2 x_i^2 - \tilde{f}(x_i)) \cdot x_i \\
\frac{\partial E(c_0, c_1, c_2)}{\partial c_2} &= \sum_{i=0}^{m} 2(c_0 + c_1 x_i + c_2 x_i^2 - \tilde{f}(x_i)) \cdot x_i^2\ ,
\end{aligned} \tag{2.66}$$

d.h. das Minimum berechnet sich aus dem linearen Gleichungssystem

$$\begin{aligned} \not{2}\sum_{i=0}^{m}(c_0 + c_1x_i + c_2x_i^2 - \tilde{f}(x_i)) &= 0 \\ \not{2}\sum_{i=0}^{m}(c_0 + c_1x_i + c_2x_i^2 - \tilde{f}(x_i))\cdot x_i &= 0 \\ \not{2}\sum_{i=0}^{m}(c_0 + c_1x_i + c_2x_i^2 - \tilde{f}(x_i))\cdot x_i^2 &= 0\ , \end{aligned} \tag{2.67}$$

das man offensichtlich auch so schreiben kann:

$$\begin{aligned} (m+1)\cdot c_0 + \Big(\sum_{i=0}^{m} x_i\Big)\cdot c_1 + \Big(\sum_{i=0}^{m} x_i^2\Big)\cdot c_2 &= \sum_{i=0}^{m}\tilde{f}(x_i) \\ \Big(\sum_{i=0}^{m} x_i\Big)\cdot c_0 + \Big(\sum_{i=0}^{m} x_i^2\Big)\cdot c_1 + \Big(\sum_{i=0}^{m} x_i^3\Big)\cdot c_2 &= \sum_{i=0}^{m} x_i\tilde{f}(x_i) \\ \Big(\sum_{i=0}^{m} x_i^2\Big)\cdot c_0 + \Big(\sum_{i=0}^{m} x_i^3\Big)\cdot c_1 + \Big(\sum_{i=0}^{m} x_i^4\Big)\cdot c_2 &= \sum_{i=0}^{m} x_i^2\tilde{f}(x_i) \end{aligned} \tag{2.68}$$

Also: 3 lineare Gleichungen für die 3 Unbekannten $c_0, c_1, c_2$. Die Gleichungsmatrix ist

$$\begin{pmatrix} m+1 & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{pmatrix} \tag{2.69}$$

und der Vektor der rechten Seite ist

$$\Big(\sum \tilde{f}(x_i), \sum x_i\tilde{f}(x_i), \sum x_i^2\tilde{f}(x_i)\Big)^\top\ .$$

**Beispiel 2.8.15.** *Ein konkretes Zahlenbeispiel dazu:* $f(x) = 2 + 4x - 6x^2 \ldots$ *und* $f$ *soll jetzt nicht wie bei obigen Beispiel durch den exakten, unverfälschten Datensatz*

$$(x_0, f(x_0)) = (0, 2), \quad (x_1, f(x_1)) = (\frac{1}{2}, \frac{5}{2}), \quad (x_2, f(x_2)) = (1, 0)$$

*festgelegt sein, sondern durch den messfehlerbehafteten Datensatz mit* 5 *Messpunkten:*

$$\begin{aligned} (x_0, \tilde{f}(x_0)) &= (0, 2.002)\ , & (x_1, \tilde{f}(x_1)) &= (0.25, 2.624) \\ (x_2, \tilde{f}(x_2)) &= (0.5, 2.498)\ , & (x_3, \tilde{f}(x_3)) &= (0.75, 1.626) \\ (x_4, \tilde{f}(x_4)) &= (1, 0.001)\ . \end{aligned} \tag{2.70}$$

*Würde man aus diesen* 5 *Messdaten die* 3 *Messdaten zu den Stellen* $x_0 = 0$, $x_2 = 0.5$, $x_4 = 1$ *auswählen und aus den verfälschten Größen* $\tilde{f}(x_0)$, $\tilde{f}(x_2)$ *und* $\tilde{f}(x_4)$ *ganz analog wie auf Seite 57 die Größen* $\tilde{c}_0$, $\tilde{c}_1$, $\tilde{c}_2$ *berechnen, ergäbe sich:*

$$\begin{array}{lrcrcrcl} x_0 = 0: & \tilde{c}_0 & & & & & = & 2.002 \\ x_2 = 0.5: & \tilde{c}_0 & + & 0.5\tilde{c}_1 & + & 0.25\tilde{c}_2 & = & 2.498 \\ x_4 = 1: & \tilde{c}_0 & + & \tilde{c}_1 & + & \tilde{c}_2 & = & 0.001 \end{array} \tag{2.71}$$

*mit der Lösung* $\tilde{c}_0 = 2.002, \tilde{c}_1 = 3.985, \tilde{c}_2 = -5.986$*; wertet man das verfälschte Polynom* $\tilde{c}_0 + \tilde{c}_1 x + \tilde{c}_2 x^2$ *für* $x = 2$ *aus, so ergibt sich*

$$\tilde{f}(2) = -13.972 \quad \textit{statt des wahren Wertes} \quad f(2) = -14,$$

*also ein Fehler*

$$\tilde{f}(2) - f(2) \;=\; 0.028 \tag{2.72}$$

*Nun Einbeziehung aller Daten ergibt für unseren Datensatz folgendes Gleichungssystem:*

$$\begin{array}{rcrcrcl} 5c_0 & + & 2.5c_1 & + & 1.875c_2 & = & 8.751 \\ 2.5c_0 & + & 1.875c_1 & + & 1.5625c_2 & = & 3.1255 \\ 1.875c_0 & + & 1.5625c_1 & + & 1.3828125c_2 & = & 1.704125 \end{array}$$

*mit der Lösung*

$$\begin{array}{rcl} c_0 &=& 2.001628568 \\ c_1 &=& 3.988571429 \\ c_2 &=& -5.98857142. \end{array} \tag{2.73}$$

*Ein Vergleich mit der exakten Lösung von (2.71) zeigt eine (allerdings nur geringe) Verbesserung (etwas näher zu der wahren Lösung* $c_0 = 2$*,* $c_1 = 4$*,* $c_2 = -6$*). Berechnet man* $\tilde{f}(2)$ *mit den Koeffizienten aus (2.73) ergibt sich* $\tilde{f}(2) = -13.97551425$ *mit dem Fehler*

$$\tilde{f}(2) - f(2) \;=\; 0.02448575 \tag{2.74}$$

*Um hier eine größere Genauigkeitssteigerung zu erreichen, hätte man noch deutlich mehr Messpunkte einbeziehen müssen.*

**Bemerkung.** Die eben beschriebene Vorgangsweise wurde zum ersten Mal von Carl Friedrich Gauß im Jahre 1801 angewendet. Von einem italienischen Astronomen war ein Planetoid entdeckt und seine Position an vielen Tagen gemessen worden (natürlich immer mit den üblichen Messfehlern). Als er wegen zu großer Sonnennähe nicht mehr beobachtet werden konnte, gelang es nachher nicht mehr, ihn wieder zu finden.
Da die Planetoidenbahn durch die Messungen bereits festgelegt ist, griff man aus der großen Zahl von Einzelbeobachtungen auf verschiedene Weise drei Beobachtungsdaten heraus, berechnete daraus die entsprechenden Laufbahnen und Positionen berechnet und suchte mit dem Fernrohr die Umgebungen der Position ab. Aufgrund der Messfehler der einzelnen Beobachtungen war dies jedoch erfolglos. Erst als Gauß entsprechend der oben beschriebenen Vorgangsweise die in den einzelnen Messungen enthaltene Gesamtinformation ausnützte, gelang es, den Planetoiden wieder zu entdecken.

Die Herleitung von (2.68) kann man sich auch folgendermaßen denken:

Zuerst betrachtet man das überbestimmte, widersprüchliche Gleichungssystem

$$\begin{array}{rcll} x_0 & \dots & c_0 + c_1 x_0 + c_2 x_0^2 & = \quad \tilde{f}(x_0) \\ x_1 & \dots & c_0 + c_1 x_1 + c_2 x_1^2 & = \quad \tilde{f}(x_1) \\ x_2 & \dots & c_0 + c_1 x_2 + c_2 x_2^2 & = \quad \tilde{f}(x_2) \\ \vdots & & & \vdots \\ x_m & \dots & c_0 + c_1 x_m + c_2 x_m^2 & = \quad \tilde{f}(x_m); \end{array} \tag{2.75}$$

mit Systemmatrix

$$A = \begin{pmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 \end{pmatrix} . \tag{2.76}$$

Offensichtlich entsteht (2.68) durch Multiplikation von (2.75) mit der transponierten Matrix

$$A^\top = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_0 & x_1 & x_2 & \cdots & x_m \\ x_0^2 & x_1^2 & x_2^2 & \cdots & x_m^2 \end{pmatrix} \tag{2.77}$$

von links. Aus (2.27) erhält man also sofort die Gaußschen Normalgleichungen (2.68) in der Form

$$A^\top A \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = A^\top \begin{pmatrix} \tilde{f}(x_0) \\ \tilde{f}(x_1) \\ \tilde{f}(x_2) \\ \vdots \\ \tilde{f}(x_m) \end{pmatrix} .$$

# Kapitel 3

# Iterative Lösung nichtlinearer Gleichungssysteme

## 3.1 Einleitung und Problemstellung

In diesem Kapitel wird das Lösen nichtlinearer Gleichungen bzw. das sogenannte *Nullstellenproblem* behandelt.

**Definition 3.1.1.** *Sei $\vec{F} : \mathbb{R}^n \to \mathbb{R}^n$ eine i.a. nichtlineare Funktion. Unter dem* **Nullstellenproblem** *versteht man die Suche nach allen Lösungen $\vec{x} \in \mathbb{R}^n$ für die $\vec{F}(\vec{x}) = \vec{0}$ gilt. Diese Lösungen $\vec{x} \in \mathbb{R}^n$ werden als* **Nullstellen** *bezeichnet.*

*Existenz* und *Eindeutigkeit* dieser Nullstellen $\vec{x}$: Nichtlineare Gleichungsysteme können oft nur in einem bestimmten Gebiet, also *lokal* eindeutig gelöst werden.
Einen Spezialfall stellt die sogenannte **affine Funktion** $F : \mathbb{R} \to \mathbb{R}$, $F(x) = ax + b,\ a, b \in \mathbb{R}$ dar: Eine eindeutige Lösung existiert für $a \neq 0$. Für $a = 0$ und $b \neq 0$ existiert keine Nullstelle und für $a = b = 0$, also $F(x) \equiv 0$, ist die Lösungsschar ganz $\mathbb{R}$.
Für $\vec{F} : \mathbb{R}^n \to \mathbb{R}^n$ sind verschiedene Glattheitsforderungen ($\vec{F}$ stetig, $\vec{F}$ stetig differenzierbar, etc.) denkbar, die Existenz und Anzahl von Lösungen beeinflussen, siehe Abb. 3.1 bis 3.6.

**Definition 3.1.2.** *Eine Nullstelle $\vec{x}$ heißt* ***isoliert****, wenn gilt*

$$\vec{F}(\vec{x}) = \vec{0} \quad \textit{und} \quad \det(\vec{F}'(\vec{x})) \neq 0$$

**Bemerkungen.**

1. Zur Veranschaulichung einer isolierten Nullstelle siehe Abbildung 3.7.

2. Im Spezialfall der skalaren Funktion $F(x) = ax + b$ entspricht wegen $F'(x) = a$ die obige Definition dem regulären Fall $a \neq 0$.

3. In Abbildung 3.8 liegt eine Nullstelle vor, die nicht isoliert ist. Eine geringfügige Störung von $F$, verändert die Situation, siehe Abbildung 3.9. Dieses Problem ist schlecht konditioniert.

4. Unter der Voraussetzung einer isolierten Nullstelle lässt sich lokale Eindeutigkeit zeigen.

Im Folgenden werden **äquivalente Formulierungen des Nullstellenproblems** angegeben.

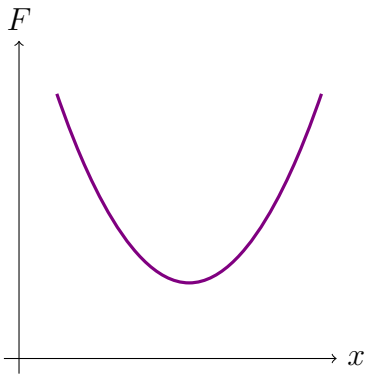



Abbildung 3.1: Beispiel für keine Nullstelle

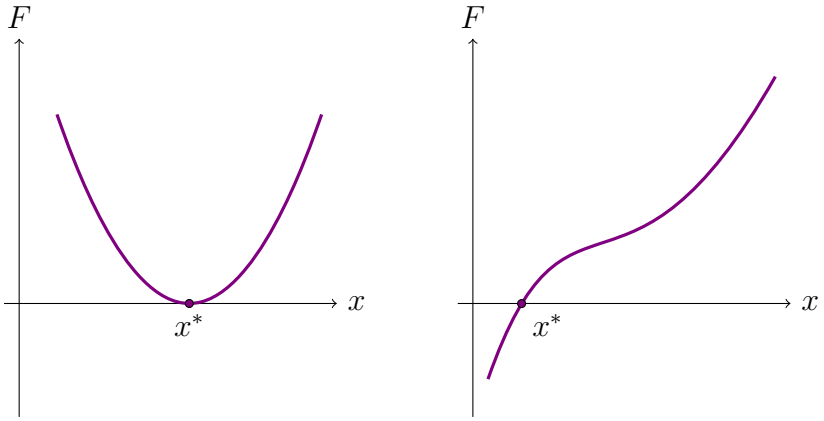



Abbildung 3.2: Beispiele für genau eine Nullstelle

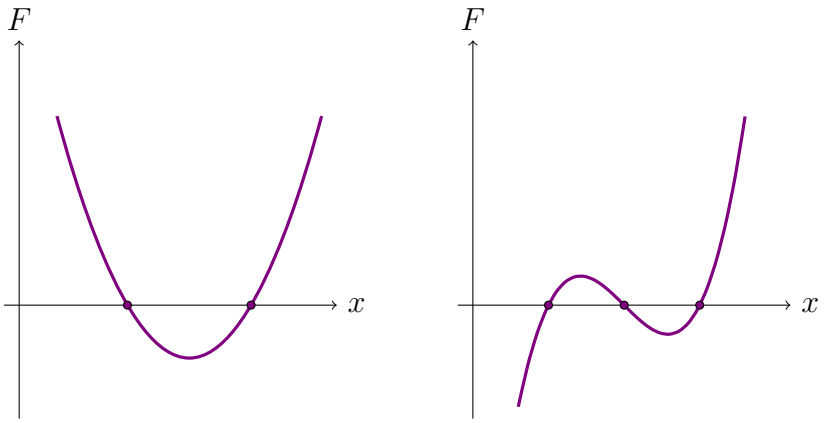



Abbildung 3.3: Beispiele für endlich viele Nullstellen

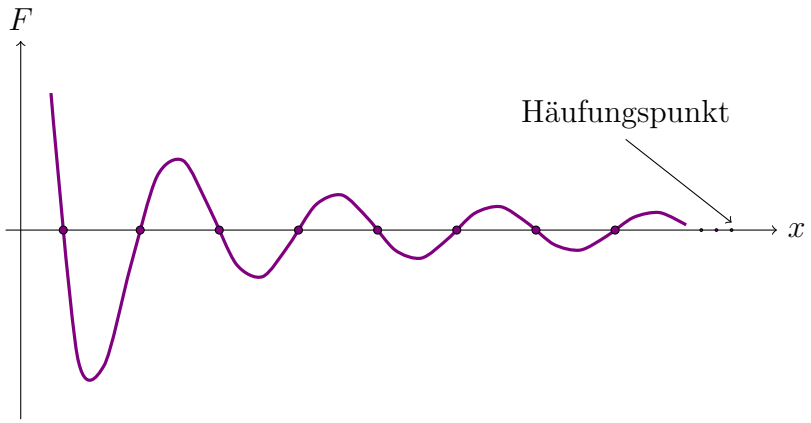



Abbildung 3.4: Beispiel für unendlich viele Nullstellen mit Häufungspunkt

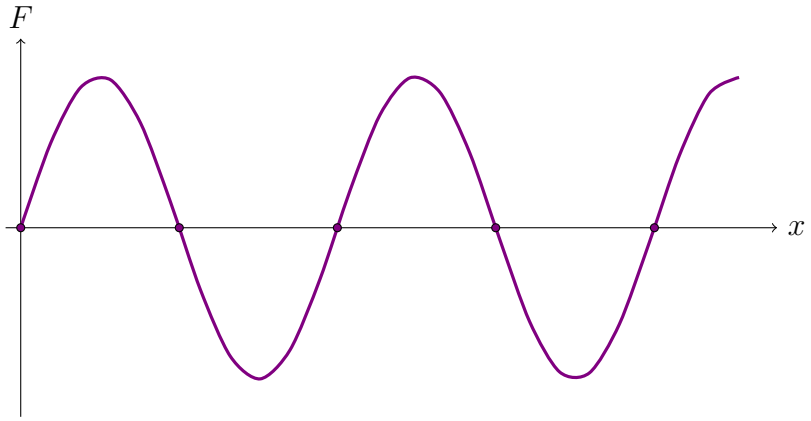



Abbildung 3.5: Beispiel für unendlich viele Nullstellen ohne Häufungspunkt

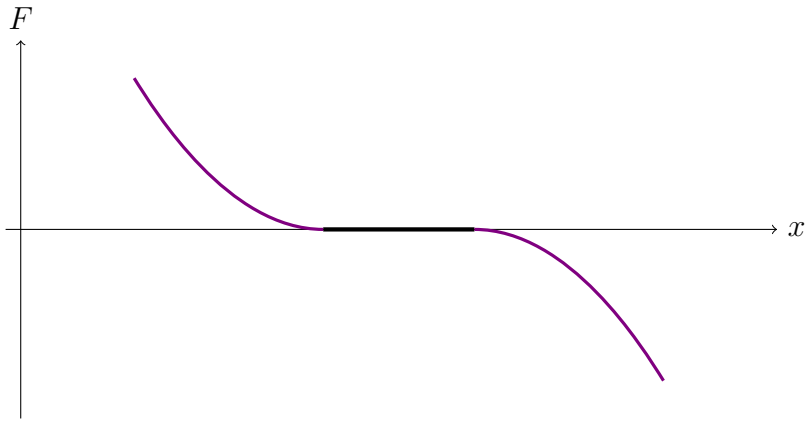



Abbildung 3.6: Beispiel für ein Kontinuum von Nullstellen

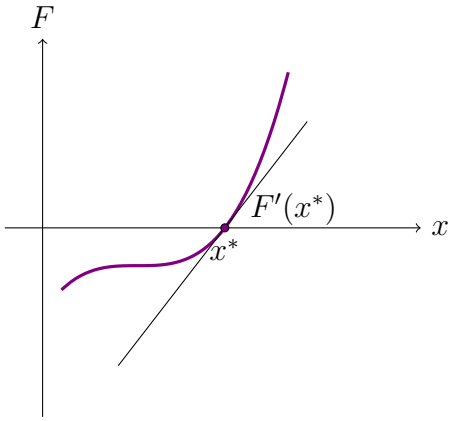



Abbildung 3.7: Beispiel für eine isolierte Nullstelle

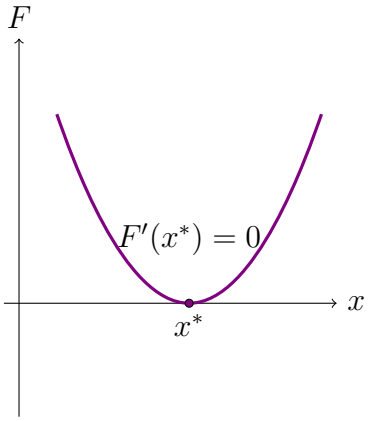



Abbildung 3.8: Beispiel für eine nicht isolierte Nullstelle

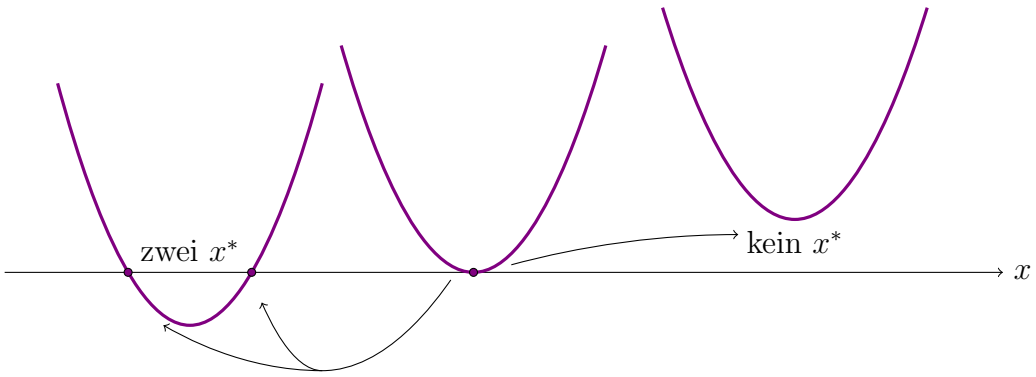



Abbildung 3.9: Einfluss von Störungen einer Funktion auf die Anzahl von Nullstellen

1. Sei $R(\vec{x})$ eine von $\vec{x} \in \mathbb{R}^n$ abhängige, reguläre $n \times n$-Matrix, dann gilt:
$$\vec{F}(\vec{x}) = \vec{0} \qquad \Longleftrightarrow \qquad R(\vec{x})\vec{F}(\vec{x}) = \vec{0} \tag{3.1}$$

2. Es gilt weiters
$$\vec{F}(\vec{x}) = \vec{0} \qquad \Longleftrightarrow \qquad \overrightarrow{T}(\vec{x}) := \vec{x} - R(\vec{x})\vec{F}(\vec{x}) = \vec{x} \tag{3.2}$$
Diese äquivalente Formulierung wird als **Fixpunktproblem** bezeichnet. Die Menge der Nullstellen von $\vec{F} : \mathbb{R}^n \to \mathbb{R}^n$ ist gleich der Menge der **Fixpunkte** von $\overrightarrow{T} : \mathbb{R}^n \to \mathbb{R}^n$, also alle $\vec{x} \in \mathbb{R}^n$ mit $\overrightarrow{T}(\vec{x}) = \vec{x}$.

Für viele Fragestellungen ist es einfacher, anstelle des Nullstellenproblems $\vec{F}(\vec{x}) = \vec{0}$ das dazu äquivalente Fixpunktproblem $\overrightarrow{T}(\vec{x}) = \vec{x}$ zu betrachten.
Für die Diskussion von Existenz und Eindeutigkeit von Lösungen ist der folgende Satz wichtig.

**Satz 3.1.3.** *Sei $I = [a, b] \subset \mathbb{R}$ und $T : \mathbb{R} \to \mathbb{R}$ stetig mit $T(I) \subset I$. Dann hat $T(x) = x$ mindestens eine Lösung in $I$, d.h. es existiert mindestens ein Fixpunkt von $T(x)$.*

*Beweis.* Das Bild $T(I) = T([a, b])$ des Intervalls $I$ unter der Abbildung $T$ liegt ganz im Intervall $I$, d.h. es gilt $T(a) \geq a$, also der Funktionswert $T(a)$ liegt oberhalb (genauer: nicht unterhalb) der Geraden $y = x$. Analog folgt aus $T(I) \subset I$ die Beziehung $T(b) \leq b$.
Da $T$ stetig ist, muss $T(x)$ mindestens einmal die Gerade $y = x$ schneiden, siehe Abb.3.10. □

Verallgemeinerung dieses Satzes im $\mathbb{R}^n$:

**Satz 3.1.4** (Satz von Brouwer)**.** *Sei $D \subset \mathbb{R}^n$ beschränkt, abgeschlossen und konvex und sei $\overrightarrow{T} : D \subset \mathbb{R}^n \to \mathbb{R}^n$ stetig mit $\overrightarrow{T}(D) \subset D$. Dann hat $\overrightarrow{T}(\vec{x}) = \vec{x}$ mindestens eine Lösung in $D$.*

*Beweis.* Siehe Literatur. □

**Bemerkung.** Eine Verallgemeinerung des Fixpunktsatzes von Brower ist der **Fixpunktsatz von Schauder**. Er ist für unendlichdimensionale Banachräume formuliert. Im Wesentlichen muss im Satz von Brouwer nur die Eigenschaft beschränkt durch *kompakt* ersetzt werden. Er wird in der Analysis oft angewendet, um die Existenz von Lösungen von Funktionalgleichungen (z.B. Differentialgleichungsprobleme, Intergralgleichungen, etc.) nachzuweisen.
Will man nicht nur die Existenz von Fixpunkten nachweisen, sondern auch deren lokale Eindeutigkeit zeigen, ist der Begriff der *Kontraktion* von Bedeutung.
In Abbildung 3.11 erkennt man, dass für $T : \mathbb{R} \to \mathbb{R}$ Eindeutigkeit bedeutet, dass der Graph von $T : \mathbb{R} \to \mathbb{R}$ flacher verläuft als die Gerade $y = x$, sonst könnte es mehrere Schnittpunkte mit der ersten Mediane, d.h. mehrere Fixpunkte geben, siehe Abbildung 3.11.

**Definition 3.1.5.** *Die Abbildung $\overrightarrow{T} : D \subset \mathbb{R}^n \to \mathbb{R}^n$ heißt **kontrahierend** auf $D$, falls $\overrightarrow{T}$ Lipschitzstetig ist*
$$\left\| \overrightarrow{T}(\vec{x}_1) - \overrightarrow{T}(\vec{x}_2) \right\| < L \left\| \vec{x}_1 - \vec{x}_2 \right\| \qquad \forall x_1, x_2 \in D \tag{3.3}$$
*mit Lipschitzkonstante $L < 1$.*

**Satz 3.1.6.** *Sei $\overrightarrow{T} : D \subset \mathbb{R}^n \to \mathbb{R}^n$ kontrahierend und $\overrightarrow{T}(D) \subset D$ so besitzt $\overrightarrow{T}(\vec{x}) = \vec{x}$ genau einen Fixpunkt in $D$.*

*Beweis.* Siehe Literatur. □

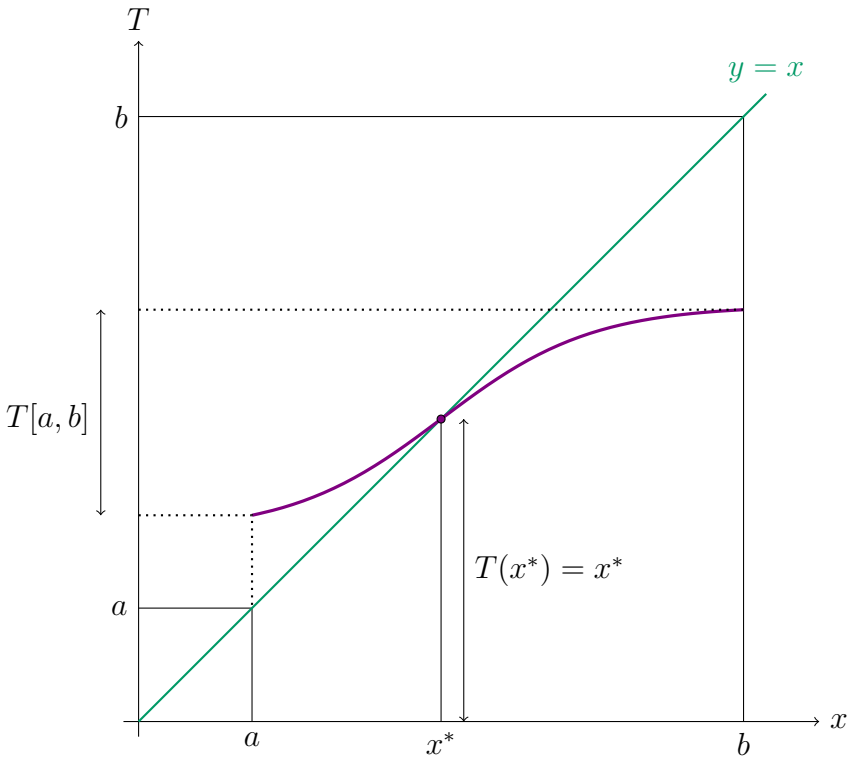



Abbildung 3.10: Beweisidee von Satz 3.1.3

## 3.2 Berechnung von Nullstellen und Fixpunkten

Im Spezialfall von linearen Gleichungssystemen erfolgt die Berechnung von Lösungen bis auf Rundungsfehler exakt durch entsprechende Formelmanipulation, etwa durch Gaußelemination. Im Gegensatz dazu können die Nullstellen nichtlinearer Gleichungen meist nur näherungsweise durch sogenannte **Iterationsverfahren** berechnet werden.
Die Idee eines Iterationsverfahrens für das Auffinden von Lösungen von $\overrightarrow{T}(\vec{x}) = \vec{x}$ ist die folgende. Wählen Sie einen Startwert $\vec{x}_0 \in D$ und berechnen Sie die folgenden Ausdrücke:

$$\begin{array}{rcl} \overrightarrow{T}(\vec{x}_0) & =: & \vec{x}_1, \\ \overrightarrow{T}(\vec{x}_1) & =: & \vec{x}_2, \\ \vdots & & \vdots \\ \overrightarrow{T}(\vec{x}_{k-1}) & =: & \vec{x}_k. \end{array} \tag{3.4}$$

Für eine kontrahierende Abbildung $\overrightarrow{T}$ folgt Konvergenz, d.h.

$$\lim_{k\to\infty} \overrightarrow{T}(\vec{x}_k) = \vec{x}^*, \qquad \overrightarrow{T}(\vec{x}^*) = \vec{x}^*. \tag{3.5}$$

**Beispiel 3.2.1.** *Sei $F(x) = e^{x-2} - x$ mit $x \in \mathbb{R}$. Es sollen die Nullstellen von $F$ bestimmt werden, also die Gleichung $F(x) = 0$ oder äquivalent das Fixpunktproblem $x = e^{x-2} =: T(x)$ gelöst werden. Als Startwert wird hier $x_0 = 0.25$ gewählt.*

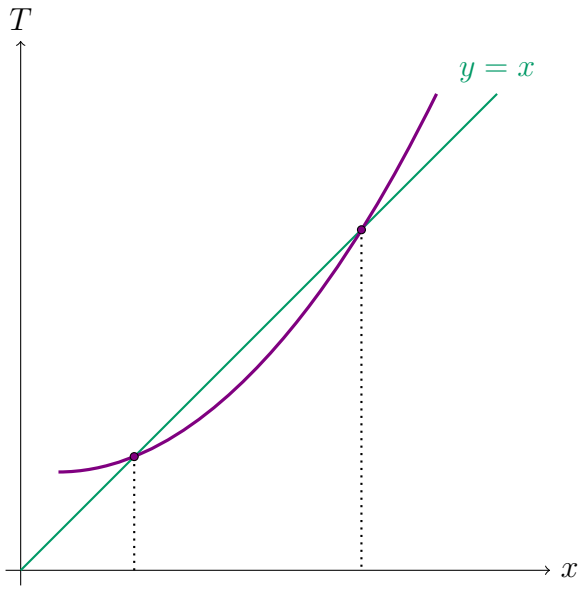



Abbildung 3.11: Zur Mehrdeutigkeit eines Fixpunktes

$$
\begin{array}{lcll}
x_0 & = & & 0.25 \\
x_1 & = & T(x_0) & = 0.1737739435\ldots \\
x_2 & = & T(x_1) & = 0.1610201033\ldots \\
x_3 & = & T(x_2) & = 0.158979519\ldots \\
x_4 & = & T(x_3) & = 0.158554386\ldots \\
x_5 & = & T(x_4) & = 0.1586040298\ldots \\
x_6 & = & T(x_5) & = 0.1585958764\ldots \\
x_7 & = & T(x_6) & = 0.1585945833\ldots \\
x_8 & = & T(x_7) & = 0.1585943782\ldots \\
x_9 & = & T(x_8) & = 0.1585943457\ldots \\
 & & & \\
x_{10} & = & T(x_9) & = 0.1585943405\ldots \\
x_{11} & = & T(x_{10}) & = 0.1585943397\ldots \\
x_{12} & = & T(x_{11}) & = 0.1585943396\ldots \\
x_{13} & = & T(x_{12}) & = 0.1585943396\ldots
\end{array}
$$

*Ab $x_{12}$ ändert sich die Iteration bis zur zehnten Nachkommastelle nicht mehr. Die Iteration berechnet Werte, die im Intervall $[x^*, x_0]$ liegen. Wegen der Monotonie der Exponentialfunktion ist die kleinstmögliche Lipschitzkonstante in diesem Intervall durch*

$$L_{opt} = T'(x_0) = e^{0{,}25-2} = 0{,}1737739435\ldots$$

*gegeben.*

**Beispiel 3.2.2.** *Ein weiteres Beispiel mit einer betragsmäßig kleineren Lipschitzkonstante ist durch*

$$F(x) = \frac{1}{2} + e^{x-10} - x = 0 \qquad \Longleftrightarrow \qquad T(x) := \frac{1}{2} + e^{x-10} = x$$

*mit $x \in \mathbb{R}$ gegeben.*

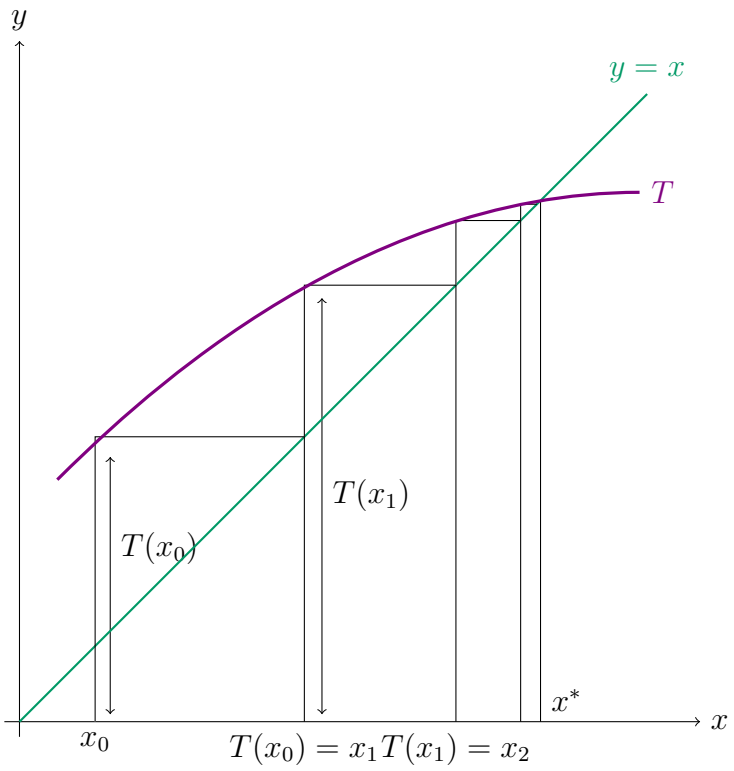



Abbildung 3.12: Visualisierung der Fixpunktiteration für $T : \mathbb{R} \to \mathbb{R}$

$$\begin{array}{lllll} x_0 & = & & & 0.5 \\ x_1 & = & T(x_0) & = & 0.5000748518\ldots \\ x_2 & = & T(x_1) & = & 0.5000748574\ldots \\ x_3 & = & T(x_2) & = & 0.5000748574\ldots \end{array}$$

*Die Iteration berechnet Werte im Intervall $[x_0, x^*]$, wo nun wegen der Monotonie der Exponentialfunktion $L_{opt}$ durch $L_{opt} = T'(x^*) = 0.00007485743331\ldots$ gegeben ist. Die Konvergenz ist hier sehr viel schneller als im vorigen Beispiel.*

**Beispiel 3.2.3.** *Das Fixpunktproblem aus Beispiel 3.2.1 kann in ein dazu äquivalentes Fixpunktproblem der Form*

$$T(x) := \ln x + 2 = x$$

*umgeformt werden. Mit einem Startwert $x_0 = 0.1586$, welcher in der Nähe von $x^*$ liegt, erhält man:*

$$\begin{array}{lllll} x_0 & = & & & 0.1586 \\ x_1 & = & T(x_0) & = & 0.158600302\ldots \\ x_2 & = & T(x_1) & = & 0.1588199578\ldots \\ x_3 & = & T(x_2) & = & 0.1600121629\ldots \\ x_4 & = & T(x_3) & = & 0.1674945515\ldots \\ x_5 & = & T(x_4) & = & 0.2131955434\ldots \end{array}$$

*Die Lipschitzkonstante für $T(x) = \ln x + 2$ ist auf dem Intervall $[x_0, 1]$ größer 1.*

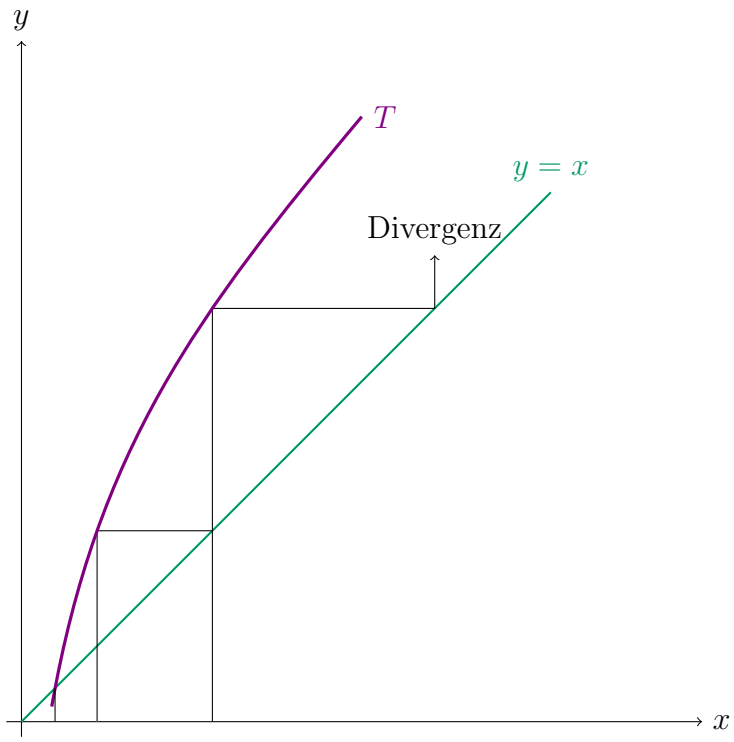



Abbildung 3.13: Visualisierung der Fixpunktiteration bei Divergenz

**Satz 3.2.4** (Kontraktionssatz)**.** *Sei* $\overrightarrow{T} : D \subset \mathbb{R}^n \to \mathbb{R}^n$ *und* $D$ *abgeschlossen, beschränkt und konvex, es gelte* $\overrightarrow{T}(D) \subset D$ *und* $\overrightarrow{T}$ *sei kontrahierend auf* $D$*. Sei* $\vec{x}_0 \in D$ *ein beliebig gewählter Startwert für die Iteration* $\vec{x}_k = \overrightarrow{T}(\vec{x}_{k-1})$ *mit* $k = 1, 2, 3, \ldots$.
*Unter diesen Voraussetzungen liegt Konvergenz gegen* $\vec{x}^*$ *vor, wobei*

$$\lim_{k\to\infty} \vec{x}_k = \vec{x}^* \tag{3.6}$$

*gilt. Weiters gilt*

$$\|\vec{x}_k - \vec{x}^*\| \leq L \|\vec{x}_{k-1} - \vec{x}^*\| \tag{3.7}$$

*und*

$$\|\vec{x}_k - \vec{x}^*\| \leq \frac{L}{1-L} \|\vec{x}_k - \vec{x}_{k-1}\| \leq \frac{L^k}{1-L} \|\vec{x}_1 - \vec{x}_0\|, \tag{3.8}$$

*mit der Lipschitzkonstante* $L < 1$ *von* $\overrightarrow{T}$ *auf* $D$*.*

**Beweis.**

1. *Eindeutigkeit.* Angenommen es gäbe zwei Fixpunkte $\vec{x}_1^*, \vec{x}_2^*$ mit $\vec{x}_1^* \neq \vec{x}_2^*$. Die Rechnung

$$\|\vec{x}_1^* - \vec{x}_2^*\| = \left\| \overrightarrow{T}(\vec{x}_1^*) - \overrightarrow{T}(\vec{x}_2^*) \right\| \leq L \|\vec{x}_1^* - \vec{x}_2^*\| < \|\vec{x}_1^* - \vec{x}_2^*\|$$

   zeigt einen Widerspruch. Die letzte Abschätzung wird durch $L < 1$ gerechtfertigt.

2. *Existenz.* Aus $\overrightarrow{T}(D) \subset D$ folgt, dass zu jedem $\vec{x}_0 \in D$ die Folge $\vec{x}_k$ existiert mit $\vec{x}_k = \overrightarrow{T}(\vec{x}_{k-1})$. Dies gilt, da aus $\vec{x}_0 \in D$ folgt $\vec{x}_1 = \overrightarrow{T}(\vec{x}_0) \in D$, daraus wieder $\vec{x}_2 = \overrightarrow{T}(\vec{x}_1) \in D$, usw. Es sind

also sämtliche $\vec{x}_k \in D$. Aus der Rechnung

$$\|\vec{x}_{k+1} - \vec{x}_k\| = \left\| \overrightarrow{T}(\vec{x}_k) - \overrightarrow{T}(\vec{x}_{k-1}) \right\| \leq L \|\vec{x}_k - \vec{x}_{k-1}\| \leq L^2 \|\vec{x}_{k-1} - \vec{x}_{k-2}\| \leq \ldots$$

$$\ldots \leq L^k \|\vec{x}_1 - \vec{x}_0\|$$

folgt,

$$\begin{aligned} \|\vec{x}_{k+r} - \vec{x}_k\| &= \|(\vec{x}_{k+r} - \vec{x}_{k+r-1}) + (\vec{x}_{k+r-1} - \vec{x}_{k+r-2}) + \ldots + (\vec{x}_{k+1} - \vec{x}_k)\| \leq \\ &\leq (1 + L + L^2 + \ldots + L^{r-1}) \|\vec{x}_{k+1} - \vec{x}_k\| < \frac{1}{1-L} \|\vec{x}_{k+1} - \vec{x}_k\| \\ &\leq \frac{L^k}{1-L} \|\vec{x}_1 - \vec{x}_0\| \qquad \forall r \in \mathbb{N} \end{aligned} \tag{3.9}$$

Gleichung (3.9) mit $k = 1$ ergibt, dass sämtliche Elemente der Folge $\vec{x}_1 = \overrightarrow{T}(\vec{x}_0)$, $\vec{x}_2 = \overrightarrow{T}(\vec{x}_1)$, ... in dem beschränkten, abgeschlossenen Bereich

$$D \cap \overline{S}\left(\vec{x}_1, \frac{L}{1-L} \|\vec{x}_1 - \vec{x}_0\|\right), \tag{3.10}$$

liegen, wobei $\overline{S}(\vec{x}_m, r)$ die abgeschlossene Kugel $\{\vec{x} \in \mathbb{R}^n : \|\vec{x} - \vec{x}_m\| \leq r\}$ mit Mittelpunkt $\vec{x}_m \in \mathbb{R}^n$ und dem Radius $r \in \mathbb{R}^+$ bezeichnet.
Wendet man Gleichung (3.9) für beliebiges $k \in \mathbb{N}$ an, so folgt wegen $L^k \to 0$ und $L < 1$, dass die Folge $\vec{x}_0, \vec{x}_1, \vec{x}_2, \ldots$ eine Chauchyfolge ist. Da eine Cauchyfolge in einem beschränkten, abgeschlossenen Bereich in $\mathbb{R}^n$ stets gegen ihren Grenzwert konvergiert, ist die Existenz des Grenzwertes $\vec{x}^* = \lim_{k\to\infty} \vec{x}_k$ sichergestellt. Es gilt daher wegen der Stetigkeit von $T$

$$\vec{x}^* = \lim_{k\to\infty} \vec{x}_k = \lim_{k\to\infty} \overrightarrow{T}(\vec{x}_{k-1}) = \overrightarrow{T}\big(\lim_{k\to\infty} \vec{x}_{k-1}\big) = \overrightarrow{T}(\vec{x}^*).$$

Der Grenzwert $\vec{x}^*$ ist der eindeutige Fixpunkt der Gleichung $\vec{x} = \overrightarrow{T}(\vec{x})$.

3. *Abschätzung.* Es gilt

$$\|\vec{x}_k - \vec{x}^*\| = \left\| \overrightarrow{T}(\vec{x}_{k-1}) - \overrightarrow{T}(\vec{x}^*) \right\| \leq L \|\vec{x}_{k-1} - \vec{x}^*\|,$$

daraus folgt die Gültigkeit von Gleichung (3.7). Weiters folgt aus Gleichung (3.9)

$$\|\vec{x}_{k+r} - \vec{x}_k\| < \frac{1}{1-L} \|\vec{x}_{k+1} - \vec{x}_k\| = \frac{1}{1-L} \left\| \overrightarrow{T}(\vec{x}_k) - \overrightarrow{T}(\vec{x}_{k-1}) \right\| \leq$$

$$\leq \frac{L}{1-L} \|\vec{x}_k - \vec{x}_{k-1}\| \leq \frac{L^k}{1-L} \|\vec{x}_1 - \vec{x}_0\|.$$

Für $r \to \infty$ erhält man

$$\|\vec{x}^* - \vec{x}_k\| = \|\vec{x}_k - \vec{x}^*\| < \frac{L}{1-L} \|\vec{x}_k - \vec{x}_{k-1}\| \leq \frac{L^k}{1-L} \|\vec{x}_1 - \vec{x}_0\|.$$

□

**Bemerkungen.**

1. Gleichung (3.7) zeigt, dass die Konvergenz tatsächlich umso rascher eintritt, je kleiner $L$ ist.
2. Gleichung (3.8) bietet die Möglichkeit, bei Kenntnis von $\vec{x}_0, \vec{x}_1$ und $L$ die Qualität der $k$-ten Näherung $\vec{x}_k$ a-priori abzuschätzen.
3. Der Kontraktionssatz gilt nicht nur im Endlichdimensionalen, sondern kann auch auf allgemeine normierte Räume übertragen werden und ist somit auf unendlichdimensionale Funktionenräume anwendbar. Er ist dann einer der zentralen Sätze der konstruktiven Mathematik und dient in dieser Form zum Nachweis der Existenz und der Eindeutigkeit von Lösungen von Funktionalgleichungen und Operatorgleichungen (Differentialgleichungsprobleme, etc.) und auch zur Gewinnung von Iterationsverfahren zur näherungsweisen Lösung dieser Probleme.

Im Folgenden wird eine *Modifikation des Kontraktionssatzes* vorgestellt, in welcher auf Implementierung und Computerarithmetik Bezug genommen wird. Unter Berücksichtigung der Tatsache, dass bei tatsächlichen Implementierungen der Iteration $\vec{x}_k = \overrightarrow{T}(\vec{x}_{k-1})$ auf einem Rechner der Ablauf durch die Computerarithmetik beeinflusst wird, muss im Rechner statt $\vec{x}_k = \overrightarrow{T}(\vec{x}_{k-1})$ die gestörte Version

$$\widetilde{\vec{x}}_k := \widetilde{\overrightarrow{T}}(\widetilde{\vec{x}}_{k-1}) \tag{3.11}$$

betrachtet werden, wobei $\widetilde{\overrightarrow{T}} : \mathbb{M}^n \to \mathbb{M}^n$, mit der Menge $\mathbb{M}$ der Maschinenzahlen und einem $n$-Tupel $\mathbb{M}^n$ von Maschinenzahlen.

**Satz 3.2.5.** *Die Abbildung $\overrightarrow{T} : D \subset \mathbb{R}^n \to \mathbb{R}^n$ genüge auf $D$ den Voraussetzungen des Kontraktionssatzes und $\widetilde{\overrightarrow{T}}$ erfülle*

$$\left\| \widetilde{\overrightarrow{T}}(\widetilde{\vec{x}}) - \overrightarrow{T}(\widetilde{\vec{x}}) \right\| \leq \varepsilon, \quad \widetilde{\vec{x}} \in D \cap \mathbb{M}^n. \tag{3.12}$$

*Für den Startwert $\vec{x}_0 \equiv \widetilde{\vec{x}}_0 \in D \cap \mathbb{M}^n$ gelte*

$$\widetilde{S}_1 := S\left( \widetilde{\vec{x}}_1, \frac{1}{1-L}\left( L \left\| \widetilde{\vec{x}}_1 - \widetilde{\vec{x}}_0 \right\| + 2\varepsilon \right) \right) \subset D.$$

*Dann gilt:*

1. *Die Folge $\left\{ \widetilde{\vec{x}}_k \right\}$ liegt ganz in der Kugel $\widetilde{S}_1$.*
2. *Es gilt:*
$$\exists k^* : \left\| \widetilde{\vec{x}}_k - \vec{x}^* \right\| \leq \delta := \frac{\varepsilon}{1-L}, \quad k = k^*, k^*+1, \ldots \tag{3.13}$$
3. *Wenn $\left\| \widetilde{\vec{x}}_{k-1} - \vec{x}^* \right\| > \delta$ ist, gilt*
$$\left\| \widetilde{\vec{x}}_k - \vec{x}^* \right\| < \left\| \widetilde{\vec{x}}_{k-1} - \vec{x}^* \right\|. \tag{3.14}$$
4. *Weiters gilt die* ***a-posteriori Fehlerabschätzung***
$$\left\| \widetilde{\vec{x}}_k - \vec{x}^* \right\| \leq \frac{\varepsilon + L \left\| \widetilde{\vec{x}}_k - \widetilde{\vec{x}}_{k-1} \right\|}{1-L}. \tag{3.15}$$

*Beweis.* Siehe Literatur. □

Die a-posteriori Abschätzung (3.15) liefert eine **Abbruchbedingung** für die Iteration. Soll die Genauigkeitsforderung

$$\left\|\widetilde{\vec{x}}_k - \vec{x}^*\right\| \leq \text{TOL},$$

mit *Toleranz* TOL erfüllt werden, betrachtet man die Differenzen $\left\|\widetilde{\vec{x}}_k - \widetilde{\vec{x}}_{k-1}\right\|$ und bricht ab, sobald die Ungleichung

$$\left\|\widetilde{\vec{x}}_k - \widetilde{\vec{x}}_{k-1}\right\| \leq \frac{1}{L}\big(TOL(1-L) - \varepsilon\big)$$

erfüllt ist. Aus (3.15) folgt:

$$\left\|\widetilde{\vec{x}}_k - \vec{x}^*\right\| \leq \frac{\varepsilon + L\frac{1}{L}\big(TOL(1-L) - \varepsilon\big)}{1-L} = TOL$$

## 3.3 Newtonverfahren

Aus der Kontraktioneigenschaft der Funktion $\overrightarrow{T}(\vec{x}) = \vec{x} - R(\vec{x})\vec{F}(\vec{x})$ folgt die Konvergenz des Iterationsverfahrens. Die Geschwindigkeit der Konvergenz hängt jedoch von der Lipschitzkonstanten $L$ ab. Je näher $L$ bei 1 liegt, desto langsamer ist die Konvergenz, für $0 < L \ll 1$ hat man rasche Konvergenz. Eine naheliegende Idee ist nun, $R(\vec{x})$ so zu wählen, dass $\overrightarrow{T}'(\vec{x})$ in einer Umgebung von $\vec{x}^*$ möglichst klein wird, was dann in der Umgebung des Fixpunktes $\vec{x}^*$ eine kleine Lipschitzkonstante und damit rasche Konvergenz zu $\vec{x}^*$ zur Folge hat. Es wird $R(\vec{x})$ in $\overrightarrow{T}(\vec{x}) = \vec{x} - R(\vec{x})\vec{F}(\vec{x})$ konkret so gewählt, dass

$$\frac{\mathrm{d}\overrightarrow{T}}{\mathrm{d}\vec{x}}(\vec{x}^*) = \overrightarrow{T}'(\vec{x}^*) = 0_{n\times n} \tag{3.16}$$

gilt. Dies wird mit der Wahl von

$$R(\vec{x}) = \big(\vec{F}'(\vec{x})\big)^{-1} \tag{3.17}$$

erreicht. Es folgt daher

$$\overrightarrow{T}(\vec{x}) = \vec{x} - \big(\vec{F}'(\vec{x})\big)^{-1}\vec{F}(\vec{x}). \tag{3.18}$$

Dass (3.16) tatsächlich gilt, ist im Fall $T : \mathbb{R} \longrightarrow \mathbb{R}$ leicht nachzuweisen:

$$T'(x) = \Big(x - \big(F'(x)\big)^{-1}F(x)\Big)' = \underbrace{1 - \underbrace{\big(F'(x)\big)^{-1}F'(x)}_{=1} + \big(F'(x)\big)^{-2}F''(x)F(x)}_{=0} \tag{3.19}$$

Schließlich ist $T'(x^*) = 0$ wegen $F(x^*) = 0$. Im Falle des $\mathbb{R}^n$, also für $\overrightarrow{T} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ gilt eine analoge Überlegung, es muss lediglich mit der vektorwertigen Funktion $\vec{F}$ und der Funktionalmatrix argumentiert werden:

$$\overrightarrow{T}'(\vec{x}) = \Big(\vec{x} - \big(\vec{F}'(\vec{x})\big)^{-1}\vec{F}(\vec{x})\Big)' = \underbrace{I - \underbrace{\big(\vec{F}'(\vec{x})\big)^{-1}\vec{F}'(\vec{x})}_{=I} + \big(\vec{F}'(\vec{x})\big)^{-1}\vec{F}''(\vec{x})\big(\vec{F}'(\vec{x})\big)^{-1}\vec{F}(\vec{x})}_{=\mathbf{0}} \tag{3.20}$$

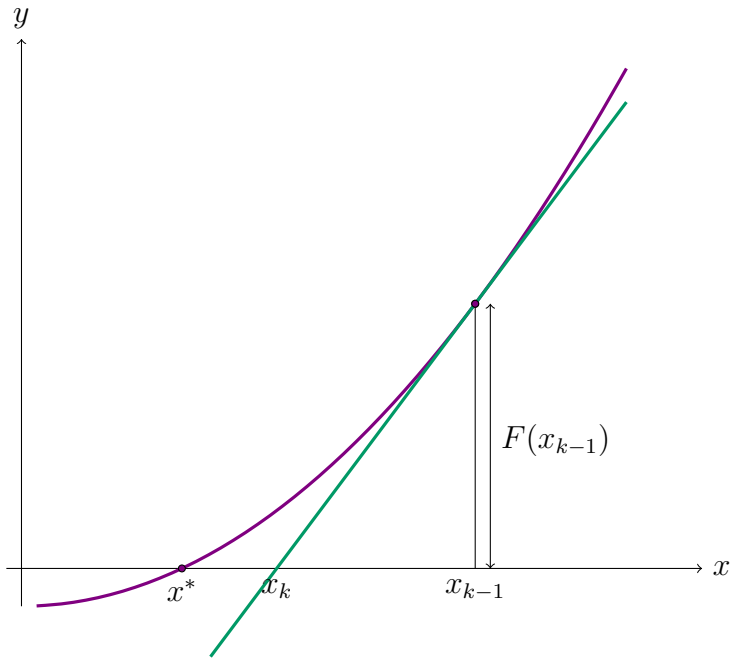



Abbildung 3.14: Geometrische Veranschaulichung des Newtonverfahrens

Der Term $\left(\vec{F}'(\vec{x})\right)^{-1}\vec{F}''(\vec{x})\left(\vec{F}'(\vec{x})\right)^{-1}\vec{F}(\vec{x})$ ist eine Matrix. Das ist auch der Grund, warum der Ausdruck sich von der Formulierung (3.19) unterscheidet. $\vec{F}(\vec{x})$ ist ein Vektor, $\left(\vec{F}'(\vec{x})\right)^{-1}$ ist die inverse Funktionalmatrix, daher ist $\left(\vec{F}'(\vec{x})\right)^{-1}\vec{F}(\vec{x})$ ebenfalls ein Vektor. $\vec{F}''(\vec{x})$ ist ein bilinearer Operator, der auf zwei Vektoren des $\mathbb{R}^n$ wirkt, also $\vec{F}'' : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$. $\vec{F}''(\vec{x})\left(\vec{F}'(\vec{x})\right)^{-1}\vec{F}(\vec{x})$ ist eine Matrix. Das Iterationsverfahren (3.4) wird mit (3.18) zum **Newtonverfahren**. Dies ist durch den Startvektor $\vec{x}_0 \in \mathbb{R}^n$ und der Iteration

$$\vec{x}_k = \overrightarrow{T}(\vec{x}_{k-1}) = \vec{x}_{k-1} - \left(\vec{F}'(\vec{x}_{k-1})\right)^{-1}\vec{F}(\vec{x}_{k-1}) \tag{3.21}$$

rekursiv definiert. Die geometrische Veranschaulichung für $n = 1$ ist in Abbildung 3.14 dargestellt.

**Beispiel 3.3.1.** *Betrachte $F(x) = e^{x-2} - x$. Für die Newtoniteration ergibt sich:*

$$x_k = x_{k-1} - \frac{e^{x_{k-1}-2} - x_{k-1}}{e^{x_{k-1}-2} - 1}$$

$$\begin{aligned} x_0 &= 0.25 \\ x_1 &= 0.1577418874\ldots \\ x_2 &= 0.1585942711\ldots \\ x_3 &= 0.1585943396\ldots \end{aligned}$$

*Die raschere Konvergenz im Vergleich zu Beispiel 3.2.1 ist offensichtlich.*

Die rasche Konvergenz des Newtonverfahrens ergibt sich für Startwerte hinreichend nahe an der gesuchten Nullstelle. Global gesehen, d.h. für beliebige Startwerte, muss das Verfahren aber nicht konvergieren.
Im Folgenden betrachten wir die reelle Funktion $F(x) = \arctan x$. Es gilt $F(0) = 0$. Der Graph dieser Funktion und der **Konvergenzbereich** , also die Menge aller Startwerte, bei denen das Iterationsverfahren konvergiert, fnden sich in Abbildung 3.15.

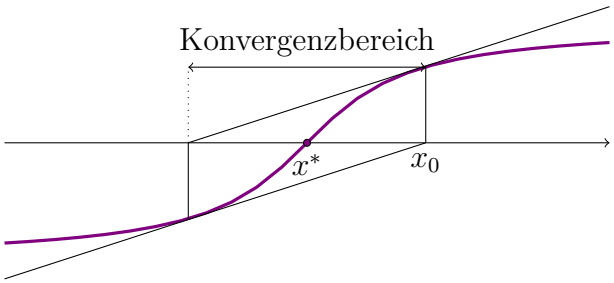



Abbildung 3.15: Konvergenzbereich von $F(x) = \arctan x$

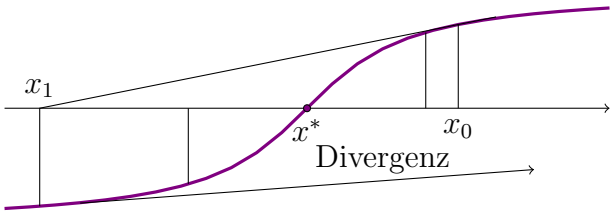



Abbildung 3.16: Divergenz von $F(x) = \arctan x$

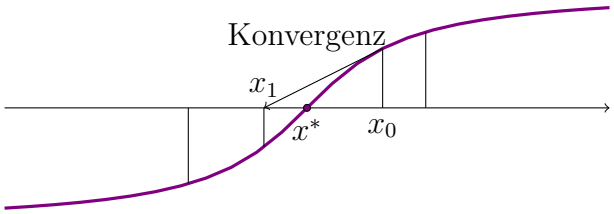



Abbildung 3.17: Konvergenz von $F(x) = \arctan x$

Die Abbildungen 3.16 und 3.17 zeigen Newtoniterationen zu verschiedenen Startwerten $x_0$. Der Startwerte im Inneren des Einzugsbereiches (Abb. 3.17) ergibt Konvergenz, außerhalb des Einzugsbereiches (Abb. 3.16) *alternierende Divergenz*, $|x_0| < |x_1| < |x_2| < \ldots$ und $x_0 > 0, x_1 < 0, x_2 > 0, \ldots$ Für Startwerte genau am Rand des Einzugsbereiches (Abb.3.15) ergibt sich die Folge

$$x_0, x_1 = -x_0, x_2 = x_0, x_3 = -x_0, \ldots$$

Im Vergleich dazu untersuchen wir die Konvergenzbereiche einer quadratischen Funktion mit Minmalstelle $\bar{x}$ und Nullstellen $x_1^*, x_2^*$, siehe Abbildung 3.18.

1. Startwert $x_0 > \overline{x}$: Konvergenz gegen $x_1^*$, der Einzugsbereich von $x_1^*$ ist also $(\overline{x}, \infty)$.
2. $x_0 < \overline{x}$: Konvergenz gegen $x_2^*$, der Einzugsbereich von $x_2^*$ ist also $(-\infty, \overline{x})$.
3. $x_0 = \overline{x}$: Das Newtonverfahren lässt sich hier nicht durchführen, da $F'(\bar{x}) = 0$.

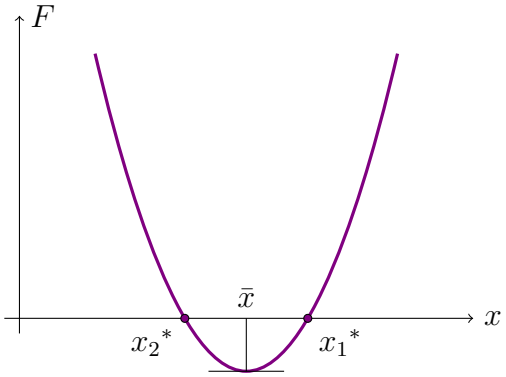



Abbildung 3.18: Konvergenzbereiche einer quadratischen Funktion

Im Falle der Divergenz des Newtonverfahrens ergibt sich

$$|F(x_0)| < |F(x_1)| < |F(x_2)| < \ldots$$

Dies liefert die Idee für das **gedämpfte Newtonverfahren**.
Man vergleicht $|F(x_k)|$ mit $|F(x_{k-1})|$. Wenn $|F(x_k)| \geq |F(x_{k-1})|$ gilt, so verkürzt man den Newtonschritt, d.h. statt

$$x_k = x_{k-1} - \big(F'(x_{k-1})\big)^{-1} F(x_{k-1})$$

betrachtet man

$$x_k^{(1)} = x_{k-1} - \lambda\big(F'(x_{k-1})\big)^{-1} F(x_{k-1}) \quad \lambda \in (0,1)$$

und untersucht, ob $|F(x_k^{(1)})| < |F(x_{k-1})|$ gilt. Falls das nicht zutrifft, betrachtet man

$$x_k^{(2)} = x_{k-1} - \frac{\lambda}{2}\big(F'(x_{k-1})\big)^{-1} F(x_{k-1})$$

usw., solange, bis für ein $j \in \mathbb{N}$

$$|F(x_k^{(j)})| < |F(x_{k-1})|$$

gilt. Dann setzt man $x_k := x_k^{(j)}$ und setzt das Newtonverfahren fort. Durch diese Strategie kann man die Konvergenzbereiche des Newtonverfahrens wesentlich vergrößern.

Das gedämpfte Newtonverfahren im Falle $\vec{F} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ wird ident formuliert, an Stelle der Beträge treten Normen, es werden die Normen $\left\|\vec{F}(\vec{x}_k)\right\|$ kontrolliert.
Im Falle des $\mathbb{R}^n$ ist außerdem zu beachten, dass die Gültigkeit der Ungleichung $\left\|\vec{F}(\vec{x}_k)\right\| < \left\|\vec{F}(\vec{x}_{k-1})\right\|$ von der Norm und der Skalierung von $\vec{F}$ abhängt. Algorithmisch kann das eine Reihe von speziellen Maßnahmen bedeuten.

## 3.4 Spezialfall: Iterative Lösung linearer Gleichungssysteme

Eine Alternative zu direkten Lösungsverfahren für lineare Gleichungssysteme (wie dem Gauß-Verfahren) liefern iterative Verfahren.
Diese werden vor allem bei großen **schwach besetzten** Matrizen angewendet, Systemen, bei denen der Großteil der Einträge verschwindet, sodass sie normalerweise mit einem Speicherbedarf von $\mathcal{O}(n)$ dargestellt werden können. Solche Matrizen entstehen etwa bei numerischen Lösungsverfahren von Differentialgleichungen wie der *Finite Elemente Methode* (FEM). Bei der Anwendung der Gauß-Elimination geht die Schwachbesetztheit und damit die Speichereffizienz im Allgemeinen verloren.
Iterative Verfahren bieten den Vorteil, dass sie auf Matrix-Vektor-Multiplikationen beruhen, deren Aufwand für schwach besetzte Matrizen nur $\mathcal{O}(n)$ ist. Die meisten iterativen Verfahren beruhen auf einer Minimumsuche für ein Funktional $\varphi(\vec{x})$, das so definiert wird, dass seine Minimalstelle $\vec{x} \in \mathbb{R}^n$ mit der Lösung von $A\vec{x} = \vec{b}$ übereinstimmt. Eine mögliche Wahl für das Funktional ist $\varphi(\vec{x}) = \|A\vec{x} - \vec{b}\|_2^2$.
Im folgenden werden verschiedene iterative Verfahren vorgestellt.

### Das Richardson-Verfahren

Das **Richardson-Verfahren** ist das einfachste iterative Verfahren. Es beruht auf dem Ansatz

$$\vec{b} = A\vec{x} = (A - I)\vec{x} + \vec{x} \Leftrightarrow \vec{x} = \vec{b} + (I - A)\vec{x}.$$

Daraus ergibt sich die Iterationsvorschrift

$$\vec{x}^{(k+1)} = \vec{b} + (I - A)\vec{x}^{(k)} = \vec{x}^{(k)} + (b - A\vec{x}^{(k)}).$$

Ausgehend von einem Startwert $\vec{x}^{(0)}$ lässt sich eindeutig die Folge $\vec{x}^{(0)}, \vec{x}^{(1)}, \vec{x}^{(2)}, \ldots$ bilden. Ist das Gleichungssystem eindeutig lösbar, so gibt es ein $\underline{\vec{x}} \in \mathbb{R}^n$ mit $\underline{\vec{x}} = A^{-1}\vec{b}$. Das Verfahren ist nur dann sinnvoll, wenn die Folge $\vec{x}^{(k)}$ für $k \to \infty$ gegen $\underline{\vec{x}}$ konvergiert. Man betrachtet daher den Fehler $\underline{\vec{x}} - \vec{x}^{(k+1)}$ nach $k+1$ Schritten. Es gilt

$$\underline{\vec{x}} - \vec{x}^{(k+1)} = \underline{\vec{x}} - \vec{x}^{(k)} - (\vec{b} - A\vec{x}^{(k)}) = \underline{\vec{x}} - \vec{x}^{(k)} - (A\underline{\vec{x}} - A\vec{x}^{(k)}) = (I - A)(\underline{\vec{x}} - \vec{x}^{(k)}).$$

Für die Norm ergibt sich daher

$$\|\underline{\vec{x}} - \vec{x}^{(k+1)}\|_2 \leq \|I - A\|_2 \|\underline{\vec{x}} - \vec{x}^{(k)}\|_2$$

und induktiv

$$\|\underline{\vec{x}} - \vec{x}^{(k+1)}\|_2 \leq \|I - A\|_2^{k+1} \|\underline{\vec{x}} - \vec{x}^{(k)}\|_2.$$

Dieser Fehler konvergiert gegen 0 und die Folge damit gegen die Lösung des Gleichungssystems genau dann, wenn $\|I - A\|_2 < 1$. In diesem Fall nennt man die Iteration eine **Kontraktion**.

**Algorithmus 3.4.1** (RICHARDSON-VERFAHREN)**.**

```
Richardson(A,b,ε)

x := 0
y := 0

while err > ε
      for i = 1,...,n
            y_i := x_i + b_i
            for j = 1,...,n
                  y_i := y_i - a_ij x_j
            end
      end
      x := y
      err := |Ax - b|
end
```

Algorithmus 3.4.1 zeigt die Implementierung des Richardson-Verfahrens. Der Wert $\epsilon$ ist eine vorher gewählte Schranke für die gewünschte Genauigkeit der Lösung.

**Beispiel 3.4.2.** *Sei*

$$A = \begin{pmatrix} 1.5 & 0.25 \\ 0.2 & 1.75 \end{pmatrix}$$

*gegeben. Die Spektralnorm von*

$$I_2 - A = \begin{pmatrix} -0.5 & -0.25 \\ -0.2 & -0.75 \end{pmatrix}$$

*ist durch*

$$\|I_2 - A\|_2 \approx 0.8812 < 1$$

*gegeben, das Richardson-Verfahren konvergiert also. Betrachtet man das Gleichungssystem $A\vec{x} = \vec{b}$ mit $\vec{b} = (1,1)^T$, so sind* 31 *Iterationsschritte notwendig, um einen Fehler der Größenordnung $err \approx 10^{-2}$ zu erhalten.*

## Das Jacobi-Verfahren

Das Richardson-Verfahren ist nur sinnvoll, wenn die Ausgangsmatrix $A$ „nahe “an der Einheitsmatrix ist. Das ist in der Praxis meist nicht der Fall. Dennoch kann das Verfahren als Ausgangspunkt besserer Methoden gewählt werden.
Um eine Matrix näher an der Einheitsmatrix zu erhalten, definiert man zunächst

$$D = diag(a_{11}, \ldots, a_{nn}).$$

Mit der Zusatzannahme $a_{ii} \neq 0$ für alle Diagonaleinträge von $A$ gilt dann

$$D^{-1} = diag\left(\frac{1}{a_{11}}, \ldots, \frac{1}{a_{nn}}\right)$$

und die Matrix $D^{-1}A$ hat als Diagonalelemente nur noch Einser. Die Gleichung $A\vec{x} = \vec{b}$ ist äquivalent zu

$$D^{-1}A\vec{x} = D^{-1}\vec{b}.$$

Das Richardson-Verfahren kann nun für die Matrix $\tilde{A} = D^{-1}A$ und den Vektor $\vec{\tilde{b}} = D^{-1}\vec{b}$ durchgeführt werden. Es ergibt sich die Iteration

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + (\vec{\tilde{b}} - \tilde{A}\vec{x}^{(k)}) = \vec{x}^{(k)} + D^{-1}(\vec{b} - A\vec{x}^{(k)})$$

bzw.

$$D\vec{x}^{(k+1)} = D\vec{x}^{(k)} + (\vec{b} - A\vec{x}^{(k)}) = \vec{b} + (D - A)\vec{x}^{(k)}.$$

Dieselbe Iterationsvorschrift erhält man durch das Abspalten der Diagonalelemente in der Form

$$\vec{b} = A\vec{x} = (D + A - D)\vec{x} = Dx + (A - D)\vec{x} \Leftrightarrow D\vec{x} = \vec{b} + (D - A)\vec{x}.$$

In dieser Formulierung spricht man vom **Jacobi-Verfahren**. Dieses ist nicht nur für eine größere Klasse von Matrizen durchführbar, sondern liefert in vielen Fällen auch eine schnellere Konvergenz als das Richardson-Verfahren.

**Algorithmus 3.4.3** (Jacobi-Verfahren)**.**

```
Jacobi(A,b,ε)
x := 0
y := 0

while err > ε
      for i = 1,...,n
            y_i := b_i
            for j = 1,...,i-1,i+1,...,n
                  y_i := y_i - a_ij x_j
            end
            y_i := y_i / a_ii
      end
      x := y
      err := |Ax - b|
end
```

Im folgenden sehr einfachen Beispiel wird der Ablauf des Algorithmus verdeutlicht.

**Beispiel 3.4.4.** *Sei das Gleichungssystem $Ax = b$ mit*

$$A = \begin{pmatrix} 1 & 3 \\ 0 & -1 \end{pmatrix} \quad \textit{und} \quad \vec{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

*gegeben. Die Diagonalmatrix $D$ hat die Form*

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

*Der erste Iterationsschritt liefert*

$$\left.\begin{array}{rrrr} y_1 = & b_1 = & & 1 \\ y_1 = & y_1 - a_{12}x_2 = & 1 - 3 \cdot 0 = & 1 \\ y_1 = & y_1/a_{11} = & 1/1 = & 1 \\ & & & \\ y_2 = & b_2 = & & 1 \\ y_2 = & y_2 - a_{21}x_1 = & 1 - 0 \cdot 0 = & 1 \\ y_2 = & y_2/a_{22} = & 1/(-1) = & -1 \end{array}\right\} \Rightarrow \vec{x} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

*Der Fehler ist* $\|A\vec{x} - \vec{b}\|_2 = 3$*, also noch recht groß, weshalb ein zweiter Iterationsschritt durchgeführt wird:*

$$\left.\begin{array}{rrrr} y_1 = & b_1 = & & 1 \\ y_1 = & y_1 - a_{12}x_2 = & 1 - 3 \cdot (-1) = & 4 \\ y_1 = & y_1/a_{11} = & 4/1 = & 4 \\ & & & \\ y_2 = & b_2 = & & 1 \\ y_2 = & y_2 - a_{21}x_1 = & 1 - 0 \cdot 1 = & 1 \\ y_2 = & y_2/a_{22} = & 1/(-1) = & -1 \end{array}\right\} \Rightarrow \vec{x} = \begin{pmatrix} 4 \\ -1 \end{pmatrix}.$$

*Diesmal ergibt sich* $A\vec{x} = \vec{b}$*, daher ist die Iteration abgeschlossen.*

Das nächste Beispiel verdeutlicht die schnellere Konvergenz des Jacobi-Verfahrens gegenüber dem Richardson-Verfahren.

**Beispiel 3.4.5.** *Sei erneut das Gleichungssystem aus Beispiel 3.4.2 gegeben. Die Lösung* $\vec{x} = (\frac{4}{7}, \frac{52}{105})^T$*, die das Jacobi-Verfahren nach nur zwei Schritten liefert, hat einen Fehler von* $\|A\vec{x} - \vec{b}\|_2 = 0.0269$*. Für einen Fehler der gleichen Größenordnung sind also nur zwei Schritte statt* 31 *notwendig.*

## Das Gauß-Seidel-Verfahren

Eine weitere Methode, um dem Verfahren zu (schnellerer) Konvergenz zu verhelfen, ist das Aufteilen oder *Splitting* von $A$ in den Diagonalanteil $D$, den unteren Dreiecksteil $L$ und den oberen Dreiecksteil $U$, also $A = L + D + U$. Wie zuvor ergibt sich aus

$$\vec{b} = A\vec{x} = (L + D + U)\vec{x} = (L + D)\vec{x} + U\vec{x}$$

die Iterationsvorschrift

$$(L + D)\vec{x}^{(k+1)} = \vec{b} - U\vec{x}^{(k)} = \vec{b} - (A - L - D)\vec{x}^{(k)} = (L + D)\vec{x}^{(k)} + (\vec{b} - A\vec{x}^{(k)})$$

bzw.

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + (L + D)^{-1}(\vec{b} - A\vec{x}^{(k)}).$$

Dieses Verfahren wird als **Gauß-Seidel-Verfahren** bezeichnet. Es muss zwar in jedem Schritt ein lineares Gleichungssystem gelöst werden, die Systemmatrix $L + D$ ist jedoch eine untere Dreiecksmatrix, was das Lösen sehr einfach macht.
Das Verfahren unterscheidet sich vom Jacobi-Verfahren dadurch, dass zur Berechnung der nächsten Iterierten $\vec{x}^{(k+1)}$ sofort die in diesem Schritt bereits vorliegenden Komponenten von $\vec{x}^{(k+1)}$ eingesetzt

werden. Das führt zu einem einfacheren Programmcode. Das Jacobi-Verfahren gehört daher zur Klasse der *Gesamtschrittverfahren*, während das Gauß-Seidel-Verfahren ein *Einzelschrittverfahren* ist.
Die Konvergenz des Verfahrens hängt ab von $\|I - (L+D)^{-1}A\|_2$. Offensichtlich entspricht das Verfahren genau dem Richardson-Verfahren angewendet auf das *präkonditionierte* Problem

$$(L+D)^{-1}A\vec{x} = (L+D)^{-1}\vec{b}.$$

Daher wird die Matrix $L+D$ als **Präkonditionierer** bezeichnet.

**Algorithmus 3.4.6** (Gauss-Seidel-Verfahren)**.**

`Gauss-Seidel(`$A$`,`$\vec{b}$`,`$\epsilon$`)`

$\vec{x} := \vec{0}$

`while` $err > \epsilon$
  `for` $i = 1, \ldots, n$
    $x_i := b_i$
    `for` $j = 1, \ldots, i-1, i+1, \ldots, n$
      $x_i := x_i - a_{ij}x_j$
    `end`
    $x_i := x_i / a_{ii}$
  `end`
  $err := |A\vec{x} - \vec{b}|$
`end`

**Beispiel 3.4.7.** *Wendet man das Gauß-Seidel-Verfahren auf das Gleichungssystem aus Beispiel 3.4.4 an, so unterscheiden sich die durchzuführenden Schritte kaum von denen des Jacobi-Verfahrens. Einzig die Umspeicherung der Werte von $\vec{y}$ in $\vec{x}$ erspart man sich.*
*1. Iterationsschritt:*

$$\begin{aligned} x_1 &= b_1 = 1 \\ x_1 &= x_1 - a_{12}x_2 = 1 - 3\cdot 0 = 1 \\ x_1 &= x_1/a_{11} = 1/1 = 1 \end{aligned}$$

$$\begin{aligned} x_2 &= b_2 = 1 \\ x_2 &= x_2 - a_{21}x_1 = 1 - 0\cdot 0 = 1 \\ x_2 &= x_2/a_{22} = 1/(-1) = -1 \end{aligned}$$

*Fehler $\|A\vec{x} - \vec{b}\|_2 = 3 \Rightarrow$ 2. Iterationsschritt:*

$$\begin{aligned} x_1 &= b_1 = 1 \\ x_1 &= x_1 - a_{12}x_2 = 1 - 3\cdot(-1) = 4 \\ x_1 &= x_1/a_{11} = 4/1 = 4 \end{aligned}$$

$$\begin{aligned} x_2 &= b_2 = 1 \\ x_2 &= x_2 - a_{21}x_1 = 1 - 0\cdot 1 = 1 \\ x_2 &= x_2/a_{22} = 1/(-1) = -1 \end{aligned}$$

Alle bisher beschriebenen Verfahren gehören zur großen Klasse der *stationären iterativen Verfahren*, da die Iterationsvorschrift unabhängig vom aktuellen Iterationsschritt ist. Im Folgenden werden zwei nicht stationäre Verfahren erwähnt, allerdings nicht näher behandelt.

## Weitere Verfahren

Für symmetrische, positiv definite Matrizen $A$ ist das **Gradientenverfahren** definiert. Dabei wird das Problem des linearen Gleichungssystems übergeführt in das Minimierungsproblem einer Funktion $F(x)$. Die Iterationsvorschrift lautet

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + \alpha_k \vec{d}^{(k)},$$

wobei die Koeffizienten $\alpha_k$ und $\vec{d}^{(k)}$ vom aktuellen Iterationsschritt abhängen. Die Iteration ist daher nicht stationär. Der Name rührt daher, dass als Suchrichtung $\vec{d}^{(k)}$ die Richtung des steilsten Abstiegs, und damit der Gradient der Funktion $F$ im Punkt $\vec{x}^{(k)}$ gewählt wird.
Ein weiteres nicht stationäres Verfahren ist die **Vektoriteration** zur Bestimmung des grössten Eigenwerts und des zugehörigen Eigenvektors. Die Iterationsvorschrift für eine symmetrische, positiv definite Matrix $A$ lautet

$$\vec{x}^{(k+1)} = \frac{A\vec{x}^{(k)}}{\|A\vec{x}^{(k)}\|}$$

für eine beliebige Matrixnorm $\|.\|$. Konvergiert die Folge, so ist

$$\lim_{k\to\infty} \vec{x}^{(k)} =: \underline{\vec{x}} = \frac{A\underline{\vec{x}}}{\|A\underline{\vec{x}}\|}$$

der Eigenvektor mit Länge Eins zum größten Eigenwert $\lambda = \|A\underline{\vec{x}}\|$.
Um alle Eigenwerte und in manchen Fällen Eigenvektoren einer quadratischen Matrix $A$ zu bestimmen, verwendet man häufig das **QR-Verfahren**, welches in der Literatur gefunden werden kann.

# Kapitel 4

# Interpolation und Approximation

## 4.1 Einleitende Betrachtungen

Bei einem **Interpolationsproblem** sind im einfachsten Fall Paare $(x_k, y_k)$ $k = 0, \ldots n$ gegeben und *einfache Funktionen* $g(x)$ mit $g(x_k) = y_k$ gesucht. Klassen einfacher Funktionen können Polynome, stückweise Polynome (Splines) oder auch rationale Funktionen sein.
Verwandt ist das Thema **Approximation**. Gegeben sind dabei eine geeignete Norm $\|\cdot\|$ und eine Funktion $f$. Gesucht ist wiederum eine *einfache Funktion* $g$, die jetzt im Sinne der Norm eine gute Approximation sein soll, z.B. mit $\|g - f\|$ minimal im Vergleich zu anderen einfachen Funktionen aus einer gegebenen Klasse von Funktionen, siehe weiterführende Literatur.

Als Beispiel für die Notwendigkeit von Interpolation betrachten wir die **Numerische Integration** (siehe Kapitel 5).

– *Trapezregel:* Der Integrand $f(x)$, dessen Stammfunktion $F(x)$ nicht geschlossen darstellbar oder nicht bekannt ist, wird durch einen interpolierenden Polygonzug $g(x)$ ersetzt. $\int_a^b g(x)\,\mathrm{d}x$ kann dann berechnet werden und ist eine Näherung für $\int_a^b f(x)\,\mathrm{d}x$, siehe Kapitel 1, Seite 14 und Abb. 4.1.

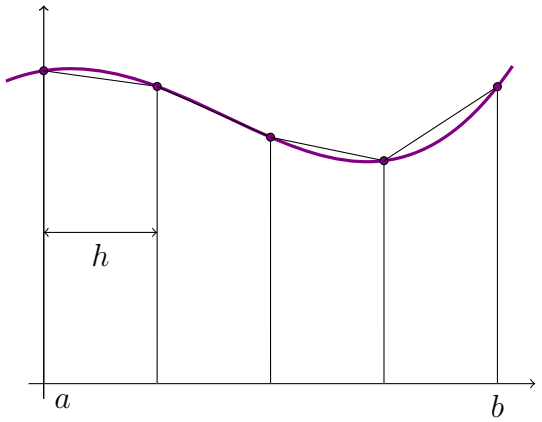



Abbildung 4.1: Trapezregel

– *Simpson Regel:* Der Integrand wird stückweise durch interpolierende quadratische Polynome ersetzt, in der Abbildung 4.2 etwa durch zwei Polynome, das erste $c_0^{(1)} + c_1^{(1)}x + c_2^{(1)}x^2$ ist durch die Punkte $(x_0, y_0 = f(x_0))$, $(x_1, y_1 = f(x_1))$, $(x_2, y_2 = f(x_2))$ festgelegt, das zweite Polynom $c_0^{(2)} + c_1^{(2)}x + c_2^{(2)}x^2$ durch die Punkte $(x_2, f(x_2))$, $(x_3, f(x_3))$, $(x_4, f(x_4))$.

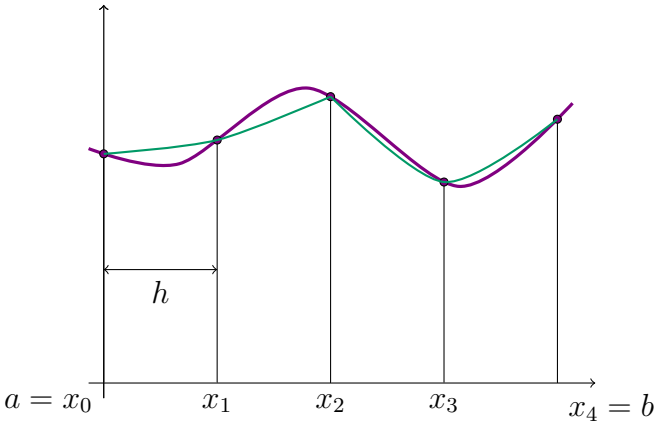



Abbildung 4.2: Simpsonregel

**Darstellung von Standardfunktionen.**
Am Computer können nur die 4 Grundrechnungsarten $*$, $/$, $+$, $-$ ausgeführt werden, d.h. es können etwa Polynome bzw. rationale Funktionen, die in endlicher Weise mit den Grundrechnungsarten aufgebaut sind, ausgewertet werden. Elementare Funktionen oder Standardfunktionen wie z.B.

$$\sin x, \quad \arcsin x, \quad e^x, \quad \ln x, \ \ldots$$

können nicht mit Hilfe der Grundrechenoperationen in endlicher Weise dargestellt werden. Um diese Funktionen am Computer auswerten zu können, muss man sie durch in endlich vielen Rechenschritten berechenbare Funktion (d.h. durch Polynome oder rationale Funktionen) ersetzen. Etwa die Taylorreihenentwicklung $e^x = \sum_{j=0}^{\infty} \frac{x^j}{j!}$ kann durch ein (endliches) Taylorpolynom $e^x \approx \sum_{j=0}^{n} \frac{x^j}{j!}$ ersetzt werden, $e^x$ nur näherungsweise darstellt.

**Anforderungen an die Ersatzfunktionen.**
Je nach Anwendungsfall werden an diese *einfachen Funktionen* oder *Ersatzfunktionen* verschiedene Anforderungen gestellt. Bei der numerischen Integration etwa sollen die Programme immer wieder auf verschiedenste Integranden $f$ angewendet werden. Das Ersetzen von $f(x)$ durch $g(x)$ muss daher einfach und unproblematisch möglich sein.
Bei der Implementierung von Standardfunktionen kann man dagegen bei der Aufstellung der Ersatzfunktion einen sehr großen Aufwand treiben.
Da so ein Programm für eine Standardfunktion dann immer wieder aufgerufen wird, ist hier die Effizienz besonders wichtig: einerseits hat man strenge Genauigkeitsforderungen (meist wird verlangt, dass man bei der Auswertung nicht mehr als einen elementaren Rundungsfehler macht), anderseits möchte man dieses Genauigkeitsniveau mit möglichst wenig Rechenoperationen erreichen (z.B. bei polynomialer Ersatzfunktion $g(x)$ mit möglichst niedrigem Polynomgrad). Um dieses schwierige Ziel zu ereichen, lohnt es sich i.A. einen größeren Entwicklungsaufwand vor der Programmierung und Implementierung in Kauf zu nehmen.

Bei der Festlegung von $g(x)$ müssen i.a. zwei Entscheidungen getroffen werden:

1. Welcher Funktionenklasse soll $g$ angehören?

   – Polynom vom Grad 3:
   $$c_0 + c_1 x + c_2 x^2 + c_3 x^3 \qquad c_i \in \mathbb{R}$$

   – Gerades Polynom vom Grad 4:
   $$c_0 + c_2 x^2 + c_4 x^4$$

   – Ungerades Polynom von Grad 5:
   $$c_1 x + c_3 x^3 + c_5 x^5$$

   – Rationale Funktion: Zählerpolynom Grad 3, Nennerpolynom Grad 5
   $$\frac{c_0 + c_1 x + c_2 x^2 + c_3 x^3}{d_0 + d_1 x + d_2 x^2 + d_3 x^3 + d_4 x^4 + d_5 x^5}$$

   – Stückweises Polynom vom Grad 1 (Polygonzug)

   – Stückweises Polynom vom Grad 2 (wie bei der Simpsonregel)

   $$\vdots$$

   *Ist die Funktion $g(x)$ ein Polynom, so wird sie meist mit $\mathbf{p}(\mathbf{x})$ bezeichnet.*

2. Wodurch soll $g$ festgelegt werden?

   – **Interpolation:** Dabei wird meist verlangt, dass die einfache Funktion $g(x)$ an gewissen Stellen $x_i$ die Werte von $f$ animmt, also
   $$\begin{aligned} g(x_0) &= f(x_0), \\ g(x_1) &= f(x_1), \\ &\vdots \\ g(x_n) &= f(x_n) \end{aligned}$$
   für eine bestimmte Menge $x_0, x_1, \ldots, x_n$ von *Interpolationsknoten*. Manchmal wird eine Interpolationsfunktion auch durch andere Interpolationsdaten, wie die Werte der Ableitung(en) an Interpolationsknoten, festgelegt. Ein Polynom $p(x)$ vom Grad 3 kann etwa durch folgende Forderungen festgelegt werden:
   $$\begin{aligned} p(x_0) &= f(x_0), \\ p'(x_0) &= f'(x_0), \\ p''(x_0) &= f''(x_0), \\ p(x_1) &= f(x_1). \end{aligned}$$

   – **Ausgleichende Interpolation:** Es gibt mehr Interpolationsdaten als Unbekannte in der zu bestimmenden Ersatzfunktion.

– **Taylorpolynom:** Das Approximationspolynom $p$ ist eine endliche Partialsumme

$$p(x) = \sum_{i=0}^{n} \frac{1}{i!} f^{(i)}(x_0)(x - x_0)^i$$

der Taylorreihe

$$\sum_{i=0}^{\infty} \frac{1}{i!} f^{(i)}(x_0)(x - x_0)^i$$

von $f$, dabei wird hinreichende Glattheit von $f$ vorausgesetzt. Ist etwa $f$ in dem betrachteten Intervall $n + 1$ mal stetig differenzierbar, so existiert nicht nur das Taylorpolynom $n$-ten Grades, sondern man kann auch noch den Fehler $p(x) - f(x)$ durch eine geeignete Restglieddarstellung des Taylorpolynoms abschätzen.

**Beispiel 4.1.1.** *Die Funktion $f(x) = \sin x$, $x \in [0, \frac{\pi}{2}]$ soll auf verschiedene Weisen interpoliert werden.*

*a)* $p(x)\ldots$ Polynom vom Grad 1, *interpoliert die Daten*

$$\begin{aligned} (x_0, f(x_0)) &= (0, \sin 0) &&= (0, 0), \\ (x_1, f(x_1)) &= \left(\tfrac{\pi}{2}, \sin \tfrac{\pi}{2}\right) &&= \left(\tfrac{\pi}{2}, 1\right). \end{aligned}$$

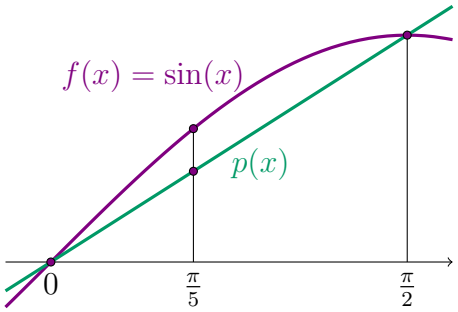



Abbildung 4.3: Vergleich von $p(x) = \frac{2}{\pi}x$ und $\sin x$ an $x = \frac{\pi}{5}$

*Es ergibt sich*

$$p(x) = \frac{2}{\pi}x \;=\; (0.6366197724\ldots)x.$$

*Vergleich von $p(x)$ und $f(x)$ etwa an der Stelle $x = \frac{\pi}{5}$, siehe Abb. 4.3:*

$$\boxed{p\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right) \;=\; -0.1877852523\ldots}$$

*b)* $p(x)\ldots$ Polynom vom Grad 2, *interpoliert die Daten*

$$\begin{aligned} (x_0, f(x_0)) &= (0, \sin 0) &&= (0, 0), \\ (x_1, f(x_1)) &= \left(\tfrac{\pi}{2}, \sin \tfrac{\pi}{2}\right) &&= \left(\tfrac{\pi}{2}, 1\right), \\ (x_2, f(x_2)) &= \left(\tfrac{\pi}{4}, \sin \tfrac{\pi}{4}\right) &&= \left(\tfrac{\pi}{4}, \tfrac{\sqrt{2}}{2}\right). \end{aligned}$$

*es kommt also zusätzlich zu den Interpolationsdaten von a) noch ein weiterer Interpolationsknoten hinzu. Es ergibt sich*

$$p(x) = (0.6366197724\ldots)x - (0.3357488674\ldots)x\left(x - \frac{\pi}{2}\right).$$

*Das ist die Newtonsche Darstellung des Interpolationspolynoms, siehe Abschnitt 4.2. Das Polynom ist das lineare Polynom aus a) plus ein zusätzliches, quadratisches Polynom. Die Newtonsche Darstellung eignet sich sehr gut, um die Interpolationsdatenmenge zu erweitern.*
*Fehler bei* $x = \frac{\pi}{5}$*:*

$$\boxed{p\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right) \;=\; 0.01103725769\ldots,}$$

*Der Fehler* $p(x) - \sin x$ *an der Stelle* $x = \frac{\pi}{5}$ *ist ungefähr um einen Faktor* 10 *kleiner als in a).*

*c)* $p(x)\ldots$ Polynom vom Grad 3, *interpoliert die Daten*

$$\begin{array}{rcccl}
(x_0, f(x_0)) & = & (0, \sin 0) & = & (0, 0), \\
(x_1, f(x_1)) & = & \left(\frac{\pi}{2}, \sin\frac{\pi}{2}\right) & = & \left(\frac{\pi}{2}, 1\right), \\
(x_2, f(x_2)) & = & \left(\frac{\pi}{4}, \sin\frac{\pi}{4}\right) & = & \left(\frac{\pi}{4}, \frac{\sqrt{2}}{2}\right), \\
(x_3, f(x_3)) & = & \left(\frac{\pi}{6}, \sin\frac{\pi}{6}\right) & = & \left(\frac{\pi}{6}, \frac{1}{2}\right).
\end{array}$$

*Es ergibt sich die* Newtonsche Darstellung

$$\begin{array}{rcl}
p(x) & = & (0.6366197724\ldots)x \\
& - & (0.3357488674\ldots)x\left(x - \frac{\pi}{2}\right) \\
& - & (\,0.1121410965\ldots)x\left(x - \frac{\pi}{2}\right)\left(x - \frac{\pi}{4}\right),
\end{array}$$

*zum quadratischen Polynom aus b) wird ein kubisches Polynom addiert..*
*Fehler bei* $x = \frac{\pi}{5}$*:*

$$\boxed{p\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right) \;=\; -0.00025624\ldots} \tag{4.1}$$

*d) Da* $\sin x$ *eine ungerade Funktion ist, liegt es nahe, mit einem* ungeraden Polynom,

$$p(x) \;=\; c_1 x + c_3 x^3,$$

*zu arbeiten. Die Konstanten* $c_1, c_3$ *werden so ermittelt, dass bezüglich der Daten*

$$\begin{array}{rcccl}
(x_0, f(x_0)) & = & \left(\frac{\pi}{6}, \sin\frac{\pi}{6}\right) & = & \left(\frac{\pi}{6}, \frac{1}{2}\right), \\
(x_1, f(x_1)) & = & \left(\frac{\pi}{3}, \sin\frac{\pi}{3}\right) & = & \left(\frac{\pi}{3}, \frac{\sqrt{3}}{2}\right)
\end{array}$$

*interpoliert wird. Das führt auf ein lineares Gleichungssystem mit* 2 *Gleichungen für die* 2 *Unbekanten* $c_1, c_3$*. Berechnet man die beiden Koeffizienten* $c_1, c_3$ *durch Lösen des Systems*

$$\begin{array}{rcl}
x = \frac{\pi}{6}: \;\; c_1\frac{\pi}{6} + c_3(\frac{\pi}{6})^3 & = & \frac{1}{2}, \\
x = \frac{\pi}{3}: \;\; c_1\frac{\pi}{3} + c_3(\frac{\pi}{3})^3 & = & \frac{\sqrt{3}}{2},
\end{array}$$

*erhält man*

$$p(x) = (0.997575097\ldots)x - (0.1555519069\ldots)x^3.$$

*Fehler bei* $x = \frac{\pi}{5}$*:*

$$\boxed{p\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right) \;=\; 0.00042298\ldots}$$

*Hier liefern nur zwei Interpolationsknoten etwa dieselbe Genauigkeit vier Interpolationsknoten in c)!*

*e)* Taylorpolynom*:* $\sin x \approx x - \frac{x^3}{3!} = p(x)$
*Fehler bei* $x = \frac{\pi}{5}$*:*

$$\boxed{p\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right) \;=\; 0.000808442\ldots}$$

*Das Taylorpolynom approximiert besonders gut in der Nähe der Entwicklungsstelle* $x = 0$*. Für größere* $x$*-Werte wird die Approximationsqualität schlechter.*

*f)* $p(x)$ *wie bei d) und e) von der Form*

$$p(x) \;=\; c_1 x + c_3 x^3,$$

*im Gegensatz zu d) jedoch durch* 3 *Interpolationsdaten*

$$\begin{array}{rcccl} (x_0, f(x_0)) & = & (\frac{\pi}{6}, \sin(\frac{\pi}{6})) & = & (\frac{\pi}{6}, \frac{1}{2}) \\ (x_1, f(x_1)) & = & (\frac{\pi}{4}, \sin(\frac{\pi}{4})) & = & (\frac{\pi}{4}, \frac{\sqrt{2}}{2}) \\ (x_2, f(x_2)) & = & (\frac{\pi}{3}, \sin(\frac{\pi}{3})) & = & (\frac{\pi}{3}, \frac{\sqrt{3}}{2}). \end{array}$$

*festgelegt. Die beiden Koeffizienten* $c_1, c_3$ *sind daher durch drei Bedingungen festgelegt, also erfolgt* ausgleichende Interpolation*. Die Quadratsumme*

$$\begin{array}{rcl} E(c_1, c_3) & := & \left(c_1 \frac{\pi}{6} + c_3 \left(\frac{\pi}{6}\right)^3 - \frac{1}{2}\right)^2 + \left(c_1 \frac{\pi}{4} + c_3 \left(\frac{\pi}{4}\right)^3 - \frac{\sqrt{2}}{2}\right)^2 \\ & + & \left(c_1 \frac{\pi}{3} + c_3 \left(\frac{\pi}{3}\right)^2 - \frac{\sqrt{3}}{2}\right)^2 \end{array}$$

*soll minimal werden. Notwendige Bedingungen für ein Minimum sind*

$$\frac{\partial E}{\partial c_1} \;=\; 0 \quad \textit{und} \quad \frac{\partial E}{\partial c_3} \;=\; 0,$$

*also*

$$\begin{array}{lcll} 2\left(c_1 \frac{\pi}{6} + c_3 (\frac{\pi}{6})^3 - \frac{1}{2}\right) \frac{\pi}{6} & + & 2\left(c_1 \frac{\pi}{4} + c_3 (\frac{\pi}{4})^3 - \frac{\sqrt{2}}{2}\right) \frac{\pi}{4} & \\ & + & 2\left(c_1 \frac{\pi}{3} + c_3 (\frac{\pi}{3})^3 - \frac{\sqrt{3}}{2}\right) \frac{\pi}{3} & = 0 \\ & \textit{und} & & \\ 2\left(c_1 \frac{\pi}{6} + c_3 (\frac{\pi}{6})^3 - \frac{1}{2}\right) \left(\frac{\pi}{6}\right)^3 & + & 2\left(c_1 \frac{\pi}{4} + c_3 (\frac{\pi}{4})^3 - \frac{\sqrt{2}}{2}\right) \left(\frac{\pi}{4}\right)^3 & \\ & + & 2\left(c_1 \frac{\pi}{3} + c_3 (\frac{\pi}{3})^3 - \frac{\sqrt{3}}{2}\right) \left(\frac{\pi}{3}\right)^3 & = 0. \end{array}$$

*Es ergibt sich das folgende lineare Gleichungssystem zur Berechnung von* $c_1, c_3$*:*

$$\begin{array}{rcl} c_1 \left[\left(\frac{\pi}{6}\right)^2 + \left(\frac{\pi}{4}\right)^2 + \left(\frac{\pi}{3}\right)^2\right] + c_3 \left[\left(\frac{\pi}{6}\right)^4 + \left(\frac{\pi}{4}\right)^4 + \left(\frac{\pi}{3}\right)^4\right] & = & \pi \left(\frac{1}{12} + \frac{\sqrt{2}}{8} + \frac{\sqrt{3}}{6}\right), \\ c_1 \left[\left(\frac{\pi}{6}\right)^4 + \left(\frac{\pi}{4}\right)^4 + \left(\frac{\pi}{3}\right)^4\right] + c_3 \left[\left(\frac{\pi}{6}\right)^6 + \left(\frac{\pi}{4}\right)^6 + \left(\frac{\pi}{3}\right)^6\right] & = & \frac{1}{2}\left(\frac{\pi}{6}\right)^3 + \frac{\sqrt{2}}{2}\left(\frac{\pi}{4}\right)^3 + \frac{\sqrt{3}}{2}\left(\frac{\pi}{3}\right)^3. \end{array}$$

*Daraus folgt*

$$p(x) = (0.9964027665\ldots)x - (0.1546327342\ldots)x^3.$$

*Fehler bei* $x = \frac{\pi}{5}$*:*

$$\boxed{p\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right) \;=\; -0.0000836\ldots}$$

## 4.2 Lagrange- und Hermiteinterpolation

Unter **Lagrangeinterpolation** versteht man die folgende Interpolationsaufgabe. An die Daten

$$(x_0, f(x_0) =: f_0),\ (x_1, f(x_1) := f_1),\ \ldots,\ (x_n, f(x_n) := f_n) \tag{4.2}$$

soll ein Interpolationspolynom vom Grad $n$,

$$p(x) = c_0 + c_1 x + \cdots + c_n x^n, \tag{4.3}$$

angepasst werden. Die $n+1$ Interpolationsbedingungen $p(x_i) = f(x_i)$, $i = 0, 1, 2, \ldots n$ legen die $n+1$ Koeffizienten $c_0, \ldots, c_n$ eindeutig fest, wenn sämtliche $x_i$ verschieden sind:

$$\begin{array}{rrcl}
x_0: & c_0 + c_1 x_0 + c_2 x_0^2 + \cdots + c_n x_0^n & = & f(x_0), \\
x_1: & c_0 + c_1 x_1 + c_2 x_1^2 + \cdots + c_n x_1^n & = & f(x_1), \\
x_2: & c_0 + c_1 x_2 + c_2 x_2^2 + \cdots + c_n x_2^n & = & f(x_2), \\
\vdots & & \vdots & \\
x_n: & c_0 + c_1 x_n + c_2 x_n^2 + \cdots + c_n x_n^n & = & f(x_n).
\end{array} \tag{4.4}$$

Das ist ein lineares Gleichungssystem mit Unbekanntenvektor $(c_0, c_1, \ldots, c_n)^\top$. Die Koeffizientenmatrix ist eine sogenannte *Vandermondematrix*

$$\begin{pmatrix}
1 & x_0 & x_0^2 & \ldots & x_0^n \\
1 & x_1 & x_1^2 & \ldots & x_1^n \\
\vdots & \vdots & \vdots & & \vdots \\
1 & x_n & x_n^2 & \ldots & x_n^n
\end{pmatrix}.$$

Aus der Regularität der Matrix folgt die eindeutige Lösbarkeit der Lagrangeschen Interpolationsaufgabe. In (4.3) ist das Interpolationspolynom $p(x)$ bezüglich der soganntен *Monombasis*

$$1, x, x^2, \ldots, x^n \tag{4.5}$$

dargestellt. Der Basisbegriff wird dabei wie in der linearen Algebra verwendet: Analog zur Darstellung eines Vektors $\vec{x} \in \mathbb{R}^3$ als Linearkombination dreier Basisvektoren, z.B.

$$\vec{x} \;=\; \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \;=\; x_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + x_3 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

ist jedes Polynom $p(x)$ eindeutig festgelegt als Linearkombination

$$p(x) \;=\; c_0 e_0 + c_1 e_1 + \cdots + c_n e_n,$$

etwa mit den Basisfunktionen

$$e_0 = e_0(x) \equiv 1, \quad e_1 = e_1(x) = x, \quad e_2 = e_2(x) = x^2, \quad \ldots, \quad e_n = e_n(x) = x^n. \tag{4.6}$$

Im Prinzip kann man das Lagrange-Interpolationspolynom tatsächlich durch Lösen des Gleichungssystems (4.4) erhalten; das ist aber i.A. nicht effizient und schlecht konditioniert.
Für andere Möglichkeiten der Berechnung werden weitere Basisdarstellungen von Polynomen vom Grad $n$ studiert. Andere Basen für den Raum der Polynome vom Grad $n$ sind z.B. die **Lagrange-Polynome**

$$\varphi_i(x) := \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} \qquad i = 0, 1, \ldots, n,$$

oder die **Newton-Polynome**

$$1, (x-x_0),\ (x-x_0)(x-x_1),\ \ldots,\ \prod_{j=0}^{n-1}(x-x_j).$$

Bei der **Hermiteinterpolation** sind im Gegensatz zur Lagrangeinterpolation an den Interpolationsknoten auch mehr oder weniger hohe Ableitungswerte vorgegeben, an die das Interpolationspolynom von entsprechend höherem Grad angepasst werden soll. Ein Beispiel ist der Datensatz $(x_0, f_0, f_0')$, $(x_1; f_1)$, $(x_2, f_2, f_2', f_2'')$, an den man ein Polynom vom Grad 5 anpassen kann, siehe Kapitel 4.2.5.

### 4.2.1 Die Lagrange - Polynome

Die $n+1$ Funktionen $\varphi_i(x),\ i = 0, \ldots, n$, gegeben durch

$$\varphi_i(x) := \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} = \frac{\prod_{j=0,j\neq i}^{n}(x-x_j)}{\prod_{j=0,j\neq i}^{n}(x_i-x_j)} \tag{4.7}$$

sind offenbar Polynome vom Grad $n$ und es gilt

$$\varphi_i(x_k) \;=\; \delta_{ik}, \quad i,k \;=\; 0, 1, \ldots, n, \tag{4.8}$$

wobei $\delta_{ik}$ das Kroneckersymbol bezeichnet, d.h. $\delta_{ik} = 1$ ist für $i = k$ und 0 für $i \neq k$. Die Identität (4.8) folgt sofort aus (4.7):

$$\varphi_i(x_i) \;=\; \frac{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} \;=\; 1$$

da für $x = x_i$ der Zähler gleich dem Nenner wird. Für $i \neq k$ verschwindet ein Faktor des Zählers

$$(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{i-1})(x_k-x_{i+1})\cdots(x_k-x_n).$$

Mithilfe von (4.8) folgt sofort, dass

$$p(x) \;\;=\;\; \sum_{i=0}^{n} f(x_i)\varphi_i(x) \tag{4.9}$$

das gesuchte Interpolationspolynom $p(x)$ zu den Interpolationsdaten $(x_i, f(x_i))$, $i = 0, 1, 2, \ldots, n$ ist, da für einen beliebigen Knotenpunkt $x_k$

$$\begin{aligned} p(x_k) &= f(x_0)\underbrace{\varphi_0(x_k)}_{\delta_{0k}=0} + f(x_1)\underbrace{\varphi_1(x_k)}_{\delta_{1k}=0} + \cdots + f(x_k)\underbrace{\varphi_k(x_k)}_{\delta_{kk}=1} + \cdots + f(x_n)\underbrace{\varphi_n(x_k)}_{\delta_{nk}=0} \\ &= f(x_k)\cdot 1 = f(x_k) \end{aligned} \tag{4.10}$$

gilt. Die $n+1$ Polynome (4.7) bilden eine Basis im Raum der Polynome vom Maximalgrad $n$, sie spannen diesen Raum auf: Jedes Polynom

$$p(x) = c_0 + c_1 x + \cdots + c_n x^n$$

kann man an den Knotenstellen $x_0$, $x_1, \ldots, x_n$ betrachten, wo es die Werte $p(x_0)$, $p(x_1), \ldots$, $p(x_n)$ annimmt, d.h. man kann es als Lagrangesches Interpolationspolynom zu dem Datensatz

$$(x_0, p(x_0)), \quad (x_1, p(x_1)), \quad \ldots, \quad (x_n, p(x_n))$$

auffassen und es daher aufgrund von (4.9) als Linearkombination

$$p(x) = p(x_0)\varphi_0(x) + p(x_1)\varphi_1(x) + \cdots + p(x_n)\varphi_n(x) \tag{4.11}$$

der $n+1$ Lagrange-Polynome schreiben. Jedes Polynom von Grad $n$ ist also tatsächlich Linearkombination der $\varphi_i(x)$. Auch die Eindeutigkeit der Darstellung folgt unmittelbar: Wenn zwei Polynome an den Knotenstellen $x_0, \ldots, x_n$ dieselben Werte haben, wenn also die Gewichte in der Linearkombination (4.11) übereinstimmen, so müssen sie identisch sein, andernfalls hätte das nichtverschwindende Differenzpolynom vom Grad $n$ die $n+1$ Nullstellen $x_0, x_1, \ldots, x_n$ im Widerspruch zum Fundamentalsatz der Algebra.

**Beispiel 4.2.1.** *Gegeben sind die Interpolationsddaten* $(0,0)$, $\left(\frac{\pi}{4}, \frac{\sqrt{2}}{2}\right)$ *und* $(\frac{\pi}{2}, 1)$. *Die zugehörigen Lagrange-Polynome und das Interpolationspolynom sind*

$$\begin{aligned} \varphi_0(x) &= \frac{(x-\frac{\pi}{4})(x-\frac{\pi}{2})}{(0-\frac{\pi}{4})(0-\frac{\pi}{2})} \\ &= 1 - (1.909859317\ldots)x + (0.8105694692\ldots)x^2, \\ \varphi_1(x) &= \frac{(x-0)(x-\frac{\pi}{2})}{(\frac{\pi}{4}-0)(\frac{\pi}{4}-\frac{\pi}{2})} \\ &= (2.546479089\ldots)x - (1.621138938\ldots)x^2, \\ \varphi_2(x) &= \frac{(x-0)(x-\frac{\pi}{4})}{(\frac{\pi}{2}-0)(\frac{\pi}{2}-\frac{\pi}{4})} \\ &= -(0.6366197724\ldots)x + (0.8105694692\ldots)x^2 \\ \\ \Rightarrow p(x) &= 0\cdot\varphi_0(x) + \frac{\sqrt{2}}{2}\varphi_1(x) + 1\cdot\varphi_2(x) \\ &= (1.16408357\ldots)x - (0.3357488674\ldots)x^2. \end{aligned}$$

Hat man die Lagrange - Polynome $\varphi_i(x)$ zu einer Knotenmenge $x_0, x_1, \ldots, x_n$ einmal aufgestellt, kann nach (4.9) sofort das Interpolationspolynom angegeben werden. Möchte man zu einer festen Knotenmenge $x_0, \ldots, x_n$ verschiedene Datensätze

$$\begin{array}{llll} (x_0, f_0^0), & (x_1, f_1^0), & \ldots, & (x_n, f_n^0), \\ (x_0, f_0^1), & (x_1, f_1^1), & \ldots, & (x_n, f_n^1), \\ \vdots & & & \vdots \\ (x_0, f_0^r), & (x_1, f_1^r), & \ldots, & (x_n, f_n^r) \end{array}$$

interpolieren, ist das Arbeiten mit Lagrange - Polynomen optimal.

Ungünstig ist hingegen, wenn man nach der Interpolation eines Datensatzes $(x_0, f_0)$, ..., $(x_n, f_n)$ feststellt, dass die Genauigkeit des Interpolationspolynoms vom Grad $n$ für den vorliegenden Zweck nicht ausreicht. Erweitert man nun den Datensatz und arbeitet mit Polynomen höheren Grades, um die Genauigkeit zu erhöhen, muss man die Lagrange - Polynome bezüglich der neuen Knotenmenge komplett neu berechnen. Für solch eine Situation ist die Darstellung des Interpolationpolynom bezüglich der *Newton - Polynome* optimal.

### 4.2.2 Die Newton - Polynome

Die **Newton - Polynome** bezüglich der Knotenmenge $x_0, x_1, \ldots, x_n$ sind

$$1,\ (x-x_0),\ (x-x_0)(x-x_1),\ (x-x_0)(x-x_1)(x-x_2),\ \ldots,\ (x-x_0)(x-x_1)\cdots(x-x_{n-1}). \quad (4.12)$$

Jedes Polynom $p(x)$ vom Grad $n$ lässt sich mit den Koeffizienten $p_0, \ldots, p_n$ als Linearkombination dieser Basispolynome schreiben,

$$\begin{array}{rcllll} p(x) & = & p_0 \cdot 1 + p_1(x-x_0) + p_2(x-x_0)(x-x_1) & + & p_3(x-x_0)(x-x_1)(x-x_2) \\ & + & \cdots & + & p_n(x-x_0)(x-x_1)\cdots(x-x_{n-1}). \end{array} \quad (4.13)$$

Um das Interpolationspolynom zum Datensatz

$$(x_0, f_0), \quad (x_1, f_1), \quad \ldots, \quad (x_n, f_n)$$

aufzustellen, können die Gewichte $p_i$, $i = 0, 1, 2, \ldots n$ in der Linearkombination (4.13) nach und nach aus den Interpolationsbedingungen berechnet werden:

$x = x_0$ in (4.13)

$$f_0 = p(x_0) = p_0 \quad (4.14)$$

$x = x_1$ in (4.13)

$$f_1 = p(x_1) = p_0 + p_1(x_1 - x_0) = f_0 + p_1(x_1 - x_0)$$
$$\Rightarrow p_1 = \frac{f_1 - f_0}{x_1 - x_0} \quad (4.15)$$

$x = x_2$ in (4.13)

$$\begin{array}{rcl} f_2 = p(x_2) & = & p_0 + p_1(x_2 - x_0) + p_2(x_2 - x_0)(x_2 - x_1) \\ & = & f_0 + \dfrac{f_1 - f_0}{x_1 - x_0}(x_2 - x_0) + p_2(x_2 - x_0)(x_2 - x_1) \end{array}$$

Weitere Umformungen ergeben

$$\begin{aligned}
f_2 - f_1 + f_1 - f_0 - \frac{f_1 - f_0}{x_1 - x_0}(x_2 - x_0) &= p_2(x_2 - x_0)(x_2 - x_1) \\
\iff \frac{f_2 - f_1}{x_2 - x_1} + \frac{(f_1 - f_0)(x_1 - x_0)}{(x_1 - x_0)(x_2 - x_1)} - \frac{(f_1 - f_0)(x_2 - x_0)}{(x_1 - x_0)(x_2 - x_1)} &= p_2(x_2 - x_0) \\
\iff \frac{f_2 - f_1}{x_2 - x_1} - \frac{(f_1 - f_0)(x_0 - x_1) + (f_1 - f_0)(x_2 - x_0)}{(x_1 - x_0)(x_2 - x_1)} &= p_2(x_2 - x_0) \\
\iff \frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0} &= p_2(x_2 - x_0)
\end{aligned}$$

und schließlich

$$\frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0} = p_2 \tag{4.16}$$

$$\vdots$$

Es ist offensichtlich, dass bei dieser Vorgangsweise die Menge der Interpolationsdaten problemlos erweitert werden kann.
(4.14), (4.15) und (4.16) legt die Definition der sogenannten **dividierten Differenzen** nahe:

Nullte dividierte Differenz

$$f[x_0] := f(x_0) = f_0$$

Erste dividierte Differenz

$$f[x_0, x_1] := \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f_1 - f_0}{x_1 - x_0} = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$$

Zweite dividierte Differenz

$$f[x_0, x_1, x_2] := \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0} = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

In analoger Weise ergibt sich offensichtlich

Dritte dividierte Differenz

$$f[x_0, x_1, x_2, x_3] := \frac{\frac{\frac{f_3 - f_2}{x_3 - x_2} - \frac{f_2 - f_1}{x_2 - x_1}}{x_3 - x_1} - \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0}}{x_3 - x_0} = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}.$$

Dies lässt folgendes Bildungsgesetz für die $k$-te dividierte Differenz $f[x_0, x_1, \ldots, x_k]$ erkennen:
Im Nenner steht die Differenz $x_k - x_0$, im Zähler die Differenz der $(k-1)$-ten dividierten Differenz, bei der sämtliche Knoten um *1 nach rechts geshiftet* sind (d.h. die Knoten $x_1, \ldots, x_k$ betrachtet werden) und der $(k-1)$-ten dividierten Differenz bezüglich der *ungeshifteten* Knoten $x_0, x_1, \ldots, x_{k-1}$.
Es ergibt sich:

**Definition 4.2.2.** *Die* **dividierten Differenzen** *sind durch*

$$f[x_i] = f(x_i) = f_i, \quad i = 1, \ldots, n$$

*und*

$$f[x_0, x_1, \ldots, x_k] := \frac{f[x_1, \ldots, x_k] - f[x_0, \ldots, x_{k-1}]}{x_k - x_0} \tag{4.17}$$

*definiert.*

**Bemerkung.** Der Zusammenhang zwischen dividierten Differenzen und Ableitungen ist offenkundig. Wir nehmen die spezielle Knotenlage $x_1 = x_0 + h$, $x_2 = x_0 + 2h$, $x_3 = x_0 + 3h, \ldots$ an und betrachten kleine $h$-Werte. Offenbar erhalten wir (exakt gelten die folgenden Beziehungen nur für $h \to 0$):

$$\begin{aligned}
f[x_0, x_1] &= \frac{f_1 - f_0}{h} \approx f'(x_0) \\
f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{2h} \approx \frac{1}{2}\frac{f'(x_1) - f'(x_0)}{h} \approx \frac{1}{2}f''(x_0) \\
f[x_0, x_1, x_2, x_3] &= \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{3h} \approx \frac{1}{3}\frac{\frac{1}{2}f''(x_1) - \frac{1}{2}f''(x_0)}{h} \approx \frac{1}{2 \cdot 3}f'''(x_0)
\end{aligned}$$

u.s.w., also allgemein

$$f[x_0, x_1, \ldots, x_k] \approx \frac{1}{k!}f^{(k)}(x_0). \tag{4.18}$$

Aus (4.14), (4.15) und (4.16) folgt

$$\begin{array}{rcccl}
p(x) &=& f[x_0] + f[x_0, x_1](x - x_0) &+& f[x_0, x_1, x_2](x - x_0)(x - x_1) \\
&+& \cdots &+& f[x_0, x_1, \ldots, x_n](x - x_0) \cdots (x - x_{n-1}),
\end{array} \tag{4.19}$$

die Gewichte $p_0, \ldots, p_n$ des Interpolationspolynom $p(x)$ in der Darstellung (4.13) lassen sich also als dividierte Differenzen rekursiv berechnen.

### 4.2.3 Das Neville-Schema

Werden nur einzelne Werte des Interpolationspolynoms benötigt, so ist es möglich, das Polynom aufzustellen und an den entsprechenden Stellen $x = \bar{x}$ auszuwerten. Für die Auswertung von (4.13) gibt es einen *Hornerartigen*[1)] Algorithmus:

$$p(\bar{x}) = \Big(\cdots\Big(p_n(\bar{x} - x_{n-1}) + p_{n-1}\Big)\Big(\bar{x} - x_{n-2}\Big) + \cdots + p_1\Big)\Big(\bar{x} - x_0\Big) + p_0. \tag{4.20}$$

Im Allgemeinen ist es jedoch ökonomischer, nicht zunächst das Interpolationspolynom aufzustellen und anschließend für $x = \bar{x}$ auszuwerten, sondern mit dem sogenannten **Neville-Schema** direkt $p(\bar{x})$ zu berechnen. Nur wenn man ein Interpolationspolynom an *vielen* Stellen $\bar{x}_i$ auswerten möchte, ist der Weg über (4.19) und (4.20) effizienter.

Wir nehmen nun an, dass die Interpolationsknoten aufsteigend indiziert sind, d.h. $x_0 < x_1 < \cdots < x_n$. Grundlage für das Nevilleschema ist die Identität

$$p_{i,i+k}(x) = \frac{(x - x_i)p_{i+1,i+k}(x) - (x - x_{i+k})p_{i,i+k-1}(x)}{x_{i+k} - x_i}, \tag{4.21}$$

[1)] Hornerschema zur Auswertung von Polynomen: Statt ein Polynom z.B. von Grad 3 direkt $p(\bar{x}) = c_0 + c_1\bar{x} + c_2\bar{x}^2 + c_3\bar{x}^3$ auszuwerten, wertet man $p(\bar{x}) = ((c_3\bar{x} + c_2)\bar{x} + c_1)\bar{x} + c_0$ aus, was Rechenoperationen erspart. Konkret bei Polynomgrad 3: nur 3 statt 5 Multiplikationen.

wobei $p_{i,i+k}(x)$ das Interpolationspolynom (vom Grad $k$) zum Datensatz $(x_i, f_i)$, ..., $(x_{i+k}, f_{i+k})$ bezeichnet und $p_{i+1,i+k}(x)$ bzw. $p_{i,i+k-1}(x)$ die Interpolationspolynome (vom Grad $k-1$) zu den Datensätzen $(x_{i+1}, f_{i+1})$, ..., $(x_{i+k}, f_{i+k})$ bzw. $(x_i, f_i)$, ..., $(x_{i+k-1}, f_{i+k-1})$.

Gleichung (4.21) ist eine Beziehung zwischen den Interpolationspolynomen $p_{i,i+k}(x)$, $p_{i,i+k-1}(x)$ und $p_{i+1,i+k}(x)$. Wertet man die Polynome jedoch bei $x = \bar{x}$ aus, erhält man eine Beziehung zwischen Polynomwerten.

**Definition 4.2.3.** *Das* **Neville-Schema**

$$\begin{aligned} p_{i,i}(\bar{x}) &= f_i && i = 1, \ldots, n \\ p_{i,i+k}(\bar{x}) &= \frac{(\bar{x} - x_i)p_{i+1,i+k}(\bar{x}) - (\bar{x} - x_{i+k})p_{i,i+k-1}(\bar{x})}{x_{i+k} - x_i} && i = 1, \ldots, n, k = 1, \ldots, n-i \end{aligned} \tag{4.22}$$

*dient der rekursiven Bestimmung der Werte der Interpolationspolynome an der Stelle $x = \bar{x}$.*

Wir erläutern das Schema zunächst für $n = 2$, d.h. wir schreiben sämtliche Neville-Identitäten, die für den Aufbau des Interpolationspolynoms $p_{0,2}$ vom Grad 2 nötig sind, in Dreiecksgestalt an:

$$\begin{array}{c|c|c|c} x_0 & f_0 \equiv p_{0,0}(x) \searrow & & \\ & & \frac{(x-x_0)p_{1,1}(x)-(x-x_1)p_{0,0}(x)}{x_1-x_0} = p_{0,1}(x) \searrow & \\ x_1 & f_1 \equiv p_{1,1}(x) \nearrow \searrow & & \frac{(x-x_0)p_{1,2}(x)-(x-x_2)p_{0,1}(x)}{x_2-x_0} = p_{0,2}(x) \\ & & \frac{(x-x_1)p_{2,2}(x)-(x-x_2)p_{1,1}(x)}{x_2-x_1} = p_{1,2}(x) \nearrow & \\ x_2 & f_2 \equiv p_{2,2}(x) \nearrow & & \end{array}$$

Setzt man nun $x = \bar{x}$ in die Polynome ein und nutzt die Identitäten für die Werte $p_{0,0}(\bar{x})\,(= f_0)$, $p_{1,1}(\bar{x})\,(= f_1)$, $p_{2,2}(\bar{x})\,(= f_2)$, $p_{0,1}(\bar{x})$, $p_{1,2}(\bar{x})$, $p_{0,2}(\bar{x})$, so kann man mithilfe dieses Dreiecksschemas den gewünschten Wert $p_{0,2}(\bar{x})$ (auf einem Computer) berechnen, ohne das Polynom $p_{0,2}$ tatsächlich aufzustellen.

Die allgemeine Dreiecksform des Neville-Schemas wird in Abbildung 4.4 gezeigt.

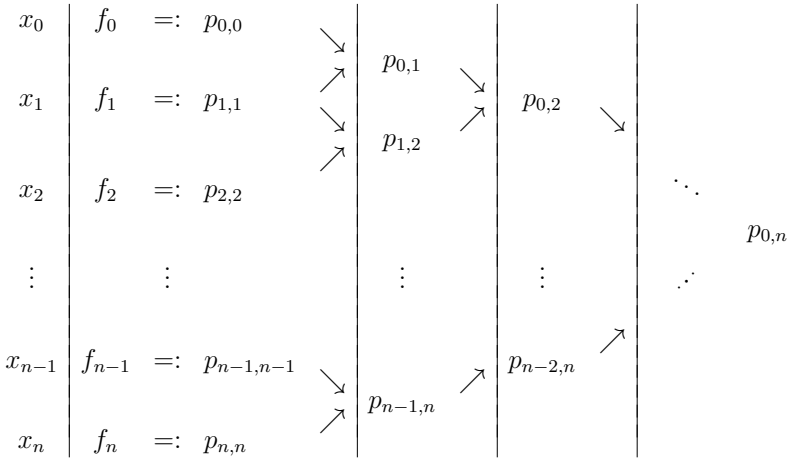
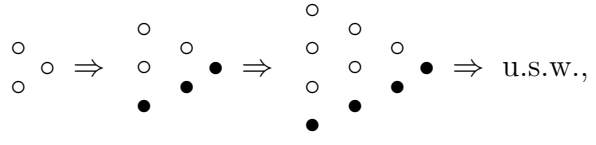
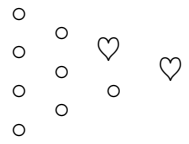



Abbildung 4.4: Die Dreiecksform des Neville-Schemas

Dabei wird das Dreieck in folgender Weise aufgebaut:



wobei • für neu hinzugekommene Werte $p_{i,i+k}$ steht. Dieses Aufbauschema ermöglicht in sehr ökonomischer Weise das Erreichen eines bestimmten, vorgegebenen Genauigkeitsniveaus: Ergibt der Vergleich von $p_{0,n}$ und $p_{0,n+1}$ bzw. ♡ in



eine nicht ausreichende Approximationsqualität (Genauigkeit) von $p_{0,n}$, kann durch Erweiterung des Datensatzes um $(x_{n+1}, f_{n+1})$ und Berechnung von $p_{0,n+1}$ die Genauigkeit gesteigert werden. Ist diese ausreichend, wird der Rechenvorgang abgebrochen.

### 4.2.4 Interpolationsfehler

Es ist klar, dass der Interpolationsfehler $p(x) - f(x)$ für $x \neq x_i$ beliebig groß werden kann, wenn man keine einschränkenden Voraussetzungen an $f$ stellt.
Der folgende Satz liefert für hinreichend glatte Funktionen $f$ eine Darstellung des bei der Polynominterpolation auftretenden Fehlers.

**Satz 4.2.4.** *Ist $f$ $(n+1)$-mal stetig differenzierbar, so gibt es zu jedem $x$ aus dem Definitionsbereich von $f$ eine Zahl $\vartheta$ aus dem kleinsten Intervall $I[x_0, \ldots, x_n; x]$, das alle $x_i$ sowie $x$ enthält, sodass für das Interpolationspolynom $p$ zum Datensatz $(x_0, f_0), \ldots, (x_n, f_n)$ die Beziehung*

$$p(x) - f(x) \;=\; -\frac{f^{(n+1)}(\vartheta)}{(n+1)!}\omega(x) \tag{4.23}$$

*gilt mit*

$$\omega(x) \;:=\; (x - x_0)(x - x_1)\cdots(x - x_n). \tag{4.24}$$

**Beweis.** Angenommen $x \neq x_i$, da für $x = x_i$ (4.23) trivial erfüllt. Dann ist $\omega(x) \neq 0$ ist und es existiert eine Größe $K_x$, sodass

$$f(x) - p(x) - K_x\omega(x) \;=\; 0.$$

$K_x = \frac{f(x)-P(x)}{\omega(x)}$ ist wohldefiniert, da der Nenner $\omega(x)$ nicht verschwindet. Mit dieser für festes $x$ festen Größe $K_x$ bilden wir jetzt die Funktion

$$F(\tau) \;:=\; f(\tau) - p(\tau) - K_x\omega(\tau), \quad \tau \in I[x_0, \ldots, x_n; x].$$

Diese besitzt in $I[x_0, \ldots, x_n; x]$ mindestens die $n + 2$ Nullstellen $x_0, \ldots, x_n; x$. Nach dem Satz von Rolle besitzt dann $\frac{dF(\tau)}{d\tau}$ in $I[x_0, \ldots, x_n; x]$ mindestens $n + 1$ Nullstellen, vergleiche Abb. 4.5.

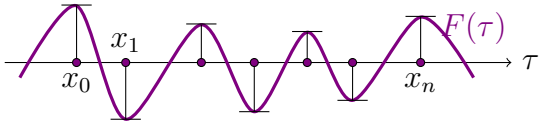



Abbildung 4.5: Nullstellen von $\frac{dF(\tau)}{d\tau}$

Daraus folgt wieder nach dem Satz von Rolle, dass $\frac{d^2F(\tau)}{d\tau^2}$ mindestens $n$ Nullstellen besitzt, ... und schließlich induktiv, dass $\frac{d^{n+1}F(\tau)}{d\tau^{n+1}}$ mindestens eine Nullstelle $\vartheta \in I[x_0, \ldots, x_n; x]$ besitzt. Da aber die $(n+1)$-te Ableitung des Interpolationspolynoms $p(x)$ vom Grad $n$ verschwindet, erhalten wir

$$\frac{d^{n+1}F(\vartheta)}{d\tau^{n+1}} \;=\; f^{(n+1)}(\vartheta) - K_x(n+1)! \;=\; 0. \tag{4.25}$$

Die dabei benützte Aussage $\omega^{(n+1)}(x) = (n+1)!$ folgt leicht durch Induktion nach der Anzahl der Linearfaktoren.

Induktionsanfang:

$$\begin{aligned} n=1: \quad \omega(x) &= (x-x_0)(x-x_1) \\ \omega'(x) &= (x-x_0)+(x-x_1) \\ \omega''(x) &= 1+1=2=2! \end{aligned}$$

Induktionsannahme:

$$\omega^{(n+1)}(x) = (n+1)! \text{ für } \omega(x) = (x-x_0)\cdots(x-x_n)$$

Induktionsschritt:

$$\begin{aligned} \omega(x) &= (x-x_0)(x-x_1)\cdots(x-x_{n+1}) \\ \Rightarrow \quad \omega'(x) &= (x-x_1)(x-x_2)\cdots(x-x_{n+1}) \\ &+ (x-x_0)(x-x_2)\cdots(x-x_{n+1}) \\ &+ \cdots \\ &+ (x-x_0)(x-x_1)\cdots(x-x_n) \quad \leftarrow \quad (n+2) \text{ Summanden} \end{aligned} \tag{4.26}$$

Die $n+2$ Summanden rechts in (4.27) entstehen dadurch, dass man auf alle möglichen Arten aus $(x-x_0)\cdots(x-x_{n+1})$ jeweils einen Faktor streicht. Differenziert man (4.27) nun $n+1$ mal, so erhält man aufgrund der Induktionsannahme

$$\begin{aligned} \omega^{(n+2)}(x) &= \underbrace{(n+1)! + (n+1)! + \cdots + (n+1)!}_{n+2 \text{ Summanden}} = \\ &= (n+2)(n+1)! \;=\; (n+2)!, \end{aligned}$$

sodass der Induktionsbeweis abgeschlossen ist.

Aus (4.25) folgt

$$K_x \;=\; \frac{f^{n+1}(\vartheta)}{(n+1)!}$$

und damit die zu (4.23) äquivalente Behauptung

$$f(x) - p(x) \;=\; K_x\omega(x) \;=\; \frac{f^{(n+1)}(\vartheta)}{(n+1)!}\omega(x).$$

□

Aus der Fehlerdarstellung (4.23), die man i.A. nicht direkt verwenden kann, da $\vartheta$ unbekannt ist, folgt sofort die Abschätzung

$$|p(x) - f(x)| \quad \leq \quad |\omega(x)| \frac{M_{n+1}}{(n+1)!} \tag{4.27}$$

wenn man über eine Schranke $M_{n+1}$ für $|f^{(n+1)}(x)|$, $x \in I[x_0, \ldots, x_n; x]$ verfügt.

**Beispiel 4.2.5.** *Gegeben ist die Funktion $f(x) = \sin x$ und die Interpolationsknoten $x_0 = 0$, $x_1 = \frac{\pi}{6}$, $x_2 = \frac{\pi}{4}$, $x_3 = \frac{\pi}{2}$ und $x = \frac{\pi}{5}$. Das interpolierende Polynom $p(x)$ hat Grad $n = 3$ und $M_{n+1} = 1$. Die Fehlerschranke ist also*

$$\boxed{\left| p\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right) \right| \leq \left| \frac{\pi}{5}\left(\frac{\pi}{5} - \frac{\pi}{6}\right)\left(\frac{\pi}{5} - \frac{\pi}{4}\right)\left(\frac{\pi}{5} - \frac{\pi}{2}\right) \right| \frac{1}{4!} = 0.002435227276\ldots}$$

*Der tatsächliche Fehler*

$$\boxed{p\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right) = -0.00025624\ldots}$$

*ist etwa um einen Faktor* 10 *kleiner, siehe Beispiel 4.1.1.*

Analog zu (4.23) gibt es auch Aussagen über die Approximation der Ableitungen $p^{(k)}(x) - f^{(k)}(x)$.

**Satz 4.2.6.** *Ist $f$ $(n+k+1)$-mal stetig differenzierbar, so gibt es zu jedem $x$ aus dem Definitionsbereich von $f$ Zahl $\vartheta_0, \ldots, \vartheta_k$ aus dem kleinsten Intervall $I[x_0, \ldots, x_n; x]$, das alle $x_i$ sowie $x$ enthält, sodass für das Interpolationspolynom $p$ zum Datensatz $(x_0, f_0), \ldots, (x_n, f_n)$ die Beziehung*

$$p^{(k)}(x) - f^{(k)}(x) \quad = \quad -\sum_{i=0}^{k} \frac{k!}{(n+k-i+1)!i!} \omega^{(i)}(x) f^{n+k-i+1}(\vartheta_i) \tag{4.28}$$

$$\vartheta_i \in I[x_0, \ldots, x_n; x]$$

*gilt.*

**Beweis.** Siehe Literatur. □

**Konvergenzbetrachtungen.** Die $(n+1)-$mal differenzierbare Funktion $f$ soll auf $[a, b]$ durch Polynome vom Grad $n$ interpoliert werden und zwar (siehe Abbildung 4.6)

1) durch ein Polynom zum Datensatz

$$(x_0, f_0), \ldots, (x_n, f_n)$$

mit den äquidistanten Knoten

$$x_0 = a, \quad x_1 = a + \frac{b-a}{n}, \quad x_2 = a + 2\frac{b-a}{n}, \quad \ldots, \quad x_n = b,$$

(Knotenabstand $h = \frac{b-a}{n}$)

2) durch zwei Polynome zu den Datensätzen

$$(x_0, f_0), \ldots, (x_n, f_n) \quad \text{und} \quad (x_n, f_n), \ldots, (x_{2n}, f_{2n})$$

mit den äquidistanten Knoten

$$x_0 = a, \quad x_1 = a + \frac{b-a}{2n}, \quad \ldots, \quad x_{2n} = b,$$

(Knotenabstand $h = \frac{b-a}{2n}$)

3) durch drei Polynome zu den Datensätzen

$$(x_0, f_0), \ldots, (x_n, f_n) \quad \text{und} \quad (x_n, f_n), \ldots, (x_{2n}, f_{2n}) \quad \text{und} \quad (x_{2n}, f_{2n}), \ldots, (x_{3n}, f_{3n})$$

mit den äquidistanten Knoten

$$x_0 = a, \quad x_1 = a + \frac{b-a}{3n}, \quad \ldots, \quad x_{3n} = b,$$

(Knotenabstand $h = \frac{b-a}{3n}$)

4) durch vier Polynome etc. (immer mehr Polynome vom Grad $n$ aneinanderstückeln).

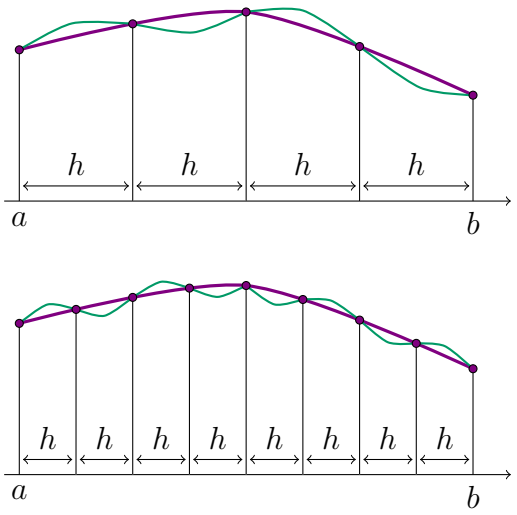



Abbildung 4.6: ein Polynom, zwei Polynome, ...

Wähle nun $x \in [a, b]$ beliebig sowie eine feste Zerlegung $x_0, \ldots, x_{mn}$ mit $m \in \mathbb{N}$. Es sei $k \in \{0, \ldots, m-1\}$ so, dass $x \in [x_{kn}, \ldots, x_{(k+1)n}]$. Dann gilt $|x - x_i| \leq \text{const} \cdot h$ für $i = kn, \ldots, (k+1)n$ und damit $|\omega(x)| \leq \text{const} \cdot h^{n+1}$.
Allgemein folgt daher für $x \in [a, b]$ aus (4.23) sofort

$$\begin{aligned} |p(x) - f(x)| &\leq \text{const} \cdot h^{n+1} \quad \text{oder} \\ p(x) - f(x) &= O(h^{n+1}). \end{aligned} \tag{4.29}$$

Für $h \to 0$ (Gitterverfeinerung, also immer mehr Interpolationspolynome vom festgehaltenen Grad $n$ auf immer kleineren Teilintervallen) ergibt sich somit Konvergenz der Interpolationsfunktion gegen $f$ mit der Ordnung $n+1$.

Bezüglich der Approximationsqualität von Ableitungen $p^{(k)}(x)$ der Interpolationsfunktion verglichen mit $f^{(k)}(x)$ ergibt sich aus (4.28) sofort

$$p^{(k)}(x) - f^{(k)}(x) \quad = \quad O(h^{n+1-k}) \quad k = 0, 1, \ldots, n, \tag{4.30}$$

denn für die höchste rechts in (4.28) auftrefende Ableitung von $\omega$ gilt [2)]

$$\omega^{(k)}(x) \quad = \quad O(h^{n+1-k}).$$

**Skalierungsdiskussion.** Aus (4.23), (4.27) und (4.29) folgt, dass der Interpolationsfehler durch genügend hohe $h$-Potenzen klein gemacht werden kann, jedoch nur für $h < 1$. Dies widerspricht auf den ersten Blick der Anschauung:

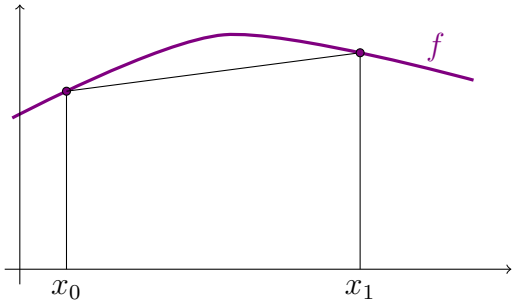



Abbildung 4.7: zeitabhängiger Vorgang interpoliert

Die Funktion $f$ beschreibe einen zeitabhängigen Vorgang und soll zwischen den Zeitpunkten $x_0$ und $x_1$ durch ein interpolierendes Geradenstück approximiert werden, siehe Abbildung 4.7. Ob der Stützstellenabstand $h = x_1 - x_0$ zahlenmässig klein ($\ll 1$) oder groß ($\gg 1$) ist, hängt von der gewählten Zeiteinheit (Jahre oder Sekunden) ab. Wie aus der Skizze ersichtlich ist, ist die Approximationsqualität $p(x) - f(x)$ von der Zeitskalierung aber unabhängig.
Dieser scheinbare Widerspruch löst sich jedoch sofort, wenn man die Fehlerformel (4.23) bzw. (4.27) betrachtet. Bei einer Umskalierung $x = c\xi$ ändert sich nicht nur $h$ und damit entsprechend auch $\omega$, sondern diese Änderung wird durch die inneren Ableitungen (Kettenregel beim Differenzieren) bei der Berechnung von $\frac{d^{n+1} f(c\cdot\xi)}{d\xi^{n+1}}$ kompensiert.

**Zusammenhang zum Taylorpolynom.** (vgl. Abb. 4.8)
Es seien eine Funktion $f(x)$ sowie ein zugehöriges lineares Interpolationspolynom $p(x)$ gegeben, welches die Stellen $(x_0, f_0)$ und $(x_1, f_1)$ interpoliert.

**Behauptung 4.2.7.** *Lässt man den zweiten Knoten $x_1$ gegen $x_0$ wandern, so konvergiert $p(x)$ offenbar gegen die Tangente $f(x_0) + (x - x_0)f'(x_0)$, also gegen das Taylorpolynom ersten Grades von $f$ mit Entwicklungsstelle $x_0$, siehe auch Abbildung 4.8.*

[2)] Es gilt $\omega(x) = (x - x_0)(x - x_1) \cdots (x - x_n) = O(h^{n+1})$, da jeder der $n+1$ Faktoren $(x - x_i)$ Größenordnung $h$ hat. Ebenso folgt $\omega'(x) = \sum_{j=1}^{n} \prod_{i=1, i\neq j}^{n} (x - x_i) = O(h^n)$, da jeder Summand $n$ Faktoren $(x - x_i)$ der Größenordnung $h$ hat.

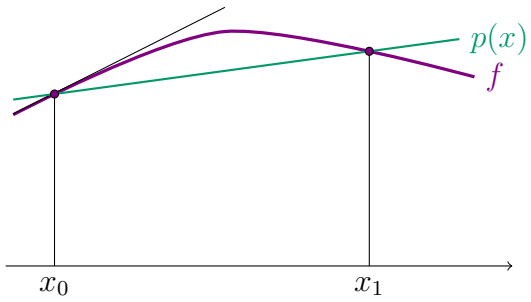



Abbildung 4.8: Sehne wird zur Tangente

**Beweis.** Betrachte $x_1 = x_0 + h$ mit $h > 0$ klein. Der erste Schritt im Neville-Schema hat damit die Form

$$p_{01}(\bar{x}) = \frac{(\bar{x} - x_0)p_{11}(\bar{x}) - (\bar{x} - x_0 - h)p_{00}(\bar{x})}{h}.$$

Da $p_{00}(\bar{x}) = f_0 = f(x_0)$ und $p_{11}(\bar{x}) = f_1 = f(x_0 + h)$ gilt, lässt sich dies umschreiben in

$$p_{01}(\bar{x}) = f(x_0 + h) + (\bar{x} - x_0)\frac{f(x_0) - f(x_0 + h)}{h}.$$

Der Grenzübergang $h \to 0$ ergibt

$$\lim_{h\to 0} p_{01}(\bar{x}) = f(x_0) + (\bar{x} - x_0)f'(x_0) = f_0 + (\bar{x} - x_0)f'_0.$$

Alternativ kann auch direkt in die Darstellungen 4.18 und 4.19 eingesetzt werden. □

**Behauptung 4.2.8.** *Allgemein strebt ein Interpolationspolynom $p(x)$ vom Grad $n$ zum Datensatz $(x_0, f_0), \ldots, (x_n, f_n)$ gegen das Taylorpolynom*

$$f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2!}f''(x_0)(x - x_0)^2 + \cdots + \frac{1}{n!}f^{(n)}(x - x_0)^n, \tag{4.31}$$

*wenn sämtliche Interpolationsknoten $x_i$ gegen $x_0$ konvergieren.*

**Beweis.** Einsetzen in (4.19) und (4.18).
Um alternativ die Konvergenz von drei Punkten zum gleichen Wert direkt über das Neville-Schema zu bestimmen, betrachtet man $x_1 = x_0+h$ und $x_2 = x_0+2h$. Nach zwei Schritten des Neville-Schemas ergibt sich

$$\begin{array}{rcl} p_{02}(\bar{x}) = & \frac{1}{2h} & [\ (\bar{x} - x_0)\left(\frac{\bar{x}-x_0-h}{h}[f(x_0 + 2h) - f(x_0 + h)] + f(x_0 + h)\right) \\ + & & (\bar{x} - x_0 - 2h)\left(\frac{\bar{x}-x_0}{h}[f(x_0 + h) - f(x_0)] + f(x_0)\right) \quad ]. \end{array}$$

Einfache algebraische Umformungen führen auf

$$\begin{array}{rcl} p_{02}(\bar{x}) = & \frac{(\bar{x}-x_0)^2}{2}\frac{[f(x_0+2h)+f(x_0)-2f(x_0+h)]}{h^2} \\ + & \frac{(\bar{x}-x_0)}{2}\left[\frac{f(x_0+h)-f(x_0+2h)}{h} + 3\frac{f(x_0+h)-f(x_0)}{h}\right] \\ + & f(x_0) \end{array}$$

und damit gilt im Grenzfall

$$\begin{array}{rcl} \lim_{h\to 0} p_{02}(\bar{x}) = & \frac{(\bar{x}-x_0)^2}{2}f''(x_0) + \frac{(\bar{x}-x_0)}{2}\left[-f'(x_0) + 3f'(x_0)\right] + f(x_0) \\ = & f_0 + (\bar{x} - x_0)f'_0 + \frac{(\bar{x}-x_0)^2}{2}f''_0. \end{array}$$

Analog wird die Rechnung für die Konvergenz von beliebig vielen Interpolationspunkten zum gleichen Wert durchgeführt. □ Parallel zur Konvergenz des Interpolationspolynoms strebt das Interpolationsrestglied (4.23) gegen das entsprechende Taylorrestglied:

$$\frac{f^{(n+1)}(\vartheta)}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n) \;\to\; \frac{f^{(n+1)}(\vartheta)}{(n+1)!}(x-x_0)^{n+1}.$$

### 4.2.5 Hermiteinterpolation

Im Gegensatz zur Lagrangeinterpolation sind an den Interpolationsknoten auch mehr oder weniger hohe Ableitungswerte vorgeschrieben, an die das Interpolationspolynom von entsprechend höherem Grad angepasst werden soll. An den Datensatz $(x_0, f_0, f_0')$, $(x_1; f_1)$, $(x_2, f_2, f_2', f_2'')$ kann man etwa ein Polynom vom Grad 5 anpassen. Um Werte des Interpolationspolynoms für $x = \bar{x}$ zu ermitteln, muss das Neville-Schema modifiziert werden. Abbildung 4.9 zeigt diese Modifikation. Die Werte ♡ sowie

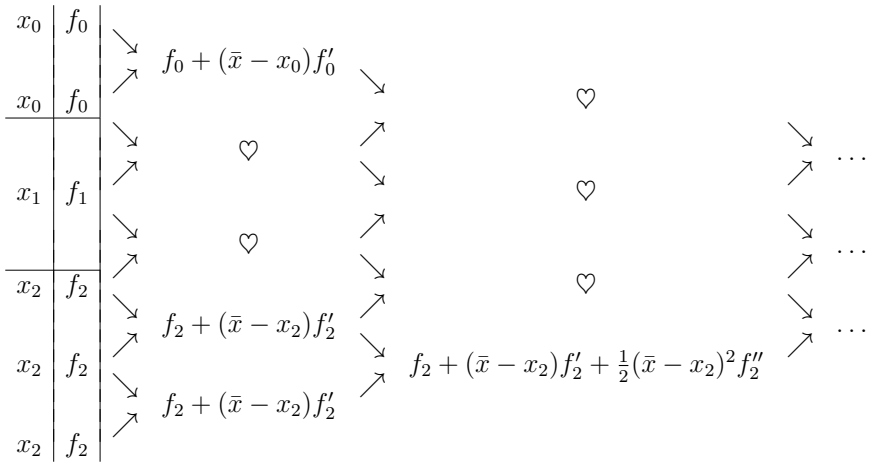



Abbildung 4.9: Die Dreiecksform des modifizierten Neville-Schemas

jene ab ... ergeben sich dabei nach dem üblichen Neville-Schema.
Die Bestimmung der Werte im modifizierten Neville-Schema kann folgendermaßen begründet werden:

- Jedes Element $p_{i,i+k}$ des Nevilleschemas ist das Interpolationspolynom zum Teildatensatz $(x_i, f_i)$, ..., $(x_{i+k}, f_{i+k})$
- Wenn die Knoten eines Interpolationspolynoms gegen einen Punkt konvergieren, dann konvergieren die Interpolationspolynome gegen die entsprechenden Taylorpolynome, siehe Behauptungen 4.2.7 und 4.2.8. Wenn also z.B. $x_1 = x_0 + h$ gegen $x_0$ konvergiert, so konvergiert $p_{0,1}$ (Wert des linearen Interpolationspolynoms zum Datensatz $(x_0, f_0)$, $(\tilde{x}_1, \tilde{f}_1)$ für $x = \bar{x}$) gegen den entsprechenden Wert $f_0 + (\bar{x} - x_0)f_0'$ des Taylorpolynoms vom Grad 1.

Auch bezüglich der Fehlerformel (4.23) ist sofort klar, wie sie aufgrund des oben beschriebenen Grenzprozesses zu modifizieren ist. Für den Datensatz

$$(x_0, f_0, f_0'), \quad (x_1, f_1), \quad (x_2, f_2, f_2', f_2'')$$

erhält man etwa

$$p(x) - f(x) \;=\; -\frac{f^{(6)}(\vartheta)}{6!}\omega(x)$$

mit $\omega(x) = (x-x_0)^2(x-x_1)(x-x_2)^3$. Für einen Datensatz der Form

$$(x_0, f_0, f_0'), \quad (x_1, f_1, f_1'), \quad \dots, \quad (x_n, f_n, f_n')$$

erhält man

$$p(x) - f(x) \;=\; -\frac{f^{(2n+2)}(\vartheta)}{(2n+2)!}\omega^2(x) \tag{4.32}$$

mit $\omega^2(x) = (x-x_0)^2(x-x_1)^2\cdots(x-x_n)^2$ und daher

$$p(x) - f(x) \;=\; O(h^{2n+2}) \tag{4.33}$$

für äquidistante Knoten $x_i - x_{i-1} = h$.

**Beispiel 4.2.9.** *Gegeben ist die Funktion $f(x) = \sin x$ mit*

$$\begin{aligned}(x_0, f_0, f_0') &= (0, \sin(0), \cos(0)) = (0,0,1),\\ (x_1, f_1, f_1') &= (\tfrac{\pi}{4}, \sin(\tfrac{\pi}{4}), \cos(\tfrac{\pi}{4})) = (\tfrac{\pi}{4}, \tfrac{\sqrt{2}}{2}, \tfrac{\sqrt{2}}{2}).\end{aligned}$$

*Das Neville-Schema bezüglich $x = \bar{x} = \frac{\pi}{5}$ ist:*

$$\begin{array}{c|c} 0 & 0 \\ 0 & 0 \\ \hline \frac{\pi}{4} & \frac{\sqrt{2}}{2} \\ \frac{\pi}{4} & \frac{\sqrt{2}}{2} \end{array} \quad \begin{array}{l} \searrow \\ \nearrow \\ \searrow \\ \nearrow \\ \searrow \\ \nearrow \end{array} \quad \begin{array}{l} 0.6283185307\ldots \\ \\ 0.5656854249\ldots \\ \\ 0.5960347077\ldots \end{array} \quad \begin{array}{l} \searrow \\ \nearrow \\ \searrow \\ \nearrow \end{array} \quad \begin{array}{l} 0.5782120461\ldots \\ \\ 0.5899648511\ldots \end{array} \quad \begin{array}{l} \searrow \\ \nearrow \end{array} \quad 0.5876142901\ldots = p(\bar{x})$$

*Der Fehler ist*

$$\boxed{p(\bar{x}) - \sin(\bar{x}) = -0.000170962131\ldots}$$

*Die Fehlerschranke (4.32)*

$$\boxed{\left|p\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right)\right| \le \frac{1}{4!}\left(\frac{\pi}{5}\right)^2\left(\frac{\pi}{5} - \frac{\pi}{4}\right)^2 = 0.0004058712126\ldots}$$

*ist also verglichen mit dem tatsächlichen Fehler etwa um den Faktor 4 zu pessimistisch.*

## 4.3 Bestapproximation

**Aufgabenstellung.** Gegeben ist eine Funktion $f$ aus einem Funktionenraum $\Phi$ (z.B. $\Phi = C^0[a,b] = \{$auf $[a,b]$ stetige Funktionen$\}$, $\Phi = C^2[0,\infty) = \{$auf $[0,\infty)$ zweimal stetig differenzierbare Funktionen$\}$, $\dots$).

Weiters betrachtet man einen **endlichdimensionalen** Teilraum $\Gamma \subset \Phi$, die Funktionen $g \in \Gamma$ können also durch endlich viele Parameter charakterisiert werden (Polynome, rationale Funktionen,..). Enthält der Raum $\Gamma$ Funktionen der Form

$$g(x) \;=\; \frac{c_0 + c_1 x + c_2 x^2}{d_0 + d_1 x + d_2 x^2 + d_3 x^3},$$

so ist er z.B. 7-dimensional, da jedes $g \in \Gamma$ durch die sieben Parameter $c_0, c_1, c_2, d_0, d_1, d_2, d_3$, festgelegt werden kann.
Sehr oft werden *lineare* endlichdimensionale Teilräume $\Gamma$ betrachtet, bei denen jedes $g \in \Gamma$ als endliche Linearkombination von Basisfunktionen geschrieben werden kann, Die Parameter, die $g \in \Gamma$ charakterisieren, sind dann die Gewichte dieser Linearkombination. Ist $\Gamma$ etwa der Raum der Polynome vom Maximalgrad 3, so hat es Dimension 4, da sich jedes $g \in \Gamma$ als $g(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3$ schreiben lässt, also als Linearkombination der Basisfunktionen $1, x, x^2, x^3$ mit den Gewichten $c_0, c_1, c_2, c_3$)

Die Räume $\Phi$ und $\Gamma$ werden mit einer Norm, etwa der **Maximumsnorm**

$$\|f\|_\infty \;=\; \max_{x\in[a,b]} |f(x)|, \tag{4.34}$$

der **$L_p$-Norm**

$$\|f\|_p \;=\; \left(\int_a^b |f(x)|^p dx\right)^{\frac{1}{p}} \quad p \geq 1 \tag{4.35}$$

oder einer allgemeinen **Skalarproduktnorm**

$$\|f\|_G \;=\; \left(\int_a^b f^2(x) G(x) dx\right)^{\frac{1}{2}} \tag{4.36}$$

mit *Gewichtsfunktion* $G(x)$ versehen.

**Definition 4.3.1.** *Die* **Bestapproximierende** $g^* \in \Gamma$ *zu* $f \in \Phi$ *ist jenes Element* $g^*$ *aus* $\Gamma$*, für das* $\|g^* - f\|$ *minimal ist, also*

$$\|g^* - f\| \;\leq\; \|g - f\| \quad \textit{für alle } g \in \Gamma. \tag{4.37}$$

Die Aufgabenstellung aus der Approximationstheorie hängt stark von der (4.37) zugrunde liegenden Norm ab.
Legt man Skalarproduktnormen (4.36) oder (4.35) mit $p = 2$ (Euklidische Norm ist eine Skalarproduktnorm mit $G(x) \equiv 1$) zugrunde legt, führt das in das weite Feld der **Fourierreihen**. Dabei kommt die Grundidee der Funktionalanalysis – einfache geometrische Konzepte aus dem Endlichdimensionalen auf unendlichdimensionale Funktionenräume zu übertragen – besonders deutlich zum Ausdruck:

**Endlichdimensionaler Raum,** z.B. $\mathbb{R}^3$:

1) Skalarprodukt:

$$\langle a, b\rangle \;=\; a_1 b_1 + a_2 b_2 + a_3 b_3 \;=\; \sum_{i=1}^{3} a_i b_i$$

2) Euklidische Norm:

$$\|a\|_2 \;=\; \langle a, a\rangle^{\frac{1}{2}} \;=\; \sqrt{\sum_{i=1}^{3} a_i^2}$$

3) Orthonormale Basis:

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad e_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

4) Orthonormalitätsrelationen bzgl. der Basiselemente:

$$\langle e_i, e_k \rangle \; = \; \delta_{ik} \quad i,k = 1,2,3$$

5) Darstellung von $a$:

$$a \;\; = \;\; a_1 e_1 + a_2 e_2 + a_3 e_3 \;\; = \;\; \langle a, e_1 \rangle e_1 + \langle a, e_2 \rangle e_2 + \langle a, e_3 \rangle e_3$$

**Funktionenraum,** z.B. $f \in C^0[-\pi, +\pi]$:

1) Skalarprodukt:

$$\langle f_1, f_2 \rangle \;\; = \;\; \int_{-\pi}^{\pi} f_1(x) f_2(x) dx$$

2) Euklidische Norm:

$$\|f\|_2 \; = \; \langle f, f \rangle^{\frac{1}{2}} \; = \; \sqrt{\int_{-\pi}^{\pi} \Big( f(x) \Big)^2 dx}$$

3) Orthornormale Basis:

$$\frac{1}{\sqrt{2\pi}}, \; \frac{1}{\sqrt{\pi}} \cos x, \; \frac{1}{\sqrt{\pi}} \sin x, \; \ldots \; \frac{1}{\sqrt{\pi}} \cos lx, \; \frac{1}{\sqrt{\pi}} \sin lx, \; \ldots$$

4) Orthogonalitätsrelationen bzgl. der Basiselemente:

$$\int_{-\pi}^{\pi} \cos lx \sin mx \, dx \; = \; 0, \quad \int_{-\pi}^{\pi} \cos lx \cos mx \, dx \; = \; \delta_{lm} \cdot \pi, \quad \int_{-\pi}^{\pi} \sin lx \sin mx \, dx \; = \; \delta_{lm} \cdot \pi.$$

(vgl. Lehrbuchliteratur)

5) Darstellung von $f$ als **Fourierreihe:**

$$\begin{aligned} f \;\; = \;\; & \langle f, \frac{1}{\sqrt{2\pi}} \rangle \frac{1}{\sqrt{2\pi}} + \langle f, \frac{1}{\sqrt{\pi}} \cos x \rangle \frac{1}{\sqrt{\pi}} \cos x + \langle f, \frac{1}{\sqrt{\pi}} \sin x \rangle \frac{1}{\sqrt{\pi}} \sin x + \cdots \\ & \cdots + \langle f, \frac{1}{\sqrt{\pi}} \cos lx \rangle \frac{1}{\sqrt{\pi}} \cos lx + \langle f, \frac{1}{\sqrt{\pi}} \sin lx \rangle \frac{1}{\sqrt{\pi}} \sin lx + \cdots \end{aligned}$$

Es gelten folgende Aussagen.

- Für jedes $f \in C^0[-\pi, \pi]$ konvergiert die Fourierreihe gegen $f$, d.h. die Vollständigkeitsrelation ist erfüllt.
- Die Vollständigkeitsrelation gilt sogar für (im Lebesgueschen Sinne) auf $[-\pi, \pi]$ quadratintegrierbare Funktionen, also unter schwächeren Voraussetzungen.

- Konvergenz der Fourierreihe gegen $f$ auch bzgl. der Maximumnorm (wenn $f$ den Dirichletschen Bedingungen genügt – vgl. Lehrbuchliteratur).
- Abgebrochene Fourierreihen (d.h. endliche Teilsummen der Fourierreihen) sind die Bestappoximierenden bzgl. der Euklid'schen Norm von $f$, der Teilraum $\Gamma \subset \Phi$ ist dabei der durch jene endlich vielen Basisfunktionen

$$\frac{1}{\sqrt{2\pi}},\ \frac{1}{\sqrt{\pi}}\cos x,\ \frac{1}{\sqrt{\pi}}\sin x,\ \ldots,\ \frac{1}{\sqrt{\pi}}\cos lx,\ \frac{1}{\sqrt{\pi}}\sin lx$$

  aufgespannte Teilraum, die der abgebrochenen Fourierreihe entsprechen.

Außer dem historisch ältesten Orthonormalsystem

$$\frac{1}{\sqrt{2\pi}},\ \frac{1}{\sqrt{\pi}}\cos x,\ \frac{1}{\sqrt{\pi}}\sin x,\ \ldots \quad \text{bezüglich } [a,b] = [-\pi, \pi]$$

gibt es noch zahlreiche weitere Orthonormalsysteme bezüglich anderer Intervalle (z.B. $[a,b] = [-1,+1]$ und $[a,b] = (-\infty, +\infty)$) und bezüglich verschiedener Skalarproduktdefinitionen $\int_a^b f_1(x) f_2(x) G(x) dx$ mit verschiedenen Gewichtsfunktionen $G(x)$. Beispiele sind Legendre-Polynome, Tschebyscheff-Polynome, Laguerre-Polynome, Hermite-Polynome, .... Die Theorie der Fourierreihen ist ein sehr umfassendes Teilgebiet der Approximationstheorie.

### 4.3.1 Tschebyscheff Approximation

Hier soll nur kurz die **Tschebyscheff–Approximation** gestreift werden, bei der in (4.37) die Maximumnorm (4.34) zugrunde gelegt wird. Zu einer vorgegebenen Funktion $f$ soll also jenes $g^*$ aus $\Gamma$ gefunden werden, für das

$$\|g^* - f\|_\infty \ \leq\ \|g - f\|_\infty \quad \forall g \in \Gamma \tag{4.38}$$

gilt oder anders gesagt jenes $g^*$ aus $\Gamma$, für das die Maximalabweichung von $f$,

$$\max_{x\in[a,b]} |g^*(x) - f(x)|$$

so klein wie möglich wird.

Ein besonders einfacher Fall ist jener einer konvexen oder konkaven Funktion.

**Satz 4.3.2.** *Sei $f$ auf $[a,b]$ konvex (oder konkav) und $\Gamma$ die Menge der linearen Polynome. Die Bestapproximierende von $f$ in $\Gamma$ erhält man durch folgendes Konstruktionsprinzip:*

*1) Bestimme die Gerade $g_1$ als Sehne durch $(a, f(a)), (b, f(b))$.*

*2) Bestimme die Gerade $g_2$ als jene Tangente, die zu $g_1$ parallel ist*

*3) Bestimme die Bestapproximierende als Mittelparallele von $g_1$ und $g_2$*

**Beweis.** Die Maximalabweichung

$$\max_{x\in[a,b]} |g^*(x) - f(x)|\,,$$

die für $g^*$ so klein wie möglich werden soll, wird bezüglich der Mittelparallelen an drei Stellen angenommen.

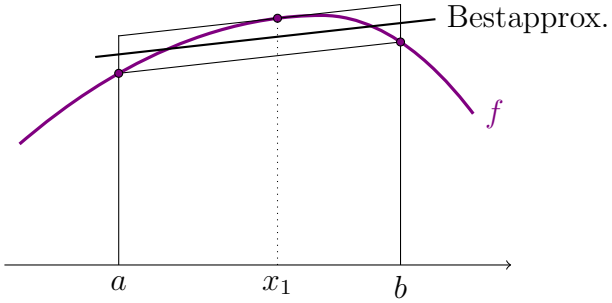



Abbildung 4.10: Alternantenpunkte

1. für $x = a$ (linkes Intervallende)
2. für $x = b$ (rechtes Intervallende)
3. für $x = x_1$ (jene Stelle, an der die zu $g_1$ parallele Tangente $g_2$ die Funktion $f$ berührt)

Betrachte dafür Abbildung 4.10. Nun folgt ein indirekter Beweis:
Stimmt die Bestapproximierende $g^*$ nicht mit der Mittelparallelen überein, so kann sie für $x = a$ nicht oberhalb der Mittelparallelen liegen, da sonst $\max_{x\in[a,b]} |g^*(x) - f(x)| \geq |g^*(a) - f(a)| >$ Maximalabweichung der Mittelparallelen, also $\|g^* - f\|_\infty > \|\text{Mittelparallele} - f\|_\infty$ gelten würde.
Aus dem selben Grund kann die Bestapproximierende auch für $x = b$ nicht oberhalb der Mittelparallelen liegen. Wenn also die Bestapproximierende von der Mittelparallelen verschieden sein soll, muss sie im Inneren von $[a, b]$ ganz unterhalb von der Mittelparallelen liegen, woraus für $x = x_1$ folgt:

$$\begin{aligned} \max_{x\in[a,b]} |g^*(x) - f(x)| &\geq |g^*(x_1) - f(x_1)| \\ &> |\text{Mittelparallele}(x_1) - f(x_1)| \\ &= \|\text{Mittelparallele} - f\|_\infty. \end{aligned}$$

Also gilt: $\|g^* - f\|_\infty > \|\text{Mittelparallele} - f\|_\infty$ und $g^*$ kann nicht die Bestapproximierende sein $\Rightarrow$ Mittelparallele ist tatsächlich die Bestapproximierende. □

Im eben besprochenen Fall wird also der Maximalabstand $\max_{x\in[a,b]} |g^*(x) - f(x)|$ in den drei Punkten $x_0 = a$, $x_1$ und $x_2 = b$ angenommen, wobei die Abweichung $g^*(x) - f(x)$ in den drei Punkten *alternierendes* Vorzeichen annimmt. Man bezeichnet diese Punkte $x_0, x_1, x_2$ deshalb als **Alternantenpunkte**.
Das bestappoximierende Polynom vom Grad 1 ist also offenbar dadurch charakterisierbar, dass die Maximalabweichung an drei Alternantenpunkten angenommen wird, an denen das Vorzeichen von $g(x)-f(x)$ alterniert. Für konkaves oder konvexes $f$ gibt es offenbar *genau* drei Alternantenpunkte, an denen die Maximalabweichung angenommen wird, wobei zwei davon die Randpunkte $x_0 = a, x_2 = b$ sind. Weist jedoch $f$ im Inneren von $[a, b]$ auch Wendepunkte auf, muss man diese Aussage etwas abschwächen: dann lässt sich nur mehr sagen, dass es *mindestens* drei Alternantenpunkte gibt, an denen die Maximalabweichung angenommen wird und $g(x) - f(x)$ alternierendes Vorzeichen annimmt. Auch die Randpunkte $a, b$ müssen nicht mehr Alternantenpunkte sein.
Diese Alternanteneigenschaft gilt auch für bestapproximierende Polynome höheren Grades (und auch noch für allgemeinere endlichdimensionale Teilräume $\Gamma \subset \Phi$ ). Für Polynome gilt:

**Satz 4.3.3.** *Zu beliebigem $f \in C[a, b]$ existiert ein eindeutig bestimmtes bestapproximierendes Polynom $P^*$ vom Maximalgrad $n$.*

Eine Charakterisierung der Bestapproximation liefert der *Alternantensatz*.

**Satz 4.3.4** (Alternantensatz)**.** *Ein beliebiges Polynom vom Grad n ist dann und nur dann bestapproximierendes Polynom im Tschebyscheff'schen Sinn, wenn (mindestens) $n+2$ Punkte*

$$a \leq x_0 < x_1 < \cdots < x_{n+1} \leq b$$

*existieren, für welche die Fehlerfunktion $g(x) - f(x)$ maximal wird und für zwei aufeinanderfolgende Punkte $x_{i-1}, x_i$ entgegengesetztes Vorzeichen annimmt.*

**Definition 4.3.5.** *Die Punkte*

$$x_0, x_1, \ldots, x_{n+1}$$

*aus Satz 4.3.4 heißen* **Alternantenpunkte**.

**Bemerkung.** Der Alternantensatz ist nicht konstruktiv. Um in konkreten Fällen die Bestapproximierende aufgrund der Charakterisierung des Alternantensatzes wirklich aufzubauen, müsste man einen extrem hohen algorithmischen Aufwand treiben (*Remez-Algorithmus*). Der Alternantensatz bietet aber die Grundlage dafür, mit viel einfacheren Mitteln eine sehr gute *Näherung* für die Bestapproximierende zu gewinnen.

**Heuristische Begründung dieser Vorgangsweise.** Da in den $n+2$ Alternantenpunkten der Fehler verschiedenes Vorzeichen aufweist, muss das bestapproximierende Polynom die Funktion $f$ jeweils zwischen zwei Alternantenpunkten schneiden – also an $n+1$ Punkten. Das bestapproximierende Polynom vom Grad $n$ ist daher auch ein Lagrange'sches Interpolationspolynom bezüglich der durch diese Schnittpunkte definierten Interpolationsdaten.
Ab sofort bezeichne $x_0^{(a)}, x_1^{(a)}, \ldots, x_{n+1}^{(a)}$ die $n+2$ Alternantenpunkte und $x_0, x_1, \ldots, x_n$ die dazwischenliegenden Knoten der Schnittpunkte von $f$ und $P^*$. Mit dieser Notation gilt

$$x_0^{(a)} < x_0 < x_1^{(a)} < x_1 < \cdots < x_n^{(a)} < x_n < x_{n+1}^{(a)}$$

und $P^*(x)$ ist Interpolationspolynom vom Grad $n$ zum Datensatz $(x_0, f_0), \ldots, (x_n, f_n)$.

Das Interpolationspolynom zu so einem Datensatz aufzustellen ist – im Gegensatz zur Aufgabenstellung *das bestapproximierende Polynom $P^*(x)$ im Tschebyscheffschen Sinn zu ermitteln* – eine sehr einfache Aufgabenstellung. Leider scheitert diese Vorgangsweise (die Bestimmung der Bestapproximierenden auf die Langrangeinterpolation zurückzuführen) daran, dass die Lage der Interpolationsknoten $x_0, \ldots, x_n$ unbekannt ist.

Die Lösung ist die näherungsweise Bestimmung dieser Interpolationsknoten. Es wird sich herausstellen, dass eine zweckmässige Wahl von $x_0, \ldots, x_n$ die (geeignet transformierten) Nullstellen der sogenannten **Tschebyscheffpolynome** ist.

Sei zunächst angenommen, jene Knoten $x_0, \ldots, x_n$, für die das Lagrangesche Interpolationspolynom zum Datensatz $(x_0, f_0), \ldots, (x_n, f_n)$ gleich dem bestapproximierenden Polynom $P^*(x)$ bezüglich $f$ ist, wären bekannt. Aufgrund von (4.23) schreibt sich der Interpolationsfehler dann als

$$P^*(x) - f(x) \;=\; -\frac{f^{(n+1)}(\vartheta)}{(n+1)!}\omega(x), \tag{4.39}$$

wobei $\omega(x)$ das Polynom $(x-x_0)\cdots(x-x_n)$ vom Grad $n+1$ ist.
Nimmt man nun an, dass $f$ so glatt ist, dass die $(n+1)$-te Ableitung von $f$ in $[a,b]$ nicht stark schwankt, sodass $f^{(n+1)}(\vartheta)$ (man beachte: $\vartheta$ hängt auch von $x$ ab – siehe den Beweis von (4.23)) annähernd konstant ist, kann man (4.39) auch schreiben als

$$P^*(x) - f(x) \approx \text{const} \cdot \omega(x).$$

Andererseits schwingt aufgrund des Alternantensatzes die Fehlerfunktion $P^*(x) - f(x)$ gleichmässig zwischen den Abweichungsmaxima. Man muss daher die Interpolationsknoten $x_0, \ldots, x_n$ so legen, dass $\omega(x) = (x-x_0)\cdots(x-x_n)$ den entsprechend gleichmässig schwingenden Verlauf aufweist.
Es wird sich herausstellen, dass das vorliegt, wenn $\omega(x)$ das Tschebyscheffpolynom vom Grad $n+1$ ist, wenn also die $x_0, \ldots, x_n$ die $n+1$ Nullstellen dieses Tschebyscheffpolynoms sind. Bis auf die Ungenauigkeit, dass $f^{(n+1)}(\vartheta(x))$ nicht konstant ist, hat dann der Interpolationsfehler genau den im Alternantensatz verlangten Verlauf, sodass das Interpolationspolynom zu den (entsprechend transformierten) Nullstellen der Tschebyscheffpolynome als Interpolationsknoten eine gute Approximation zur Bestapproximierenden sein sollte.

**Tschebyscheffpolynome.** Einen gleichmässig schwingenden Verlauf weist zum Beispiel die Funktion $\cos\varphi$ auf, die auf dem Intervall $[0,(n+1)\pi]$ die $(n+1)$ Nullstellen $\frac{\pi}{2}, \frac{3\pi}{2}, \frac{5\pi}{2}, \ldots, \frac{(2n+1)\pi}{2}$ hat.
Die Variable $\psi$, definiert durch $\psi = \frac{1}{n+1}\varphi$ (d.h. $\varphi = (n+1)\psi$), durchläuft das Intervall $[0,\pi]$, wenn $\varphi$ das Intervall $[0,(n+1)\pi]$ durchläuft.
Für $\psi \in [0,\pi]$ hat die Funktion $\cos((n+1)\psi)$ also $n+1$ Nullstellen (und $n+2$ Extrema) und damit den idealschwingenden Verlauf, ist aber leider kein Polynom. Aufgrund der Identität

$$\cos k\psi \;=\; (\cos\psi)^k - \binom{k}{2}(\cos\psi)^{k-2}(1-\cos^2\psi) + \binom{k}{4}\cos^{k-4}\psi(1-\cos^2\psi)^2 - + \cdots$$

ist jedoch $\cos(n+1)\psi$ ein Polynom vom Grad $n+1$ in $\cos\psi$. Die Transformation $t = \cos\psi$ beziehungsweise $\psi = \arccos t$ führt somit auf das Polynom

$$T_{n+1}(t) \;=\; \cos((n+1)\arccos t)$$

in $t$, das Grad $n+1$ hat. Offenbar durchläuft $\psi$ das Intervall $[0,\pi]$, wenn $t$ das Intervall $[-1,+1]$ durchläuft, d.h. sämtliche Nullstellen von $T_{n+1}(t)$ liegen in dem Intervall $[-1,+1]$.

**Definition 4.3.6.** *Das* **Tschebyscheffpolynom** *vom Grad $k$ ist definiert durch*

$$T_k(t) \;=\; \cos(k\arccos t).$$

Wegen $\cos(0\cdot\arccos t) = \cos 0 = 1$ gilt $T_0(t) \equiv 1$. Ebenso folgt wegen $\cos(1\cdot\arccos t) = t$, dass $T_1(t) = t$.

**Satz 4.3.7.** *Die Tschebyscheffpolynome erfüllen die Rekursion*

$$T_{k+1}(t) = 2t\cdot T_k(t) - T_{k-1}(t).$$

**Beweis.** Setze dafür in die Identität

$$\cos((k+1)\psi) = 2\cos\psi\cos k\psi - \cos((k-1)\psi)^{3)}$$

ein. □

Es ergibt sich

$$T_2(t) = 2t^2 - 1, \quad T_3(t) = 4t^3 - 3t, ...$$

Die Tschebyscheffpolynome zeigen also den ideal gleichmäßigen Schwingungsverlauf, d.h. die Nullstellen von $T_{n+1}$ sind die idealen Nullstellen für das gleichmäßig ausschwingende Polynom $\omega$ vom Grad $n+1$. Dabei ist aber zu beachten, dass $T_{n+1}(t)$ bezüglich des Intervalls $[-1,+1]$ betrachtet wurde (d.h. die Nullstellen symmetrisch um $t=0$ in diesem Intervall liegen), während $\omega$ sich auf das Intervall $a \le x \le b$ bezieht. Die Nullstellen [4] $t_0, t_1, \ldots, t_n \in [-1,+1]$ von $T_{n+1}(t)$ müssen daher noch transformiert werden gemäß

$$x_i = \frac{b-a}{2} \cdot t_i + \frac{a+b}{2}$$

.

**Beispiel 4.3.8.** *Es sei die Funktion $f(x) = \sin x$ für $x \in [0, \frac{\pi}{2}]$ gegeben. Das bestapproximierende Polynom vom Grad 1 wird gemäß Satz 4.3.2 konstruiert.*
*Die erste Gerade ist*

$$g_1 = \frac{2}{\pi} \cdot x = (0.6366197724\ldots) \cdot x.$$

*Da der Anstieg von $g_2$ ebenfalls $\frac{2}{\pi}$ ist und wegen $\frac{d \sin x}{dx} = \cos x$, ist $x_1^{(a)}$ festgelegt durch*

$$\cos x_1^{(a)} = \frac{2}{\pi}$$

*d.h. $x_1^{(a)} = \arccos(\frac{2}{\pi}) = 0.8806892354\ldots$. Das approximierende Polynom hat daher die Form*

$$\begin{aligned} P^*(x) &= \frac{2}{\pi}x + \frac{1}{2}\left(\sin x_1^{(a)} - \frac{2}{\pi}x_1^{(a)}\right) \\ &= (0.6366197724\ldots) \cdot x + (0.1052568312\ldots). \end{aligned}$$

*Die Maximalabweichung von $P^*(x)$ und $\sin x$ ist gegeben durch $\frac{1}{2}(\sin x_1^{(a)} - \frac{2}{\pi}x_1^{(a)}) = 0.1052568312\ldots$.*
*Im Gegensatz zum Interpolationspolynom vom Grad 1 zum Datensatz $(0, \sin 0), (\frac{\pi}{2}, \sin\frac{\pi}{2})$ (vgl. Beispiel 4.1.1 a)) für das sich*

$$g\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right) = -0.1877852523\ldots$$

*ergeben hat, erhalten wir jetzt*

$$P^*\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right) = -0.08252842109\ldots.$$

**Beispiel 4.3.9.** *Die Funktion $f(x) = \sin x$ soll für $x \in [0, \frac{\pi}{2}]$ mit einem Lagrangeinterpolationspolynom vom Grad 3 interpoliert werden, wobei die 4 transformierten Nullstellen des Tschebyscheffpolynoms vom Grad 4 als Interpolationsknoten dienen.*
*Die 4 Nullstellen von $T_4(t)$ in $(-1,+1)$ sind*

$$\begin{aligned} t_0 &= \cos(\tfrac{7\pi}{8}), & t_1 &= \cos(\tfrac{5\pi}{8}), \\ t_2 &= \cos(\tfrac{3\pi}{8}), & t_3 &= \cos(\tfrac{\pi}{8}), \end{aligned}$$

[3] Folgt für $\alpha = k\psi$ und $\beta = \psi$ durch Addition der bekannten Formeln $\cos(\alpha+\beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$ und $\cos(\alpha-\beta) = \cos\alpha\cos\beta + \sin\alpha\sin\beta$.

[4] Die Nullstellen von $T_k(t)$ sind offenbar $\cos(\frac{2k+1-2i}{2k}\pi)$, $i = 1(1)k$.

*die transformierten Nullstellen aus* $[0, \frac{\pi}{2}]$ *sind also*

$$\begin{aligned} x_0 &= \tfrac{\pi}{4} \cdot \cos(\tfrac{7\pi}{8}) + \tfrac{\pi}{4} &&= 0.05978487536\ldots \\ x_1 &= \tfrac{\pi}{4} \cdot \cos(\tfrac{5\pi}{8}) + \tfrac{\pi}{4} &&= 0.4848392985\ldots \\ x_2 &= \tfrac{\pi}{4} \cdot \cos(\tfrac{3\pi}{8}) + \tfrac{\pi}{4} &&= 1.085957028\ldots \\ x_3 &= \tfrac{\pi}{4} \cdot \cos(\tfrac{\pi}{8}) + \tfrac{\pi}{4} &&= 1.511011451\ldots \end{aligned}$$

*Mit dem Nevilleschema berechnen wir nun* $p\left(\frac{\pi}{5}\right)$*, wobei* $p$ *das Interpolationspolynom vom Grad* 3 *zu diesen Interpolationsknoten bezeichnet:*

$$\begin{array}{lllll} x_0 & 0.059749\ldots & & & \\ & & 0.6032204\ldots & & \\ x_1 & 0.466066\ldots & & 0.582599\ldots & \\ & & 0.5660007\ldots & & 0.586865\ldots \\ x_2 & 0.884749\ldots & & 0.593487\ldots & \\ & & 0.7625883\ldots & & \\ x_3 & 0.998213\ldots & & & \end{array}$$

*Für den Fehler folgt*

$$p\left(\frac{\pi}{5}\right) - \sin\left(\frac{\pi}{5}\right) = -0.000920249794\ldots.$$

# Kapitel 5

# Numerische Integration

## 5.1 Motivation

Das Ziel der **numerischen Integration** oder **numerischen Quadratur** ist das Auffinden von Näherungswerten $I$ für bestimmte Integrale der Form

$$\int_a^b f(t)\,\mathrm{dt}$$

von Funktionen $f : [a, b] \subset \mathbb{R} \to \mathbb{R}$ oder mehrdimensionale Integrale

$$\int_{a_1}^{b_1} \cdots \int_{a_n}^{b_n} \vec{f}(t_1, \ldots, t_n)\,\mathrm{dt}_n \ldots \mathrm{dt}_1$$

von Funktionen $\vec{f} : [a_1, b_1] \times \ldots \times [a_n, b_n] \to \mathbb{R}^n$ mit $n \in \mathbb{N}$.
Im $\mathbb{R}^n$ werden oft auch Integrale über ein Gebiet $\mathbb{G}$ der Form

$$\int_{\mathbb{G}} \vec{f}(t_1, \ldots, t_n)\,\mathrm{dt}_1 \ldots \mathrm{dt}_n$$

berechnet.
Der Näherungswert $I$ soll einer bestimmten Genauigkeitsbedingung genügen, wie beispielsweise für Integrale über $\mathbb{R}$

$$\left| I - \int_a^b f(t)\,\mathrm{dt} \right| < \varepsilon, \qquad \varepsilon > 0. \tag{5.1}$$

In diesem Kapitel wird nahezu ausschließlich der eindimensionale Fall besprochen.
Die Grundidee für numerische Quadraturverfahren ist, dass der Integrand $f(t)$ auf dem Intervall $[a, b]$ durch eine einfache Funktion $g(t)$ ersetzt wird und $\int_a^b g(t)\,\mathrm{dt}$ als Näherungswert für $\int_a^b f(t)\,\mathrm{dt}$ genommen wird.
Für die Wahl der Funktion $g(t)$ gibt es im Wesentlichen zwei zu beachtende Aspekte:

1. Das Integral $\int_a^b g(t)\,\mathrm{dt}$ soll leicht zu berechnen sein, $g$ ist daher meistens ein Polynom oder eine stückweise polynomiale Funktion.

2. Um die Funktion $g$ selbst leicht berechnen zu können, kommt sehr häufig das Interpolationsprinzip zur Anwendung.

Wir werden $g$ als stückweise polynomiale Funktion konstruieren, welche die Funktionswerte $f(t_i) = f_i,\ i = 0, 1, \ldots, n$, für eine Zerlegung $(t_i)_{i\in\mathbb{N}}$ des Intervalls $[a, b]$, interpoliert. Einzelne Verfahren arbeiten auch mit *einem* Interpolationspolynom auf dem ganzen Intervall. Um für so ein Polynom die Genauigkeitsanforderung zu erfüllen, muss jedoch ein ausreichend hoher Polynomgrad ausgewählt werden.

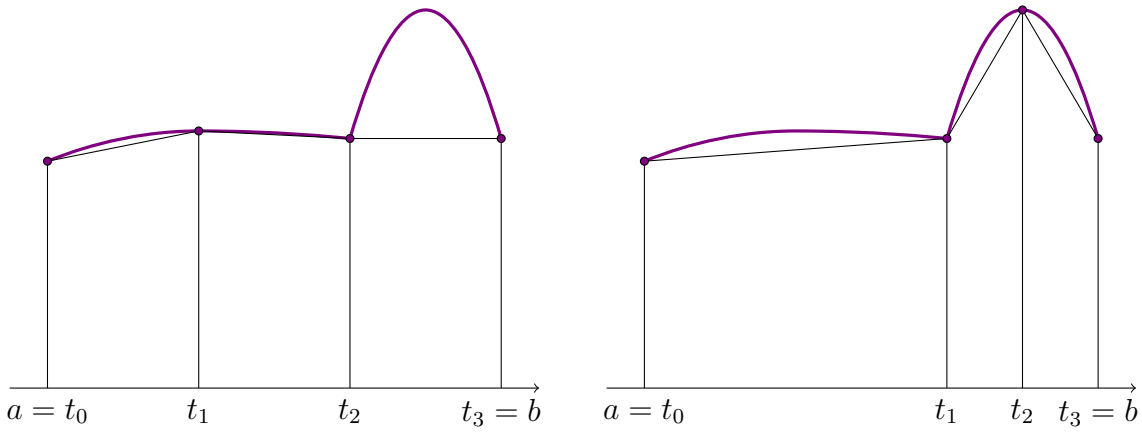



Abbildung 5.1: Äquidistantes und angepasstes Gitter

Um die Genauigkeitsanforderung (5.1) aus Effizienzgründen mit möglichst wenigen Funktionsauswertungen erfüllen zu können wird i.A. nicht an äquidistanten Knoten interpoliert. Gute Algorithmen generieren automatisch optimale Gitter, siehe Abbildung 5.1.

## 5.2 Newton - Cotes - Formeln

Zunächst werden wir die **Newton-Cotes-Formeln** betrachten, bei denen die Stützstellen äquidistant verteilt sind.

**Die Trapezregel.** Die Trapezregel ist die einfachste und damit ungenaueste der Newton-Cotes-Formeln. Es werden die beiden Endpunkte $a$ und $b$ des Intervalls als Stützstellen für die Interpolation durch ein lineares Polynom herangezogen. Mithilfe der Lagrange-Polynome $\varphi_0(t) = \frac{b-t}{b-a}$ und $\varphi_1(t) = \frac{t-a}{b-a}$ (siehe (4.7)) erhalten wir

$$g(t) = f(a)\frac{b-t}{b-a} + f(b)\frac{t-a}{b-a}$$

und damit

$$T_{b-a}(f) = \frac{1}{b-a}\left(f(a)\int_a^b (b-t)\,\mathrm{d}t + f(b)\int_a^b (t-a)\,\mathrm{d}t\right) = (b-a)\frac{f(a)+f(b)}{2}.$$

**Die Simpsonregel.** Bei der Simpsonregel werden drei äquidistante Stützpunkte herangezogen und die Funktion durch ein Polynom zweiten Grades interpoliert. Zur Vereinfachung betrachten wir das Standardintervall $[0, 1]$ anstelle von $[a, b]$. Die äquidistanten Stützstellen sind damit $t_0 = 0$, $t_1 = \frac{1}{2}$ und $t_2 = 1$.

Mit den Lagrange-Polynomen $\varphi_i$, $i = 0, 1, 2$ (siehe (4.7)) erhalten wir also für die Näherung des Integrals

$$\begin{aligned} S_h(f) &= \int_0^1 \left(f(0)\varphi_0(t) + f\left(\frac{1}{2}\right)\varphi_1(t) + f(1)\varphi_2(t)\right) \mathrm{dt} \\ &= f(0)\int_0^1 \frac{(t-t_1)(t-t_2)}{(t_0-t_1)(t_0-t_2)}\,\mathrm{dt} + f\left(\frac{1}{2}\right)\int_0^1 \frac{(t-t_0)(t-t_2)}{(t_1-t_0)(t_1-t_2)}\,\mathrm{dt} + f(1)\int_0^1 \frac{(t-t_0)(t-t_1)}{(t_2-t_0)(t_2-t_1)}\,\mathrm{dt} \end{aligned}$$

Einsetzen der Knotenpunkte $t_i$ für $i = 0, 1, 2$ liefert

$$\int_0^1 \frac{(t-\frac{1}{2})(t-1)}{(-\frac{1}{2})(-1)}\,\mathrm{dt} = \int_0^1 (2t^2 - 3t + 1)\,\mathrm{dt} = \frac{1}{6}, \qquad \int_0^1 \frac{t(t-1)}{\frac{1}{2}(-\frac{1}{2})}\,\mathrm{dt} = \frac{4}{6}, \qquad \int_0^1 2t(t-\frac{1}{2})\,\mathrm{dt} = \frac{1}{6}.$$

Das ergibt die *Simpsonregel* für das Standardintervall $[0, 1]$

$$\int_0^1 f(t)\mathrm{d}t \approx \frac{1}{6}f(0) + \frac{4}{6}f\left(\frac{1}{2}\right) + \frac{1}{6}f(1),$$

die mithilfe der Transformation $[0, 1] \ni t \mapsto a + t(b-a) \in [a, b]$ auf allgemeine Intervalle der Form $[a, b]$ übertragen werden kann. Es gilt also (wende die Substitutionsregel für das Integral sowie die Simpsonregel für das Standardintervall an)

$$\int_a^b f(t)\mathrm{d}t \approx (b-a)\left[\frac{1}{6}f(a) + \frac{4}{6}f\left(\frac{a+b}{2}\right) + \frac{1}{6}f(b)\right].$$

Eine *alternative Herleitung* ist die folgende: Wir betrachten (erneut auf dem Standardintervall $[0, 1]$) die Quadraturformel

$$\int_0^1 f(t)\,\mathrm{dt} \approx c_0 f(0) + c_1 f\left(\frac{1}{2}\right) + c_2 f(1) \tag{5.2}$$

mit noch nicht festgelegten Koeffizienten $c_0, c_1, c_2$. Diese sogenannten Gewichte $c_i$ werden dadurch festgelegt, dass die Quadraturformel für Polynome mit Grad kleiner gleich zwei exakt ist. Speziell ergibt sich

$$\begin{array}{lrcll} f(t) \equiv 1: & 1c_0 + 1c_1 + 1c_2 & = & \int_0^1 1\,\mathrm{dt} & = 1, \\ f(t) \equiv t: & \frac{1}{2}c_1 + 1c_2 & = & \int_0^1 t\,\mathrm{dt} & = \frac{1}{2}, \\ f(t) \equiv t^2: & \frac{1}{4}c_1 + 1c_2 & = & \int_0^1 t^2\,\mathrm{dt} & = \frac{1}{3} \end{array} \tag{5.3}$$

und als Lösung des linearen Gleichungssystem $c_0 = \frac{1}{6}$, $c_1 = \frac{4}{6}$ und $c_2 = \frac{1}{6}$.

**Allgemein.** Die vorgestellten Methoden lassen sich für Polynome $n$-ten Grades verallgemeinern. Es werden Daten $\big(a = t_0, f(t_0)\big)$, $\big(t_1, f(t_1)\big), \ldots, \big(t_n = b, f(t_n)\big)$, $n \in \mathbb{N}$ mit Knotenabstand $t_i - t_{i-1} = \frac{b-a}{n}$, interpoliert und das Interpolationspolynom integriert. Diese Quadraturformeln sind die *abgeschlossenen Newton-Cotes-Formeln*. Abgeschlossen werden sie genannt, da die beiden Endpunkte des Intervalls ebenfalls Interpolationspunkte sind. Anderenfalls spricht man von *offenen Newton-Cotes-Formeln*.

Für den Verfahrensfehler gilt folgender

**Satz 5.2.1.** *Sei $f(t)$ eine $(n+2)$-mal stetig differenzierbare Funktion, falls $n$ gerade, oder $(n+1)$-mal stetig differenzierbar, falls $n$ ungerade. Dann gilt*

$$\int_a^b g(t)\,dt - \int_a^b f(t)\,dt = \begin{cases} -\frac{M_{n,g}}{(n+2)!}h^{n+3}f^{(n+2)}(\xi) & n \text{ gerade} \\ -\frac{M_{n,u}}{(n+1)!}h^{n+2}f^{(n+1)}(\xi) & n \text{ ungerade} \end{cases} \tag{5.4}$$

*wobei $\xi \in [a,b]$ und $h = \frac{b-a}{n}$ gilt, sowie*

$$M_{n,g} = \int_0^n t^2(t-1)(t-2)\dots(t-n)\,dt$$

*und*

$$M_{n,u} = \int_0^n t(t-1)(t-2)\dots(t-n)\,dt.$$

**Beweis.** Siehe Isaacson-Keller, Analysis of Numerical Methods. □

Die Gewichte für spezielle Newton - Cotes - Formeln sind in Tabelle 5.1 für das Intervall $[a,b]$ mit der Länge $L = b-a$ und die äquidistanten Knoten $t_i = a+ih = a+i\frac{L}{n}$, $i = 0,1,2,\dots n$, zusammengestellt.

| $n$ | Gewichte $c_i$ | | | | | | | Fehler | Name |
|---|---|---|---|---|---|---|---|---|---|
| 1 | $L\frac{1}{2}$ | $L\frac{1}{2}$ | | | | | | $\frac{L^3}{12}f''(\xi)$ | Trapezregel |
| 2 | $L\frac{1}{6}$ | $L\frac{4}{6}$ | $L\frac{1}{6}$ | | | | | $\frac{(\frac{L}{2})^5}{90}f^{(iv)}(\xi)$ | Simpsonregel |
| 3 | $L\frac{1}{8}$ | $L\frac{3}{8}$ | $L\frac{3}{8}$ | $L\frac{1}{8}$ | | | | $\frac{3(\frac{L}{3})^5}{80}f^{(iv)}(\xi)$ | 3/8 - Regel oder Pulcherrima |
| 4 | $L\frac{7}{90}$ | $L\frac{32}{90}$ | $L\frac{12}{90}$ | $L\frac{32}{90}$ | $L\frac{7}{90}$ | | | $\frac{8(\frac{L}{4})^7}{945}f^{(vi)}(\xi)$ | Milneregel |
| 5 | $L\frac{19}{288}$ | $L\frac{75}{288}$ | $L\frac{50}{288}$ | $L\frac{50}{288}$ | $L\frac{75}{288}$ | $L\frac{19}{288}$ | | $\frac{275(\frac{L}{5})^7}{12096}f^{(vi)}(\xi)$ | 6 - Punkt - Regel |
| 6 | $L\frac{41}{840}$ | $L\frac{216}{840}$ | $L\frac{27}{840}$ | $L\frac{272}{840}$ | $L\frac{27}{840}$ | $L\frac{216}{840}$ | $L\frac{41}{840}$ | $\frac{9(\frac{L}{6})^9}{1400}f^{(viii)}(\xi)$ | Weddleregel |

Tabelle 5.1: Gewichte und Fehlerschranken einiger Newton - Cotes - Formeln

Für die Ordnungsaussage (5.4) in Satz 5.2.1 bzw. in der obigen Tabelle gibt es ein einfaches heuristisches Argument. Der Integrationsfehler

$$\int_a^b \big(g(t) - f(t)\big)\mathrm{d}t,$$

kann leicht mittels des Interpolationsfehlers, siehe (4.23)

$$-\int_a^b \frac{f^{(n+1)}(\theta(t))}{(n+1)!}\omega(t)\mathrm{d}t \tag{5.5}$$

dargestellt werden. Nimmt man nun an, dass $\theta(t)$ eine glatte Funktion ist mit $f^{(n+1)}(\theta(t)) = O(1)$, so ist der Integrand in der Größenordnung von $\omega(t)$, d.h. $O(L^{n+1})$. Da die Länge des Integrationsintervalls $b-a = O(L)$ ist, ist das Integral (5.5) von der Größenordnung $O(L^{n+2})$, was für ungerades $n$ in Übereinstimmung mit (5.4) bzw. mit der Tabelle oben ist. Dass für gerades $n$ noch eine Potenz

von $L$ mehr möglich ist, lässt sich heuristisch aus den Symmetrieeigenschaften von $\omega$ begründen, da in diesen Fällen $\omega$ schiefsymmetrisch bezüglich des Intervallmittelpunkt $\frac{a+b}{2}$ verläuft so, dass

$$\int_a^b \omega(t)\,\mathrm{d}t = 0$$

gilt. Wäre $f^{(n+1)}(t)$ konstant, also unabhängig von $t$, so würde das Integral (5.5) verschwinden und die numerische Integration wäre exakt. Aus Satz 5.2.1 folgt also insbesondere:

**Satz 5.2.2.** *Die Newton - Cotes - Formel vom Grad $n$ ist für Polynome vom Grad $n+1$ (wenn $n$ gerade) bzw. Polynome vom Grad $n$ (wenn $n$ ungerade) exakt.*

**Zusammengesetzte Newton-Cotes-Formeln.** Für große $n$ sind die abgeschlossenen Newton-Cotes-Formeln aus praktischer Sicht unbrauchbar, da viele Funktionsauswertungen notwendig sind. Dabei kommt es vermehrt zu Rundungsfehlern und Auslöschung. Ab $n = 7$ treten in etlichen Formeln sogar negative Gewichte auf.
Aus diesem Grund werden in der Praxis meist die *zusammengesetzten Newton-Cotes-Formeln* eingesetzt, bei denen das Integrationsintervall $[a, b]$ in Teilintervalle $[a_i, b_i]$, $i = 1, ..., N$ mit $a_1 = a$ und $b_N = b$ unterteilt wird, auf denen jeweils Newton-Cotes-Formeln verwendet werden. Theoretisch können die Teilintervalle beliebig (mit Länge $L_i$) und für die verschiedenen Intervalle verschiedene Schrittweiten $h_i$ gewählt werden, wir betrachten jedoch nur den Fall, in dem

$$L_i = \hat{L}\ \forall i, \quad h_i = h\ \forall i \quad \text{sowie} \quad \hat{L} = nh$$

gilt.

Für die *zusammengesetzte Trapezregel* (also $n = 1$) mit $N \in \mathbb{N}$ Teilintervallen (also $\hat{L} = \frac{L}{N}$), $h = \frac{\hat{L}}{1} = \hat{L}$ und $t_j = a + jh$ erhalten wir damit

$$T_N(f) = T_h(f) = h\left(\frac{1}{2}f(a) + \sum_{j=1}^{N-1} f(t_j) + \frac{1}{2}f(b)\right).$$

Mithilfe der Fehlerabschätzungen aus Tabelle 5.1 für einzelne Interpolationsintervalle ist es möglich, Fehlerabschätzungen für das gesamte Intervall $[a, b]$ zu finden:

$$\left|T_h(f) - \int_a^b f(t)\mathrm{d}t\right| = \sum_{r=1}^{N} \frac{h^3}{12} f''(\xi_r) \le N\frac{h^3}{12}\max_{t\in[a,b]}|f''(t)| = \frac{(b-a)h^2}{12}\max_{t\in[a,b]}|f''(t)|. \tag{5.6}$$

Analog gilt für die *zusammengesetzte Simpsonregel* (also $n = 2$) mit $N \in N$ Teilintervallen (also $\hat{L} = \frac{L}{N}$), $h = \frac{\hat{L}}{2}$ und $t_j = a + jh$

$$S_N(f) = S_h(f) = 2h\sum_{j=0}^{N-1}\left(\frac{1}{6}f(t_{2j}) + \frac{4}{6}f(t_{2j+1}) + \frac{1}{6}f(t_{2j+2})\right)$$

und damit die Fehlerabschätzung

$$\begin{aligned}\left|S_h(f) - \int_a^b f(t)\,\mathrm{dt}\right| &= \sum_{r=1}^{N} h^5\frac{1}{90}f^{(iv)}(\xi_r) \le Nh^5\frac{1}{90}\max_{t\in[a,b]}|f^{(iv)}(t)| \\ &= h^4\frac{(b-a)}{180}\max_{t\in[a,b]}|f^{(iv)}(t)|.\end{aligned} \tag{5.7}$$

Analog kann auch für Newton-Cotes-Formeln höherer Ordnung ($n \ge 3$) und $N \in \mathbb{N}$ Teilintervalle vorgegangen werden.

### 5.2.1 Daten- und Rundungsfehler

Der Effekt von verfälschten Funktionswerten $\widetilde{f}(t_i)$ lässt sich folgendermaßen abschätzen: Sei $i \in \mathbb{N}$ und $c_i > 0$, dann gilt

$$\begin{aligned}\left|\sum_i c_i \widetilde{f}(t_i) - \sum_i c_i f(t_i)\right| &\leq \sum_i |c_i| \left|\widetilde{f}(t_i) - f(t_i)\right| \\ &\leq \max_i \left|\widetilde{f}(t_i) - f(t_i)\right| (b-a).\end{aligned} \tag{5.8}$$

Auch in diesem allgemeinen Fall gilt $\sum_i |c_i| = \sum_i c_i = b - a$, da die Quadratur von $f(t) \equiv 1$ exakt ist. Allgemein gilt

$$\int_a^b f(t)\mathrm{d}t \approx c_0 f(t_0) + c_1 f(t_1) + \ldots + c_N f(t_N)$$

und, für $f(t) \equiv 1$,

$$\int_a^b 1\,\mathrm{d}t = c_0 + c_1 + \ldots + c_N.$$

Vergleicht man (5.8) mit

$$\left|\int_a^b \widetilde{f}(t)\mathrm{d}t - \int_a^b f(t)\mathrm{d}t\right| \leq (b-a) \max_{t\in[a,b]} |\widetilde{f}(t) - f(t)|, \tag{5.9}$$

so erkennt man, dass die Empfindlichkeit bezüglich Funktionsverfälschungen im Fall der exakten Integration und der numerischen Integration gleich ist.
Bezüglich der Rechenfehler ist i.A. nur der Additionsfehler wesentlich. Da der Gesamtadditionsfehler mit der Anzahl der Summanden wächst, nimmt der Gesamtrechenfehler mit kleinerwedendem $h$ zu. Er kann jedoch durch Verwendung partieller doppelter Genauigkeit i.A. hinreichend klein gehalten werden. bei monotonem Verlauf von $f$ empfiehlt es sich, die Summation von der Seite der absolut kleineren Werten her zu beginnen.

### 5.2.2 Effizienz der Newton - Cotes - Formeln

Konvergenzordnungen der Newton - Cotes - Formeln:

$$\begin{array}{rrcll} \text{Trapezregel} & T_h(f) - \int_a^b f(t)\mathrm{d}t & = & O(h^2) & (5.6) \\ \text{Simpsonregel} & S_h(f) - \int_a^b f(t)\mathrm{d}t & = & O(h^4) & (5.7) \\ \text{3/8 - Regel oder Pulcherrima} & P_h(f) - \int_a^b f(t)\mathrm{d}t & = & O(h^4) & \\ \text{Milneregel} & M_h(f) - \int_a^b f(t)\mathrm{d}t & = & O(h^6) & \end{array}$$

Diese Ordnungsaussagen gelten natürlich nur für hinreichend glatte, also hinreichend oft stetig differenzierbare Funktionen.
Anhand des folgenden Beispiels soll nun die Effizienz der unterschiedlichen Newton - Cotes - Formeln für verschiedene Genauigkeitsniveaus analysiert werden.

**Beispiel 5.2.3.** *Es soll $\int_{-1}^{1} e^{8t}\,dt$ auf ein Genauigkeitsniveau von*

**(a)** $10^{-1}$

**(b)** $10^{-10}$

*berechnet werden. Wie ist die Schrittweite zu wählen, sodass dieses Genauigkeitsniveau mit*

**(i)** *Trapezregel*

**(ii)** *Simpsonregel.*

*erreicht wird? Dabei soll nur der Verfahrensfehler berücksichtigt werden.*

**(a)** **(i)** *Die Fehlerschranke nach (5.6),*

$$\frac{b-a}{12}h^2 \max_{t\in[a,b]} |f''(t)| = h^2\frac{2}{12}64\ e^8 \approx 3{,}18\cdot 10^4 h^2,$$

*hat*

$$3{,}18\cdot 10^4 h^2 \approx 10^{-1}$$

*und damit* $h \approx 0{,}002$ *zur Folge.*

**(ii)** *Die Fehlerschranke nach (5.7),*

$$\frac{b-a}{180}h^4 \max_{t\in[a,b]} |f^{(iv)}| h^4\frac{2}{180}\,4096\ e^8 \approx 1{,}36\cdot 10^5 h^4,$$

*hat*

$$1{,}36\cdot 10^5 h^4 \approx 10^{-1}$$

*und damit* $h \approx 0{,}03$ *zur Folge.*

*Für das Verfahren höherer Ordnung wird also eine um den Faktor* 10 *kleinere Schrittweite benötigt.*

**(b)** **(i)** *Analog folgt*

$$3{,}18\cdot 10^4 h^2 \approx 10^{-10}$$

*und damit* $h \approx 6\cdot 10^{-8}$.

**(ii)** *Analog folgt*

$$1{,}36\cdot 10^5 \cdot h^4 \approx 10^{-10}$$

*und damit* $h \approx 2\cdot 10^{-4}$.

*In diesem Fall ist die benötigte Schrittweite für das Verfahren höherer Ordnung sogar um den Faktor* $10^{-4}$ *kleiner.*

***Fazit:*** *Das Beispiel zeigt, dass erst bei großer Genauigkeitsanforderung ein Verfahren höherer Ordnung effizienter ist (benötigte Schrittweite h ist viel kleiner). Auch wenn ein Beispiel kein Beweis ist, zumal nur die Fehlerschranken und nicht die tatsächlichen Quadraturformeln herangezogen wurden, ist diese Aussage tatsächlich richtig, nicht nur für Quadraturverfahren, sondern auch für gewisse Verfahren zur Lösung von Differentialgleichungen, die auf der Quadratur-Idee aufbauen.*

Vom Effizienzstandpunkt aus betrachtet, wünscht man sich für starke Genauigkeitsforderungen nicht nur Verfahren hoher Ordnung, sondern auch möglichst wenige Knoten, also Funktionsauswertungen, in einem Interpolationsintervall.
Quadraturverfahren zu konstruieren, bei denen mit möglichst wenigen Funktionsauswertungen im Interpolationsintervall möglichst hohe Ordnungen erzielt werden, ist das Ziel der *Gauß-Quadratur.*

## 5.3 Gauß - Quadratur

**Grundidee:** *Bei den Newton - Cotes - Formeln sind die Interpolationsknoten äquidistant im Interpolationsintervall. Die Ordnungen werden nur aufgrund der geeigneten Definition der Gewichte erreicht. Es liegt daher nahe, zusätzlich auch noch die Interpolationsknoten raffinierter zu wählen, um noch höhere Ordnung zu erreichen.*

Bei den Gauß - Formeln liegen die Knoten innerhalb der Interpolationsintervalle im Gegensatz zu den Newton - Cotes - Formeln nicht äquidistant. Es gibt daher keine feste Schrittweite $h$. Wenn man im Zusammenhang mit Gaußformeln von Ordnungsaussagen *Fehlerniveau* $= O(h^{\text{Potenz}})$ spricht, versteht man unter $h$ eine mittlere Schrittweite oder eventuell die Länge eines einzelnen Interpolationsintervalls. Ein weiterer Gegensatz zu den abgeschlossenen Newton - Cotes - Formeln besteht darin, dass die Randpunkte der Interpolationsintervalle keine Gitterpunkte der Quadratur sind.

Um die relative Lage der nicht äquidistanten Knoten der Gaußquadratur zu beschreiben, wird erneut ein Standardintervall betrachtet, in diesem Fall das Intervall $[-1, 1]$. Jedes beliebige Intervall $[a, b]$ kann durch die lineare Transformation $t \mapsto -1 + \frac{2}{b-a}(t - a)$ auf $[-1, 1]$ transformiert werden. Nur im Zusammenhang mit Ordnungsaussagen (*Fehlerniveau* $= O(h^{\text{Potenz}})$) wird nicht an das Standardintervall $[-1, 1]$ gedacht, da für festen Polynomgrad $n$ eine asymptotische Betrachtung $h \to 0$ inkompatibel mit einem festgehaltenen Intervall $[-1, 1]$ wären, siehe etwa Formel (5.16).

Man möchte also die *Gewichte* $c_i$ und die *Knotenstellen* $t_i$ in der Quadraturformel

$$I := \sum_{i=0}^{n} c_i f(t_i) \approx \int_{-1}^{+1} f(t)dt \tag{5.10}$$

so bestimmen, dass die Ordnung optimal wird.

Bei den Newton - Cotes - Formeln wurde mit Lagrangeinterpolation gearbeitet, die Gauß-Quadratur basiert dagegen auf der Hermiteinterpolation, siehe Abschnitt 4.2.5. Konkret sei $g(t)$ ein Polynom vom Grad $2n + 1$, das den Datensatz

$$\big(t_0, f(t_0), f'(t_0)\big),\ \big(t_1, f(t_1), f'(t_1)\big),\ \ldots,\ \big(t_n, f(t_n), f'(t_n)\big)$$

interpoliert.

Wir bezeichnen die entsprechenden Basispolynome mit $\psi_i(t)$ und $\rho_i(t)$, es sind also Polynome vom Grad $2n + 1$, für die

$$\begin{array}{llll} \psi_i(t_k) = \delta_{ik}, & \psi_i'(t_k) = 0, & & \\ \rho_i(t_k) = 0, & \rho_i'(t_k) = \delta_{ik}, & i, k = 0, 1, \ldots, n \end{array} \tag{5.11}$$

gilt, wobei $\delta_{ik}$ das Kroneckersymbol bezeichnet, also $\delta_{ik} = 0$ für $i \neq k$ und $\delta_{ik} = 1$ für $i = k$. Aus

$$g(t) \;=\; \sum_{i=0}^{n} f(t_i)\psi_i(t) + \sum_{i=0}^{n} f'(t_i)\rho_i(t) \tag{5.12}$$

folgt die Quadraturformel

$$I \;=\; \int_{-1}^{+1} g(t)\mathrm{d}t \;=\; \sum_{i=0}^{n} f(t_i) \int_{-1}^{+1} \psi_i(t)\mathrm{d}t + \sum_{i=0}^{n} f'(t_i) \int_{-1}^{+1} \rho_i(t)\mathrm{d}t \tag{5.13}$$

mit noch unbekannten Knoten $t_i$ für $i = 0, 1, 2, \ldots n$. Wenn es nun gelingt, spezielle Knoten $t_0, t_1, \ldots, t_n$ so zu finden, dass

$$\int_{-1}^{+1} \rho_i(t)\mathrm{d}t \;=\; 0, \quad i = 0, 1, \ldots, n \tag{5.14}$$

gilt, entsteht wieder eine Quadraturformel von der üblichen Gestalt

$$I = \sum_{i=0}^{n} f(t_i) \underbrace{\int_{-1}^{+1} \psi_i(t)\mathrm{d}t.}_{c_i} \tag{5.15}$$

Verglichen mit den Newton - Cotes - Formeln ist das asymptotische Ordnungsniveau natürlich höher: Für den Quadraturfehler gilt

$$\int\limits_{\substack{\text{Interpolations-}\\ \text{intervall}}} \Big(g(t) - f(t)\Big)\,\mathrm{d}t \;=\; \int\limits_{\substack{\text{Interpolations-}\\ \text{intervall}}} \frac{\omega^2(t) f^{(2n+2)}(\vartheta(t))}{(2n+2)!}\mathrm{d}t \tag{5.16}$$

mit

$$\omega^2(t) \;=\; (t-t_0)^2(t-t_1)^2\cdots(t-t_n)^2 \tag{5.17}$$

Bezeichnet man mit $h$ den mittleren Knotenabstand, etwa definiert durch $h = \frac{\mid \text{Interpolationsintervall} \mid}{n}$, so ist offenbar $\omega^2(t) = O(h^{2n+2})$. Da die Länge des Interpolationsintervalls $O(h)$ entspricht, hat das Integral (5.16) die Größenordnung $O(h^{2n+3})$. Diese Ordnungsaussage gilt natürlich nur für eine ausreichend glatte Funktion $f$ und bezieht sich auf ein Interpolationsintervall. Bei Aufsummation über alle Interpolationsintervalle verliert man eine $h$-Potenz, sodass die Gauß - Formeln insgesamt von der Ordnung $2n + 2$ sind.

Die entscheidende Frage ist nun, ob es wirklich gelingt, Knoten $t_0, \ldots, t_n \in [-1, 1]$ zu finden, sodass (5.14) gilt.

Die Basispolynome $\psi_i(t), \rho_i(t)$ der Hermiteinterpolation aus (5.11) lassen sich mithilfe der Basispolynome $\varphi_i(t)$ der Lagrangeinterpolation (4.7) als

$$\begin{aligned} \psi_i(t) &= (1 - 2\varphi_i'(t_i)(t - t_i))[\varphi_i(t)]^2 \\ \rho_i(t) &= (t - t_i)[\varphi_i(t)]^2 \end{aligned} \tag{5.18}$$

anschreiben. Man verifiziert sofort die Gültigkeit von (5.11). Die Basisfunktion $\rho_i(t)$ schreibt sich ausführlich als

$$\begin{aligned} \rho_i(t) &= (t - t_i)\left[\frac{(t-t_0)\cdots(t-t_{i-1})(t-t_{i+1})\cdots(t-t_n)}{(t_i-t_0)\cdots(t_i-t_{i-1})(t_i-t_{i+1})\cdots(t_i-t_n)}\right]^2 = \\ &= \prod_{k=0}^{n}(t-t_k)\frac{(t-t_0)\cdots(t-t_{i-1})(t-t_{i+1})\cdots(t-t_n)}{[(t_i-t_0)\cdots(t_i-t_{i-1})(t_i-t_{i+1})\cdots(t_i-t_n)]^2} = \\ &\quad\uparrow \qquad\qquad\qquad\uparrow \\ &= \omega(t)\cdot \text{ Polynom vom Grad } n \end{aligned}$$

Es ist also (5.14) sicher erfüllt, wenn

$$\int_{-1}^{+1} \omega(t)p(t)dt \;=\; 0 \tag{5.19}$$

für *jedes* Polynom $p(t)$ vom Grad $\leq n$ gilt.

---

Ein kurzer Einschub über die sogenannten **Legendre - Polynomen**, die durch

$$P_0(t) = 1, \quad P_1(t) = t, \quad P_2(t) = \frac{1}{2}(3t^2 - 1)$$

und durch die Rekursionsformel

$$P_{n+1}(t) = \frac{2n+1}{n+1}\, t\, P_n(t) - \frac{n}{n+1}\, P_{n-1}(t)$$

definiert sind. Die Legendre - Polynome bilden ein sogenanntes Orthogonalsystem, d.h. es gilt

$$\langle P_i, P_j \rangle \;=\; \int_{-1}^{+1} P_i(t) P_j(t) dt \;=\; \begin{cases} 0 & i \neq j, \\ \frac{2}{2i+1} & i = j. \end{cases}$$

Da die Legendre - Polynome auch linear unabhängig sind, bilden die ersten $n+1$ Legendre - Polynome eine orthogonale Basis im Raum der Polynome vom Maximalgrad $n$, d.h. jedes Polynom $p(t)$ vom Grad $n$ läßt sich in eindeutiger Weise als Linearkombination von $P_0(t), \ldots, P_n(t)$ schreiben. Wegen

$$\begin{aligned} \int_{-1}^{+1} P_{n+1}(t) p(t) \mathrm{d}t &= \int_{-1}^{+1} P_{n+1}(t) \Big( \lambda_0 P_0(t) + \cdots + \lambda_n P_n(t) \Big) \mathrm{d}t \\ &= \lambda_0 \int_{-1}^{+1} P_{n+1}(t) P_0(t) \mathrm{d}t + \lambda_1 \int_{-1}^{+1} P_{n+1}(t) P_1(t) \mathrm{d}t + \cdots + \lambda_n \int_{-1}^{+1} P_{n+1}(t) P_n(t) \mathrm{d}t \\ &= 0 \end{aligned} \tag{5.20}$$

mit $\lambda_i \in \mathbb{R}$, $i = 0, ..., n$ steht damit $P_{n+1}$ orthogonal auf alle Polynome vom Grad $\leq n$. Die Nullstellen der Legendre-Polynome liegen alle im Intervall $[-1, +1]$.

**Satz 5.3.1.** *Das Legendre-Polynom $P_n$ ($n \in \mathbb{N}$) vom Grad $n$ hat in $[-1, +1]$ genau $n$ paarweise verschiedene Nullstellen.*

**Beweis.** Es seien $t_j$, $j = 1, 2, \ldots, l$ die verschiedenen Nullstellen von $P_n(t)$ in $[-1, +1]$ mit den Vielfachheiten $\alpha_j$, also

$$P_n(t) = (t - t_1)^{\alpha_1} (t - t_2)^{\alpha_2} \cdots (t - t_l)^{\alpha_l} Q_n(t),$$

wobei $Q_n(t) \neq 0$ in $(-1, 1)$ ein Polynom vom Grad $< n$ ist. Definiere

$$\beta_j := \begin{cases} 0 \\ 1 \end{cases} \quad \text{falls} \quad \alpha_j \quad \begin{matrix} \text{gerade} \\ \text{ungerade} \end{matrix}$$

und

$$P_n^*(t) := (t - t_1)^{\beta_1} (t - t_2)^{\beta_2} \cdots (t - t_l)^{\beta_l}.$$

Der Grad von $P_n^*$ ist damit sicher $\leq l$.

Falls $l < n$, folgt aus (5.20) die Identität

$$0 \;=\; \int_{-1}^{+1} P_n(t) P_n^*(t) \mathrm{d}t \;=\; \int_{-1}^{+1} (t - t_1)^{\alpha_1 + \beta_1} \cdots (t - t_l)^{\alpha_l + \beta_l} Q_n(t) \mathrm{d}t.$$

Das ist ein Widerspruch, da der Integrand rechts in $[-1, 1]$ das Vorzeichen nicht wechselt, aber auch nicht identisch Null ist. Es muss also $l = n$ gelten und damit $\alpha_j = 1,\ j = 1, 2, \ldots n$. □

---

Wählt man nun die $n+1$ Knoten $t_0, \ldots, t_n$ aus $[-1, +1]$ als die Nullstellen des Legendre - Polynoms $P_{n+1}(t)$ vom Grad $n+1$, so stimmt $\omega(t)$ in (5.19) bis auf einen multiplikativen Faktor mit $P_{n+1}(t)$ überein. Da $p(t)$ in (5.19) ein beliebiges Polynom vom Maximalgrad $n$ ist, ist (5.19) wegen (5.20) tatsächlich erfüllt.

Die niedrigsten Gauß - Formeln sind

$$\begin{aligned}
\int_{-1}^{+1} f(t)\mathrm{d}t &\approx 2f(0) \\
\int_{-1}^{+1} f(t)\mathrm{d}t &\approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) \\
\int_{-1}^{+1} f(t)\mathrm{d}t &\approx \frac{5}{9} f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\sqrt{\frac{3}{5}}\right) \\
&\vdots
\end{aligned} \tag{5.21}$$

Bei anderen Integrationsintervallen als $[-1, 1]$ bzw. bei Unterteilung des Integrationsintervalls in Teilintervalle und Anwendung der Gauß - Formeln auf jedes Teilintervall müssen die Quadraturgewichte $c_i$ und die Knotenstellen $t_i$ natürlich entsprechend transformiert werden.

**Bemerkungen.** Es gibt noch weitere Quadraturformeln, die auf ähnlichen Ideen basieren wie die Gauß - Formeln:

1. **Radau-Formeln:** Es werden die Randpunkte $-1$ oder $1$ in die Menge der Knoten $t_i$ aufgenommen und alle weiteren Knoten und Gewichte dann so gewählt, dass eine Quadraturformel möglichst hoher Ordnung entsteht. Die Ordnung ist um 1 niedriger als die der entsprechenden Gauß-Formel mit gleicher Knotenanzahl.

2. **Lobatto-Formeln:** Es werden die Randpunkte $-1$ und $+1$ in die Menge der Knoten $t_i$ aufgenommen und alle weiteren Knoten und Gewichte bezüglich der erreichbaren Ordnung optimal gewählt. Die Ordnung ist um 2 niedriger als die der entsprechenden Gauß-Formel.

3. **Gauß-Kronrod-Formeln:** Bei adaptiven Schrittweitenstrategien ist die Idee, nach jeder Schrittweitenverfeinerung Fehlerschätzungen durchzuführen, um zu bestimmen, ob für die gewünschte Genauigkeit weitere Stützstellen notwendig sind, siehe Abschnitt 5.6. Dabei bestimmt man im Allgemeinen die Differenz aus beiden Quadraturwerten (*genauer - ungenauer*) und schließt daraus auf das Fehlerniveau.

   Dabei kann man entweder das ursprüngliche Teilintervall halbieren und dieselbe Quadraturformel wie zuvor auf das neue Teilintervall anwenden oder einfach eine genauere Quadraturformel wählen.

   In beiden Fällen erweisen sich die irrationalen Gaußknoten als Nachteil:

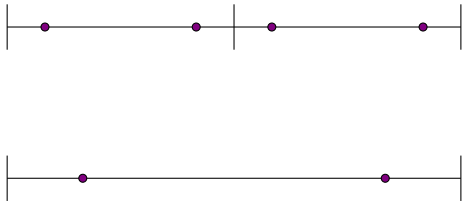


Abbildung 5.2: Intervallhalbierung

a) Betrachte etwa die Gaußknoten $-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}$. Bei Halbierung des Intervalls sind Funktionsauswertungen an völlig neuen Stellen notwendig, siehe Abbildung 5.2. Die alten Funktionswerte können für die Berechnung der genaueren Näherung *nicht* verwendet werden.

b) Auch bei Verwendung von Formelpaaren gilt wieder, dass die Gaußknoten der genaueren Gaußformel völlig verschieden sind von den Knoten der ungenaueren Gaußformel:

$$\begin{aligned} n=2: &\quad f\left(-\frac{1}{\sqrt{3}}\right)+f\left(\frac{1}{\sqrt{3}}\right), \\ n=3: &\quad \frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right)-\frac{8}{9}f(0)+\frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right). \end{aligned}$$

$\Longrightarrow$ **Gauß-Kronrod-Formelpaar:** Die $(n+1)$-punktige Gaußformel (ungenauere Formel des Paares) wird durch weitere Knoten und Gewichte zu einer weiteren Quadraturformel (genauere Formel des Paares) ergänzt, d.h. die Funktionsauswertungen in der Gaußformel werden auch in der genaueren Formel verwendet.

## 5.4 Asymptotische Fehlerentwicklungen

### 5.4.1 Euler - Maclaurinsche Summenformel

**Satz 5.4.1.** *Für $f \in C^{2m+2}[a,b]$ besitzt die Trapezsumme die Entwicklung*

$$T_h(f) = \tau_0 + \tau_1 h^2 + \tau_2 h^4 + \cdots + \tau_m h^{2m} + \alpha_{m+1}(h)h^{2m+2} \tag{5.22}$$

*mit $\tau_0 := \int_a^b f(t)dt$. Dabei sind die $\tau_i(f)$ von $h$ unabhängig und $\alpha_{m+1}(h)$ ist eine beschränkte Funktion von $h$, d.h. $|\alpha_{m+1}(h)| \le M$ für alle $h = \hat{L} = \frac{L}{N} = \frac{b-a}{N}$, $N \in \mathbb{N}$.*

**Bemerkungen.**

1. Die Koeffizienten $\tau_j$ können explizit mit Hilfe der sogenannten Bernoulli-Polynome $B_k(t)$ als

$$\tau_j = \frac{B_{2j}(0)}{(2j)!}\left(f^{(2j-1)}(b) - f^{(2j-1)}(a)\right)$$

angegeben werden.

Die Bernoulli-Polynome $B_k(t)$, $k = 1, 2, \ldots$ sind rekursiv definiert durch

$$\begin{aligned} B_0(t) &\equiv 1, \\ B_k'(t) &= kB_{k-1}(t), \qquad k \ge 1, \\ \textstyle\int_0^1 B_k(t)\mathrm{d}t &= 0, \qquad k \ge 1. \end{aligned} \tag{5.23}$$

Wegen (5.23) hat jedes $B_k(t)$ eine Darstellung der Form

$$B_k(t) \;=\; A_k + k\int_0^t B_{k-1}(\tau)\mathrm{d}\tau, \quad k \geq 1,$$

wobei die Konstante $A_k = B_k(0)$ so zu bestimmen ist, dass $\int_0^1 B_k(t)dt = 0$ gilt. Es folgt sofort, dass $B_k(t)$ ein Polynom $k$-ten Grades ist. Insbesondere findet man

$$\begin{aligned} B_1(t) &= t - \tfrac{1}{2}, & B_2(t) &= t^2 - t + \tfrac{1}{6}, \\ B_3(t) &= t^3 - \tfrac{3}{2}t^2 + \tfrac{1}{2}t, & B_4(t) &= t^4 - 2t^3 + t^2 - \tfrac{1}{30}. \end{aligned}$$

Es gelten folgende wichtige Eigenschaften:

$$\begin{array}{ll} \text{(i)} & B_k(0) = B_k(1) \quad \text{für } k \geq 2 \\ \text{(ii)} & \text{Die } B_k(t) \text{ sind für gerades } k \text{ gerade Funktionen} \\ & \text{und für ungerades } k \text{ ungerade Funktionen} \\ & \text{bezüglich der Stelle } t = \frac{1}{2} \\ \text{(iii)} & B_{2k+1}(0) = B_{2k+1}(1) = 0 \quad \text{für alle } k \geq 1 \end{array} \tag{5.24}$$

2. Setzt man $g(t) := f(a+th)$ für $0 \leq t \leq n$ (das entspricht einer Intervalltransformation $[a,b] \to [0,n]$), dann ist (5.22) äquivalent zur sogenannten **Euler - MacLaurinschen Summenformel**

$$\begin{aligned} &\frac{g(0)}{2} + g(1) + \cdots + g(n-1) + \frac{g(n)}{2} - \int_0^n g(\tau)\mathrm{d}\tau \\ =\;& \sum_{k=1}^{m} \frac{B_{2k}(0)}{(2k)!}[g^{(2k-1)}(n) - g^{(2k-1)}(0)] + R_{m+1} \end{aligned} \tag{5.25}$$

mit

$$R_{m+1} \;=\; -\frac{1}{(2m+2)!}\int_0^n [S_{2m+2}(\tau) - S_{2m+2}(0)]\, g^{(2m+2)}(\tau)\mathrm{d}\tau,$$

wobei $S_k(t) := B_k(t-j)$ für $j \leq t < j+1$ und $j = 0, 1, 2, \ldots$

3. Im Gegensatz zu (1.16), wo der Verfahrensfehler der Trapezregel nur abgeschätzt wird, beschreibt die Gleichung (5.22) die Struktur des Fehlers. Man spricht von einer **asymptotischen Entwicklung** des Verfahrensfehlers der Trapezregel. Ein viel einfacheres Beispiel einer asymptotischen Entwicklung hatten wir schon in (1.12), siehe S. 12 kennengelernt.

Es sei erwähnt, dass auch bezüglich anderer Quadraturformeln, etwa bezüglich aller Newton - Cotes - Formeln und aller Gauß - Formeln solche asymptotischen Fehlerentwicklungen existieren.

### 5.4.2 Hauptanwendung asymptotischer Fehlerentwicklungen

Die Hauptanwendung asymptotischer Fehlerentwicklungen sind **Extrapolationsalgorithmen**. Im Folgenden wird dies am Beispiel der Trapezregel besprochen.
Betrachte $T_h(f) = T_f(h)$ als Funktion von $h$. Dann gilt

$$\int_a^b f(t)\,\mathrm{dt} \;=\; \lim_{h\to 0} T_f(h) \;=\; T_f(0).$$

$T_f(0)$ ist dabei nicht als Trapezsumme definiert, sonder nur über den Grenzwert $\lim_{h\to 0} T_f(h)$. Aus $\lim_{h\to 0} T_f(h) = T_f(0)$ könnte man schließen, dass es sinnvoll ist, Trapezsummen mit sehr kleinen Schrittweiten (=Teilintervalllängen) als Approximation des gesuchten Integrals zu nehmen. Aus folgendem (bekannten) Grund scheitert dies aber: Für $h \to 0$ steigt der Rechenaufwand und damit auch der Rundungsfehler wegen der immer mehr werdenden Funktionsauswertungen gegen Unendlich.

Es liegt daher folgender Gedanke nahe: Man ersetzt die Funktion $T_f(h)$ durch eine Ersatzfunktion $\bar{T}_f(h)$, die für $h = 0$ leicht auszuwerten ist und nimmt $\bar{T}_f(0)$ als Näherungswert für das gesuchte Integral. Dies führt zur folgenden **Trapezsummenextrapolation** oder **Rombergintegration** (siehe Abb. 5.3).

Man berechnet für einige Schrittweiten, in Abbildung 5.3 für $h_0 > h_1 > h_2 > 0$, die Trapezsummen

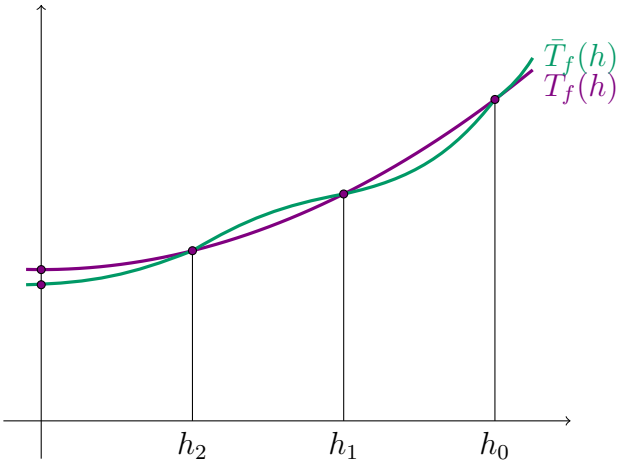



Abbildung 5.3: Trapezsummenextrapolation

$T_f(h_i)$ und interpoliert die Daten $(h_i, T_f(h_i))$ mit einem Interpolationspolynom $\bar{T}_f(h)$. Die Auswertung von $\bar{T}_f(h)$ erfolgt mit dem Neville-Schema. Da der Wert 0 außerhalb des die Knoten umfassenden Intervalls liegt, spricht man hier von *Extrapolation.*

**Bemerkungen.**

1. Genaugenommen ist $T_f(h)$ keine stetige Funktion, wie dies in Abbildung 5.3 dargestellt ist, sondern nur für die Argumentwerte $h = \frac{b-a}{N}$, $N \in \mathbb{N}$ definiert, die sich bei 0 häufen. Dies ist aber offensichtlich irrelevant für die oben beschriebene Idee, die der Trapezsummenextrapolation zu Grunde liegt.

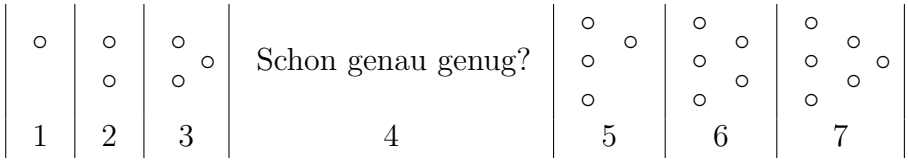
2. Wie aus (5.22) ersichtlich, ist $T_f(h)$ bis auf das Restglied von der Ordnung $O(h^{2m+2})$ ein Polynom in $h^2$. Dem entspricht auch die waagrechte Tangente von $T_f(h)$ für $h = 0$ in Abbildung 5.3. Es liegt daher nahe, $\bar{T}_f$ nicht als Polynom in $h$, sondern gleich als Polynom in $h^2$ anzusetzen, sodass das Neville-Schema konkret folgende Gestalt annimmt, siehe (4.21):

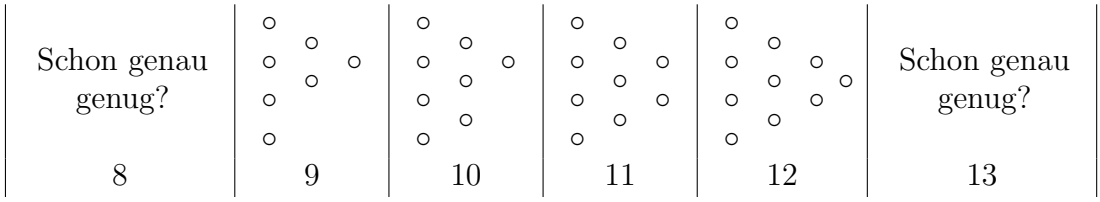
$$\begin{array}{rcll} T_{i,i} & := & T_f(h_i), & i = 1,2,\ldots,n \\ T_{i,i+k} & := & T_{i,i+k-1} + (-h_i^2)\frac{T_{i,i+k-1}-T_{i+1,i+k}}{h_{i+k}^2-h_i^2} & i = 1,2,\ldots,n, \quad k = 1,2,\ldots,n-i \\ \bar{T}_f(0) & := & T_{0,n} & \end{array} \tag{5.26}$$

   Das Schema (5.26) ensteht aus (4.21) durch folgende Substitutionen:

$$p_{i,i+k} \to T_{i,i+k}, \quad f_i \to T_f(h_i), \quad \bar{x} \to 0, \quad x_i \to h_i^2, \quad x_{i+k} \to h_{i+k}^2.$$

Natürlich wird man in der praktischen Rechnung $n$ nicht von vornherein festsetzen, da man ohne unnötigen Rechenaufwand ein bestimmtes Genaugkeitsniveau erreichen will. Der zeitliche Ablauf in der praktischen Rechnung ist wie folgt.




Man berechnet also immer erst dann eine neue Trapezsumme, die auf dem feineren Gitter zusätzliche Funktionsauswertungen kostet, wenn das Genauigkeitsniveau noch nicht erreicht ist.

Ein weiterer für den Algorithmus wichtiger Aspekt ist die Frage, wie die Schrittweitenfolge $h_0, h_1, h_2, \ldots$ zweckmäßig zu wählen ist. Einerseits sollte man darauf achten, dass bereits vorliegende Funktionsauswertungen möglichst auch in den späteren Trapezsummen mit feineren Gittern verwendet werden, was z.B. durch $h_0 = b - a$, $h_1 = \frac{h_0}{2}$, $h_2 = \frac{h_1}{2}$, $h_3 = \frac{h_2}{2}, \ldots$ gegeben ist, andererseits sollte die Gitterfeinheit nicht zu rasch zunehmen, um Rechenarbeit für die Berechnung neuer $T_f(h_i)$ zu sparen. Man wählt daher oft die Folge

$$h_0 = b - a,\ h_1 = \frac{h_0}{2},\ h_2 = \frac{h_0}{3},\ h_3 = \frac{h_0}{4},\ h_4 = \frac{h_0}{6},\ h_5 = \frac{h_0}{8},\ \ldots$$

Es sei hier nur bemerkt, dass mit den Newton - Cotes - Formeln, den Gauß - Radau - Lobatto - und Gauß - Kronrod - Formeln und der Rombergintegration noch nicht alle eindimensionalen Quadraturformeln besprochen worden sind.

Ein Formeltyp von allerdings nur theoretischer Bedeutung sind Qudraturformeln mit gleichen Koeffizienten $c_i = c$. In

$$\int_a^b f(t)\,\mathrm{dt} \approx c \sum_{i=0}^{n} f(t_i)$$

werden nur die Knoten $t_i$ so gewählt, dass sich hohe Ordnungen ergeben. Für diese und alle weiteren eindimensionalen Quadraturformeln wird auf weiterführende Literatur verwiesen.

Ein anderes wichtiges Kapitel, das hier ganz unberücksichtigt geblieben ist, sind *uneigentliche Integrale*, also Integrale, die existieren, aber entweder ein unendliches Integrationsintervall haben oder einen Integrand mit einer oder mehreren Singularitäten. Singularitäten und unendliches Integrationsintervall können natürlich auch gemeinsam auftreten. Auch hierfür wird auf die weiterführende Literatur verwiesen.

## 5.5 Mehrdimensionale Integrale

In diesem Abschnitt werden einfache Quadraturideen für den zweidimensionalen Fall besprochen. Wir betrachten, siehe Abbidlung 5.4, das zweidimensionale Integral

$$\iint_{\mathbb{G}} f(t_1, t_2)\ \mathrm{dt}_1\,\mathrm{dt}_2. \tag{5.27}$$

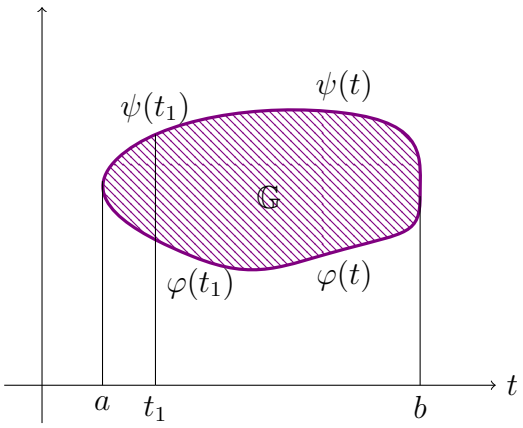



Abbildung 5.4: Integral über das Gebiet $\mathbb{G}$

Es gilt

$$\iint_{\mathbb{G}} f(t_1, t_2)\ \mathrm{dt}_1\,\mathrm{dt}_2 \;=\; \int_a^b \Big[\int_{\varphi(t_1)}^{\psi(t_1)} f(t_1, t_2) dt_2\Big]\,\mathrm{dt}_1, \tag{5.28}$$

die zweidimensionale Quadratur kann also auf die Berechnung von zwei eindimensionalen Integralen

$$\int_a^b g(t_1) dt_1 \quad \text{mit} \quad g(t_1) = \int_{\varphi(t_1)}^{\psi(t_1)} f(t_1, t_2)\,\mathrm{dt}_2$$

zurückgeführt werden. Man kann daher grundsätzlich mit den bereits bekannten Quadraturformeln das Integral $\int_a^b g(t_1)\,\mathrm{dt}_1$ berechnen:

$$\int_a^b g(t_1)\,\mathrm{dt}_1 \;\approx\; c_0 g(t_1^0) + c_1 g(t_1^1) + \cdots + c_n g(t_1^n),$$

wobei bei jedem Funktionsaufruf $g(t_1^i)$ das Integral

$$\int_{\varphi(t_1^i)}^{\psi(t_1^i)} f(t_1^i, t_2)\,\mathrm{dt}_2$$

berechnet wird, d.h. wieder eine – meist dieselbe – Quadraturformel verwendet werden muss.

Den Fehler, den man bei der numerischen Berechnung des inneren Integrals $\int_{\varphi(t_1^i)}^{\psi(t_1^i)} f(t_1^i, t_2)\,\mathrm{dt}_2$ macht, kann man als Datenfehler bei der Funktionsauswertung von $g(t_1^i)$ betrachten und kann somit leicht den Gesamtverfahrensfehler abschätzen, wobei man nur Datenfehlerabschätzungen und Verfahrensfehlersabschätzungen der eindimensionalen Quadratur benötigt.

**Beispiel 5.5.1.** *Es soll* $\int_{-1}^{+1}\int_{-1}^{+1} e^{\frac{1}{10}(t_1+t_2)}\,dt_1\,dt_2$ *mit Hilfe der dreipunktigen Gauß - Formel berechnet werden, also*

$$\int_{-1}^{+1} f(t)\,dt \;\approx\; \frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right)+\frac{8}{9}f(0)+\frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right).$$

*Es folgt*

$$\begin{aligned}
&\int_{-1}^{+1}\int_{-1}^{+1} e^{\frac{1}{10}(t_1+t_2)}\,dt_1\,dt_2 \approx\\
&\approx \frac{5}{9}\int_{-1}^{+1} e^{\frac{1}{10}(-\sqrt{\frac{3}{5}}+t_2)}\,dt_2+\frac{8}{9}\int_{-1}^{+1} e^{\frac{1}{10}t_2}\,dt_2+\frac{5}{9}\int_{-1}^{+1} e^{\frac{1}{10}(\sqrt{\frac{3}{5}}+t_2)}\,dt_2 \approx\\
&\approx \frac{5}{9}\left[\frac{5}{9}e^{\frac{2}{10}(-\sqrt{\frac{3}{5}})}+\frac{8}{9}e^{\frac{1}{10}(-\sqrt{\frac{3}{5}})}+\frac{5}{9}e^{0}\right]+\\
&\quad+\frac{8}{9}\left[\frac{5}{9}e^{\frac{1}{10}(-\sqrt{\frac{3}{5}})}+\frac{8}{9}e^{0}+\frac{5}{9}e^{\frac{1}{10}(\sqrt{\frac{3}{5}})}\right]+\\
&\quad+\frac{5}{9}\left[\frac{5}{9}e^{0}+\frac{8}{9}e^{\frac{1}{10}\sqrt{\frac{3}{5}}}+\frac{5}{9}e^{\frac{2}{10}\sqrt{\frac{3}{5}}}\right]=\\
&= \frac{25}{81}e^{-\frac{1}{5}\sqrt{\frac{3}{5}}}+\frac{80}{81}e^{-\frac{1}{10}\sqrt{\frac{3}{5}}}+\frac{114}{81}e^{0}+\frac{80}{81}e^{\frac{1}{10}\sqrt{\frac{3}{5}}}+\frac{25}{81}e^{\frac{1}{5}\sqrt{\frac{3}{5}}}=\\
&= 4.013351122\ldots
\end{aligned}$$

*Der exakte Integralwert ist 4.013351122...*

Bei höherdimensionalen Integralen steigt allerdings die Anzahl der Funktionsauswertungen rasch an. Arbeitet man etwa mit einer $(n+1)$-punktigen Gauß - Formel, so hat man bei einem $k$-dimensionalen Integral $(n+1)^k$ Funktionsauswertungen (z.B. $n=5$ und $k=10$ ergibt $(n+1)^k \approx 6\cdot 10^7$!). Bei hochdimensionalen Integralen scheitern daher wegen der hohen Rechenzeit und dem schlechten Rundungsfehlerniveau diese einfachen Verfahren. Alternativen sind hier:

(i) Monte - Carlo - Methoden, sogenannte randomisierte Algorithmen,

(ii) Methoden, die mit Hilfe von zahlentheoretischen Betrachtungen (Gleichverteilung) hergeleitet werden, siehe Literatur.

## 5.6 Aspekte bezüglich praktischer Implementierungen

In den vorhergehenden Abschnitten haben wir zahlreiche Quadraturformeln kennengelernt. Um gute Quadratur-Software zu produzieren, genügt es nicht, einfach diese Quadraturverfahren zu programmieren. Man muss auch umfangreiche Überlegungen anstellen, um u.a. sicherzustellen, dass

(i) das gewünschte Genauigkeitsniveau mit sehr hoher Warscheinlichkeit auch wirklich erreicht wird und dass

(ii) dies ohne unnötigen Rechenaufwand, also *effizient* geschieht.

Ein wichtiger Aspekt bezüglich (ii) sind die sogenannten **adaptiven Schrittweitensteuerungen**, die automatisch a-posteriori, also während des Rechnungsverlaufes, dafür sorgen, dass in den Bereichen, wo der Integrand welliger ist d.h. größere Ableitungen hat, die Gitterpunkte dichter liegen. Wir deuten all diese Fragen im Folgenden nur an:

### 5.6.1 Fehlerschätzungen

Es soll nicht nur der Näherungswert $I$ für $\int_a^b f(t)\,\mathrm{dt}$ produziert werden, sondern auch eine Schätzung für $e(I) := I - \int_a^b f(t)\,\mathrm{dt}$. Dafür sind im Wesentlichen zwei Vorgangsweisen verbreitet:

- **($\mathbf{h - \frac{h}{2}}$)-Kriterium:** Wie bereits erwähnt, besitzen praktisch alle Quadraturformeln eine asymptotische Fehlerentwicklung, d.h. für ein Verfahren der Ordnung $p$ gilt

$$e(I) := I_h - \int_a^b f(t)\,\mathrm{dt} \;=\; \tau_p h^p + O(h^{p+1}). \tag{5.29}$$

  Der Faktor $\tau_p$ hängt vom Integranden $f$ und von den Integrationsgrenzen $a, b$ ab und ist daher in konkreten Fällen nicht bekannt. Zur Schätzung des Fehlers liegt nun folgende Vorgangsweise nahe:
  Man berechnet mit derselben Quadraturformel eine Näherung $I_{\frac{h}{2}}$ basierend auf der Schrittweite $\frac{h}{2}$. Für diese Näherung gilt dann

$$I_{\frac{h}{2}} - \int_a^b f(t)\,\mathrm{dt} \;=\; \tau_p \left(\frac{h}{2}\right)^p + O(h^{p+1}). \tag{5.30}$$

  Bildet man die Differenz (5.29) − (5.30), so folgt

$$I_h - I_{\frac{h}{2}} \;=\; \tau_p \underbrace{(h^p - \frac{h^p}{2^p})}_{\frac{2^p-1}{2^p}h^p} + O(h^{p+1}) \tag{5.31}$$

  und damit [1)]

$$(I_h - I_{\frac{h}{2}})\frac{1}{2^p-1} \;=\; \left(\frac{h}{2}\right)^p \tau_p + O(h^{p+1}) \;=\; I_{\frac{h}{2}} - \int_a^b f(t)\,\mathrm{dt} + O(h^{p+1}). \tag{5.32}$$

  Man hat also zusätzlich zum genaueren Näherungswert $I := I_{\frac{h}{2}}$ auch die Fehlerschätzung

$$e(I) \;=\; \frac{1}{2^p-1}(I_h - I_{\frac{h}{2}}) \tag{5.33}$$

  zur Verfügung.

- **Formelpaare,** z.B. Gauß-Kronrod: Man veschafft sich zwei Näherungswerte $I^{(1)}$ und $I^{(2)}$ für $\int_a^b f(t)\,\mathrm{dt}$, die auf zwei verschiedenen Quadraturformeln mit unterschiedlicher Ordnung basieren. Zweckmäßigerweise nimmt man immer an, dass in der genaueren Formel auch alle Funktionsauswertungen der ungenaueren Formel verwendet werden und natürlich auch noch einige zusätzliche. Für hinreichend kleine Schrittweiten ist dann

$$I^{(1)} - I^{(2)} \;\approx\; I^{(1)} - \int_a^b f(t)\,\mathrm{dt}$$

  In der Praxis nimmt man auch hier die genauere Näherung als endgültigen Näherungswert des Integrals, also $I := I^{(2)}$. Die Größe

$$e(I) \;:=\; I^{(1)} - I^{(2)} \tag{5.34}$$

  ist dann i.a. eine starke *Überschätzung* des Fehlers.

[1)] Der Summand $O(h^{p+1})$ verschwindet nicht, da es sich nicht um zwei gleiche exakte Werte handelt, die voneinander abgezogen werden, sondern um zwei (möglicherweise) verschiedene Werte der selben Größsenordnung.

### 5.6.2 Schrittweitensteuerungen

Hier sind zwei Varianten verbreitet: die *globale Strategie* und die *lokale Strategie*.

- **Globale Strategie:**

  1. Schritt: Berechnung von $I_{[a,b]}$ und $e(I_{[a,b]})$

     $$\text{Abfrage:} \quad |e(I_{[a,b]})| < \varepsilon$$

     falls ja: fertig, falls nein:

  2. Schritt: Berechnung von $I_{[a,\frac{a+b}{2}]},\ I_{[\frac{a+b}{2},b]}$ und $e(I_{[a,\frac{a+b}{2}]}),\ e(I_{[\frac{a+b}{2},b]})$.

     $$\text{Abfrage:} \quad |e(I_{[a,\frac{a+b}{2}]})| + |e(I_{[\frac{a+b}{2},b]})| < \varepsilon$$

     falls ja: fertig,[2)] falls nein:

  3. Schritt: Es wird jenes Intervall nochmals halbiert, dem die betragsgrößere Fehlerschätzung entspricht usw.

- **Lokale Strategie:**

  1. Schritt: wie bei globaler Strategie

  2. Schritt: Berechnung von $I_{[a,\frac{a+b}{2}]},\ I_{[\frac{a+b}{2},b]}$ und $e(I_{[a,\frac{a+b}{2}]}),\ e(I_{[\frac{a+b}{2},b]})$

     $$\begin{aligned} \text{Abfragen:} \quad |e(I_{[a,\frac{a+b}{2}]})| &< \frac{1}{2}\varepsilon \\ |e(I_{[\frac{a+b}{2},b]})| &< \frac{1}{2}\varepsilon \end{aligned}$$

     Wenn beide Bedingungen erfüllt: fertig.
     Wenn nur eine von beiden erfüllt: Das andere Intervall wird weiter halbiert.
     Wenn beide nicht erfüllt: Beide Intervalle werden weiter halbiert usw.

Der Vorteil der lokalen Strategie ist ein geringerer Organisationsaufwand, es muss nicht immer wie bei der globalen Strategie ein Sortiervorgang erfolgen. Dies ist aber bei der Geschwindigkeit moderner Rechner kein wesentlicher Aspekt.

Der Nachteil der lokalen Strategie ist, dass unter gewissen Umständen lokal eine ganz sinnlos hohe Genauigkeit erreicht wird, da die Größe $\varepsilon$ immer mit der lokalen Intervalllänge gewichtet ist. In den Unstetigkeitsstellen, wo wegen der geringen Glattheit des Integranden die Quadraturformeln schlecht arbeiten, werden ganz kurze Intervalle entstehen. Die Abfrage $|e(.)| <$ (extrem kurze Länge) $\cdot\, \varepsilon$ wird sehr schwer zu erfülllen sein und es entstehen dort immer noch kürzere Intervalle, obwohl der Beitrag dieser Intervalle zum Gesamtintegral längst vernachlässigbar ist.

---

[2)] Falls tatsächlich vorzeichenbehaftete Schätzung vorliegt: Abfrage eventuell

$$|e(I_{[a,\frac{a+b}{2}]}) + e(I_{[\frac{a+b}{2},b]})| < \varepsilon$$

## 5.7 Ein abschliessendes Zahlenbeispiel

Das Intgral

$$\int_0^1 e^t \,\mathrm{dt} \;=\; e^t\Big|_0^1 \;=\; e-1 \;=\; 1.718281828\ldots$$

soll numerisch berechnet werden. Wir vergleichen dabei numerische Verfahren mit annähernd gleichem Rechenaufwand und jeweils 3 Funktionsauswertungen.

**Zusammengesetzte Trapezregel.** Verwende 2 Interpolationsintervalle $[0, \frac{1}{2}]$ und $[\frac{1}{2}, 1]$, also $h = \frac{1}{2}$, $t_0 = 0$, $t_1 = h = \frac{1}{2}$ und $t_2 = 2h = 1$.
Es folgt

$$\begin{aligned}\int_0^1 e^t \,\mathrm{dt} \;&\approx\; \frac{1}{2}he^0 + he^{\frac{1}{2}} + \frac{1}{2}he^1 \;= \\ &=\; \underline{1.7}53991092\ldots\end{aligned}$$

**Simpsonregel.** Setze $h = \frac{1}{2}$, $t_0 = 0$, $t_1 = \frac{1}{2}$ und $t_2 = 1$. Es folgt

$$\begin{aligned}\int_0^1 e^t \,\mathrm{dt} \;&\approx\; \frac{1}{6}he^0 + \frac{4}{6}he^{\frac{1}{2}} + \frac{1}{6}he^1 \;= \\ &=\; \underline{1.718}861152\ldots\end{aligned}$$

**Trapezsummenextrapolation.** Wähle $h_0 = 1$ und $h_1 = \frac{1}{2}$. Damit lautet das Neville-Schema

$$\begin{array}{lcl} T_{0,0} &=& 1.859140914\ldots \\ & & \qquad\qquad T_{0,1} \;=\; \underline{1.718}861151\ldots \\ T_{1,1} &=& \underline{1.7}53991092\ldots \end{array}$$

Bis auf Rundungsfehler ergeben Trapezsummenextrapolation und Simpsonregel denselben Wert.

**Gaußformel.** Mit drei Punkten lautet die Formel bezüglich $[-1, +1]$

$$\int_{-1}^{+1} f(x)\,\mathrm{dx} \;\approx\; \frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9}f(0) + \frac{5}{9}f\left(\sqrt{\frac{2}{5}}\right).$$

Transformation des Intervalls: $-1 \le x \le 1 \;\rightarrow\; a = 0 \le t \le 1 = b$

Transformation der Knoten: $t_i = \frac{a+b}{2} + \frac{b-a}{2}x_i = \frac{1}{2} + \frac{1}{2}x_i$.
Es folgt

$$\begin{array}{rcrcrcl} x_0 &=& -\sqrt{\frac{3}{5}} &\rightarrow& t_0 &=& \frac{1}{2} - \frac{1}{2}\sqrt{\frac{3}{5}}, \\ x_1 &=& 0 &\rightarrow& t_1 &=& \frac{1}{2}, \\ x_2 &=& \sqrt{\frac{3}{5}} &\rightarrow& t_2 &=& \frac{1}{2} + \frac{1}{2}\sqrt{\frac{3}{5}}. \end{array}$$

Transformation der Gewichte:

$$\begin{aligned} c_{i,[-1,+1]} \;&=\; \int_{-1}^{+1} \psi_i(x)\,\mathrm{d}x, \quad \text{siehe (5.15)}, \\ c_{i,[a,b]} \;&=\; \int_a^b \psi_{i,[a,b]}(t)\,\mathrm{d}t. \end{aligned}$$

Riemannsumme bezüglich $\psi_i(x)$ und analoge Riemannsumme bezüglich $\psi_{i,[a,b]}(t)$ liefern $c_{i,[a,b]} : c_{i,[-1,+1]} = (b-a) : 2$ und damit

$$c_{i,[a,b]} = \frac{b-a}{2} c_{i,[-1,+1]} = \frac{1}{2} c_{i,[-1,+1]}.$$

Es folgt

$$\begin{aligned} \int_0^1 f(t)\,\mathrm{d}t &\approx \frac{5}{9}\cdot\frac{1}{2} f\left(\frac{1}{2}-\frac{1}{2}\sqrt{\frac{3}{5}}\right) + \frac{8}{9}\cdot\frac{1}{2} f\left(\frac{1}{2}\right) + \frac{5}{9}\cdot\frac{1}{2} f\left(\frac{1}{2}+\frac{1}{2}\sqrt{\frac{3}{5}}\right) = \\ &= \frac{5}{18} e^{(\frac{1}{2}-\frac{1}{2}\sqrt{\frac{3}{5}})} + \frac{4}{9} e^{\frac{1}{2}} + \frac{5}{18} e^{(\frac{1}{2}+\frac{1}{2}\sqrt{\frac{3}{5}})} = \underline{1.7182809}051\ldots \end{aligned}$$

# Kapitel 6

# Numerische Lösung von Differentialgleichungen

## 6.1 Anfangswertprobleme

Wir betrachten Anfangswertprobleme von gewöhnlichen Differentialgleichungen (ordinary differential equations, ODEs) **erster Ordnung**, in den Gleichungen

$$\begin{aligned} y'(t) = \frac{dy}{dt} &= f(t, y(t)) \quad \text{mit } f : [t_0, t_{end}] \times \mathbb{R}^n \to \mathbb{R}^n \,, \\ y(t_0) &= y_0, \quad y_0 \in \mathbb{R}^n \end{aligned} \tag{6.1}$$

mit Lösung

$$y(t) : [t_0, t_{end}] \to \mathbb{R}^n$$

tritt also nur die erste Ableitung der gesuchten Funktion $y$ auf.

**Bemerkung.** Ein System von Differentialgleichungen $k$**-ter Ordnung** ($k \in \mathbb{N}$), also

$$\begin{aligned} y^{(k)}(t) &= f(t, y(t), y'(t), y''(t), ..., y^{(k-1)}), \\ y(t_0) = y_0,\ y'(t_0) &= y_1,\ ...,\ y^{(k-1)}(t_0) = y_{k-1}, \end{aligned}$$

lässt sich in ein (größeres) System erster Ordnung überführen.

Der folgende Satz ist eine Aussage über die Existenz und Eindeutigkeit von Lösungen von (6.1).

**Satz 6.1.1** (Picard-Lindelöf)**.** *Sei* $\mathcal{D} := \{(t, y) : t_0 \le t \le a, \|y - y_0\| \le b\}$ *und sei die Funktion* $f : \mathcal{D} \to \mathbb{R}^n$ *stetig und beschränkt mit* $\|f(t, y)\| \le M$ *in* $\mathcal{D}$*. Erfüllt* $f$ *eine Lipschitzbedingung in* $y$ *mit Lipschitzkonstante* $L$*, also* $\|f(t, y_1) - f(t, y_2)\| \le L\|y_1 - y_2\|$ *für* $(t, y_i) \in \mathcal{D}$ *,* $i = 1, 2$ *, so existiert genau eine Lösung* $y(t)$ *von* (6.1) *für* $t_0 \le t \le T := \min(a, \frac{b}{M})$*.*

**Beweis.** Siehe weiterführende Literatur. □

Im Folgenden nehmen wir an, dass die Funktion $f$ die Voraussetzungen des Satzes von Picard-Lindelöf erfüllt.

Die Lipschitzkonstante $L$ ist auch ein wichtiger Parameter für die Sensitivität der Lösung gegenüber Störungen der Funktion $f$ und der Anfangsbedingung $y_0$ .

**Satz 6.1.2** (Konditionsabschätzung für Anfangswertprobleme)**.** *Betrachten wir das System von Differentialgleichungen* (6.1) *und ein gestörtes Problem*

$$\begin{aligned} \tilde{y}'(t) &= \frac{d\tilde{y}}{dt} = \tilde{f}(t, \tilde{y}(t))\,, \\ \tilde{y}(t_0) &= \tilde{y}_0\,. \end{aligned} \tag{6.2}$$

*Für die Funktionen $f, \widetilde{f} : \mathcal{D} \to \mathbb{R}^n$ sind die Voraussetzungen des Satzes von Picard-Lindelöf erfüllt. Es gelte weiters für $\delta, \delta_0 \in \mathbb{R}_{\geq 0}$*

$$\begin{aligned} \|y_0 - \widetilde{y_0}\| &\leq \delta_0\,, \\ \|f(t,y) - \widetilde{f}(t,y)\| &\leq \delta \quad \text{in } \mathcal{D}\,. \end{aligned}$$

*Dann gilt für die beiden Lösungen $y$ und $\tilde{y}$ die Abschätzung*

$$\|y(t) - \widetilde{y}(t)\| \leq e^{Lt}\delta_0 + \frac{e^{Lt}-1}{L}\,\delta\,, \qquad t \in [t_0, T]\,. \tag{6.3}$$

**Beweis.** Lemma von Gronwall. □

**Lemma 6.1.3.** (Gronwall) *Angenommen die Funktion $v : [0,T] \to \mathbb{R}$ erfüllt*

$$\begin{aligned} v'(t) &\leq \omega v(t) + \delta\,, \qquad t \in [0,T] \\ v(0) &\leq \delta_0 \end{aligned} \tag{6.4}$$

*mit $\delta$, $\delta_0 \geq 0$ und $\omega \in \mathbb{R}$. Dann gilt*

$$v(t) \leq e^{\omega t}\delta_0 + \frac{e^{\omega t}-1}{\omega}\,\delta\,, \qquad t \in [0,T]\,.$$

**Bemerkung.** Im Fall $\omega = 0$ gilt $v(t) \leq \delta_0 + \delta t$, da $\lim_{\omega\to 0} \frac{e^{\omega t}-1}{\omega} = t$.

**Beweis.** Multiplikation von (6.4) mit $e^{-\omega t}$ führt auf $(e^{-\omega t}v(t))' \leq e^{-\omega t}\delta$, Integration und danach Multiplikation mit $e^{\omega t}$ liefert die Behauptung. □

**Beispiel 6.1.4.** *Gegeben sei das skalare Problem*

$$\begin{aligned} y' &= \lambda y\,, \\ y(0) &= y_0 \end{aligned} \tag{6.5}$$

*mit $\lambda \in \mathbb{R}$ und einer Störung $\delta_0 \in \mathbb{R}_{\geq 0}$ in der Anfangsbedingung,*

$$\tilde{y}(0) = y_0 + \delta_0.$$

*Die exakte Lösung ist $y(t) = y_0 e^{\lambda t}$, die gestörte Lösung $\widetilde{y}(t) = (y_0 + \delta_0)e^{\lambda t}$.*
*Aus $L = |\lambda|$ und* (6.3) *ergibt sich wegen $\delta = 0$ die Abschätzung*

$$|y(t) - \widetilde{y}(t)| = |y_0 e^{\lambda t} - (y_0 + \delta_0)e^{\lambda t}| \leq e^{|\lambda| t}\delta_0\,,$$

*die für $\lambda \ll 0$ sehr pessimistisch und daher unrealistisch ist.*
*Betrachte dafür Abbildung 6.1, die für $\lambda < 0$ illustriert, wie die Lösungen für $t \to \infty$ zusammenlaufen.*

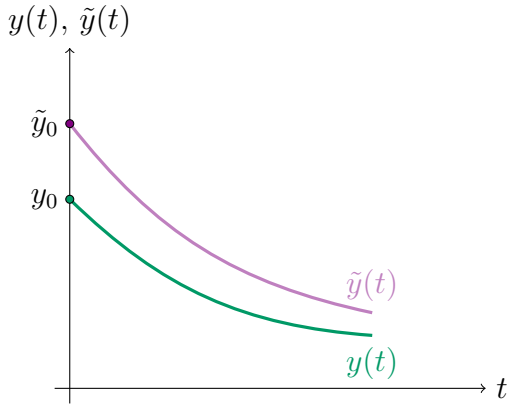



Abbildung 6.1: Lösung $y(t)$ und gestörte Lösung $\tilde{y}(t)$ des Anfangswertproblems (6.5)

Das Anfangswertproblem

$$y'(t) = \lambda y(t), \qquad y(0) = y_0 \qquad \text{mit } \lambda \ll 0$$

ist ein sehr einfaches sogenanntes *steifes* Problem. Es zeigt das folgende charakteristische Verhalten:

(i) Die Lipschitzkonstante $L = |\lambda|$ ist sehr groß.

(ii) Für $y_0 = 0$ ist die Lösung $y(t) \equiv 0$ glatt.

(iii) Für $y_0 \neq 0$ fällt die Lösung $y(t) = e^{\lambda t} y_0$ unmittelbar nach dem Start rasch ab, ist also unglatt (dh. die Ableitungen sind betragsmäßig sehr groß). Weiter entfernt von $t = 0$ wird die Lösung wieder schnell glatt und nähert sich der glatten Lösung $y(t) \equiv 0$. Abbildung 6.2 illustriert dieses Verhalten.

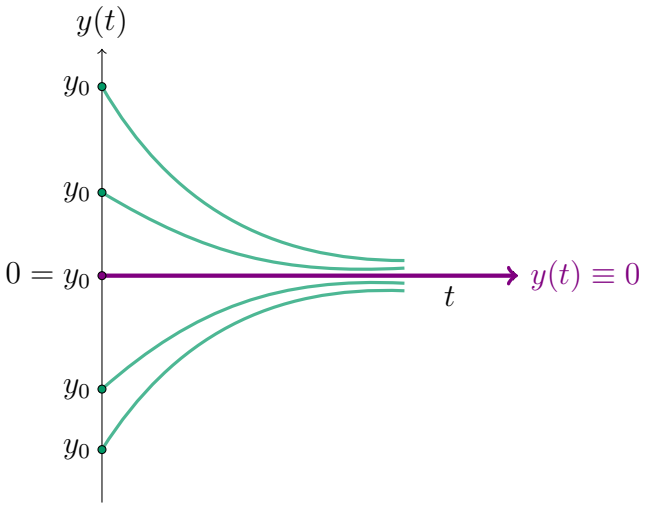



Abbildung 6.2: Lösungen für verschiedene Anfangswerte $y_0 \in \mathbb{R}$, $y_0 \neq 0$ nähern sich der glatten Lösung $y(t) \equiv 0$ an

Für allgemeine Probleme kann *Steifheit* folgendermaßen definiert werden:

**Definition 6.1.5.** *Ein System von gewöhnlichen Differentialgleichungen (6.1) heißt* **steif**, *wenn die Jacobimatrix $f_y = (\frac{\partial f_i}{\partial y_j})$ für $i, j = 1, 2, 3, ..., n$ in der Nähe der Lösung $y$ Eigenwerte $\lambda$ mit Realteil $\lambda \ll 0$ hat, neben Eigenwerten von moderater Größenordnung.*

Steife Probleme sind genau jene, für die Konditionsabschätzungen basierend auf der üblichen Lipschitzkonstante $L$ extrem unrealistisch sind, da diese für steife Probleme im Allgemeinen sehr groß ist, obwohl die Probleme sehr gute Kondition haben. Störungen werden sehr schnell weggedämpft. Ein etwas allgemeineres steifes Problem ist durch

$$y'(t) = \lambda\Big(y(t) - g(t)\Big) + g'(t), \qquad y(0) = y_0 \tag{6.6}$$

mit einer glatten Funktion $g(t)$ und $\lambda << 0$ gegeben. Der Fall $y_0 = g(0)$ liefert als Lösung die Funktion $g(t)$. Abbildung 6.3 zeigt das Verhalten der Lösungen zu verschiedenen Anfangsbedingungen $y_0 \in \mathbb{R}$.

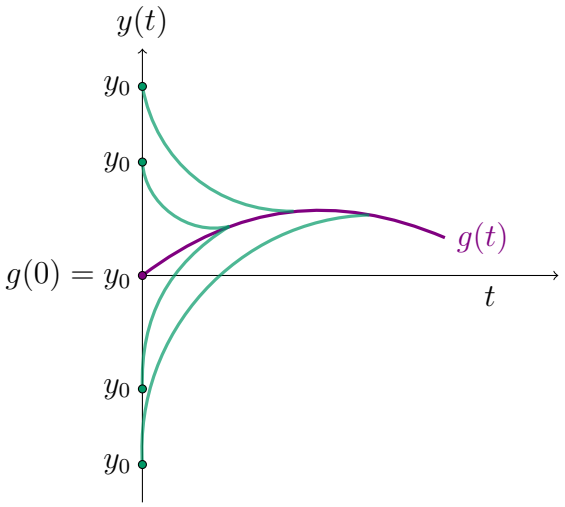



Abbildung 6.3: Lösungen von (6.6) für verschiedene Anfangswerte $y_0 \in \mathbb{R}$, $y_0 \neq g(0)$ nähern sich der glatten Lösung $y(t) = g(t)$ an

Für Abschätzungen im Zusammenhang mit steifen Probleme wird die sogenannte *einseitige Lipschitzkonstante* $m \in \mathbb{R}$ verwendet:

**Definition 6.1.6.** *Eine Funktion $f : \mathcal{D} \to \mathbb{R}^n$ erfüllt eine einseitige Lipschitzbedingung bezüglich $y$ in $\mathcal{D} \subseteq [t_0, t_{end}] \times \mathbb{R}^n$ mit* **einseitiger Lipschitzkonstante** *$m \in \mathbb{R}$, falls*

$$\langle y - \tilde{y}, f(t, y) - f(t, \tilde{y})\rangle \leq m\|y - \tilde{y}\|_2^2$$

*für $(t, y), (t, \tilde{y}) \in \mathcal{D}$ und ein Skalarprdukt $<,>$ auf $\mathbb{R}^n$ .*

Zur Lösung steifer Probleme sind **implizite numerische Verfahren**, die im nächsten Abschnitt vorgestellt werden, besser geeignet.

## 6.2 Euler-Verfahren; Konsistenz, Stabilität, Konvergenz

Ein einfaches Verfahren zur Lösung von Anfangswertproblemen gewöhnlicher Differentialgleichungen ist das **Euler-Verfahren**: Entwickeln wir die gesuchte Lösung $y$ von (6.1) an der Stelle $t + h$ in eine Taylorreihe und setzen $y'(t) = f(t, y(t))$ ein, so erhalten wir

$$y(t + h) = y(t) + hf(t, y(t)) + O(h^2) \approx y(t) + hf(t, y(t)).$$

Das motiviert das folgende einfache Diskretisierungsverfahren

$$\begin{aligned} y_0 &= y_0 \\ y_1 &= y_0 + hf(t_0, y_0) \\ &\vdots \\ y_n &= y_{n-1} + hf(t_{n-1}, y_{n-1})\,, \quad n = 1, 2, \ldots \qquad t_j = t_0 + jh\,, \qquad j = 0, 1, 2, \ldots\,. \end{aligned} \tag{6.7}$$

Die Schrittweite $h$ kann dabei auch variabel sein.
Es sind $y_1 \approx y(t_1), y_2 \approx y(t_2), \ldots, y_n \approx y(t_n), \ldots$ Näherungen an die exakte Lösung $y$ zu den Zeitpunkten $t_1$, $t_2$,..., $t_n$, ... . Zur Illustration dieses *expliziten Euler-Verfahrens* betrachte Abbildung 6.4.

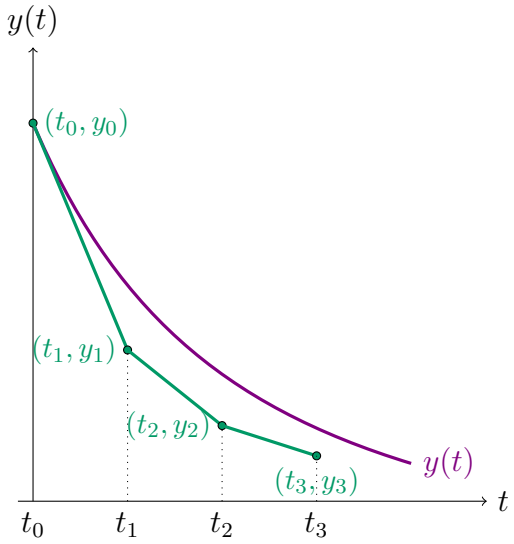



Abbildung 6.4: Das explizite Euler-Verfahren

**Definition 6.2.1.** *Ein* **explizites** *numerisches Verfahren schließt von $y_{n-1}$ auf $y_n$ .*

**Definition 6.2.2.** *Der Fehler $l_n$, der in einem Schritt entsteht, wird als* **lokaler Fehler** *oder* **lokaler Diskretisierungsfehler** *bezeichnet, also*

$$y(t_n) - y(t_{n-1}) - hf(t_{n-1}, y(t_{n-1})) = l_n h\,, \tag{6.8}$$

$$\frac{y(t_n) - y(t_{n-1})}{h} - f(t_{n-1}, y(t_{n-1})) = l_n\,. \tag{6.9}$$

Das explizite Euler-Verfahren ist **konsistent**, d.h. $\|l_n\| \to 0$ für $h \to 0$. Falls die Lösung $y$ zweimal stetig differenzierbar ist, gilt

$$l_n = \frac{y(t_n) - y(t_{n-1})}{h} - y'(t_{n-1}) = h \int_0^1 y''(t_{n-1} + \theta h)(1 - \theta)\mathrm{d}\theta\,, \qquad 0 < \theta < 1$$

und daher

$$\|l_n\| = O(h) = O(h^p) \quad \text{mit } p = 1\,. \tag{6.10}$$

Die **Konsistenzordnung** $p$ für das explizite Euler-Verfahren ist daher 1, es gilt

$$\|l_n\| \le \frac{M_2}{2} h\,,$$

wobei $M_2$ eine Schranke für $\|y''(t)\|$ ist.

Der **globale (Diskretisierungs-)fehler** ist $e_n = y_n - y(t_n)$.

Abbildung 6.5 zeigt den Unterschied von lokalem und globalem Diskretisierungsfehler. Der rot gezeichnete Fehler trägt zum lokalen Fehler bei, während die gestrichelte Linie den globalen Fehler darstellt. Dabei bezeichnet $y(t_0) = y_0$ die exakte Lösung des Problems.

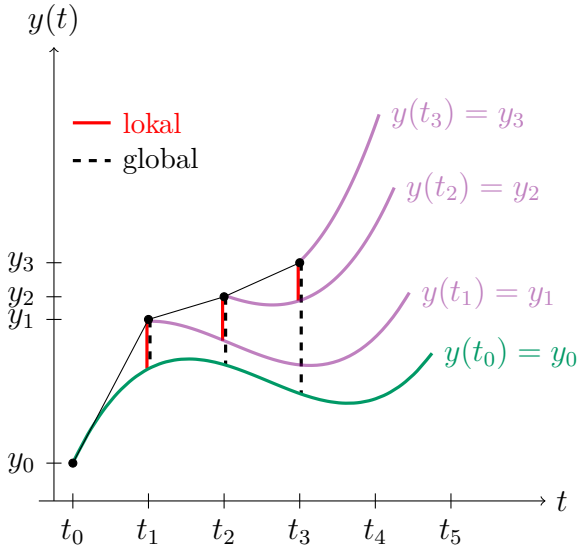



Abbildung 6.5: Lokaler und globaler Diskretisierungsfehler im Vergleich

**Definition 6.2.3.** *Ein Diskretisierungsverfahren heißt* **konvergent**, *falls* $\|e_n\| \to 0$ *für* $h \to 0$, *die* **Konvergenzordnung** $p$ *ist durch*

$$\|e_n\| = O(h^p) \quad p = 1, 2, ...$$

*gegeben.*

Für das explizite Euler-Verfahren ist $p = 1$.
Aus der Konsistenz eines Diskretisierungsverfahrens folgt nicht notwendigerweise Konvergenz. Man braucht zusätzlich die Eigenschaft der **Stabilität**, dass also der globale Effekt der lokalen Fehler gleichmäßig für $h \to 0$ beschränkt bleibt.
Betrachten wir zwei parallele Schritte eines Diskretisierungsverfahrens:

$$(t_{n-1}, y_{n-1}) \to (t_n, y_n)$$
$$(t_{n-1}, \tilde{y}_{n-1}) \to (t_n, \tilde{y}_n)$$

**Definition 6.2.4.** *Das Verfahren heißt* **stabil**, *falls*

$$\|y_n - \tilde{y}_n\| \leq (1 + Sh)\|y_{n-1} - \tilde{y}_{n-1}\|$$

*gleichmäßig für* $h < h_0$ *gilt, wobei die Konstante* $S$ *unabhängig von* $h$ *ist. Für das Euler-Verfahren gilt* $S = L$.

**Satz 6.2.5** (Konvergenz des Euler-Verfahrens). *Sei* $y(t) \in C^2[0, t_{end}]$ *und* $M_2 = \sup_{t\in[0,t_{end}]} \|y''(t)\|$. *Dann gilt*

$$\|e_n\| = \|y_n - y(t_n)\| \leq e^{Lt_n}\|e_0\| + \frac{e^{Lt_n} - 1}{L}\frac{M_2}{2}\, h$$

*wobei* $e_0 = \eta_0 - y(t_0) = O(h)$.

**Beweis.** Lemma von Gronwall. □

**Bemerkungen.**

1. Wachstumsfaktoren wie in Konditionsabschätzungen, siehe Satz 1.1.2.
2. Es gilt: Konsistenz + Stabilität ⇒ Konvergenz.
3. Auch für variables $h$ sind entsprechende Abschätzungen möglich, siehe Literatur.

**Lemma 6.2.6** (Gronwall, diskrete Version). *Angenommen die nichtnegative Folge* $(v_n)$ $n = 0, 1, 2, ...$ *erfüllt*

$$\begin{aligned} v_0 &\leq \delta_0, \\ v_n &\leq (1+\omega)v_{n-1} + \delta, \qquad n = 1, 2, 3, ... \end{aligned}$$

*mit* $\omega$, $\delta_0$, $\delta \geq 0$. *Dann gilt*

$$v_n \leq e^{n\omega}\delta_0 + \frac{e^{n\omega} - 1}{\omega}\,\delta \qquad \textit{für alle } n.$$

**Beweis.** Die Rekursion

$$\begin{aligned} v_1 &\leq (1+\omega)\delta_0 + \delta \\ v_2 &\leq (1+\omega)v_1 + \delta \leq (1+\omega)^2\delta_0 + (1+\omega)\delta + \delta \\ &\vdots \end{aligned}$$

führt auf

$$\begin{aligned} v_n &\leq (1+\omega)^n\delta_0 + (1 + (1+\omega) + ... + (1+\omega)^{n-1})\delta \\ &= (1+\omega)^n\delta_0 + \frac{(1+\omega)^n - 1}{\omega}\delta \\ &\leq e^{n\omega}\delta_0 + \frac{e^{n\omega} - 1}{\omega}\delta \end{aligned}$$

für $1 + \omega \leq e^\omega$ und $\omega > 0$. □

Beim **impliziten Euler-Verfahren**

$$y_n = y_{n-1} + hf(t_n, y_n) \qquad n = 1, 2, ... \tag{6.11}$$

muss in jedem Schritt ein (nicht)lineares Gleichungssystem gelöst werden. Wendet man dieses Verfahren auf das Modellproblem $y'(t) = \lambda y(t)$, $\lambda \ll 0$ und Anfangswert $\delta_0$ an, ergibt sich

$$y_n = \left(\frac{1}{1 - h\lambda}\right)^n \delta_0$$

Für $\lambda \ll 0$ ist $1/(1 - h\lambda)^n$ klein, auch für nicht so kleine Schrittweiten $h$ und daher spiegelt (6.11) das Verhalten von $e^{\lambda t}\delta_0$ sehr gut wider.

## 6.3 Einschrittverfahren allgemein

Ein allgemeines explizites Einschrittverfahren ist durch

$$\begin{aligned} y_0 &= y_0\,, \qquad t_n = t_0 + hn \quad \text{(oder auch variables } h\text{)} \\ \frac{y_n - y_{n-1}}{h} &= \underbrace{\varphi(t_{n-1}, y_{n-1}; h)}_{\text{Inkrementfunktion z.B. } \varphi(t,y;h) = f(t,y)} \qquad n = 1, 2, ... \end{aligned} \tag{6.12}$$

mit geeigneter Wahl von $\varphi$ gegeben, sodass das Verfahren die Konvergenzordnung $p > 1$ hat.
Die Definition des allgemeinen Einschrittverfahrens (6.12) erlaubt es, eine Rekursion für $y_{n-1} \to y_n$ in der Form

$$y_n = y_{n-1} + h\varphi(t_{n-1}, y_{n-1}; h)$$

anzugeben.
Dabei ergibt sich der lokale (Diskretisierungs)fehler

$$l_n = \frac{y(t_n) - y(t_{n-1})}{h} - \varphi((t_{n-1}), y(t_{n-1}); h)\,.$$

Das Produkt $hl_n$ ist die Differenz zwischen exaktem Lösungswert $y(t_n)$ und Näherung $y_n$ mit Startwert $y(t_{n-1})$.
Die Wahl der *Inkrementfunktion* $\varphi$ erlaubt es, höhere Konvergenzordnungen $p > 1$ zu realisieren. Die Konstruktion von Inkrementfunktionen basiert auf der Approximation des Integrals in der zu (6.1) äquivalenten Integralgleichung

$$y(t_n) = y(t_{n-1}) + \int_{t_{n-1}}^{t_n} f(t, y(t))\mathrm{d}t.$$

Das explizite Euler-Verfahren arbeitet mit der einfachen Approximation

$$\int_{t_{n-1}}^{t_n} f(t, y(t))dt \approx hf(t_{n-1}, y(t_{n-1}))\,,$$

das implizite Euler-Verfahren mit

$$\int_{t_{n-1}}^{t_n} f(t, y(t))dt \approx hf(t_n, y(t_n))\,.$$

Eine bessere Approximation kann durch zusätzliche Funktionsauswertungen an Zwischenwerten im Intervall $[t_{n-1}, t_n]$ erreicht werden.
Ein einfaches Beispiel dafür ist das **verbesserte Euler-Verfahren**

$$\begin{aligned} Y_1 &= y_{n-1} + \frac{h}{2} f(t_{n-1}, y_{n-1})), \\ y_n &= y_{n-1} + hf(t_{n-1} + \frac{h}{2}, Y_1) \end{aligned}$$

mit $\varphi(t, y; h) = f(t + \frac{h}{2}, y + \frac{h}{2} f(t, y))$.

Das **Verfahren von Heun** ist ein explizites Einschrittverfahren zweiter Ordnung

$$\begin{aligned} Y_1 &= f(t_{n-1}, y_{n-1}) \\ Y_2 &= f(t_{n-1} + h, y_{n-1} + hY_1) \\ y_n &= y_{n-1} + h\varphi(t_{n-1}, y_{n-1}, h) \\ \varphi(t_{n-1}, y_{n-1}, h) &= \frac{Y_1 + Y_2}{2} \end{aligned}$$

Die Inkrementfunktion $\varphi$ ist also ein Mittelwert zweier Anstiege.

Eine Verallgemeinerung dieser Idee führt auf die **(expliziten) Runge-Kutta-Verfahren**. Hier werden für die Berechnung $y_{n-1} \to y_n$, $s \in \mathbb{N}$ definierte Zwischenapproximationen oder *Stufen* $Y_i$ an den Stellen $\tau_i = t_{n-1} + c_i h,\ i = 1, 2, ...s,\ c_i \in \mathbb{R}\,,\ c_1 = 0$ und $a_{ij} \in \mathbb{R}$ verwendet.

$$\begin{aligned} Y_1 &= y_{n-1} \\ Y_2 &= y_{n-1} + ha_{21} f(\tau_1, Y_1) \\ Y_3 &= y_{n-1} + h(a_{31} f(\tau_1, Y_1) + a_{32} f(\tau_2, Y_2)) \\ &\vdots \\ Y_s &= y_{n-1} + h(a_{s1} f(\tau_1, Y_1) + ha_{s2} f(\tau_2, Y_2) + ... + ha_{s,s-1} f(\tau_{s-1}, Y_{s-1})) \\ y_n &= y_{n-1} + h \underbrace{(b_1 f(\tau_1, Y_1) + b_2 f(\tau_2, Y_2) + ... + b_s f(\tau_s, Y_s))}_{\varphi(t_{n-1}, y_{n-1}; h)} \end{aligned}$$

Die Methode ist durch die reellen Koeffizienten $a_{ij}, b_i, c_j$ und durch die natürliche Zahl $s$ festgelegt und wird durch das sogenannte **Butcher Tableau**

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}$$

dargestellt, wobei

$$A = \begin{pmatrix} 0 & \dots & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ a_{31} & a_{32} & 0 & \dots \\ \vdots & \vdots & \ddots & 0 \\ a_{s1} & a_{s2} & \dots & a_{s,s-1} \end{pmatrix}, \quad \begin{pmatrix} 0 = c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_s \end{pmatrix} \quad \text{und} \quad b = (b_1, b_2, ..., b_s).$$

Im folgenden ist das Butcher Tableau für das explizite Euler-Verfahren ($s = 1$) und das klassische 4-stufige Runge-Kutta-Verfahren der Ordnung $p = 4$ angegeben:

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array} \qquad \begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Ein **implizites $s$-stufiges Runge-Kutta-Verfahren** mit den Stufen $Y_i$ an den Stellen $\tau_i = t_{n-1} + c_i h,\ i = 1, 2, ...s,\ c_1 = 0$ ist durch das folgende $n \times s$ System von nichtlinearen Gleichungen gegeben:

$$\begin{aligned}
Y_1 &= y_{n-1} + h(a_{11}f(\tau_1, Y_1) + ha_{12}f(\tau_2, Y_2) + ... + ha_{1s}f(\tau_s, Y_s)) \\
Y_2 &= y_{n-1} + h(a_{21}f(\tau_1, Y_1) + ha_{22}f(\tau_2, Y_2) + ... + ha_{2s}f(\tau_s, Y_s)) \\
&\vdots \\
Y_s &= y_{n-1} + h(a_{s1}f(\tau_1, Y_1) + ha_{s2}f(\tau_2, Y_2) + ... + ha_{ss}f(\tau_s, Y_s)) \\
y_n &= y_{n-1} + h \underbrace{(hb_1 f(\tau_1, Y_1) + b_2 f(\tau_2, Y_2) + ... + b_s f(\tau_s, Y_s))}_{\varphi(t_{n-1}, y_n - 1; h)}
\end{aligned}$$

Das implizite Euler-Verfahren wird durch das Butcher-Tableau

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

charakterisiert. Der **impliziten Mittelpunktsregel** mit $y_n = y_{n-1} + hf(t_{n-1} + \frac{h}{2}, \frac{1}{2}(y_n + y_{n-1}))$ entspricht das Butcher Tableau

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}.$$

Anhand des folgenden Beispiels sollen die verschiedenen Einschrittverfahren miteinander verglichen werden.

**Beispiel 6.3.1.** *Gegeben ist das Anfangswertproblem*

$$y'(t) = -ty(t), \qquad t \in (0, 1], \qquad y(0) = 1$$

*mit der exakten Lösung $y(t) = e^{-\frac{t^2}{2}}$. Die folgende Tabelle liefert die Werte für das explizite Euler-Verfahren 1. Ordnung, das Verfahren von Heun 2. Ordnung und das Runge-Kutta-Verfahren 4. Ordnung mit $h = 0.2$ sowie die entsprechenden Werte der exakten Lösung. Die übereinstimmenden Nachkommastellen sind dabei fett gedruckt.*

| Gitterpunkt | Euler | Heun | Runge-Kutta | exakte Lösung |
|---|---|---|---|---|
| 0.0 | **1.0000000000** | **1.0000000000** | **1.0000000000** | 1.0000000000 |
| 0.2 | 1.0000000000 | **0.9**799999997 | **0.9801986**654 | 0.9801986733 |
| 0.4 | **0.9**599999994 | **0.92**25579989 | **0.923116**2833 | 0.9231163464 |
| 0.6 | **0.8**831999983 | **0.83**49204842 | **0.835270**0715 | 0.8352702144 |
| 0.8 | **0.7**772159969 | **0.726**0468515 | **0.726149**0026 | 0.7261490371 |
| 1.0 | **0.6**528614356 | **0.6069**751663 | **0.60653**13426 | 0.6065306597 |

## 6.4 Lineare Mehrschrittverfahren allgemein

Einschrittverfahren basieren auf der einfachen Rekursion $y_{n-1} \to y_n$ und verwenden keine Informationen aus den vorhergehenden Schritten. Um höhere Ordnungen zu erreichen, sind zusätzliche

Auswertungen der rechten Seite $f(t,y)$ in (6.1) notwendig. Das ist aber ineffizient, falls Funktionsauswertungen sehr aufwändig sind.

Ein **lineares Mehrschrittverfahren** zur Lösung eines Anfangswertproblems ist ein Verfahren der Form

$$\underbrace{\frac{1}{h}(\alpha_0 y_{n-k} + ... + \alpha_{k-1} y_{n-1} + \alpha_k y_n)}_{\approx y'} = \underbrace{\beta_0 f(t_{n-k}, y_{n-k}) + ... + \beta_{k-1} f(t_{n-1}, y_{n-1}) + \beta_k f(t_n, y_n)}_{\varphi(t_{n-k}, y_{n-k}, ..., t_n, y_n; h)}$$

Das ist eine $k$-Schritt Rekursion $y_{n-k}, ..., y_{n-1} \to y_n$. Dieses Verfahren heißt *linear*, weil die Terme $f(t_i, y_i)$ und $y_i$ auf der linken Seite nur linear vorkommen. Es gilt $\sum \alpha_i = 0$, $\alpha_k \neq 0$ und $\varphi(t_{n-k}, y_{n-k}, ..., t_n, y_n; h)$ ist die Inkrementfunktion. Für dieses $k$-**Schrittverfahren** ist nur eine weitere Funktionsauswertung pro Schritt erforderlich. Es werden aber $k-1$ Startwerte benötigt, die üblicherweise durch Einschrittverfahren passender Ordnung berechnet werden. Für $\beta_k = 0$ heißt das Verfahren **explizit**, für $\beta_k \neq 0$ **implizit**. Einfache Beispiele für Mehrschrittverfahren sind das

$$\text{explizite Euler-Verfahren:} \quad \frac{y_n - y_{n-1}}{h} = f(t_{n-1}, y_{n-1}) \quad \text{und das} \tag{6.13}$$

$$\text{implizite Euler-Verfahren:} \quad \frac{y_n - y_{n-1}}{h} = f(t_n, y_n)\,. \tag{6.14}$$

Für $\alpha_k = 1$, $\alpha_{k-1} = -1$ und alle übrigen $\alpha_i = 0$ erhält man eine wichtige Spezialklasse von linearen Mehrschrittverfahren, die sogenannten **Adamsverfahren**

$$\frac{y_n - y_{n-1}}{h} = \beta_0 f(y_{n-k}, t_{n-k}) + \cdots + \beta_{k-1} f(y_{n-1}, t_{n-1}) + \beta_k f(y_n, t_n).$$

Explizite **Adams-Bashford-Verfahren** haben die Konsistenzordnung $p = k$:

| $k$ | $\beta_0$ | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\beta_4$ | $p$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | | | | 1 |
| 2 | $-\frac{1}{2}$ | $\frac{3}{2}$ | 0 | | | 2 |
| 3 | $\frac{5}{12}$ | $-\frac{16}{12}$ | $\frac{23}{12}$ | 0 | | 3 |
| 4 | $-\frac{9}{24}$ | $\frac{37}{24}$ | $-\frac{59}{24}$ | $\frac{55}{24}$ | 0 | 4 |

Implizite **Adams-Moulton-Verfahren** haben die Konsistenzordnung $p = k+1$:

| $k$ | $\beta_0$ | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\beta_4$ | $p$ |
|---|---|---|---|---|---|---|
| 1 | $\frac{1}{2}$ | $\frac{1}{2}$ | | | | 2 |
| 2 | $-\frac{1}{12}$ | $\frac{8}{12}$ | $\frac{5}{12}$ | | | 3 |
| 3 | $\frac{1}{24}$ | $-\frac{5}{24}$ | $\frac{19}{24}$ | $\frac{9}{24}$ | | 4 |
| 4 | $-\frac{19}{720}$ | $\frac{106}{720}$ | $-\frac{264}{720}$ | $\frac{646}{720}$ | $\frac{251}{720}$ | 5 |

**Beispiel 6.4.1.** *Gegeben ist die lineare, skalare inhomogene Differentialgleichung*

$$\begin{aligned} y'(t) &= y(t) + t \qquad t \in [0,1], \\ y(0) &= 1 \end{aligned}$$

*mit exakter Lösung* $y(t) = 2e^t - (t+1)$.
*Wir verwenden die Schrittweite* $h = 0.2$ *und berechnen* 5 *Schritte mit dem impliziten Adams-Verfahren der Ordnung 2 (*$k = 1$*) sowie* 4 *Schritte mit dem expliziten Adams-Verfahren der Ordnung 2 (*$k = 2$*), wobei wir bei letzterem als zweiten Anfangswert den Wert* $y(0.2)$ *der exakten Lösung heranziehen.*

| $t_n$ | implizites Adams-Verfahren | explizites Adams-Verfahren | exakte Lösung |
|---|---|---|---|
| 0.0 | **1.0000000000** | **1.0000000000** | 1.0000000000 |
| 0.2 | **1.24**44444445 | **1.2428055163** | 1.2428055163 |
| 0.4 | **1.58**76543209 | **1.57**56471712 | 1.5836493952 |
| 0.6 | **2.0**515775033 | **2.0**240607709 | 2.0442376007 |
| 0.8 | **2.6**630391707 | **2.6**137142851 | 2.6510818569 |
| 1.0 | **3.4**548256531 | **3.3**754224935 | 3.4365636569 |

Lässt sich die Gleichung für $y_{n+1}$ des impliziten Verfahrens nicht explizit machen, dann kann man sie durch ein Iterationsverfahren lösen. Für einen relativ genauen Startwert $y_{n+1}^{(0)}$ wird bereits eine Iteration zu einer ziemlich exakten Lösung $y_{n+1}$ der impliziten Gleichung führen. Den Startwert $y_{n+1}^{(0)}$ bestimmt man mit einem expliziten Verfahren. Das ergibt z.B.

$$\begin{aligned} y_{n+1}^{(0)} &= y_n + \frac{h}{2}(3(f(t_n, y_n) - f(t_{n-1}y_{n-1})), \\ y_{n+1} &= y_n + \frac{h}{12}(5f(t_{n+1}, y_{n+1}^{(0)}) + 8f(t_n, y_n) - f(t_{n-1}, y_{n-1})). \end{aligned}$$

Die erste Gleichung heißt Prediktor, die zweite Korrektor, das Verfahren heißt **Prediktor-Korrektor-Verfahren**. (Aus den beiden Werten $y_{n+1}^{(0)}$ und $y_{n+1}$ kann man den lokalen Fehler abschätzen.)

Eine wichtige Klasse von implziten Mehrschrittverfahren zur numerischen Lösung von steifen Problemen sind die **Backward Differentiation Formulas (BDF)**

$$\frac{1}{h}\sum_{j=0}^{k} \alpha_j y_{n-k+j} = f(t_n, y_n)\,,$$

mit $\alpha_i$, $i = 1, 2, ..., k$ entsprechend der folgenden Tabelle:

| $k$ | $\alpha_0$ | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | $\alpha_4$ | $\alpha_5$ | $\alpha_6$ |
|---|---|---|---|---|---|---|---|
| 1 | $-1$ | $1$ | | | | | |
| 2 | $\frac{1}{2}$ | $-2$ | $\frac{3}{2}$ | | | | |
| 3 | $-\frac{1}{3}$ | $\frac{3}{2}$ | $-3$ | $\frac{11}{6}$ | | | |
| 4 | $\frac{1}{4}$ | $-\frac{4}{3}$ | $3$ | $-4$ | $\frac{25}{12}$ | | |
| 5 | $-\frac{1}{5}$ | $\frac{5}{4}$ | $-\frac{10}{3}$ | $5$ | $-5$ | $\frac{137}{60}$ | |
| 6 | $\frac{1}{6}$ | $-\frac{6}{5}$ | $\frac{15}{4}$ | $-\frac{20}{3}$ | $\frac{15}{2}$ | $-6$ | $\frac{147}{60}$ |

Die Konsistenz, also die Ordnung des lokalen Diskretisierungfehlers $l_n$, ist bei Mehrschrittverfahren im Gegensatz zu Einschrittverfahren im Allgemeinen relativ einfach zu studieren (Theorie von Differenzengleichungen).

# Index