# Numerik

Vorlesungsskript
Studiengang Technische Informatik
(Bachelor, 3.Semester)
Hochschule Albstadt-Sigmaringen

Prof.Dr. Andreas Knoblauch

1. Oktober 2016

# Inhaltsverzeichnis

# Kapitel 1

# Rechnerarithmetik

## 1.1 Gleitpunktzahlen

Der Zweck von Gleitpunktzahlen ist die effiziente Darstellung von reellen Zahlen $\mathbb{R}$ in einem digitalen Computer.

**Definition 1.1: Gleitpunktzahlen**

Die Gleitpunktzahlendarstellung zerlegt jede reelle Zahl $r \in \mathbb{R}$ in drei Bestandteile, [1]

I) ein Vorzeichen $(-1)^v$ mit $v \in \{0, 1\}$,

II) eine Mantisse $m \in \mathbb{R}$ und

III) einen Exponenten $e \in \mathbb{Z}$,

sodass

$$r = (-1)^v \cdot m \cdot 2^e \tag{1.1}$$

**Bemerkung:** Die Darstellung nach (1.1) legt die Mantisse und den Exponenten nicht eindeutig fest, z.B. $16 = (-1)^0 \cdot 0.5 \cdot 2^5 = (-1)^0 \cdot 1 \cdot 2^4 = (-1)^0 \cdot 2 \cdot 2^3 = \ldots$. Für eine eindeutige Darstellung kann man (1.1) normalisieren: $e$ wird so gewählt, dass $m$ genau eine von Null verschiedene Stelle (unmittelbar) vor dem Komma hat.

**Definition 1.2: Normalisierte Gleitpunktzahlen**

Genügt die reelle Zahl $r \in \mathbb{R}$ der Darstellung

2

$$r = (-1)^v \sum_{i=0}^{t-1} r_i \cdot 2^{-i} \cdot 2^e \, mit \, r_0 = 1 \tag{1.2}$$

so gehört die Zahl $r$ zur Menge der normierten Gleitpunktzahlen $\mathbb{M}_N$.
Das Vorzeichen $(-1)^v$ wird durch das Vorzeichen-Bit $v \in \{0,1\}$, der Wert der Mantisse $m$ durch die binären Ziffern $r_i \in \{0,1\}$, $i = 1, \ldots, t-1$, und der Wert des Exponenten $e$ durch die ganze Zahl $e \in \mathbb{Z}$, $e_{\min} < e < e_{\max}$ festgelegt.

**Bemerkungen:**

I) $\sum_{i=0}^{t-1} r_i \cdot 2^{-i} = (1.r_1 r_2 \ldots r_{t-1})_2$ entspricht der Binärdarstellung der Mantisse. Hierbei ist die Ziffer vor dem Binärpunkt "." stets $r_0 = 1$. Die Mantisse hat genau $t$ Ziffern (inklusive $r_0$).

II) Die Fälle $e = e_{\min}$ und $e = e_{\max}$ definieren spezielle "Sonderwerte", dazu gleich mehr ...

III) Eine Gleitpunktzahl wird i.d.R. als eine Zeichenkette von Bits $s := [v, e, m]$ abgespeichert. Auf diese Weise ist (in Reihenfolge von links nach rechts) ein effizienter Vergleich zweier Gleitpunktzahlen möglich.

IV) Der begrenzte Speicher erzwingt ein endliches Speicherformat, z.B. durch Festlegung der Anzahl von Stellen für die Binärzahlendarstellung von Mantisse und Exponent.
**Beispiel**: IEEE-Standardformate für Gleitpunktzahlen:

- "Single format": $m$ mit 23 Bit und $e$ mit 8 Bit, sodass sich eine Gesamtgröße von 32 Bit ergibt (inkl. Vorzeichen-Bit)
- "Double format": $m$ mit 52 Bit und $e$ mit 11 Bit , Gesamtgröße 64 Bit

V) Die darstellbaren Gleitpunktzahlen sind nicht äquidistant in ihrem Zahlenbereich verteilt. Zum Beispiel haben zwei benachbarte Gleitpunktzahlen mit gleichem Exponent $e = 1$ den doppelten Abstand wie zwei andere benachbarte Gleitpunktzahlen mit $e = 0$. Die folgende Skizze stellt alle Gleitpunktzahlen für Mantissen mit $t = 4$ bit und Exponenten mit 3 bit dar.

also $v = 0$ (da die Zahl positiv ist), $m = (1.1011001\ldots)_2$ und $e = 3$.

**Beispiel 1.3: Umwandlung der Dezimalzahl 13.6 in das Format (1.2)**

- Zuerst machen wir die Umwandlung von 13.6 ins Binärformat indem wir die Zahl als Summe von Zweierpotenzen schreiben. Dabei ziehen wir in jedem Schritt aus der Zahl (oder ihrem Rest) die jeweils größte Zweierpotenz heraus:

$$\begin{array}{rcl} 13.6 & = & 2^3 + \\ 5.6 & = & 2^3 + 2^2 + 1.6 \\ & = & 2^3 + 2^2 + 2^0 + 0.6 \\ & = & 2^3 + 2^2 + 2^0 + 2^{-1} + 0.1 \\ & = & 2^3 + 2^2 + 2^0 + 2^{-1} + 2^{-4} + 0.0375 \\ & = & \ldots \end{array}$$

Also können wir die Dezimalzahl $(13.6)_{10}$ als Binärzahl $(1101.1001\ldots)_2$ darstellen.

3

- Nun führen wir die Normalisierung durch, indem wir den Exponenten $e$ so wählen, dass für die Mantisse genau eine Stelle vor dem (Binär-) Punkt erforderlich ist:

$$(13.6)_{10} \quad = \quad (1101.1001\ldots)_2 = (-1)^0 \cdot (1.1011001\ldots)_2 \cdot 2^3 \ ,$$

also $v = 0$ (da die Zahl positiv ist), $m = (1.1011001\ldots)_2$ und $e = 3$.

4

**Darstellung der Null:** Um die Zahl $r = 0$ als Gleitpunktzahl im Format (1.1) darzustellen muss die Mantisse $m = 0$ sein. Da für normierte Gleitpunktzahlen im Format (1.2) die Mantisse immer von Null verschieden ist (da $r_0 = 1$), erfordert die Zahl $r = 0$ ein gesondertes Format:

**Definition 1.4: Subnormale Gleitpunktzahlen, $e = e_{\min}$, Darstellung der Null**

5 Eine reelle Zahl $r \in \mathbb{R}$ mit der Darstellung

$$r = (-1)^v \cdot \sum_{i=0}^{t-1} r_i \cdot 2^{-i} \cdot 2^e \text{ mit } r_0 = 0 \tag{1.3}$$

(3)

gehört der Menge der nicht-normierten oder subnormalen Gleitpunktzahlen $\mathbb{M}_S$ an.

6 Es gilt analog zu Definition 1.2: Das Vorzeichen $(-1)^v$ wird durch das Vorzeichen-Bit $v \in \{0, 1\}$, der Wert der Mantisse $m$ durch die Ziffern $r_i \in \{0, 1\}$, $i = 1, \ldots, t-1$ und der Wert des Exponenten durch eine ganze Zahl $e \in \mathbb{Z}$, $e = e_{\min}$ festgelegt.

**Bemerkungen:**

- Die Zahl Null ist eine spezielle subnormale Gleitpunktzahl und wird einheitlich – aber nicht eindeutig – durch

7 $$0 = (-1)^v \cdot 0 \cdot 2^{e_{\min}} \tag{1.4}$$

  repräsentiert, d.h., mit $r_i = 0$ für $i = 0, \ldots, t-1$ und $e = e_{\min}$.
- Zahlen mit maximalem Exponent $e = e_{\max}$ sind für die Darstellung von Sonderwerten reserviert:

**Definition 1.5: Sonderwerte Unendlich $\infty$ und Not-a-Number, NaN, $e = e_{\max}$**

8

- Die Belegung der Mantissen-Bits $r_i = 0$ für $i = 0, \ldots, t-1$ und der Exponent $e = e_{\max}$ identifizieren die Werte $(-1)^v \infty$.
- Alle restlichen Kombinationen, bei denen mindestens ein Bit $r_i \neq 0$ ist für ein $i \in \{1, \ldots, t-1\}$ und $e = e_{\max}$ gilt, sind für Sonderwerte vorgesehen: Not-a-Number, kurz NaN.

**Bemerkungen:** (Rechnen mit Sonderwerten $\infty$ und NaN)

I) Ein Überlauf des Bereichs der normalisierten Gleitpunktzahlen führt zum Ergebnis $\infty$.

II) Für eine positive, normalisierte Gleitpunktzahl $r > 0$ gilt:
$0 + r = r$, $0 \cdot r = 0$, $r/0 = \infty$, $0/0 =$NaN
$\infty + r = \infty$, $\infty \cdot r = \infty$, $r/\infty = 0$, $\infty \cdot 0 =$NaN
$\infty \cdot \infty = \infty$, $\infty + \infty = \infty$, $\infty - \infty =$NaN, $\infty/\infty =$NaN.

## 1.2 Runden und Rundungsfehler

Für eine Gleitpunktzahl im Format (1.2) gilt im Hinblick auf Runden:

- Das Vorzeichen $(-1)^v$ wird durch $v = 0$ oder $v = 1$ exakt dargestellt.
- Solange der Exponent $e = \mathbb{Z}$ im Bereich $e_{\min} \leq e < e_{\max}$ liegt, sind keine Rundungsfehler zu beachten (für die Darstellung des Exponenten).
- Im Falle eines Exponenten-Überlaufs ($e \geq e_{\max}$) terminieren Programme i.d.R. mit Meldung eines Überlauffehlers.
- Im Falle eines Exponenten-Unterlaufs ($e < e_{\min}$) wird kommentarlos auf 0 abgerundet. D.h., Unterläufe sind i.d.R. unproblematisch.

Dies reduziert die Betrachtung auf die Mantisse:

**Definition 1.6: Standardrundung**

9

Die Mantisse $m = 1.r_1 r_2 \dots r_{t-1} | r_t \dots$ einer normierten Gleitpunktzahl wird in Abhängigkeit des letzen Mantissen-Bits $r_{t-1}$ und der ersten nicht mehr gespeicherten Stelle $r_t$ gemäß folgender Fallunterscheidung gerundet ($\mathrm{rd}(m)$):

I) Im Falle $r_t = 0$ wird abgerundet: $\mathrm{rd}(m) = 1.r_1 r_2 \dots r_{t-1}$.

II) Im Falle $r_t = 1$ werden folgende Möglichkeiten unterschieden:

a) Im Falle $m = 1.r_1 r_2 \dots r_{t-1} | 1000 \dots$ (d.h. $m$ liegt genau zwischen zwei normierten Gleitpunktzahlen) wird ...
(a0) ... für $r_{t-1} = 0$ abgerundet: $\mathrm{rd}(m) = 1.r_1 r_2 \dots r_{t-2} 0$
(a1) ... für $r_{t-1} = 1$ aufgerundet: $\mathrm{rd}(m) = 1.r_1 r_2 \dots r_{t-2} 1 + 2^{-(t-1)}$

b) In allen anderen Fällen mit $r_t = 1$ wird aufgerundet:
$\mathrm{rd}(m) = 1.r_1 r_2 \dots r_{t-1} + 2^{-(t-1)}$

Also, anschaulich ...

10

| | $1.r_1$ | ... | $r_{t-2}$ | $r_{t-1}$ | $r_t$ | $r_{t+1} \dots$ | Rundung |
|---|---|---|---|---|---|---|---|
| Fall I | 1.x | ... | x | x | 0 | x... | $\downarrow$ |
| Fall IIa0 | 1.x | ... | x | 0 | 1 | 000... | $\downarrow$ |
| Fall IIa1 | 1.x | ... | x | 1 | 1 | 000... | $\uparrow$ |
| Fall IIb | 1.x | ... | x | x | 1 | $<\neq 000 \dots >$ | $\uparrow$ |

Hierbei bedeutet $\downarrow$ abrunden (d.h. nach $r_{t-1}$ bei | abschneiden)
und $\uparrow$ bedeutet aufrunden (d.h. bei | abschneiden und $2^{-(t-1)}$ addieren).

**Beispiel 1.7: Runden von $(13.6)_{10}$ auf $t$ Stellen der Mantisse**

In Beispiel 1.3 haben wir $(13.6)_{10}$ als normalisierte Gleitpunktzahl $(-1)^0 \cdot (1.1011001\ldots)_2 \cdot 2^3$ dargestellt. Nun wollen wir die Mantisse $(1.1011001\ldots)_2$ auf $t$ Stellen runden (für verschiedene $t$):

| t | $r$ | Fall-Nr. | $\uparrow$ / $\downarrow$ | $m$ | Fehler $\lvert\delta_r\rvert$ |
|---|---|---|---|---|---|
| 2 | $(-1)^0 \cdot (1.1\lvert 011001\ldots)_2 \cdot 2^3$ | Fall I | ab | 1.1 | $(1.6)_{10}$ |
| 3 | $(-1)^0 \cdot (1.10\lvert 11001\ldots)_2 \cdot 2^3$ | Fall IIb | auf | 1.11 | $(0.4)_{10}$ |
| 4 | $(-1)^0 \cdot (1.101\lvert 1001\ldots)_2 \cdot 2^3$ | Fall IIb | auf | 1.110 | $(0.4)_{10}$ |
| 5 | $(-1)^0 \cdot (1.1011\lvert 001\ldots)_2 \cdot 2^3$ | Fall I | ab | 1.1011 | $(0.1)_{10}$ |
| 6 | $(-1)^0 \cdot (1.10110\lvert 01\ldots)_2 \cdot 2^3$ | Fall I | ab | 1.10110 | $(0.1)_{10}$ |
| 7 | $(-1)^0 \cdot (1.101100\lvert 1\ldots)_2 \cdot 2^3$ | Fall IIb | auf | 1.101101 | $(0.025)_{10}$ |

11

Wir möchten nun den maximalen (absoluten) Rundungsfehler $|\delta_r| := |\operatorname{rd}(r) - r|$ beim Runden einer beliebigen (normalisierten) Zahl $r = (-1)^v \cdot (1.r_1 r_2 \ldots r_{t-1} | r_t r_{t+1} \ldots) \cdot 2^e$ auf $t$ Mantissenstellen bestimmen:

- Beim Abrunden nach Fall I ist $r_t r_{t+1} \ldots = 0x\ldots$, d.h. der Fehler ist höchstens $(\sum_{k=t+1}^{\infty} 2^{-k}) \cdot 2^e$ (wenn nach der 0 alle ubekannten Stellen 1 wären). Mit der geometrischen Reihenformel (siehe z.B. Vorlesung Mathematik I) ist wegen

$$\sum_{k=t+1}^{\infty} 2^{-k} = \sum_{k=t+1}^{\infty} 0.5^k = \sum_{k'=0}^{\infty} 0.5^{k'+t+1} = 0.5^{t+1} \sum_{k'=0}^{\infty} 0.5^{k'} = 2^{-(t+1)} \frac{1}{1-0.5} = 2^{-t}$$

  der maximale Fehler also höchstens $2^{-t} \cdot 2^e = 2^{e-t}$.
- Im Fall IIa ist $r_t r_{t+1} \ldots = 100\ldots$ und deshalb der Fehler beim Auf- oder Abrunden genau $2^{-t} \cdot 2^e = 2^{e-t}$.
- Beim Aufrunden nach Fall IIb ist $r_t r_{t+1} \ldots = 1x\ldots$ und deshalb der Fehler höchstens $(\sum_{k=t+1}^{\infty} 2^{-k}) \cdot 2^e$ (wenn nach der 1 alle unbekannten Stellen 0 wären), also wie bei Fall I wieder höchstens $2^{e-t}$.

In jedem Fall ist der maximale absolute Rundungsfehler also höchstens $|\delta_r| \leq 2^{e-t}$.

Jedoch ist der absolute Rundungsfehler zur Beurteilung eines Fehlers nicht immer aussagekräftig genug! (Z.B. ist $\delta_r = 0.5$ für $r = 3$ viel, aber für $r = 150000$ wenig ...). Daher definiert man auch den relativen Rundungsfehler $\epsilon_r$:

**Definition 1.8: Absoluter und Relativer Rundungsfehler**

Der absolute Rundungsfehler $\delta_r \in \mathbb{R}$ ist die Differenz

12
$$f_{\mathrm{rd}}(r) := \delta_r = \mathrm{rd}(r) - r \ .$$

Der relative Rundungsfehler $\epsilon_r \in \mathbb{R}$ ist der Quotient

13
$$f_{\mathrm{rel}}(r) := \epsilon_r := \frac{f_{\mathrm{rd}}(r)}{r} = \frac{\delta_r}{r} = \frac{\mathrm{rd}(r) - r}{r} \quad \text{für } r \neq 0 \ .$$

**Bemerkungen:**

- Auflösen obiger Definition des relativen Fehlers $\epsilon_r$ nach $\mathrm{rd}(r)$ ergibt

14
$$\mathrm{rd}(r) = r(1 + \epsilon_r) \ . \tag{1.5}$$

- Für beliebige (normalisierte) Zahlen $r = (-1)^v \cdot (1.r_1 r_2 \dots r_{t-1} | r_t r_{t+1} \cdots) \cdot 2^e$ gilt $|r| \geq 2^e$. Deshalb ergibt sich mit der (bereits oben gezeigten) Abschätzung $|\delta_r| \leq 2^{e-t}$ folgende Abschätzung des relativen Rundungsfehlers:

15
$$|\epsilon_r| = |\frac{\delta_r}{r}| = \frac{|\delta_r|}{|r|} \leq \frac{2^{e-t}}{2^e} = 2^{-t}$$

Aus der letzen Bemerkung ergibt sich also folgendes Lemma:

**Lemma 1.9: Abschätzung der Rundungsfehler**

Für absolute und relative Rundungsfehler beim Runden einer Zahl $r = (-1)^v \cdot (1.r_1 r_2 \dots r_{t-1} | r_t r_{t+1} \dots) \cdot 2^e$ mit $t$-stelliger Mantisse gelten die folgenden oberen Schranken:

16
$$|\delta_r| \leq 2^{e-t} \qquad \text{und} \qquad |\epsilon_r| \leq 2^{-t} \ . \tag{1.6}$$

Die obere Schranke des relativen Rundungsfehlers $|\epsilon_r|$ ist eine wichtige numerische Größe und bekommt deshalb einen eigenen Namen:

**Definition 1.10: Maschinengenauigkeit**

Die obere Schranke für den maximalen relativen Rundungsfehler, der bei Rundung einer Zahl $r \in \mathbb{R}$ in eine Maschinenzahl mit $t$-stelliger Mantisse auftreten kann, heißt Maschinengenauigkeit $\epsilon$ und ergibt sich zu

$$\epsilon = 2^{-t} \tag{1.7}$$

17

**Bemerkungen:**

- Die Maschinengenauigkeit $\epsilon$ ist die größte positive Zahl für die gilt $1 +_{\mathbb{M}} \epsilon = 1$, d.h. die bei der Maschinenaddition $+_{\mathbb{M}}$ zu 1 gerade noch "weggerundet" wird.
- Die vorgestellte Rundungsart ist eine von vier möglichen, welche der IEEE-Standard 754 vorsieht.

## 1.3 Gleitpunktarithmetik und Fehlerfortpflanzung

Im folgenden beschäftigen wir uns mit der Realisierung der Grundrechenoperationen für Maschinenzahlen $\mathbb{M}$ und den dabei entstehenden Effekten.

**Definition 1.11: Maschinenoperationen**

Für die Grundrechenarten $+, -, \cdot, /$ bezeichnen $+_{\mathbb{M}}, -_{\mathbb{M}}, \cdot_{\mathbb{M}}, /_{\mathbb{M}}$ jeweils die entsprechenden Maschinenoperationen, d.h. die Anwendung der jeweiligen Operation auf Zahlen im normierten Gleitpunktzahlenformat gemäß Definition 1.2.

18

**Bespiele:** $200 + 5 = 205$ bezeichnet die übliche Addition auf $\mathbb{R}$. Hingegen bezeichnet $200 +_{\mathbb{M}} 5 =?$ die entsprechende Maschinenaddition. Das Ergebnis kann hierbei vom exakten Wert abweichen.

**Definition 1.12: Realisierung der Maschinenoperationen**

Das allgemeine Vorgehen bei der Realisierung der Maschinenoperationen ist wie folgt:

19

I) Verknüpfung der Maschinenzahlen mit höherer (ausreichend hoher) Genauigkeit

II) Runden des Ergebnisses auf eine Maschinenzahl

D.h. für Maschinenzahlen $r, s \in \mathbb{M}$ gilt für jede Operation $\circ \in \{+, -, \cdot, /\}$

20

$$r \circ_{\mathbb{M}} s := \mathrm{rd}(r \circ s).$$

$\Rightarrow$ D.h. ein Rechenfehler kann erst im Schritt II (Rundung) entstehen, wenn das exakte Ergebnis $r \circ s$ auf eine Maschinenzahl "eingestampft" wird.

Im folgenden betrachten wir beispielhaft den Algorithmus für die Maschinenaddition etwas genauer:

**Algorithmus 1.13: Maschinenaddition**

21

Die Addition zweier Maschinenzahlen $r$ und $s$ mit $t$-stelliger Mantisse wird wie folgt durchgeführt:

1) Die Darstellungen von $r$ und $s$ werden so angepaßt, dass beide denselben Exponenten haben.

2) Die resultierenden Mantissen werden in höherer Genauigkeit addiert.

3) Das Ergebnis wird normalisiert (also so lange verschoben, bis nur noch eine führende Eins vor dem Komma steht; siehe Def. 1.2).

4) Die resultierende Mantisse wird gemäß Def. 1.6 auf eine Maschinenzahl gerundet.

$\Rightarrow$ Erst im 4-ten Schritt des Algorithmus werden Rundungsfehler in das Ergebnis eingeschleppt.

**Beispiel 1.14: Addition zweier Maschinenzahlen**

Wir wollen für $t = 3$ die beiden Maschinenzahlen $r = \frac{7}{4} = (1.11)_2 \cdot 2^0$ und $s = \frac{3}{8} = (1.10)_2 \cdot 2^{-2}$ mit dem Algorithmus für Maschinenaddition addieren, 22

$$\begin{aligned} r +_{\mathbb{M}} s = (1.11)_2 \cdot 2^0 +_{\mathbb{M}} (1.10)_2 \cdot 2^{-2} &\overset{1.}{=} (111)_2 \cdot 2^{-2} + (1.10)_2 \cdot 2^{-2} \\ &\overset{2.}{=} (1000.10)_2 \cdot 2^{-2} \\ &\overset{3.}{=} (1.00010)_2 \cdot 2^1 \\ &\overset{4.}{=} (1.00)_2 \cdot 2^1 = (2)_{10}, \end{aligned}$$

wohingegen das exakte Ergebnis $r + s = \frac{7}{4} + \frac{3}{8} = \frac{17}{8}$ ist.
Der absolute Rundungsfehler ist also

$$f_{\text{rd}} = (r +_{\mathbb{M}} s) - (r + s) = 2 - \frac{17}{8} = -\frac{1}{8}$$

und der relative Rundungsfehler

$$|f_{\text{rel}}| = \left| \frac{(r +_{\mathbb{M}} s) - (r + s)}{r + s} \right| = \frac{1/8}{17/8} = \frac{1}{17} = 0.0588\ldots \approx 6\%.$$

Das ist relativ gut, denn der maximale Rundungsfehler bei einer 3-stelligen Mantisse ist $\epsilon = 2^{-3} = 12.5\%$.

Der relative Rundungsfehler $\epsilon_+ \in \mathbb{R}$, der im vierten Schritt eingeschleppt wird,

$$r +_{\mathbb{M}} s = \text{rd}(r + s) \overset{(1.5)}{=} (r + s)(1 + \epsilon_+) \tag{1.8}$$

kann nach (1.6) und (1.7) durch die Maschinengenauigkeit $\epsilon$ nach oben abgeschätzt werden:

$$|\epsilon_+| = |f_{\text{rel}}| \leq \epsilon\ . \tag{1.9}$$

Diese Analyse gilt unabhängig von der betrachteten Operation:

**Lemma 1.15: Relativer Fehler bei Gleitpunktoperationen**

23

Seien $r$ und $s$ zwei Gleitpunktzahlen und $\circ_{\mathbb{M}}$ die der Operation $\circ \in \{+, -, \cdot, /\}$ entsprechende Gleitpunkt- (oder Maschinen-) Operation.
Wegen Def. 1.12 liefert jede Gleitzahloperation $\circ$ als Ergebnis

$$r \circ_{\mathbb{M}} s = \mathrm{rd}(r \circ s) = (r \circ s) \cdot (1 + \epsilon_\circ)$$

wobei der relative Fehler $\epsilon_\circ$ der Gleitpunktoperation wegen Def. 1.10 stets durch die Maschinengenauigkeit beschränkt ist,

$$\epsilon_\circ := \frac{(r \circ_{\mathbb{M}} s) - (r \circ s)}{r \circ s} = \frac{\mathrm{rd}(r \circ s) - r \circ s}{r \circ s} \leq \epsilon.$$

### Fehlerfortpflanzung

**Ziel:** Analyse des Fehlers bei Folgen von Operationen.

**Beispiel:** Berechnung der Summe $y = a + b + c$ dreier Maschinenzahlen $a$, $b$, $c$.
Die Berechnung wird in zwei Teilschritten ausgeführt,

$$1)\ \tilde{x} = a +_{\mathbb{M}} b \qquad \text{und} \qquad 2)\ \tilde{y} = \tilde{x} +_{\mathbb{M}} c\ , \tag{1.10}$$

wobei $\epsilon_1$ und $\epsilon_2$ die relativen Fehler in Schritt 1 und Schritt zwei sind (mit $|\epsilon_1|, |\epsilon_2| \leq \epsilon$).

24

Es gilt:

$$\begin{aligned}
\tilde{y} &\overset{(1.10)}{=} \tilde{x} +_{\mathbb{M}} c \\
&\overset{(1.8)}{=} (\tilde{x} + c)(1 + \epsilon_2) \\
&\overset{(1.10)}{=} ((a +_{\mathbb{M}} b) + c)(1 + \epsilon_2) \\
&\overset{(1.8)}{=} ((a + b)(1 + \epsilon_1) + c)(1 + \epsilon_2) \\
&= a + b + c + (a + b)\epsilon_1 + (a + b + c)\epsilon_2 + (a + b)\epsilon_1\epsilon_2
\end{aligned}$$

Da $\epsilon_1 \cdot \epsilon_2$ sehr klein ist, kann der Term $(a+b)\epsilon_1\epsilon_2$ für die Rundungsfehleranalyse vernachlässigt werden. Somit gilt

$$\tilde{y} \doteq a + b + c + (a+b)\epsilon_1 + (a+b+c)\epsilon_2 \ . \tag{25}$$

Hierbei bedeutet der Punkt über dem Gleichheitszeichen "ist in erster Annäherung gleich" (d.h. "$\doteq$" zeigt an, dass Terme höherer Ordnung vernachlässigt wurden).
Damit gilt für den relativen Fehler

$$\begin{aligned} f_{\text{rel}}(y) &= \frac{\tilde{y} - y}{y} \\ &= \frac{(a+b+c+(a+b)\epsilon_1 + (a+b+c)\epsilon_2) - (a+b+c)}{a+b+c} \\ &= \frac{a+b}{a+b+c}\epsilon_1 + \epsilon_2 \ , \end{aligned} \tag{26}$$

und mit der Dreiecksungleichung folgt die Abschätzung

$$|f_{\text{rel}}(y)| = |\frac{a+b}{a+b+c}\epsilon_1 + \epsilon_2| \leq |\frac{a+b}{a+b+c}\epsilon_1| + |\epsilon_2| = |\frac{a+b}{a+b+c}| \cdot |\epsilon_1| + |\epsilon_2| \tag{27}$$

Wegen $\epsilon_1 \leq \epsilon$ und $\epsilon_2 \leq \epsilon$ (Lemma. 1.15) gilt also folgendes Lemma:

**Lemma 1.16: Relativer Fehler bei drei Summanden**

Für den relativen Fehler $f_{\text{rel}}(y)$ bei der Berechnung der Summe $y = a + b + c$ durch die Maschinenaddition $\tilde{y} = (a +_{\mathbb{M}} b) +_{\mathbb{M}} c$ mit Maschinengenauigkeit $\epsilon$ gilt die Abschätzung

$$|f_{\text{rel}}(y)| \ \leq \ (1 + |\frac{a+b}{a+b+c}|) \cdot \epsilon = (1 + \frac{1}{|1 + \frac{c}{a+b}|}) \cdot \epsilon \tag{28}$$

**Folgerungen aus Lemma 1.16:**

- Der relative Fehler liegt i.A. in der Größenordnung der Maschinengenauigkeit $\epsilon$ (falls $|\frac{a+b}{a+b+c}|$ nicht zu groß ist).
- Der relative Fehler kann aber auch viel größer als die Maschinengenauigkeit werden falls

29

$$c \approx -(a+b)$$

ist. Denn für $c \to -(a+b)$ ist $|a+b|/|a+b+c| \to \infty$ und damit wird auch die obere Schranke von $|f_{\text{rel}}(y)|$ beliebig (unendlich) groß. In diesem Fall spricht man von Auslöschung.

- Offensichtlich spielt die Reihenfolge der Operator-Anwendung eine wesentliche Rolle! Die Maschinenaddition $+_{\mathbb{M}}$ ist also (im Gegensatz zu $+$) nicht assoziativ (aber immerhin kommutativ)!

Welches ist für die die Addition $y = u + v + w$ die beste Reihenfolge die Summanden $u$, $v$, $w$ zu addieren?
Es gibt drei Möglichkeiten, wobei nach Lemma 1.16 jeweils drei unterschiedliche obere Schranken $f_{\text{rel}}(y)$ für den relativen Fehler resultieren:

30

$$\begin{aligned}
&1)\ (u+v)+w \quad &\Rightarrow \quad &|f_{\text{rel}}(y)| \le (1+|\frac{u+v}{u+v+w}|)\cdot\epsilon \\
&2)\ (u+w)+v \quad &\Rightarrow \quad &|f_{\text{rel}}(y)| \le (1+|\frac{u+w}{u+v+w}|)\cdot\epsilon \\
&3)\ (v+w)+u \quad &\Rightarrow \quad &|f_{\text{rel}}(y)| \le (1+|\frac{v+w}{u+v+w}|)\cdot\epsilon
\end{aligned}$$

Offensichtlich erhält man die kleinste Fehler-Schranke falls man zuerst diejenigen Zahlen addiert deren Summe betragsmässig am kleinsten ist.

**Beispiel 1.17: Optimale Addition dreier Maschinenzahlen**

Wir wollen drei Zahlen $u = 10$, $v = 1$, $w = -11.1$ addieren, d.h. $y := u + v + w = -0.1$. 31
Es gilt $u + v = 11$, $u + w = -1.1$ und $v + w = -10.1$. Die Summe $u + w$ ist also betragsmässig am kleinsten und deshalb ist die Reihenfolge $(u +_{\mathbb{M}} w) +_{\mathbb{M}} v$ optimal. Dafür erhält man eine minimale Fehlerschranke von

$$|f_{\text{rel}}(y)| \leq (1 + 1.1/0.1)\epsilon = 12\epsilon \qquad \text{für} \qquad y = (u +_{\mathbb{M}} w) +_{\mathbb{M}} v$$

Hätte man eine der beiden anderen Reihenfolgen zu addieren gewählt, so müsste man mit viel (fast zehnmal) größeren Fehlerschranken leben:

$$|f_{\text{rel}}(y)| \leq (1 + 11/0.1)\epsilon = 111\epsilon \qquad \text{für} \qquad y = (u +_{\mathbb{M}} v) +_{\mathbb{M}} w$$
$$|f_{\text{rel}}(y)| \leq (1 + 10.1/0.1)\epsilon = 102\epsilon \qquad \text{für} \qquad y = (v +_{\mathbb{M}} w) +_{\mathbb{M}} u$$

**Bemerkung:** Die Unterschiede der Fehlerschranken können beliebig groß werden, falls z.B. $v \ll u$ und $w \approx -(u + v)$.
Man erhält z.B. noch extremere Beispiele für $u = 100$, $v = 1$, $w = -101.1$; oder $u = 1000$, $v = 1$, $w = -1001.1$, etc.

## Auslöschung

Die eben bei der Addition dreier Zahlen beobachtete Auslöschung tritt allgemein immer dann auf, wenn ungefähr gleich große bereits mit Fehlern behaftete Zahlen voneinander abgezogen werden.
In diesem Fall können signifikante Mantissenstellen “ausgelöscht” werden.

**Beispiel 1.18: Auslöschung**

32

Die zwei reellen Zahlen $r = \frac{3}{5}$ und $s = \frac{4}{7}$ mit normierten gerundeten Maschinendarstellung $(\underline{1.0011})_2 \cdot 2^{-1}$ und $(\underline{1.0010})_2 \cdot 2^{-1}$ mit 5-stelliger Mantisse haben die Differenz

$$r - s = \frac{3}{5} - \frac{4}{7} = \frac{3 \cdot 7 - 4 \cdot 5}{35} = \frac{1}{35}.$$

Ausführung der Maschinenoperation mit den Gleitpunktzahlen liefert hingegen

$$\begin{aligned}(1.0011)_2 \cdot 2^{-1} - (1.0010)_2 \cdot 2^{-1} &= (0.0001)_2 \cdot 2^{-1} \\ &= (1.0000)_2 \cdot 2^{-5}\end{aligned}$$

Das Ergebnis $2^{-5} = 1/32$ ist behaftet mit einem relativen Fehler

$$\frac{1/35 - 1/32}{1/35} = -0.09375,$$

also ca. 9.4% (und somit etwas das dreifache der Maschinengenauigkeit $\epsilon = 0.031 = 3.1\%$).

**Analyse des "Auslöschungseffekts:"**
Der relative Fehler der Differenz $y = a - b$ zweier Zahlen $a$ und $b$ mit kleinen relativen Fehlern $\epsilon_a$ und $\epsilon_b$ ergibt sich nach Lemma 1.15 und (1.5) zu

$$\begin{aligned}\epsilon_y &= \frac{a - b - (a(1+\epsilon_a) - b(1+\epsilon_b))(1+\epsilon_-)}{a-b} \\ &= \frac{a - b - (a + a\epsilon_a - b - b\epsilon_b)(1+\epsilon_-)}{a-b} \\ &= \frac{a - b - a - a\epsilon_a + b + b\epsilon_b - a\epsilon_- - a\epsilon_a\epsilon_- + b\epsilon_- + b\epsilon_b\epsilon_-}{a-b} \\ &= \frac{-a\epsilon_a + b\epsilon_b - (a-b)\epsilon_- - a\epsilon_a\epsilon_- + b\epsilon_b\epsilon_-}{a-b} \\ &\doteq -\frac{a}{a-b}\epsilon_a + \frac{b}{a-b}\epsilon_b - \epsilon_- \qquad (1.11)\end{aligned}$$

Daraus ergibt sich folgendes Lemma:

**Lemma 1.19: Auslöschung bei Subtraktion**

Beim Berechnen der Differenz $a - b$ wird es nach (1.11) eine extreme Fehlerverstärkung geben, falls $|a|$ oder $|b|$ sehr groß ist verglichen mit $|a - b|$, d.h. falls

$$|a - b| \ll \min\{|a|, |b|\} .$$

33

⇒ Also müssen Differenzen annähernd gleichgroßer, fehlerbehafteter Zahlen vermieden werden (etwa durch Anwendung des Assoziativ- oder Kommutativ-Gesetzes; vgl. Beispiel 1.17).

**Beispiel 1.20: Numerische Berechnung der Exponentialfunktion**

Reihenentwicklung (siehe Analysis-Vorlesung):

$$\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \dots$$

Funktion zur Berechnung von $\exp(x)$ in Pseudo-Code:

```
FUNCTION e_hoch(INT x) {
   z:=0.0; y:=1.0; k:=1
   WHILE (z != (z+(x*y/k)) DO {
      z := z+y;
      y := x*y/k;
      k := k+1
   }
   RETURN z;
}
```

Beobachtung: Die Funktion e_hoch($x$) liefert für positive Argumente $x$ gute Er-gebnisse, jedoch völlig falsche Ergebnisse für negative Argumente.
Grund: Summanden haben bei negativen Argumenten alternierende Vorzeichen.
⇒ Es folgt Auslöschung!

34

## Einschub: Taylor-Reihe

Wir wiederholen die Definition der Taylor-Reihe aus der Mathematik-Vorlesung des 1.Semesters:

**Definition 1.21: Taylor-Reihe**

Sei $I \subseteq \mathbb{R}$ ein Intervall und $f : I \to \mathbb{R}$ eine beliebig oft stetig differenzierbare Funktion. Dann gilt für $x_0 \in I$ und $x \in I$:

35

$$f(x) = \sum_{j=0}^{\infty} \frac{f^{(j)}(x_0)}{j!} \cdot (x - x_0)^j$$
$$= f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 + \frac{f'''(x_0)}{6}(x - x_0)^3 + \dots$$

Diese Darstellung von $f$ heißt Taylor-Reihe(nentwicklung) von $f$ in der Stelle $x_0$.

36

**Beispiel:** Taylor-Reihe der Exponential-Funktion an der Stelle $x_0 = 0$:
Für $f(x) := \exp(x)$ gilt $f^{(j)}(x) = e^x$ für alle $j = 0, 1, 2, 3, \dots$ und somit $f^{(j)}(x_0) = f^{(j)}(0) = e^0 = 1$ für alle $j = 0, 1, 2, 3, \dots$.
Somit lautet die Taylorreihe von $e^x$ um $x_0 = 0$

$$e^x = \sum_{j=0}^{\infty} \frac{1}{j!} \cdot (x - 0)^j = \sum_{j=0}^{\infty} \frac{x^j}{j!}$$

**Bemerkungen:**

- $\sum_{j=0}^{k} \frac{f^{(j)}(x_0)}{j!} \cdot (x - x_0)^j$ heißt Taylor-Polynom $k$-ten Grades.
  $\sum_{j=k+1}^{\infty} \frac{f^{(j)}(x_0)}{j!} \cdot (x - x_0)^j$ heißt Taylor-Restglied.
- Falls eine Potenz-Reihe für alle $x$ mit $|x - x_0| < R$ konvergiert sagt man, die Reihe hat Konvergenzradius $R$. Der Konvergenzradius einer Taylor-Reihe ist nicht notwendig größer 0.

- Viele numerische Verfahren beruhen darauf, eine gegebene "hinreichend freundliche" Funktion $f$ durch ein Taylor-Polynom kleinen Grades (z.B. $k = 1$ oder $k = 2$) zu ersetzen und das gestellte Problem (Integration, Nullstellenbestimmung, etc.) für diese wesentlich einfachere Funktion zu lösen.
  Der auftretende Fehler kann durch die Größe des Taylor-Restglieds abgeschätzt werden.

## 1.4 Kondition und Stabilität

Ein numerisches Berechnungsverfahren für ein Problem $f$ ermittelt in einer Folge von Berechnungen (z.B. Additionen, ...) aus den Eingangsdaten $x \in \mathbb{R}^n$, $n \in \mathbb{N}^*$ eine Näherung für das Ergebnis $y = f(x) \in \mathbb{R}$.

Hierbei sind die Eingangsdaten $x_i$ i.d.R. mit einem (absoluten) Rundungsfehler $\delta_{x_i}$ behaftet.

Eine zentrale Frage der Numerik lautet: In welcher Weise können sich diese Eingabefehler im Laufe des Berechnungsverfahrens verstärken?

37

Den quantitativen Zusammenhang zwischen dem absoluten Fehler $\delta_y$ im Ergebnis $y$ und dem absoluten Fehler $\delta_x$ in der Eingabe $x$ ergibt sich für hinreichend glatte Funktionen $f$ aus der Taylor-Reihenentwicklung

$$y + \delta_y = f(x + \delta_x) = f(x) + f'(x)\delta_x + o(\delta_x^2)$$

Unter Vernachlässigung der Terme höherer Ordnung gilt also:

$$\delta_y \doteq f'(x) \cdot \delta_x$$

Die Verstärkung des relativen Rundungsfehlers $\epsilon_y$ des Resultats $y$ ergibt sich dann für $x, y \neq 0$

$$\begin{aligned} \epsilon_y = f_{\text{rel}}(y) = \frac{\delta_y}{y} &\doteq \frac{f'(x) \cdot \delta_x}{y} = \frac{x \cdot f'(x)}{y} \cdot \frac{\delta_x}{x} \\ &= \frac{x \cdot f'(x)}{y} \cdot f_{\text{rel}}(x) = \frac{x \cdot f'(x)}{y} \epsilon_x \end{aligned} \tag{8}$$

aus dem relativen Rundungsfehler $\epsilon_x$ der Eingabe $x$. (1.12)

**Definition 1.22: Kondition**

38

Die Kondition der Funktion $y = f(x)$ ergibt sich aus dem Verstärkungsfaktor

$$K_f(x) := \left| \frac{x \cdot f'(x)}{f(x)} \right| \tag{1.13}$$

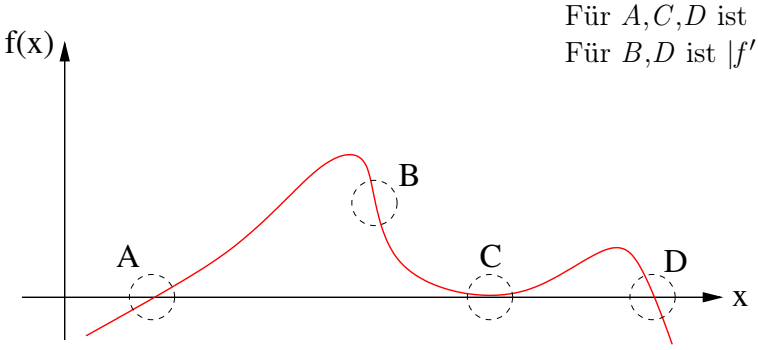
zwischen den relativen Rundungsfehlern aus (1.12).

**Bemerkungen:**

- Die Kondition ist eine Maßzahl für die Sensitivität des Resultats $y$ in Bezug auf die Eingabe $x$.
- Wir sprechen von einer bzgl. $x$ gut konditionierten Aufgabenstellung $f(x)$, wenn kleine relative Eingabefehler $\epsilon_x$ bei exakter Arithmetik (also ohne Einfluß von Rechenfehlern) zu kleinen relativen Fehlern $\epsilon_y$ im Resultat führen.
- Die Kondition ist eine Funktion der Eingabe $x$ und nach (1.13) eine Eigenschaft der Aufgabenstellung $f$ und nicht des Berechnungsverfahrens.

**Beispiel:** Die Kreise kennzeichnen Bereiche mit schlechter Kondition (d.h. großer Konditionszahl $K_f$):

Für $A$,$C$,$D$ ist $|x/f(x)| \gg 1$.
Für $B$,$D$ ist $|f'(x)| \gg 1$.



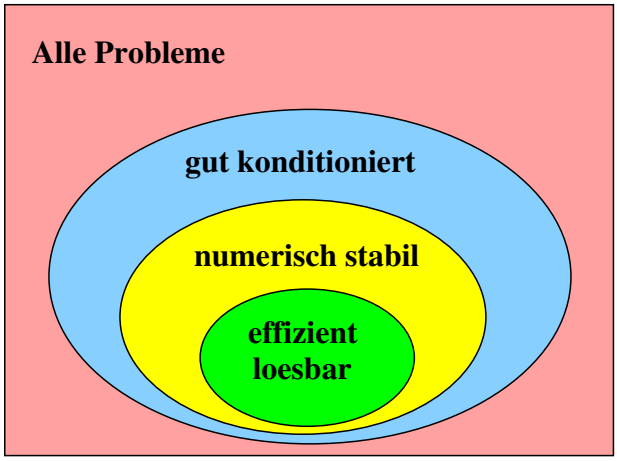
**Definition 1.23: Numerisch stabiles Verfahren**

Wir sprechen von einem numerisch stabilen Berechnungsverfahren, falls die relativen Eingabefehler eines gut konditionierten Problems $y = f(x)$ durch das Berechnungsvefahren nicht vergrößert werden.
Oder umgekehrt: Ein Verfahren, das trotz kleiner (d.h. guter) Kondition große relative Fehler im Ergebnis produziert, heißt numerisch instabil.

39

**Bemerkungen:**

- Während die Kondition $K_f$ eine Eigenschaft des zu lösenden numerischen Problems $f$ ist, ist Stabilität eine Eigenschaft des Verfahrens zur Lösung von $f$.
- Daraus abgeleitet nennt man ein numerisches Problem $f$ stabil, falls dafür ein stabiles Berechnungsverfahren existiert.
- Die folgende Illustration zeigt den Zusammenhang zwischen
  - gut konditionierter Aufgabenstellung
  - numerisch stabilem Berechnungsverfahren, und
  - effizientem Algorithmus:



# Kapitel 2

# Lösung linearer Gleichungssysteme

Lineare Gleichungen und Gleichungssysteme gehören zu den elementarsten – zugleich aber auch zu den wichtigsten und in der Praxis am häufigsten auftretenden numerischen Problemen.

- Es existieren zwei Klassen von Lösungsverfahren:
  1. Direkte Verfahren, die nach endlich vielen Operationen eine (evtl. mit Rundungsfehlern behaftete) Lösung liefern.
  2. Iterative Verfahren, die beginnend mit einer Anfangsnäherung durch eine Folge von Iterationsschritten eine verbesserte Näherungslösung generieren.
- Fragestellungen
  bei 1.: Benötigte Rechenoperationen, Rechengenauigkeit
  bei 2.: Konvergenzgeschwindigkeit, Kosten der Iterationsschritte
- In der Praxis oft Kombination aus 1. und 2.

Im folgenden: 1. (Direkte Verfahren). Wir betrachten zunächst eine Teilklasse von linearen Gleichungssystemen, die einfach zu lösen sind:

## 2.1 Auflösen von Dreiecksgleichungssystemen

Bei Dreiecksgleichungssystemen kommen nur bei der ersten Gleichung alle Unbekannten $x_1, x_2, \ldots x_n$ vor. Bei der zweiten Gleichung fehlt die erste Unbekannte $x_1$, bei der dritten Gleichung fehlen die ersten beiden Unbekannten $x_1, x_2$, und so fort, und die letze Gleichung hat nur noch eine einzige Unbekannte $x_n$. Deshalb lassen sich Dreiecksgleichungssysteme besonders einfach in einem Zug "von unten nach oben" lösen, wie folgendes Beispiel illustriert:

**Beispiel 2.1: Dreiecksgleichungssystem**

40 Gegeben sei folgendes lineares Gleichungssystem in drei Unbekannten $x_1$, $x_2$ und $x_3$:

$$\begin{array}{rcrcrcr} 10\,x_1 & - & 7\,x_2 & + & 0\,x_3 & = & 7 \\ & & 2.5\,x_2 & + & 5\,x_3 & = & 2.5 \\ & & & & 6.2\,x_3 & = & 6.2 \end{array}$$

... oder in Matrixschreibweise:

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 2.5 \\ 6.2 \end{pmatrix}$$

41 Wegen der Dreiecksform läßt sich dieses Gleichungssystem einfach von unten her auflösen:

$$\begin{aligned} x_3 &= \frac{6.2}{6.2} = 1 \\ x_2 &= \frac{2.5 - 5x_3}{2.5} = \frac{2.5 - 5 \cdot 1}{2.5} = \frac{-2.5}{2.5} = -1 \\ x_3 &= \frac{7 - 0x_3 + 7x_2}{10} = \frac{7 - 0 \cdot 1 + 7 \cdot (-1)}{10} = \frac{0}{10} = 0 \end{aligned}$$

Somit resultiert der Lösungsvektor $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$.

Allgemeiner Algorithmus zum Lösen von linearen Gleichungssystemen, die durch eine obere $n \times n$ Dreiecksmatrix

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2n} \\ & & \ddots & \vdots \\ 0 & & & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

gegeben sind:

42

I) Berechnung von $x_n = \frac{b_n}{a_{nn}}$.

II) Schrittweise Berechnung von

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} \cdot x_j}{a_{ii}}$$

für $i = n-1, n-2, \ldots, 3, 2, 1$.

**Bemerkung:** Die Elemente $a_{ii}$, $i = 1, 2, \ldots, n$ bilden die Hauptdiagonale der Matrix. Ist einer der Hauptdiagonaleneinträge $a_{ii} = 0$, so ist das Gleichungssystem nicht eindeutig lösbar.

Im nächsten Schritt wird versucht, das Lösen von allgemeinen quadratischen linearen Gleichungssystemen auf das Lösen von Dreiecksgleichungssystemen zurückzuführen.

## 2.2 Die Gauß - Elimination

Wir wollen nun allgemeine lineare Gleichungssysteme der folgenden Form lösen:

$$\begin{array}{ccccccccc}
a_{11}\,x_1 & + & a_{12}\,x_2 & + & \ldots & + & a_{1n} & = & b_1 \\
a_{21}\,x_1 & + & a_{22}\,x_2 & + & \ldots & + & a_{2n} & = & b_2 \\
\vdots & & & & & & & & \\
a_{n1}\,x_1 & + & a_{n2}\,x_2 & + & \ldots & + & a_{nn} & = & b_n
\end{array}$$

... oder in Matrixschreibweise:

43

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

**Lösungsstrategie der Gauß - Elimination:** Wir gehen schrittweise zu äquivalenten, einfacheren Gleichungssystemen über, die jeweils genau die gleiche Lösung besitzen. Ziel ist dabei, das Gleichungssystem bzw. die Matrix auf Dreiecksgestalt zu bringen.

**Erlaubte Umformungen, die die Lösung nicht verändern:**

1) Multiplizieren einer Zeile mit einer Zahl verschieden von Null.
2) Addieren eines Vielfachen einer Zeile zu einer anderen Zeile.
3) Vertauschen von Zeilen bzw. Spalten.

Die letzte Umformung ist offensichtlich eine Äquivalenzumformung, da die jeweiligen Aussagen der Gleichungen sich durch die Vertauschungen nicht ändern. Die ersten beiden Umformungen sind auch Äquivalenzumformungen, denn für $a_i(x) := a_{i1}x_1 + \ldots + a_{in}x_n$ gilt für $i, j \in \{1, 2, \ldots, n\}$ und $c \neq 0$

$$\begin{array}{lcl} (a_i(x) = b_i) & \Leftrightarrow & (c \cdot a_i(x) = c \cdot b_i) \\ (a_i(x) = b_i) \text{ und } (a_j(x) = b_j) & \Leftrightarrow & (a_i(x) = b_i) \text{ und } (a_i(x) + a_j(x) = b_i + b_j) \end{array}$$

**Beachte:** Operationen sind nicht nur an der Matrix, sondern auch konsistent am Vektor der rechten Seite $(b)$ bzw. am Lösungsvektor $(x)$ durchzuführen.

**Beispiel 2.2: Gauß -Elimination**

Folgendes Gleichungssystem soll gelöst werden:

$$\begin{array}{rcrcrcl} 10\,x_1 & - & 7\,x_2 & + & & = & 7 \\ -3\,x_1 & + & 2\,x_2 & + & 6\,x_3 & = & 4 \\ 5\,x_1 & - & x_2 & + & 5x_3 & = & 6 \end{array}$$

44 ... oder in Matrixschreibweise:

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \\ 6 \end{pmatrix}$$

**Lösung:** Um die Dreiecksform zu erhalten versuchen wir zuerst die Elemente (hier $a_{21} = -3$ und $a_{31} = 5$) der ersten Spalte zu eliminieren. Danach eliminieren wir ensprechend die Elemente der zweiten Spalte (hier nur $a_{32} = -1$), und so fort. Wir führen also die folgenden Schritte durch:

1) Um $a_{21}$ zu eliminieren addieren wir das $\frac{3}{10}$-fache der ersten Zeile zur zweiten Zeile:

45

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.1 \\ 2.5 \end{pmatrix}$$

2) Um $a_{31}$ zu eliminieren addieren wir das $\frac{-5}{10} = -\frac{1}{2}$-fache der ersten Zeile zur dritten Zeile:

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 2.5 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.1 \\ 2.5 \end{pmatrix} \quad (46)$$

3) Um $a_{32}$ zu eliminieren addieren wir das $\frac{-2.5}{-0.1} = 25$-fache der zweiten Zeile zur dritten Zeile:

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 0 & 155 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.1 \\ 155 \end{pmatrix} \quad (47)$$

4) Das entstandene Dreiecksgleichungssystem kann nun mit dem in Kapitel 2.1 entwickelten Verfahren gelöst werden:

$$\begin{aligned} x_3 &= \frac{155}{155} = 1 \\ x_2 &= \frac{6.1 - 6 \cdot 1}{-0.1} = -1 \\ x_1 &= \frac{7 - (-7) \cdot (-1) - 0 \cdot 1}{10} = 0 \end{aligned} \quad (48)$$

**Bemerkungen:**

- Bei der Transformation kann eine Situation auftreten, die uns zwingt, das Verfahren etwas zu modifizieren:
  Beispiel: Besitzt im ursprünglichen System das Element $a_{22}$ den Wert 2.1 anstatt 2, so entsteht nach den ersten beiden Schritten das Gleichungssystem

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & 0 & 6 \\ 0 & 2.5 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.1 \\ 2.5 \end{pmatrix} \quad (49)$$

  Somit ist es unmöglich, unter Benutzung der zweiten Zeile die Dreiecksgestalt zu erreichen.
  Ausweg: Vertauschung der zweiten und dritten Zeile (dritte Umformungsregel). Danach kann wie gewohnt weiter verfahren werden.

- Es ist oftmals ungünstig, wenn während eines numerischen Berechnungsverfahrens sehr große (oder sehr kleine) Zwischenergebnisse entstehen (dies kann auf einen numerisch instabilen Algorithmus hindeuten). Lösung:

  – Man sollte versuchen, an den Diagonal-Positionen $a_{ii}$ möglichst betragsgroße Elemente zu erhalten (da man durch diese teilen muss).

  – Dies kann durch Vertauschung innerhalb der Zeilen $i$ bis $n$ erreicht werden (dritte Umformungsregel). Der Vorgang heißt Zeilen-Pivotsuche, das betragsmäßig größte Element $a_{ji}$, $j = i, \ldots, n$ heißt Pivotelement. Zu beachten ist, dass man die Positionen im Vektor $b$ (nicht aber in $x$) entsprechend vertauschen muss.

  – Statt Zeilen kann man auch Spalten tauschen – zu beachten ist, dass in diesem Fall die Positionen in $x$ (nicht aber in $b$) entsprechend getauscht werden müssen.

D.h. in obigem Beispiel sollte nach dem zweiten Schritt

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 2.5 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.1 \\ 2.5 \end{pmatrix}$$

die zweite und dritte Zeile vertauscht werden, da $|a_{32}| = |2.5| > |-0.1| = |a_{22}|$ ist. Also erhält man

50

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & -0.1 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 2.5 \\ 6.1 \end{pmatrix}$$

und daraus resultiert durch addieren des 0.1/2.5-fachen von Zeile 2 zu Zeile 3 die Dreiecksform

51

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 2.5 \\ 6.2 \end{pmatrix}$$

## Einschub: Vektoren und Matrizen (vgl. Skript Mathe I)

Eine $n \times m$ Matrix $A$ beschreibt eine lineare Abbildung, die einem Vektor $x \in \mathbb{R}^m$ einen anderen Vektor $y \in \mathbb{R}^n$ zuordnet durch die Festlegung $y = A \cdot x$ oder

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^m a_{1j} x_j \\ \vdots \\ \sum_{j=1}^m a_{nj} x_j \end{pmatrix}$$

Zu zwei Matrizen $B$ und $A$ wird die Hintereinanderausführung der zugehörigen Abbildung $y = A \cdot (B \cdot x) = (AB)x$ beschrieben durch die Matrix $A \cdot B$, das sogenannte Matrixprodukt von $A$ und $B$:

**Definition 2.3: Matrixprodukt**

Für eine $k \times n$ Matrix $A$ und eine $n \times m$ Matrix $B$ 52

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kn} \end{pmatrix} \quad \text{und} \quad B = \begin{pmatrix} b_{11} & \dots & b_{1m} \\ \vdots & & \vdots \\ b_{n1} & \dots & b_{nm} \end{pmatrix}$$

ergibt sich das Matrixprodukt $A \cdot B$ durch die $k \times m$ Matrix

$$A \cdot B = \begin{pmatrix} \sum_{j=1}^n a_{1j} \cdot b_{j1} & \dots & \sum_{j=1}^n a_{1j} \cdot b_{jm} \\ \vdots & & \vdots \\ \sum_{j=1}^n a_{kj} \cdot b_{j1} & \dots & \sum_{j=1}^n a_{kj} \cdot b_{jm} \end{pmatrix}$$

d.h. die Ergebnismatrix $AB$ hat in Zeile/Reihe $r$, Spalte $s$ den Wert

$$(AB)_{rs} = \sum_{j=1}^{n} a_{rj} b_{js}. \qquad 53$$

**Bemerkungen:**

1) Man beachte, dass die Größen der Matrizen zueinander passen müssen – $A$ muss soviele Spalten haben wie $B$ Zeilen hat.
2) Es folgt dass die Matrizenmultiplikation im allgemeinen nicht kommutativ ist, d.h. $A \cdot B \neq B \cdot A$. Dafür ist sie aber wenigstens assoziativ und distributiv, denn durch Nachrechnen sieht man, dass

54

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$
$$A \cdot (B + C) = A \cdot B + A \cdot C$$

3) Es gibt zwei wichtige Spezialfälle der Matrizenmultiplikation:

- Das Skalarprodukt oder innere Produkt zweier Vektoren $x, y \in \mathbb{R}^n$ ergibt den Skalar (d.h. die einfache reelle Zahl)

55

$$C = x^T \cdot y = (x_1, \ldots, x_n) \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \sum_{j=1}^{n} x_j \cdot y_j \qquad \in \mathbb{R}$$

  Hierbei ist $x^T$ (sprich: $x$ transponiert) der dem Spaltenvektor $x$ entsprechende Zeilenvektor.

- Das äußere Produkt zweier Vektoren $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ ergibt die $n \times m$ Matrix

56

$$C = x \cdot y^T = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \cdot (y_1, \ldots, y_m) = \begin{pmatrix} x_1 y_1 & \ldots & x_1 y_m \\ \vdots & & \vdots \\ x_n y_1 & \ldots & x_n y_m \end{pmatrix}$$

4) Die Elemente $c_{rs}$ der Matrix $C = A \cdot B$ ergeben sich als Skalarprodukt der $r$-ten Zeile von $A$ und der $s$-ten Spalte von $B$.

**Beispiel 2.4: Matrizenmultiplikation**

Führen Sie die folgenden Matrizenmultiplikationen durch:

57

a) $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 6 \end{pmatrix} = \begin{pmatrix} 1 \cdot 5 + 2 \cdot 6 \\ 3 \cdot 5 + 4 \cdot 6 \end{pmatrix} = \begin{pmatrix} 17 \\ 39 \end{pmatrix}$

b) $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 \cdot 5 + 2 \cdot 7 & 1 \cdot 6 + 2 \cdot 8 \\ 3 \cdot 5 + 4 \cdot 7 & 3 \cdot 6 + 4 \cdot 8 \end{pmatrix} = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$

c) $\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = 1 \cdot 3 + 2 \cdot 4 = 11$

d) $\begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 \cdot 3 & 1 \cdot 4 \\ 2 \cdot 3 & 2 \cdot 4 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 6 & 8 \end{pmatrix}$

**Definition 2.5: Transponierte Matrix**

Die Matrix $A^T$ bezeichnet die Transponierte der Matrix $A$ mit

$$A^T = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kn} \end{pmatrix}^T = \begin{pmatrix} a_{11} & \dots & a_{k1} \\ \vdots & & \vdots \\ a_{1n} & \dots & a_{kn} \end{pmatrix}$$

58

**Bemerkungen:**

- $A^T$ erhält man aus $A$ durch Spiegelung der Matrix-Koeffizienten an der (verlängerten) ersten Hauptdiagonalen.
- Wenn $A$ eine $k \times n$ Matrix ist, dann ist $A^T$ eine $n \times k$ Matrix.

**Beispiel 2.6: Transponierte Matrizen**

Bestimmen Sie die folgenden Transponierten:

59

a) $$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

b) $$\begin{pmatrix} 5 & 6 & 7 \\ 8 & 9 & 0 \end{pmatrix}^T = \begin{pmatrix} 5 & 8 \\ 6 & 9 \\ 7 & 0 \end{pmatrix}$$

**Lemma 2.7: Rechenregel Transponierte**

Seien $A$ und $B$ jeweils $k \times n$ Matrizen, $C$ eine $n \times m$ Matrix und $k \in \mathbb{R}$ ein Skalar. Dann gilt:

60

$$\begin{aligned} (A+B)^T &= A^T + B^T \\ (k \cdot A)^T &= k \cdot A^T \\ (A^T)^T &= A \\ (A \cdot C)^T &= C^T \cdot A^T \end{aligned}$$

**Beweis:** Einfach nachrechnen, z.B. folgt die erste Gleichung aus

$$\left(\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kn} \end{pmatrix} + \begin{pmatrix} b_{11} & \dots & b_{1n} \\ \vdots & & \vdots \\ b_{k1} & \dots & b_{kn} \end{pmatrix}\right)^T = \begin{pmatrix} a_{11}+b_{11} & \dots & a_{1n}+b_{1n} \\ \vdots & & \vdots \\ a_{k1}+b_{k1} & \dots & a_{kn}+b_{kn} \end{pmatrix}^T$$
$$= \begin{pmatrix} a_{11}+b_{11} & \dots & a_{k1}+b_{k1} \\ \vdots & & \vdots \\ a_{1n}+b_{1n} & \dots & a_{kn}+b_{kn} \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{k1} \\ \vdots & & \vdots \\ a_{1n} & \dots & a_{kn} \end{pmatrix} + \begin{pmatrix} b_{11} & \dots & b_{k1} \\ \vdots & & \vdots \\ b_{1n} & \dots & b_{kn} \end{pmatrix}$$

und der Rest folgt entsprechend. □

Wir schließen den Einschub mit der Zusammenfassung weiterer Eigenschaften von Matrizen ab:

- Eine Matrix $A$ mit der Eigenschaft $A^T = A$ heißt symmetrisch. Z.B. ist für jede beliebige $n \times n$ Matrix $A$ die Matrix $A \cdot A^T$ symmetrisch, denn

61

$$(AA^T)^T = (A^T)^T A^T = AA^T$$

- Der Rang einer Matrix $A$ (bzw. eines Gleichungssystems) bezeichnet die Anzahl der linear unabhängigen Zeilen von $A$.
  Der Rang einer Matrix spielt für die Lösbarkeit eines Gleichungssystems eine entscheidende Rolle:
  Die Gleichung $Ax = b$ ist genau dann lösbar, wenn $\text{rang}(A) = \text{rang}(A|b)$ gilt.

- Eine quadratische $n \times n$ Matrix $A$ mit vollem Rang $\text{rang}(A) = n$ ist invertierbar, d.h. es existiert eine eindeutig bestimmte $n \times n$ Matrix $A^{-1}$, die *Inverse* von $A$, mit

$$A \cdot A^{-1} = A^{-1} \cdot A = I := \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

  wobei $I$ die Einheitsmatrix ist. Bei der Einheitsmatrix sind alle Koeffizienten gleich 0 außer den Elementen der ersten Hauptdiagonalen, die den Wert 1 haben.

- Wenn $A$ eine Inverse besitzt (d.h. den vollen Rang hat), so heißt $A$ regulär, andernfalls singulär.

## 2.3 Lineare Ausgleichsrechnung

Bisher wurden lineare Gleichungssysteme betrachtet, bei denen die Anzahl der Bedingungen (=Gleichungen) mit der Anzahl der Unbekannten übereinstimmt und zu einer regulären (= in Dreiecksform überführbaren) $n \times n$ Matrix führt.

Im folgenden soll der Fall betrachtet werden, dass mehr Gleichungen als Unbekannte vorliegen - man sagt dann "das Gleichungssystem ist überbestimmt". In Matrix-Schreibweise führt das auf eine Fragestellung

$$A \cdot x = b \quad \text{mit} \quad A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \quad \text{für } m > n. \tag{62}$$

Dieses Gleichungssystem ist im Allgemeinen nicht lösbar! Im folgenden definieren wir eine Methode mit der man wenigstens einen Vektor $x$ finden kann, der die Abweichung zwischen $Ax$ und $b$ so klein wie möglich macht.

**Definition 2.8: Lineare Ausgleichsrechnung**

Ziel der linearen Ausgleichsrechnung ist es die oben genannte Fragestellung "so gut wie möglich" zu lösen. D.h. wir suchen ein $x$ welches den Abstand $Ax - b$ bezüglich der Euklid'schen Norm minimiert, also der Gleichung (63)

$$x = \arg\min_{x'} ||Ax' - b||_2^2$$

genügt. Man bezeichnet dieses Verfahren auch als Methode der kleinsten Quadrate.

**Bemerkungen:**

- Zur Vereinfachung der Rechnungen (siehe unten) verwendet man statt der eignentlichen Euklid'schen Distanz $||Ax' - b||_2$ das Quadrat der Euklid'schen Distanz $||Ax' - b||_2^2$ (denn dann kommen keine Wurzeln vor).
- Bei der Minimierung macht das aber keinen Unterschied, da $||Ax' - b||_2$ genau dann minimal ist, wenn $||Ax' - b||_2^2$ minimal ist (denn die Quadrierung von positiven zahlen ist eine streng monoton steigende Abbildung).

Im folgenden leiten wir die Lösung her, indem wir $f(x) := ||Ax - b||_2^2$ minimieren.

64 Mit

$$Ax - b = \begin{pmatrix} \sum_{j=1}^{n} a_{1j} \cdot x_j - b_1 \\ \vdots \\ \sum_{j=1}^{n} a_{mj} \cdot x_j - b_m \end{pmatrix}$$

folgt für die quadratische Fehlerfunktion

65

$$f(x) := ||Ax - b||_2^2 = (Ax - b)^T \cdot (Ax - b) = \sum_{k=1}^{m} \left( \sum_{j=1}^{n} a_{kj} \cdot x_j - b_k \right)^2$$

Die Summe nimmt ihr Minimum an, wenn alle partiellen Ableitungen gleich Null sind:

66

$$0 \stackrel{!}{=} \frac{df}{dx_i} = 2 \cdot \sum_{k=1}^{m} \left( \sum_{j=1}^{n} a_{kj} \cdot x_j - b_k \right) a_{ki} \qquad \text{für } i = 1, \ldots, n.$$

$$\Leftrightarrow \qquad \sum_{k=1}^{m} a_{ki} \cdot \sum_{j=1}^{n} a_{kj} \cdot x_j = \sum_{k=1}^{m} a_{ki} \cdot b_k \qquad \text{für } i = 1, \ldots, n.$$

Die Matrix-Notation der letzen Gleichung lautet:

67

$$(A^T Ax)_i = (A^T b)_i \qquad \text{für } i = 1, \ldots, n.$$

Fassen wir diese $n$ Gleichungen $(i = 1, \ldots n)$ zu einem System zusammen so erhalten wir ein lösbares lineares Gleichungssytem aus $n$ Gleichungen und $n$ Unbekannten:

**Lemma 2.9: Normalgleichung**

68 Die Lösung der linearen Ausgleichsrechnung für das überbestimmte lineare Gleichungssystem $Ax = b$ erhält man durch Lösen von

$$A^T Ax = A^T b \ .$$

Diese Gleichung wird als Normalgleichung zu $A$ und $b$ bezeichnet.

**Beispiel 2.10: Normalgleichung**

Wir betrachten das überbestimmte Gleichungssystem

$$\begin{pmatrix} 1 & 1 \\ 2 & 1 \\ -1 & 4 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} .$$

Die dazugehörige Normalgleichung lautet

$$\begin{pmatrix} 1 & 2 & -1 \\ 1 & 1 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ -1 & 4 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & -1 \\ 1 & 1 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} .$$

. 69

Ausmultiplizieren ergibt

$$\begin{pmatrix} 6 & -1 \\ -1 & 18 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} . \quad 70$$

Lösen des nunmehr quadratischen Gleichungssystems liefert die Lösung

$$x = \begin{pmatrix} \frac{13}{107} \\ \frac{7}{107} \end{pmatrix} . \quad 71$$

**Bemerkungen:**

- Beachten Sie dass $A^T A$ symmetrisch ist und Sie beim Ausmultiplizieren nur (ungefähr) die Hälfte der Einträge berechnen müssen.
- Multipliziert man die Normalgleichung $A^T Ax = A^T b$ beidseitig mit der Inversen $(A^T A)^{-1}$ von links ergibt sich allgemein die Lösung

$$x = A^+ b \qquad \text{für } A^+ := (A^T A)^{-1} A^T .$$

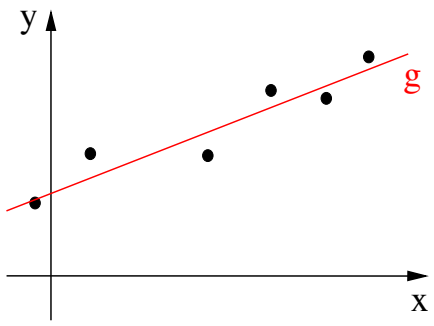
$A^+$ nennt man hierbei die Pseudoinverse oder Moore-Penrose-Inverse von $A$.

**Anwendung:**

Mit dem Verfahren der Ausgleichsrechnung kann zu gegebenen Punkten $(x_j, y_j)$, $j = 1, \ldots, n$ in der $(x, y)$-Ebene eine Gerade

72 $$g(x) = a + bx$$

gefunden werden, die möglichst nahe an den vorgegebenen Punkten liegt. Diese Gerade $g(x)$ bezeichnet man als lineare Ausgleichsgerade.




Die (zunächst unbekannten) Parameter $a$ und $b$ der Gerade sollen nun so gewählt werden, dass die Bedingungen $g(x_j) \approx y_j$ für $j = 1, \ldots, n$ so genau wie möglich erfüllt werden. Dies ergibt das Gleichungssystem

73 $$\begin{pmatrix} a + bx_1 \\ a + bx_2 \\ \vdots \\ a + bx_n \end{pmatrix} \approx \begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{pmatrix}$$

74 oder in Matrixschreibweise

$$A \cdot \begin{pmatrix} a \\ b \end{pmatrix} \approx y \qquad \text{mit} \quad A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

Die zugehörige Normalgleichung lautet also nach Lemma 2.9

75 $$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \end{pmatrix} \cdot \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

oder ausmultipliziert

76 $$\begin{pmatrix} n & \sum_{j=1}^n x_j \\ \sum_{j=1}^n x_j & \sum_{j=1}^n x_j^2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^n y_j \\ \sum_{j=1}^n x_j y_j \end{pmatrix} \qquad (10) \qquad (2.1)$$

Die gesuchte Gerade $y = a + bx$ erhält man in gewohnter Weise durch Lösen dieses Gleichungssystems.

**Beispiel 2.11: Lineare Ausgleichsrechnung / Regression**

Wir wollen zu den Punkten $P_1 = (1, 2)$, $P_2 = (2, 1)$, $P_3 = (4, 3)$ die Ausgleichsgerade $g(x) = a + bx$ berechnen:
$\rightarrow$ **Gesucht:** Geradenparameter $a$ und $b$, so dass $g(x_i) \approx y_i$ für $i = 1, 2, 3$ gilt:

$$\begin{aligned} P_1: &\quad a + b \cdot 1 \approx 2 \\ P_2: &\quad a + b \cdot 2 \approx 1 \\ P_3: &\quad a + b \cdot 4 \approx 3 \end{aligned} \tag{77}$$

oder in Matrixschreibweise

$$\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 4 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix} \tag{78}$$

Die zugehörige Normalgleichung lautet

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 4 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix} \tag{79}$$

oder ausmultipliziert

$$\begin{pmatrix} 3 & 7 \\ 7 & 21 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 6 \\ 16 \end{pmatrix} \tag{80}$$

Dieselbe Gleichung hätte man auch direkt mit (2.1) erhalten. Mit der üblichen Gauß-Elimination erhält man (durch Addieren von $(-7/3)$ mal Zeile 1 zu Zeile 2)

$$\begin{pmatrix} 3 & 7 \\ 0 & \frac{14}{3} \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 6 \\ 2 \end{pmatrix} \tag{81}$$

und damit $b = 3/7$ und $a = 1$. Die gesuchte Lösung ist somit

$$g(x) = 1 + \frac{3}{7}x \ . \tag{82}$$

Mit dieser Technik können wir auch das allgemeine Problem lösen:

**Definition 2.12: Allgemeines Ausgleichsproblem**

83 Gegeben sei eine Familie von Ansatzfunktionen $g_k(x)$ für $k = 1, \ldots, m$ und eine Menge von Punkten $(x_j, y_j)$ für $j = 1, \ldots, n$ mit $n > m$.

- Gesucht ist diejenige Funktion $f(x) = \sum_{k=1}^{m} a_k g_k(x)$ als Linear-Kombination der Ansatzfunktionen $g_k(x)$, die möglichst nahe an den vorgegebenen Punkten liegt.
- Das Finden der Funktion $f(x)$ bezeichnet man als allgemeines Ausgleichsproblem.

**Bemerkung:** Für den vorher betrachteten Spezialfall der linearen Ausgleichsgeraden hatte man $m = 2$ Ansatzfunktionen $g_1(x) = 1$ und $g_2(x) = x$ sodass $f(x) = a_1 + a_2 x$ die Geradengleichung ergab.

Lösungsweg für das Allgemeine Ausgleichsproblem: Mit der $n \times m$ Matrix

84
$$G = \begin{pmatrix} g_1(x_1) & \cdots & g_m(x_1) \\ \vdots & & \vdots \\ g_1(x_n) & \cdots & g_m(x_n) \end{pmatrix}$$

ist die optimale Funktion $f$ wieder bestimmt durch die Koeffizienten $a^T = (a_1, \ldots, a_m)^T$, die $||Ga - y||_2$ minimieren. Diese erhält man als Lösung der Normalgleichung

85
$$G^T \cdot G \cdot \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} = G^T \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

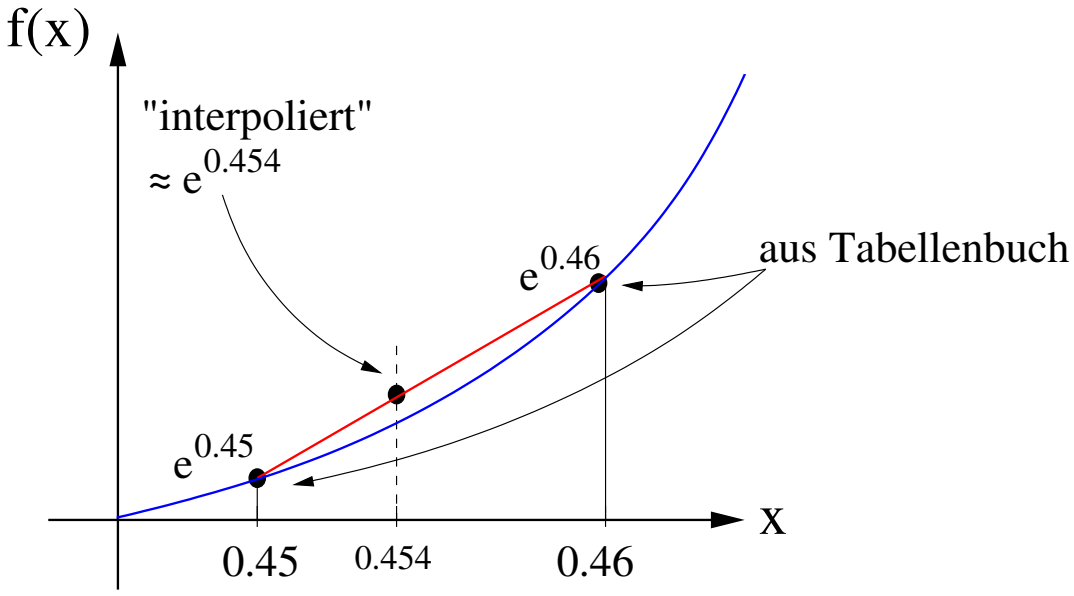
Für Beispiele siehe Übungen.

# Kapitel 3

# Interpolation und Integration

**Vorbemerkung:**
Interpolation war schon vor dem Computerzeitalter ein wichtiges Thema!
→ Es gab Tafelwerke (Tabellenbücher), in denen die Funktionswerte mathematischer Funktionen wie Logarithmus, Exponentialfunktion, Sinus oder Cosinus an vorgegebenen Stützstellen tabelliert waren.

**Beispiel:** Ermittlung des Funktionswerts $e^{0.454}$.



## 3.1 Interpolation

**Definition 3.1: Interpolationsproblem**

86

Unter einem Interpolationsproblem verstehen wir das Problem, dass zu

- paarweise verschiedenen Stellen $x_j$ und dazu gehörigen vorgegebenen Werten $y_j$ (mit $j = 0, \ldots, n$), die zu einer unbekannten Funktion $f(x)$ mit $f(x_j) = y_j$ für alle $j = 0, \ldots, n$ gehören,
- und gegebenen Ansatzfunktionen $g_0(x)$, $\ldots$, $g_n(x)$

Koeffizienten $c_0, \ldots, c_n \in \mathbb{R}$ zu finden sind, so dass die Interpolationsfunktion

$$G(x) := \sum_{k=0}^{n} c_k \cdot g_k(x_j)$$

für alle $j = 0, \ldots, n$ die Punkte $(x_j, y_j)$ interpoliert (d.h. durch diese Punkte verläuft), so dass gilt:

$$G(x_j) \stackrel{!}{=} y_j \quad \text{für alle } j = 0, \ldots, n.$$

**Bemerkungen:**

- Als Ansatzfunktionen $g_k(x)$ könnte man z.B. $x^k$ oder $\cos(kx)$ wählen.
- Ziel ist es also, eine gegebene Funktion durch Linearkombination der Ansatzfunktionen so anzunähern, dass diese an den Stützstellen $(x_j, y_j)$ genau übereinstimmen.
  $\rightarrow$ Diese $n+1$ Bedingungen führen auf ein lineares $(n+1) \times (n+1)$ Gleichungssystem ($\rightarrow$ siehe Übungen)

87

$$\begin{pmatrix} g_0(x_0) & g_1(x_0) & \cdots & g_n(x_0) \\ \vdots & \vdots & & \vdots \\ g_0(x_n) & g_1(x_n) & \cdots & g_n(x_n) \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}.$$

- Zur Unterscheidung sei noch auf das eng mit dem Interpolationsproblem verwandte Ausgleichs- oder Approximationsproblem verwiesen (vgl. Kapitel 2.3, Def. 2.12): Bei der Approximation suchen wir ebenfalls Koeffizienten $c_0, \ldots, c_m$ zu Ansatzfunktionen $g_0(x), \ldots, g_m(x)$ – man beachte: hier kann $m \neq n$ gelten - so dass bezüglich einer gegebenen Norm – z.B. der Euklid'schen Norm (oder deren Quadrat) – die Abstände zwischen $G(x_j)$ und $y_j$

$$\sum_{j=0}^{n} \left( \sum_{k=0}^{m} c_k g_k(x_j) - y_j \right)^2$$

minimiert werden, also ähnlich wie in Def. 2.8

$$(c_0, c_1, \ldots, c_m) \;=\; \arg \min_{(c'_0, c'_1, \ldots, c'_m)} \left|\left| \begin{pmatrix} \sum_{k=0}^{m} c'_k g_k(x_0) - y_0 \\ \cdots \\ \sum_{k=0}^{m} c'_k g_k(x_n) - y_n \end{pmatrix} \right|\right|_2 .$$

Damit die Minimierungsaufgabe effizient lösbar wird, wird i.d.R. die Euklid'sche Norm verwendet.

### Interpolation mit Polynomen

Bei der Interpolation mit Polynomen sind die Ansatzfunktionen von der Form

$$g_k(x) = x^k \tag{88}$$

für k=1,. . . ,n. Wir suchen also das Polynom

$$p(x) = \sum_{k=0}^{n} c_k \cdot x^k \tag{89}$$

für welches

$$p(x_j) = y_j \tag{90}$$

für alle $j = 0, 1, \ldots, n$ gilt. Man könnte die Koeffizienten $c_k$ durch Lösen eines linearen Gleichungssystems erhalten. Dieses Gleichungssystem hätte $n+1$ Gleichungen (für jeden Punkt eine) und $n+1$ unbekannte Koeffizienten $c_0, c_1, \ldots, c_n$, d.h. es gibt im allgemeinen (falls die Stützstellen $x_j$ paarweise verschieden sind) eine eindeutige Lösung. Anstatt ein lineares Gleichungssystem aufzustellen und zu lösen kann man das gesuchte Polynom aber auch einfacher ermitteln:

**Definition 3.2: Lagrange-Polynom**

Zu n+1 vorgegebenen paarweise verschiedenen Stützstellen $x_0, x_1, \ldots, x_n$ ist für $j = 0, 1, \ldots, n$ das $j$-te Lagrange-Polynom definiert durch

91
$$L_j(x) := \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i} = \frac{(x - x_0)\cdots(x - x_{j-1})(x - x_{j+1})\cdots(x - x_n)}{(x_j - x_0)\cdots(x_j - x_{j-1})(x_j - x_{j+1})\cdots(x_j - x_n)} .$$

**Eigenschaften des Lagrange-Polynoms:** $L_j(x)$ ist ein Polynom vom Grad $n$ (da Produkt aus $n$ Linearfaktoren $x - x_i$). Es nimmt an jeder Stützstelle $x_i$ außer $x_j$ den Wert 0 an. An der Stützstelle $x_j$ nimme es den Wert 1 an. Deswegen gilt:

92
$$L_j(x_i) = \begin{cases} 0 & \text{für } i = 0, 1, \ldots, n \text{ mit } i \neq j \\ 1 & \text{für } i = j \end{cases}$$

Daraus ergibt sich der folgende Satz:

**Satz 3.3: Interpolation mit Lagrange-Polynomen**

Für $n + 1$ paarweise verschiedene Punkte $(x_j, y_j)$, $j = 0, 1, \ldots, n$ bilden wir ein Polynom $p(x)$ als die mit den $y_j$ gewichtete Summe der den Stützstellen $x_j$ entsprechenden Lagrange-Polynome,

93
$$p(x) = \sum_{i=0}^{n} y_j \cdot L_j(x).$$

Dann hat $p(x)$ Grad $n$ und erfüllt die Interpolationsbedingung $p(x_i) = y_i$ für alle $i = 0, 1, \ldots, n$. Somit ist $p(x)$ eine (und damit die einzige) Lösung des Interpolationsproblems mit Polynomen.

**Beweis:** Wegen obiger Herleitung gilt offensichtlich $p(x_i) = \sum_{j=0}^{n} y_j \cdot L_j(x_i) = y_i$ (nur ein Summand ist $\neq 0$).
Die Eindeutigkeit folgt, da (1) jedes Polynom $p(x) = c_0 + c_1 x + \ldots + c_n x^n$ vom Grad $n$ durch die $n + 1$ Koeffizienten $c_j$ eindeutig bestimmt ist, und (2) ein Polynom $n$-ten Grades höchstens $n$ Nullstellen haben kann außer es ist das Nullpolynom (siehe Mathe I, Fundamentalsatz der Algebra): Denn nehmen wir an, es gäbe noch ein weiteres Polynom $p'(x) = c'_0 + c'_1 x + \ldots + c'_n x^n$ welches dieselben Punkte interpoliert, also $p(x_i) = p'(x_i) = y_i$ für $i = 0, 1, \ldots, n$. Dann hätte das Differenzpolynom $q := p - p'$

mindestens $n+1$ Nullstellen, nämlich $q(x_i) = 0$ für $i = 0, 1, \ldots, n$. Da $q$ höchstens Grad $n$ hat (falls $c_n \neq c'_n$) muss $q$ also das Null-Polynom sein, d.h. $p - p' = 0$ und damit $p = p'$. □

**Beispiel 3.4: Lagrange-Polynom**

Wir wollen die Funktion $f(x) = 2^x$ durch ein quadratisches Interpolationspolynom mit Stützstellen $-1, 0, 1$ approximieren. Dann können wir z.B. an der Stelle $x = 2$ den Wert des Interpolationspolynoms mit dem exakten Wert $f(2) = 2^2 = 4$ vergleichen.

94

Zu den Stützpunkten $(-1, \frac{1}{2})$, $(0, 1)$, $(1, 2)$ gehören die Lagrange-Polynome

$$L_0(x) = \frac{(x-0)(x-1)}{(-1-0)(-1-1)} = \frac{x(x-1)}{2} = \frac{1}{2}x^2 - \frac{1}{2}x$$
$$L_1(x) = \frac{(x+1)(x-1)}{(0+1)(0-1)} = \frac{(x+1)(x-1)}{-1} = -x^2 + 1$$
$$L_2(x) = \frac{(x+1)(x-0)}{(1+1)(1-0)} = \frac{(x+1)x}{2} = \frac{1}{2}x^2 + \frac{1}{2}x$$

Damit ist das Interpolationspolynom

$$p(x) = \frac{1}{2}L_0(x) + 1 \cdot L_1(x) + 2 \cdot L_2(x) = \frac{1}{4}x^2 + \frac{3}{4}x + 1 \ .$$

Für die Stelle $x = 2$ gilt damit

$$p(2) = 1 + \frac{3}{2} + 1 = 3.5 \qquad (\Rightarrow \text{ Fehler: } f(2) - p(2) = 4 - 3.5 = 0.5) \ .$$

Zur numerischen Bestimmung des Interpolationspolynoms wird normalerweise nicht die Lagrange-Darstellung verwendet, da diese

- keine genaue Rundungsfehleranalyse zuläßt,
- aufwändig ist, und
- inflexibel in Bezug auf Veränderung der Stützstellen ist.

Im folgenden behandeln wir eine effizientere Darstellung.

**Satz 3.5: Rekursive Berechnung des Interpolations-Polynoms**

Gegeben seien $l+1$ Stützpunkte $(x_i, y_i), \ldots, (x_{i+l}, y_{i+l})$. Für rekursive Definitionen

95
$$p_i(x) := y_i \qquad (11) \quad (3.1)$$
$$p_{i,\ldots,i+l}(x) := \frac{(x - x_i) \cdot p_{i+1,\ldots,i+l}(x) - (x - x_{i+l}) \cdot p_{i,\ldots,i+l-1}(x)}{x_{i+l} - x_i} \qquad (12) \quad (3.2)$$

ist $p_{i,\ldots,i+l}(x)$ das (eindeutige) Interpolationspolynom zu den Stützpunkten $(x_i, y_i), \ldots, (x_{i+l}, y_{i+l})$.
D.h. man kann die Berechnung des Polynoms $p_{i,\ldots,i+l}(x)$ vom Grad $l$ auf die Berechnung der beiden Polynome $p_{i+1,\ldots,i+l}(x)$ und $p_{i,\ldots,i+l-1}(x)$ vom Grad $l-1$ zurückführen.

**Beweis:** Wir müssen also für $l = 0, 1, 2, \ldots$ zeigen, dass das Polynom $p_{i,\ldots,i+l}(x)$ die Punkte $(x_i, y_i), \ldots, (x_{i+l}, y_{i+l})$ interpoliert, also $p_{i,\ldots,i+l}(x_j) = y_j$ für $j = i, i+1, \ldots, i+l$. Wir führen den Beweis mit vollständiger Induktion über die Anzahl $l$ der Stützpunkte:
**I.A.** $(l = 0)$**:** Offensichtlich gilt $p_i(x_i) = y_i$ nach (3.1).
**I.S.** $(l \to l+1)$**:** Wir setzen als I.V. den Satz für $l$ Stützpunkte als bewiesen voraus, und zeigen damit, dass er auch für $l+1$ Stützpunkte gilt. Nach I.V. gilt also insbesondere $p_{i+1,\ldots,i+l}(x_j) = y_j$ für $j = i+1, i+2, \ldots, i+l$ und $p_{i,\ldots,i+l-1}(x_j) = y_j$ für $j = i, i+1, \ldots, i+l-1$. Daraus folgt, dass auch $p_{i,\ldots,i+l}(x_j) = y_j$ für alle $j = i, i+1, \ldots, i+l$ gilt, denn:

- $j = i$: Da nach der I.V. $p_{i,\ldots,i+l-1}(x_i) = y_i$ gilt, folgt mit (3.2)

96
$$p_{i,\ldots,i+l}(x_i) = \frac{0 - (x_i - x_{i+l}) \cdot y_i}{x_{i+l} - x_i} = y_i$$

- $j = i + l$: Da nach der I.V. $p_{i+1,\ldots,i+l}(x_{i+l}) = y_{i+l}$ gilt, folgt mit (3.2)

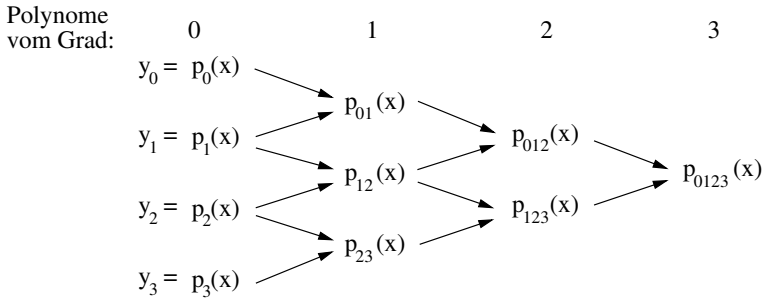
97
$$p_{i,\ldots,i+l}(x_{i+l}) = \frac{(x_{i+l} - x_i) \cdot y_{i+l} - 0}{x_{i+l} - x_i} = y_{i+l}$$

- Da für $i < j < i + l$ nach der I.V. $p_{i+1,\ldots,i+l}(x_j) = y_j$ und $p_{i,\ldots,i+l-1}(x_j) = y_j$ gilt, folgt mit (3.2)

98
$$p_{i,\ldots,i+l}(x_j) = \frac{(x_j - x_i) \cdot y_j - (x_j - x_{i+l}) \cdot y_j}{x_{i+l} - x_i} = \frac{(x_j - x_j + x_{i+l} - x_i) \cdot y_j}{x_{i+l} - x_i} = y_j$$

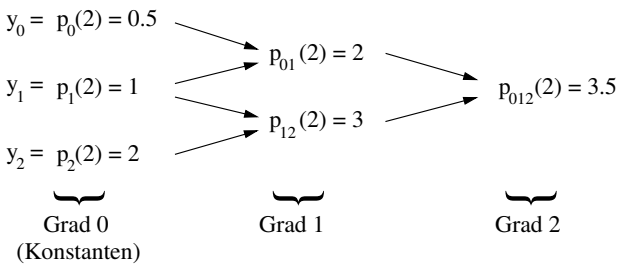
Daraus folgt: $p_{i,\ldots,i+l}(x)$ ist das (und damit das eindeutige) Interpolationspolynom zu den Punkten $(x_i, y_i), \ldots, (x_{i+l}, y_{i+l})$. □

**Bemerkung:** Um das Interpolationspolynom $p(x)$ an einer einzelnen Stelle $x$ auszuwerten, kann unter Anwendung von (3.2) ein Tableau von Funktionsauswertungen von Interpolationspolynomen wachsenden Grades an der Stelle $x$ aufgebaut werden. Man spricht dabei vom Neville-Tableau bzw. Neville-Schema an der Stelle $x$. Die folgende Figur illustriert das Neville-Tableau für eine Interpolation mit vier Stützpunkten:




**Beispiel 3.6: Neville-Schema**

Wir berechnen nun das Interpolationspolynom von Beispiel 3.4 an der Stelle $x = 2$ mit Hilfe des Neville-Schemas. die Stützpunkte sind wieder $(-1, \frac{1}{2})$, $(0, 1)$, $(1, 2)$ und damit ergibt sich mit dem Neville-Tableau




99

$$p_{01}(2) = \frac{(2-(-1))\cdot p_1(2) - (2-0)\cdot p_0(2)}{0-(-1)} = \frac{3\cdot 1 - 2\cdot \frac{1}{2}}{1} = 2$$

$$p_{12}(2) = \frac{(2-0)\cdot p_2(2) - (2-1)\cdot p_1(2)}{1-0} = \frac{2\cdot 2 - 1\cdot 1}{1} = 3$$

$$p_{012}(2) = \frac{(2-(-1))\cdot p_{12}(2) - (2-1)\cdot p_{01}(2)}{1-(-1)} = \frac{3\cdot 3 - 1\cdot 2}{2} = \frac{7}{2} = 3.5$$

Wir haben also das gleiche Ergebnis wie vorher erhalten (Eindeutigkeit von Interpolationspolynomen), d.h. Fehler $f(2) - p_{012}(2) = 4 - 3.5 = 0.5$.

### Weitere Interpolationsansätze

Die bisher betrachteten Interpolationsansätze besitzen folgende Nachteile:

- Sie neigen bei höherem Polynomgrad zu starken Oszillationen, ...
  - ... da man zwischen den Stützstellen "keine Kontrolle über die Polynome hat"
  - ... da die Polynome i.d.R. viele reelle Nullstellen haben.
- Es besteht eine starke (globale) Abhängigkeit der Lösung von einer Stützstelle!

Fazit: Interpolation mit Polynomen ist daher nur für kleine Polynom-Grade, also eine kleine Anzahl von Stützstellen sinnvoll!

### Die Hermite-Interpolation

Um Oszillationen zu vermeiden, werden bei der sogenannten Hermite-Interpolation an den Stützstellen nicht nur Funktionswerte, sondern auch die Ableitungen angegeben.

**Definition 3.7: Hermite-Interpolation**

Die Hermite-Interpolation löst im einfachsten Fall – bei Berücksichtigung der ersten Ableitung – das Interpolations-Problem für $n+1$ Punkte $(x_0, y_0), \ldots, (x_n, y_n)$ durch finden eines Polynoms vom Grad $2n+1$ mit

100

$$p(x_j) = y_j \qquad \text{und} \qquad p'(x_j) = y'_j \qquad \text{für } j = 0, \ldots, n$$

Dieses Problem kann wieder auf ein reguläres lineares Gleichungssystem zurückgeführt werden.

**Bemerkung:** Ähnlich zu Lagrange-Polynomen kann das Hermite-Polynom mittels Ansatz-Polynomen direkt angegeben werden.

Für Beispiele siehe Übungen...

### Die Spline-Interpolation

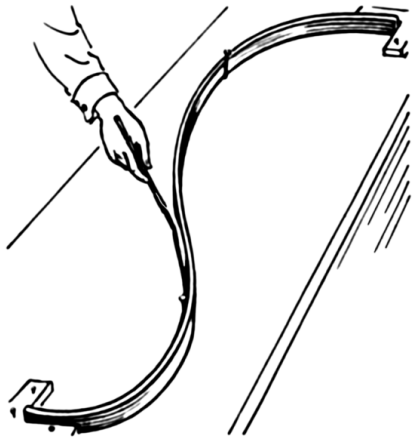
Ein Spline ist eine Funktion, die stückweise aus Polynomen (niedrigen Grades) zusammengesetzt ist. An den Stellen, an denen zwei Polynomstücke zusammenstossen, werden zusätzliche Bedingungen (z.B. Stetigkeit, Differenzierbarkeit) gestellt.

**Definition 3.8: Spline**

Unter einer Splinefunktion vom Grad $k$ zu Stützstellen $x_0, \ldots, x_n$ verstehen wir eine Funktion $S$ mit folgenden Eigenschaften: 101

- $S(x)$ ist $(k-1)$-mal stetig differenzierbar und ...
- $S(x)$ ist auf den Intervallen $[x_i, x_{i+1}]$, $i = 0, \ldots, n-1$ ein Polynom $p_{x_i,x_{i+1}}(x)$ vom Grad $k$.

Zur Wortherkunft: Der Begriff stammt aus dem Schiffbau: Eine lange dünne Latte (Straklatte, engl. "spline"), die an einzelnen Punkten durch Gwichte ("Molche") fixiert wird, biegt sich genau wie ein kubischer Spline mit natürlicher Randbedingung (wobei die Spannungsenergie minimal wird). Ähnlich verwendeten (vor dem Computerzeitalter) Technische Zeichner zwischen Nägeln (den "Stützpunkten") eingeklemmte biegsame Metallstreifen, um "weiche" Kurven zu zeichnen, siehe nebenstehende Skizze:



**Bemerkungen:**

- Nach obiger Definition hat eine Splinefunktion also $n \cdot (k+1)$ Parameter:
  - Die Polynome vom Grad $k$ haben jeweils $k+1$ Koeffizienten. 102
  - Es gibt $n$ Polynomzüge (für jedes Teilintervall eines).
- "$(k-1)$-mal stetig differenzierbar" bedeutet für die $j$-te Ableitung der Polynome $p_{x_i,x_{i+1}}(x)$:

  $$p^{(j)}_{x_i,x_{i+1}}(x_{i+1}) = p^{(j)}_{x_{i+1},x_{i+2}}(x_{i+1}) \text{ für } j = 0, 1, \ldots, k-1 \text{ und } i = 0, 1, \ldots, n-2$$ 103

  (wobei die 0-te Ableitung dem Funktionswert entspricht).
  $\rightarrow$ Dies liefert $(n-1) \cdot k$ Gleichungen.

- Die eigentlichen Interpolationsbedingungen an den Stützstellen sind

104
$$S(x_i) = y_i \qquad \text{für } i = 0, 1, \ldots, n \ .$$

→ Dies liefert weitere $n+1$ Gleichungen.

- Insgesamt verfügt die Splinefunktion damit über $n(k+1)$ Parameter, die

105
$$(n-1)k + (n+1) = nk - k + n + 1 = n(k+1) - (k-1)$$

Gleichungen erfüllen müssen.
Somit können noch $k-1$ zusätzliche Forderungen an die Splinefunktion $S(x)$ gestellt werden, z.B.

  – Durch explizite Vorgabe möglichst vieler Ableitungen im Anfangs- und Endpunkt $x_0$ und $x_n$, z.B.

106
$$S^{(j)}(x_0) = y^{(j)}(x_0) \quad \text{und} \quad S^{(j)}(x_n) = y^{(j)}(x_n) \qquad \text{für } j = 1, \ldots, \frac{k-1}{2}$$

Dies führt auf ein quadratisches lineares Gleichungssystem mit $n(k+1)$ Unbekannten, welches i.Allg. eindeutig gelöst werden kann.

Für Beispiele siehe Übungen...

## Einschub: Binomialkoeffizienten und Binomischer Lehrsatz

**Definition 3.9: Binomialkoeffizienten**

Wir definieren für $k, n \in \mathbb{N}_0$ die Binomialkoeffizienten

107
$$\binom{n}{k} := \frac{n!}{k!(n-k)!} = \frac{n(n-1)\ldots(n-k+1)}{k(k-1)\ldots 1} \qquad \text{für } k \leq n$$

Für $k > n$ wird der Binomialkoeffizient $=0$ gesetzt.

**Bemerkungen:**

- Anschaulich ist $\binom{n}{k}$ die Anzahl der $k$-elementigen Teilmengen einer $n$-elementigen Menge, also $\binom{n}{k} = |\{B \subseteq \{1, 2, \ldots, n\} : |B| = k\}|$.

- Ein Binom (lat. bi, zwei; nomen, Name) ist in der Mathematik ein Polynom mit zwei Gliedern. Der Name "Binomialkoeffizient" (lat. bi=zwei; nom=Namen) stammt daher, dass die verschiedenen Potenzen $(x+y)$, $(x+y)^2$, $(x+y)^3$, ... des Binoms $(x+y)$ durch den Binomialkoeffizienten ausgedrückt werden können (siehe Binomischen Lehrsatz).

**Satz 3.10: Eigenschaften von Binomialkoeffizienten**

$$\binom{n}{0} = \binom{n}{n} = 1$$
$$\binom{n}{1} = \binom{n}{n-1} = n$$
$$\binom{n+1}{k+1} = \binom{n}{k} + \binom{n}{k+1}$$

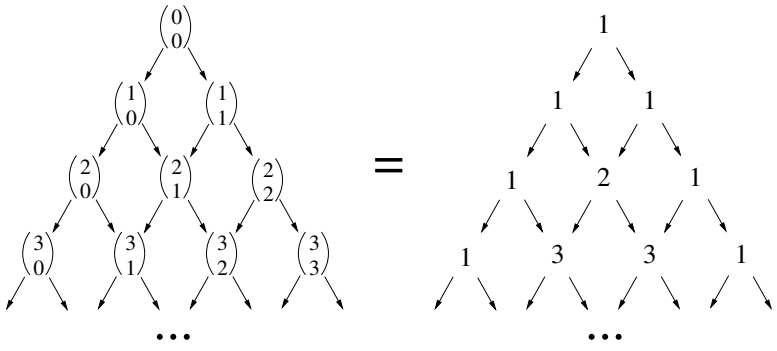
108

**Beweis:**

$$\binom{n}{0} := \frac{n!}{0!n!} = 1 \qquad \text{und} \qquad \binom{n}{n} := \frac{n!}{n!0!} = 1$$
$$\binom{n}{1} := \frac{n!}{1!(n-1)!} = n \qquad \text{und} \qquad \binom{n}{n-1} := \frac{n!}{(n-1)!1!} = n$$
$$\begin{aligned}\binom{n}{k} + \binom{n}{k+1} &:= \frac{n(n-1)\dots(n-k+1)}{k!} + \frac{n(n-1)\dots(n-k)}{(k+1)!} \\ &= \frac{n(n-1)\dots(n-k+1)(k+1) + n(n-1)\dots(n-k)}{(k+1)!} \\ &= \frac{n(n-1)\dots(n-k+1)(k+1+n-k)}{(k+1)!} = \binom{n+1}{k+1}\end{aligned}$$

□

Die in dem letzen Satz gefundenen Rekursionsgleichungen entsprechen dem Pascal'schen Dreieck (siehe folgende Figur):



Damit läßt sich der folgende sogenannte binomische Lehrsatz beweisen:

**Satz 3.11: binomische Summenformel oder binomischer Lehrsatz**

In Ringen (und erst recht in Körpern) gilt für $n \in \mathbb{N}_0$:

109

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k} \tag{13}$$

**Beweis:** Beweis durch vollständige Induktion über $n$:

I.A., $n = 0$ : $\quad (a+b)^0 = 1 = \sum_{k=0}^{0} \binom{0}{0} a^0 b^0$

I.S., $n \to n+1$ : $\quad (a+b)^{n+1} = (a+b)^n (a+b) \overset{I.V.}{=} \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k} (a+b)$

$$= \sum_{k=0}^{n} \binom{n}{k} a^{k+1} b^{n-k} + \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k+1}$$

$$= a^{n+1} + \sum_{k=0}^{n-1} \binom{n}{k} a^{k+1} b^{n-k} + \sum_{k=1}^{n} \binom{n}{k} a^k b^{n-k+1} + b^{n+1}$$

$$= a^{n+1} + \sum_{k=1}^{n} \binom{n}{k-1} a^k b^{n-k+1} + \sum_{k=1}^{n} \binom{n}{k} a^k b^{n-k+1} + b^{n+1}$$

$$= a^{n+1} + \sum_{k=1}^{n} \left( \binom{n}{k-1} + \binom{n}{k} \right) a^k b^{n-k+1} + b^{n+1}$$

$$= a^{n+1} + \sum_{k=1}^{n} \binom{n+1}{k} a^k b^{n-k+1} + b^{n+1}$$

$$= \sum_{k=0}^{n+1} \binom{n+1}{k} a^k b^{n+1-k}$$

□

## 3.2 Bezier-Kurven

In der Computergrafik ist man häufig an einer leicht und einfach steuerbaren Darstellung von Kurven interessiert, die nicht unbedingt interpolierend sein müssen.
Dazu eignen sich sogenannte Bezier-Kurven. Die "Bausteine" von Bezier-Kurven sind die sogenannten Bernstein-Polynome:

**Definition 3.12: Bernstein-Polynom**

Für $i = 0, 1, \ldots, n$ heißen

$$B_i^{(n)}(t) := \binom{n}{i} \cdot t^i \cdot (1-t)^{n-i} \qquad (110)$$

die $n+1$ Bernsteinpolynome vom Grad $n$.

Die Eigenschaften der Bernstein-Polynome werden durch folgenden Satz beschrieben:

**Satz 3.13: Eigenschaften der Bernstein-Polynome**

I) $B_i^{(n)}(t)$ besitzt Nullstellen bei 0 (mit Vielfachheit $i$) und bei 1 (mit Vielfachheit $n-i$).

II) $B_i^{(n)}(t) \geq 0$ für $0 \leq t \leq 1$

III) $\sum_{i=0}^{n} B_i^{(n)}(t) = 1$ für beliebiges $t$.
$\Rightarrow$ Die Bernstein-Polynome sind geeignete Gewichtsfunktionen.

IV) Der Wert eines Bernstein-Polynoms an der Stelle $t$ kann rekursiv bestimmt werden:

$$B_i^{(n)}(t) = t \cdot B_{i-1}^{(n-1)}(t) + (1-t) \cdot B_i^{(n-1)}(t)$$

V) Das Maximum von $B_i^{(n)}(t)$ auf $t \in [0,1]$ liegt bei $t = i/n$. Dort gilt sogar $B_i^{(n)}(i/n) > B_j^{(n)}(i/n)$ für alle $j \neq i$.

**Beweis:**

I) Folgt direkt aus Def. 3.12.

II) Folgt direkt aus Def. 3.12 da $t \geq 0$ und $1 - t \geq 0$.

III) Folgt aus dem binomischen Lehrsatz (Satz 3.11), denn

$$\sum_{i=0}^{n} B_i^{(n)}(t) = \sum_{i=0}^{n} \binom{n}{i} \cdot t^i \cdot (1-t)^{n-i} = (t + 1 - t)^n = 1 \ . \qquad (111)$$

IV) Folgt aus den Eigenschaften der Binomialkoeffizienten (Satz 3.10), denn

112
$$
\begin{aligned}
& t \cdot B_{i-1}^{(n-1)}(t) + (1-t) \cdot B_i^{(n-1)}(t) \\
&= t\binom{n-1}{i-1} \cdot t^{i-1}(1-t)^{n-1-(i-1)} + (1-t)\binom{n-1}{i} t^i (1-t)^{n-1-i} \\
&= \binom{n-1}{i-1} \cdot t^i (1-t)^{n-i} + \binom{n-1}{i} t^i (1-t)^{n-i} \\
&= \left(\binom{n-1}{i-1} + \binom{n-1}{i}\right) \cdot t^i (1-t)^{n-i} \\
&= \binom{n}{i} t^i (1-t)^{n-i} = B_i^{(n)}(t)
\end{aligned}
$$

V) Ableiten von $B_i^{(n)}(t)$ ergibt

113
$$
B_i^{(n)'}(t) = \binom{n}{i} \left( i(1-t)^{n-i} t^{i-1} - (n-i)(1-t)^{n-i-1} t^i \right)
$$

sodass man das Maximum für $B_i^{(n)'}(t) = 0$ erhält, bzw. äquivalent für

114
$$
\begin{aligned}
i(1-t)^{n-i} t^{i-1} = (n-i)(1-t)^{n-i-1} t^i \quad &\Leftrightarrow \quad i(1-t) = (n-i)t \\
&\Leftrightarrow \quad t = \frac{i}{n}
\end{aligned}
$$

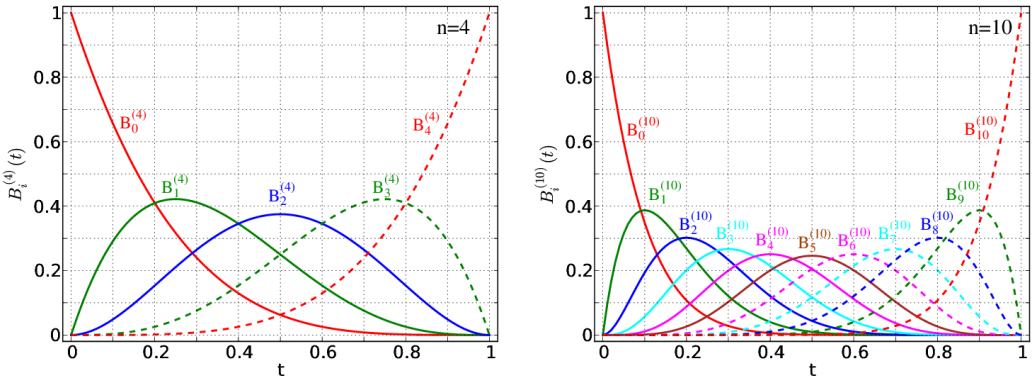
Den Rest kann man ähnlich zeigen. □

**Beispiel 3.14: Bernstein-Polynom**

Bernstein-Polynome vom Grad 4 (davon gibt es fünf Stück):

115
$$
\begin{aligned}
B_0^{(4)} &= \binom{4}{0} \cdot t^0 \cdot (1-t)^{4-0} = (1-t)^4 \\
B_1^{(4)} &= \binom{4}{1} \cdot t^1 \cdot (1-t)^{4-1} = 4t(1-t)^3 \\
B_2^{(4)} &= \binom{4}{2} \cdot t^2 \cdot (1-t)^{4-2} = 6t^2(1-t)^2 \\
B_3^{(4)} &= \binom{4}{3} \cdot t^3 \cdot (1-t)^{4-3} = 4t^3(1-t) \\
B_4^{(4)} &= \binom{4}{4} \cdot t^4 \cdot (1-t)^{4-4} = t^4
\end{aligned}
$$

Folgende Figur zeigt die Bernsteinpolynome $B_i^n(t)$ für $n = 4$ (links) und $n = 10$ (rechts):




Wie im vorigen Satz erwähnt, eigenen sich die Bernsteinpolynome als Gewichte, da sie positiv sind und zu 1 aufsummieren. Hierbei ist die Idee, die Punkte einer gegebenen Punktemenge mit den Werten der Bernsteinpolynome zu gewichten. Damit erhält man eine Kurve die den ersten und letzten Punkt interpoliert und von den restlichen Punkten angezogen wird:

**Definition 3.15: Bezier-Kurve**

Zu gegebenen Kontroll-Punkten $\vec{b_i} \in \mathbb{R}^k$, $i = 0, 1, \ldots, n$ definieren wir die zugehörige Bezier-Kurve mittels

$$\vec{x}(t) := \sum_{i=0}^{n} B_i^{(n)}(t) \cdot \vec{b_i} \qquad \text{mit } 0 \leq t \leq 1$$

116

**Bemerkungen:** Mit Satz 3.13 folgen für Bezier-Kurven folgende Eigenschaften:

- $\vec{x}(t)$ ist gewichtete Summe der Kontrollpunkte $\vec{b_i}$ mit den Gewichten $B_i^{(n)}(t)$.
- Beim Durchlaufen des Parameters $t$ von 0 nach 1 startet die Kurve im ersten Kontrollpunkt $\vec{x}(0) = \vec{b_0}$ und endet im letzten Kontrollpunkt $\vec{x}(1) = \vec{b_n}$.
- Dabei wird die Kurve $\vec{x}(t)$ von allen Kontrollpunkten $\vec{b_i}$ der Reihe nach "angezogen", denn für $t = i/n$ hat $\vec{b_i}$ das größte Gewicht. D.h. ist Parameter $t$ in einem Bereich um $i/n$, so liefert $B_i^n(t)$ das größte Gewicht und bewirkt, sodass hier der Kontrollpunkt $\vec{b_i}$ den Kurvenverlauf am stärksten bestimmt.
- Der Verlauf der Kurve wird nur durch die Kontrollpunkte bestimmt, d.h. es gibt keine unerwünschten Oszillationen.

**Beispiel 3.16: Bezier-Kurve**

Bestimmen Sie die Bezierkurve $x(\vec{t})$ zu den Kontrollpunkten

$$\vec{b}_0 = \begin{pmatrix}1\\1\end{pmatrix},\ \vec{b}_1 = \begin{pmatrix}1\\5\end{pmatrix},\ \vec{b}_2 = \begin{pmatrix}5\\5\end{pmatrix},\ \vec{b}_3 = \begin{pmatrix}10\\5\end{pmatrix},\ \vec{b}_4 = \begin{pmatrix}10\\10\end{pmatrix}$$

und bestimmen Sie die Punkte $\vec{x}(i/4)$ auf der Bezier-Kurve für $i = 0, 1, 2, 3, 4$.
Lösung: Da fünf Kontrollpunkte $\vec{b}_i$ für $i = 0, 1, 2, 3, 4$ gegeben sind ist $n = 4$. Die zugehörigen Bernsteinpolynome $B_i^{(4)}(t)$ haben wir bereits in Beispiel 3.14 auf Seite 56 berechnet. Damit ergibt sich mit Def. 3.15 die Bezier-Kurve

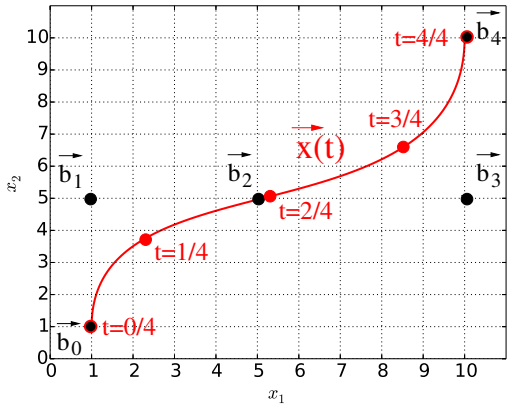
117

$$\begin{aligned}\vec{x}(t) &= B_0^{(4)}(t)\cdot\vec{b}_0 + B_1^{(4)}(t)\cdot\vec{b}_1 + B_2^{(4)}(t)\cdot\vec{b}_2 + B_3^{(4)}(t)\cdot\vec{b}_3 + B_4^{(4)}(t)\cdot\vec{b}_4\\ &= (1-t)^4\begin{pmatrix}1\\1\end{pmatrix} + 4t(1-t)^3\begin{pmatrix}1\\5\end{pmatrix} + 6t^2(1-t)^2\begin{pmatrix}5\\5\end{pmatrix} + 4t^3(1-t)\begin{pmatrix}10\\5\end{pmatrix} + t^4\begin{pmatrix}10\\10\end{pmatrix}\end{aligned}$$

und durch Einsetzen von $t = i/4$ für $i = 0, 1, 2, 3, 4$ ergibt sich (mit dem Taschenrechner oder Computer)

$$\vec{x}(0) = \begin{pmatrix}1\\1\end{pmatrix},\ \vec{x}(1/4) \approx \begin{pmatrix}2.30\\3.75\end{pmatrix},\ \vec{x}(2/4) \approx \begin{pmatrix}5.31\\5.06\end{pmatrix},\ \vec{x}(3/4) \approx \begin{pmatrix}8.49\\6.57\end{pmatrix},\ \vec{x}(1) = \begin{pmatrix}10\\10\end{pmatrix}$$

Die folgende Skizze zeigt diese Punkte (rote Kreise) auf der Bezier-Kurve (rote Linie) neben den Kontrollpunkten (schwarze Kreise):



## 3.3 Integration

Ziel der numerischen Integration ist die Berechnung des bestimmten Integrals

$$I(f) := \int_a^b f(x)dx \ . \tag{118}$$

Diese Aufgabe kann für eine kleine Klasse von Funktionen $f(x)$ durch explizite Angabe einer Stammfunktion $F(x)$ exakt gelöst werden:

$$I(f) = \int_a^b f(x)dx = F(b) - F(a) \tag{119}$$

**Beispiel 3.17: Integration mit Stammfunktion**

Für Potenzfunktionen $f(x) = x^k$ gilt

$$I(f) = \int_a^b x^k dx = \left[\frac{x^{k+1}}{k+1}\right]_a^b = \frac{1}{k+1} \cdot \left(b^{k+1} - a^{k+1}\right) \tag{120}$$

Leider lässt sich die Stammfunktion in den meisten Fällen aber nicht in geschlossener Form angeben und man ist auf ein (numerisches) Näherungsverfahren angewiesen.

### Numerische Quadratur

Ansatzpunkt der numerischen Quadratur ist die Definition des (Riemann-) Integrals als Grenzfall einer (unendlichen) Summe (siehe Mathe I). Hierbei zerlegt man das Interval $[a, b]$ in $n+1$ Teilintervalle der Längen $w_i$ und summiert für $i = 0, 1, \ldots, n+1$ die "Teil-Flächen" $f(x_i) \cdot w_i$ auf (wobei $x_i$ im $i$-ten Teilintervall liegt). Daraus folgt die folgende Quadraturregel:

**Satz 3.18: Quadraturregel**

Die Funktion $f(x)$ wird an bestimmten Stellen $x_i$ (wieder Stützstellen genannt) ausgewertet und mittels dieser Werte und zugehöriger Gewichte $w_i$ ein Näherungswert mit der Quadraturregel

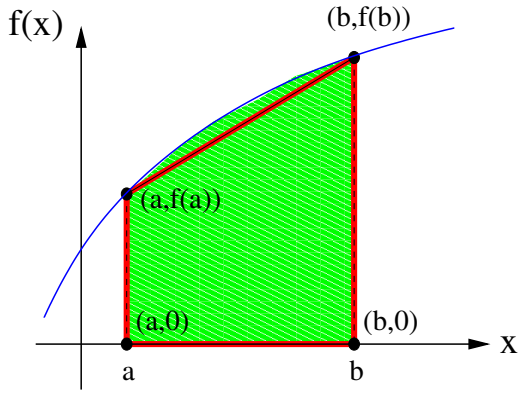
$$I(f) = \sum_a^b f(x)dx \approx \sum_{i=0}^n w_i f(x_i) =: I_n(f) \tag{121}$$

bestimmt. Hierbei gilt $I(f) = \lim_{n\to\infty} I_n(f)$.

Die Fragestellung lautet:
Wie sind die $x_i$ und $w_i$ zu wählen, um bei vorgegebener – möglichst kleiner – Anzahl $n$ von Stützstellen einen brauchbaren Näherungswert $I_n(f)$ zu erhalten?
**Bemerkung:** Um dem Problem der Auslöschung aus dem Weg zu gehen, sollten alle Gewichte $w_i$ positiv sein.

### Quadraturregeln aus Flächenbestimmung

Das bestimmte Integral $I(f)$ läßt sich interpretieren als die Fläche unter dem Funktionsgraph von $f(x)$. $I(f)$ ist genau die Größe der von $f(x)$ und der $x$-Achse begrenzten Fläche zwischen $a$ und $b$ (siehe untenstehende Skizze).




122 Mit Hilfe dieser Interpretation können wir $I(f)$ annähern, in dem wir die zugehörige Fläche approximieren: Eine erste Möglichkeit besteht darin, die von $f(x)$ begrenzte Fläche durch das Trapez $(a,0)$, $(b,0)$, $(b,f(b))$, $(a,f(a))$ anzunähern – dies führt zur Trapezregel

$$I(f) \;=\; \int_a^b f(x)dx \;\approx\; \frac{f(a)+f(b)}{2}(b-a)$$

bzw. der folgenden Definition...

**Definition 3.19: Trapez-Regel**

Die Trapez-Regel approximiert das Integral $I(f) \;=\; \int_a^b f(x)dx$ mittels

123
$$I(f) \;\approx\; \frac{b-a}{2}f(a)+\frac{b-a}{2}f(b)$$

### Quadraturregeln aus der Interpolation

Es liegt nahe, die zu untersuchende Funktion $f(x)$ durch ein interpolierendes Polynom $p(x)$ zu ersetzen – denn das Integral über die im Polynom auftretenden Terme $x^k$ kann einfach mittels der Stammfunktion berechnet werden. Daraus resultiert folgendes Verfahren:

- Wähle Stützstellen $x_0, \ldots, x_n$ (der Einfachheit halber äquidistant)

$$x_i = a + i \cdot h \qquad \text{für} \quad i = 0, \ldots, n \quad \text{und} \quad h := \frac{b-a}{n} \tag{124}$$

  mit $x_0 = a$ und $x_n = b$.

- Die Gewichte $w_0, \ldots, w_n$ ergeben sich dann aus der Quadratur des interpolierenden Polynoms aus der Lagrange-Darstellung

$$p(x) = \sum_{i=0}^{n} f(x_i) \cdot L_i(x) \tag{125}$$

  mit der Näherung

$$\int_a^b f(x)dx \;\approx\; \int_a^b p(x)dx = \sum_{i=0}^{n} f(x_i) \cdot \int_a^b L_i(x)dx = \sum_{i=0}^{n} f(x_i) w_i \tag{126}$$

Anwendung des Verfahrens:

- Für $n = 1$ ergibt sich $x_0 = a$, $x_1 = b$, $h = b - a$ und

$$p(x) = f(a) + \frac{f(b) - f(a)}{b - a} \cdot (x - a) \tag{127}$$

  Damit wird der Wert des Integrals angenähert durch

$$\begin{aligned} I(f) = \int_a^b f(x)dx \approx \int_a^b p(x)dx &= \left[ f(a) \cdot x + \frac{f(b) - f(a)}{b - a} \cdot \frac{(x-a)^2}{2} \right]_a^b \\ &= f(a) \cdot (b - a) + \frac{f(b) - f(a)}{b - a} \cdot \frac{(b-a)^2}{2} \\ &= (b - a) \cdot \left( f(a) + \frac{f(b) - f(a)}{2} \right) \\ &= \frac{h}{2}(f(a) + f(b)) \end{aligned} \tag{128}$$

  mit Gewichten $w_0 = w_1 = \frac{h}{2} = \frac{b-a}{2}$.
  $\Rightarrow$ Dies entspricht genau der Trapezregel!

- Ähnlich kann man für $n = 2$ mit

129

$$x_0 = a, x_1 = \frac{a+b}{2}, x_2 = b$$

die sogenannte Simpson-Regel herleiten:

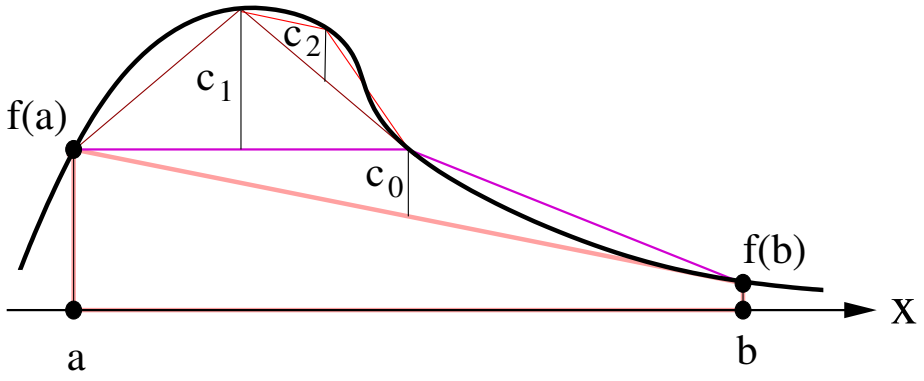
130

$$I(f) = \int_a^b f(x)dx \approx \frac{b-a}{6} \cdot \left( f(a) + 4 \cdot f(\frac{a+b}{2}) + f(b) \right).$$

### Adaptive Verfahren

Die Trapezregel kann wie folgt zu einem adaptiven Verfahren ausgebaut werden:

- Im ersten Schritt erfolgt eine Näherung von $I(f)$ auf dem intervall $[a, b]$ durch lineare Interpolation mit den Intervallgrenzen als Stützstellen ($\Leftrightarrow$ Anwendung der Trapezregel).
- In weiteren Schritten erfolgt eine Auswertung des Fehlers in der Intervallmitte und Aufaddieren einer korrigierenden "Hütchenfunktion" mit der Höhe des Fehlers auf die bisherige Approximation (siehe Skizze)




  Dieses Verfahren läßt sich auf Teilintervallen fortsetzen und bricht ab, wenn die Höhe der korrigierenden Hütchenfunktion eine gewisse Fehlerschranke unterschreitet.

- Das Integral $I(f)$ kann näherungsweise berechnet werden, indem die Flächen der Hütchenkorrekturen berechnet und aufsummiert werden.

# Kapitel 4

# Iterative Verfahren

Oft läßt sich die Lösung eines Problems nicht direkt berechnen, sondern nur iterativ approximieren.
Bei iterativen Vefahren wird mittels einer geeigneten Iterationsvorschrift eine Folge konstruiert, deren Grenzwert die gesuchte Lösung ist.

## 4.1 Fixpunktiteration

**Definition 4.1: Allgemeine Iteration**

Unter einer allgemeinen Iteration verstehen wir ein Verfahren der Form

$$x_0 \in \mathbb{R} \qquad (\textit{Startwert})$$
$$x_{n+1} = \Phi(x_n) \in \mathbb{R} \text{ für } n = 0, 1, 2, \ldots \qquad (\Phi\text{: Iterationsfunktion})$$



Falls die so definierte Folge der $x_n$ konvergiert, also

$$x_n \to \bar{x} \quad \text{bzw.} \quad ||x_n - \bar{x}|| \to 0 \quad \text{für } n \to \infty,$$

dann gilt für eine stetige Iterationsfunktion $\Phi$, dass

$$\bar{x} = \Phi(\bar{x})$$

Ein $\bar{x}$ mit dieser Eigenschaft heißt Fixpunkt von $\Phi$.

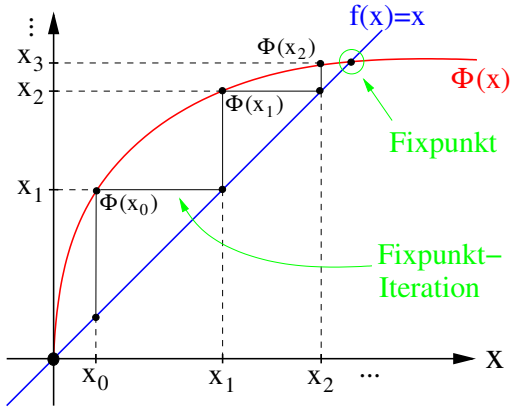
**Definition 4.2: Fixpunkt**

Für eine Funktion $\Phi$ heißt eine Gleichung der Form

132 $$\Phi(x) = x$$

Fixpunktgleichung.
Ihre Lösungen, also diejenigen $\bar{x}$ für die $\Phi(\bar{x}) = \bar{x}$ gilt, heißen Fixpunkte.

Geometrische Deutung: Ein Fixpunkt ist der Schnittpunkt der Geraden $f(x) = x$ mit der Iterationsfunktion $\Phi(x)$ (siehe nebenstehende Illustration).




**Beispiel 4.3: Fixpunktgleichung**

Zur Iterationsfunktion $\Phi(x) = x^2$ lautet die entsprechende Iterationsvorschrift

133 $$x_{n+1} = \Phi(x_n) = x_n^2 \ .$$

Damit gilt:

134 $$x_n = x_{n-1}^2 = x_{n-2}^4 = \ldots = x_0^{2^n} \ .$$

Das Konvergenzverhalten dieser Folge hängt vom Startwert $x_0$ ab:

135
$$\begin{aligned}
x_0 = 0 \quad &\Rightarrow \quad x_n = 0 \text{ für } n = 1, 2, \ldots \quad \Rightarrow \quad \bar{x} = 0 \\
x_0 = \pm 1 \quad &\Rightarrow \quad x_n = 1 \text{ für } n = 1, 2, \ldots \quad \Rightarrow \quad \bar{x} = 1 \\
0 < |x_0| < 1 \quad &\Rightarrow \quad x_n = x_0^{2^n} \to 0 \text{ für } n \to \infty \quad \Rightarrow \quad \bar{x} = 0 \\
|x_0| > 1 \quad &\Rightarrow \quad x_n = x_0^{2^n} \to \infty \text{ für } n \to \infty
\end{aligned}$$

Die Folge konvergiert also für fast alle Startwerte entweder gegen den Fixpunkt 0 oder gegen $\infty$ (in gewisser Weise ebenfalls ein Fixpunkt wegen $\infty = \infty^2$).

Offensichtlich ist 1 ebenfalls ein Fixpunkt, aber keine der erzeugten Folgen konvergiert gegen 1, es sei denn, der Startwert ist exakt $\pm 1$. Ein solcher Fixpunkt heißt abstoßender Fixpunkt.

**Satz 4.4: Anziehende und abstoßende Fixpunkte**

Sei $\Phi : [a, b] \to \mathbb{R}$ eine Funktion mit stetiger Ableitung $\Phi'$ und $\bar{x} \in [a, b]$ ein Fixpunkt von $\Phi$. Dann gilt für die Fixpunktiteration $x_{n+1} = \Phi(x_n)$:

- Ist $|\Phi'(\bar{x})| < 1$, so konvergiert $x_n$ gegen $\bar{x}$, falls der Startwert $x_0$ nahe genug bei $\bar{x}$ liegt.
  $\bar{x}$ heißt dann anziehender Fixpunkt.
- Ist $|\Phi'(\bar{x})| > 1$, dann konvergiert $x_n$ für keinen Startwert $x_0 \neq \bar{x}$ gegen $\bar{x}$.
  $\bar{x}$ heißt dann abstoßender Fixpunkt.

136

**Beweis:** Für $n = 0, 1, 2, \ldots$ sei $\epsilon_n := x_n - \bar{x}$ der "Fehler" von $x_n$, d.h. die Differenz zwischen $x_n$ und dem Fixpunkt $\bar{x}$. Dann gilt mit dem Mittelwertsatz (bzw. dem Satz von Taylor; siehe Einschub/Mathe I):

$$x_{n+1} = \Phi(x_n) = \Phi(\bar{x} + \epsilon_n) \overset{MWS/Taylor}{=} \Phi(\bar{x}) + \epsilon_n \Phi'(\xi) \tag{137}$$

für ein $\xi \in (\bar{x} - |\epsilon_n|, \bar{x} + |\epsilon_n|)$. Da $\Phi'$ stetig ist gilt $\Phi'(\xi) \to \Phi'(\bar{x})$ für $\epsilon_n \to 0$. Also ist $\Phi'(\xi) \approx \Phi'(\bar{x})$ falls $\epsilon_n$ genügend klein ist (d.h. $x_n$ nahe genug bei $\bar{k}$ liegt). Also folgt mit $\Phi(\bar{x}) = \bar{x}$ ($\bar{x}$ ist ja Fixpunkt) in sehr guter Näherung (d.h. kleiner relativer Fehler)

$$x_{n+1} \approx \bar{x} + \epsilon_n \Phi'(\bar{x}) \tag{138}$$

und deshalb gilt in guter Näherung für $|\epsilon_n| = |x_n - \bar{x}|$ die Rekursion

$$|\epsilon_{n+1}| = |\Phi'(\bar{x})| \cdot |\epsilon_n| \ . \tag{139}$$

D.h. für $|\Phi'(\bar{x})| < 1$ gilt $|\epsilon_{n+1}| < |\epsilon_n|$ und damit verkleinert sich der Abstand von $x_n$ zu $\bar{x}$ in jeder Iteration, sodass $x_n \to \bar{x}$ für $n \to \infty$.
Für $\Phi'(\bar{x}) > 1$ gilt entsprechend $|\epsilon_{n+1}| > |\epsilon_n|$ und damit kann $x_n$ nicht gegen $\bar{x}$ konvergieren da sich der Abstand $|x_n - \bar{x}|$ in jeder Iteration vergrößert. □

Der folgende Banach'sche Fixpunktsatz präzisiert die Konvergenzbedingungen (welche Startwerte, Fehlerabschätzung):

**Satz 4.5: Banach'scher Fixpunktsatz**

Sei $\Phi : [a,b] \to [a,b]$ (d.h. $\Phi$ bildet $[a,b]$ auf sich selber ab) und $\Phi$ eine kontrahierende Abbildung (d.h. es gibt eine Konstante $0 < L < 1$ mit $|\Phi(x) - \Phi(y)| \leq L \cdot |x-y|$ für alle $x, y \in [a,b]$). Dann gilt:

140

I) $\Phi$ hat genau einen Fixpunkt $\bar{x}$ in $[a,b]$

II) Die Fixpunktiteration $x_{n+1} = \Phi(x_n)$ konvergiert gegen $\bar{x}$ für alle Startwerte $x_0 \in [a,b]$.

III) Es gelten die Fehlerabschätzungen:

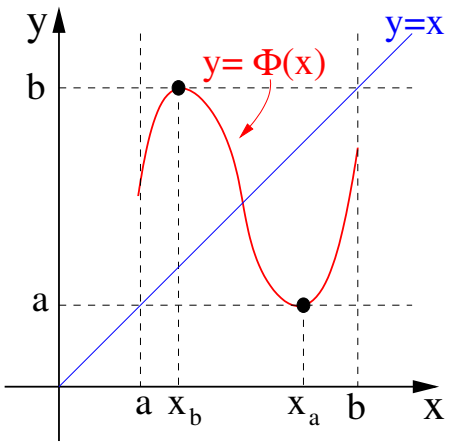
$$|x_n - \bar{x}| \leq \frac{L^n}{1-L} \cdot |x_1 - x_0| \qquad \text{(a-priori-Abschätzung)}$$

$$|x_n - \bar{x}| \leq \frac{L}{1-L} \cdot |x_n - x_{n-1}| \qquad \text{(a-posteriori-Abschätzung)}$$

**Bemerkung:** Ein Intervall zu finden, das unter $\Phi$ in sich selbst abgebildet wird, kann schwierig sein - ist es aber gefunden, dann ist obiger Satz sehr nützlich.

**Beweis:**

- Es gibt mindestens eine Lösung $\bar{x}$ der Fixpunktgleichung:
  Denn da $\Phi : [a,b] \to [a,b]$ gibt es (mindestens) ein $x_a \in [a,b]$ mit $\Phi(x_a) = a$ und ein $x_b \in [a,b]$ mit $\Phi(x_b) = b$.
  Da $\Phi$ stetig ist, ist auch $\delta(x) := \phi(x) - x$ stetig und es gilt $\delta(x_a) = a - x_a \leq 0$ (da $x_a \geq a$) und $\delta(x_b) = b - x_b \geq 0$ (da $x_b \leq b$). Also gibt es mit dem Zwischenwertsatz (siehe Mathe I) mindestens eine Stelle $\bar{x} \in [x_a, x_b]$ (oder $[x_b, x_a]$) mit $\delta(\bar{x}) = 0$ bzw. $\Phi(\bar{x}) = \bar{x}$, d.h. mindestens einen Fixpunkt $\bar{x}$.




- Sei nun $\bar{x}$ so ein Fixpunkt. Wir zeigen, dass es der einzige ist: Für $\epsilon_n := x_n - \bar{x}$ folgt aus dem Mittelwertsatz (wie im vorigen Beweis)

141
$$x_{n+1} = \Phi(\bar{x} + \epsilon_n) \overset{MWS}{=} \Phi(\bar{x}) + \epsilon_n \cdot \Phi'(\xi) = \bar{x} + \epsilon_n \cdot \Phi'(\xi)$$

für ein $\xi \in (a, b)$. Da nach Voraussetzung für den Differenzenquotienten $|\frac{\Phi(x)-\Phi(y)}{x-y}| \leq L$ für alle $x, y \in (a, b)$ so gilt erst recht $|\Phi'(\xi)| \leq L$ (da $\Phi'$ der Grenzwert eines Differenzenquotienten ist). Damit folgt wegen $\epsilon_{n+1} := x_{n+1} - \bar{x} = \epsilon_n \cdot \Phi'(\xi)$

142
$$|\epsilon_{n+1}| \leq L \cdot |\epsilon_n| \ . \qquad (4.1)$$

Da $L < 1$ konvergiert $\epsilon_{n+1} \to 0$ und damit $x_n \to \bar{x}$ für $n \to \infty$, d.h. $\bar{x}$ ist der einzige Fixpunkt (I).

- Wegen (4.1) gilt

143
$$\begin{aligned} |x_n - x_{n-1}| &= |(x_n - \bar{x}) - (x_{n-1} - \bar{x})| = |\epsilon_n - \epsilon_{n-1}| \\ &\geq ||\epsilon_n| - |\epsilon_{n-1}|| = |\epsilon_{n-1}| - |\epsilon_n| \overset{(4.1)}{\geq} |\epsilon_{n-1}| - L \cdot |\epsilon_{n-1}| = (1 - L)|\epsilon_{n-1}| \end{aligned}$$

und deshalb

144
$$|\epsilon_{n-1}| \leq \frac{|x_n - x_{n-1}|}{1 - L} \ . \qquad (4.2)$$

Damit folgt (III), denn

145
$$\begin{aligned} |x_n - \bar{x}| = |\epsilon_n| &\overset{(4.1)}{\leq} L|\epsilon_{n-1}| \overset{(4.2)}{\leq} \frac{L}{1-L}|x_n - x_{n-1}| \quad \text{(a-posteriori)} \\ |x_n - \bar{x}| = |\epsilon_n| &\overset{(4.1)}{\leq} L|\epsilon_{n-1}| \overset{(4.1)}{\leq} \ldots \overset{(4.1)}{\leq} L^n|\epsilon_0| \overset{(4.2)}{\leq} \frac{L^n}{1-L}|x_1 - x_0| \quad \text{(a-priori)} \end{aligned}$$

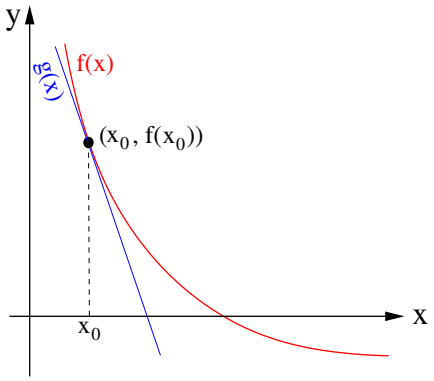
Da (III) unabhängig vom Startwert $x_0 \in [a, b]$ gilt, folgt also auch (II).

□

## 4.2 Das Newton-Verfahren zur Nullstellenbestimmung

**Problemstellung**: Gegeben sei eine differenzierbare Funktion $f : \mathbb{R} \to \mathbb{R}$, gesucht ist eine Nullstelle von $f$, also ein $\bar{x} \in \mathbb{R}$ mit $f(\bar{x}) = 0$.

Hierzu "linearisieren" wir zunächst die Funktion, d.h. wir ersetzen $f$ durch eine lineare Funktion (d.h. eine Gerade) $g$, die der ursprünglichen Funktion $f$ möglichst "gut" entspricht – und lösen das Problem zunächst für diese Ersatzfunktion (siehe nebenstehende Skizze)




Linearisierung der Funktion $f(x)$ im Punkt $x_0$: Taylor-Entwicklung von $f$ in $x_0$ liefert

146
$$f(x) \;=\; f(x_0) + f'(x_0)(x - x_0) + o((x - x_0)^2) \;\approx\; f(x_0) + f'(x_0)(x - x_0)$$

Daraus resultiert die Gerade

147
$$g(x) \;=\; f(x_0) + f'(x_0)(x - x_0)$$

Die Gerade $g(x)$ ist eine gute Näherung von $f(x)$ in der Nähe von $x_0$. Die Abweichung zwischen $g(x)$ und $f(x)$ wird i.d.R. umso größer, je weiter wir uns von der Stelle $x_0$ entfernen.

Lösung unseres Problems (d.h. Bestimmung der Nullstelle) für die Ersatzfunktion:

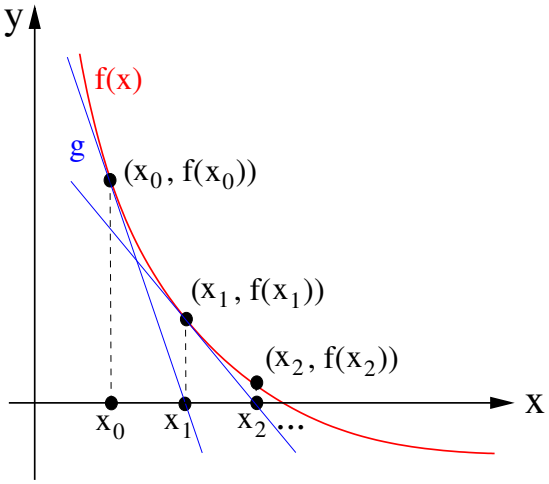
148
$$\begin{aligned} & f(x_0) + f'(x_0)(x - x_0) = 0 \\ \Leftrightarrow \quad & f'(x_0)(x - x_0) = -f(x_0) \\ \Leftrightarrow \quad & x = x_0 - \frac{f(x_0)}{f'(x_0)} \end{aligned}$$

Diese Stelle $x$ stellt somit eine bessere Näherung an die Nullstelle von $f$ dar.
$\Rightarrow$ Wir machen dasselbe nochmal, starten aber jetzt mit

$$x_1 := x_0 - \frac{f(x_0)}{f'(x_0)}$$

149

$\Rightarrow$ usw. (siehe folgende Skizze...)




Dies führt auf die folgende, als Newton-Verfahren bezeichnete Iterationsvorschrift:

**Definition 4.6: Newton-Verfahren**

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \text{ für } n = 0, 1, 2, \ldots$$

150

**Bemerkungen:**

- Nach Def. 4.2 löst das Newton-Verfahren also die Fixpunktgleichung $\Phi(x) = x$ mit Iterationsfunktion $\Phi(x) = x - f(x)/f'(x)$.
- Der Startpunkt $x_0$ sollte in der Nähe der Nullstelle liegen, um Aussicht auf schnelle Konvergenz zu haben.
- Nachteile des Newton-Vefahrens:
  - Wir benötigen Ableitungen der Funktion $f$.
  - Konvergenz (insb. in der Nähe lokaler Extrema) ist nicht immer garantiert.

### Das vereinfachte Newtonverfahren

Statt in jedem Iterationsschritt $f'(x_n)$ zu berechnen, kann stets $f'(x_0)$ verwendet werden. Daraus resultiert das vereinfachte Newton-Vefahren:

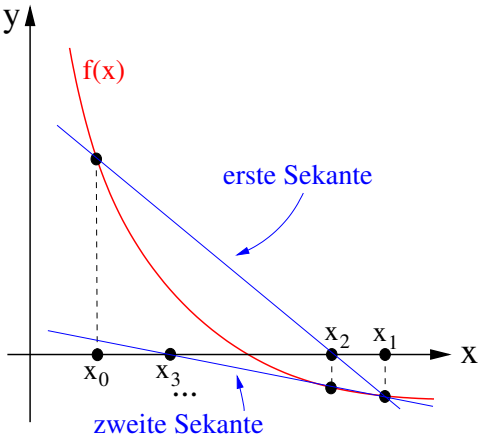
**Definition 4.7: Vereinfachtes Newton-Verfahren**

151

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)} \text{ für } n = 0, 1, 2, \ldots$$

### Das Sekantenverfahren

Beim sogenannten Sekanten-Verfahren wird im Newton-Verfahren die Tangente im Punkt $(x_0, f(x_0))$ ersetzt durch die Sekantengerade durch die Punkte $(x_0, f(x_0))$ und $(x_1, f(x_1))$ und als Näherung für die Nullstelle von $f$ die Nullstelle der Sekantengerade berechnet.




$\Rightarrow$ D.h. Die Ableitung $f'(x_0)$ des Newton-Vefahrens wird im ersten Schritt des Sekantenverfahrens durch die Steigung der Sekantengerade $\frac{f(x_1)-f(x_0)}{x_1-x_0}$ ersetzt, also:

152

$$x_2 = x_0 - \frac{f(x_0)}{\frac{f(x_1)-f(x_0)}{x_1-x_0}} = x_0 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} \cdot f(x_0)$$

Die weiteren Schritte sind analog, sodass sich das folgende Sekantenverfahren ergibt:

**Definition 4.8: Sekanten-Verfahren**

153

$$x_{n+1} = x_{n-1} - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \cdot f(x_{n-1}) \quad \text{für } n = 1, 2, 3, \ldots$$

**Bemerkungen:**

- Das Sekantenverfahren benötigt zwei Startstellen $x_0$ und $x_1$.
- Vorteile des Sekantenverfahrens:
  - Das Verfahren ist billig (denn jeder Iterationsschritt kostet nur eine (neue) Funktionsauswertung $f(x_n)$).
  - Es brauchen keine Ableitungen berechnet zu werden.

## 4.3 Konvergenzgeschwindigkeit

Wir wollen im folgenden die Konvergenzgeschwindigkeit verschiedener iterativer Verfahren beurteilen. Dafür betrachten wir die Konvergenz einer Folge von Werten $\{x_n\}$ zu einem Fixpunkt $\bar{x}$.

**Definition 4.9: Konvergenzordnung**

Sei $\{x_n\}$ eine Folge mit $\lim_{n\to\infty} x_n = \bar{x}$. Man sagt, das Verfahren besitzt eine Konvergenzordnung $q \geq 1$, wenn es eine Konstante $c > 0$ gibt mit

$$|x_{n+1} - \bar{x}| \leq c \cdot |x_n - \bar{x}|^q \quad \text{für alle } n = 0, 1, 2, \dots \ . \tag{154}$$

Falls $q = 1$ gilt, verlangt man zusätzlich $c < 1$.

**Bemerkungen:**

- Im Fall $q = 1$ spricht man von linearer Konvergenz.
- Im Fall $q = 2$ spricht man von quadratischer Konvergenz.

**Satz 4.10: Konvergenzordnung von Newton- und Sekanten-Verfahren**

Für einfache Nullstellen von $f$ konvergiert

- das Newton-Verfahren quadratisch ($q = 2$),
- das vereinfachte Newton-Verfahren linear ($q = 1$)
- und für das Sekanten-Verfahren gilt

$$q = \frac{1 + \sqrt{5}}{2} \approx 1.618 \ . \tag{155}$$

**Bemerkung:** Im Falle mehrfacher Nullstellen von $f$ konvergiert das Newton-Verfahren nur noch linear. Will man die quadratische Konvergenz erhalten, muss die Iterationsvorschrift wie folgt modifiziert werden:

**Satz 4.11: Modifiziertes Newton-Verfahren**

156

$$x_{n+1} = x_n - m \cdot \frac{f(x_n)}{f'(x_n)}$$

Hierbei bezeichnet $m \in \mathbb{N}$ die Vielfachheit der Nullstelle.

**Bemerkung:** Leider kennt man die Vielfachheit der gesuchten Nullstelle oft nicht.

**Beispiele:** Siehe Übungen...

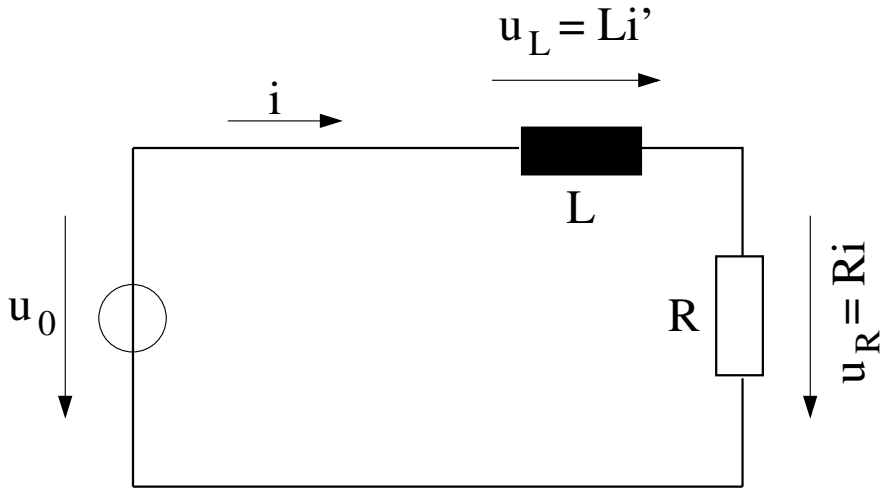
# Kapitel 5

# Gewöhnliche Differentialgleichungen

## 5.1 Problemstellung

In vielen technischen/physikalischen/ökonomischen Vorgängen – wie etwa der Beschreibung elektrischer Schaltkreise (siehe Skizze und folgendes Beispiel) – sind relevante Größen mit ihren zeitlichen und/oder räumlichen Veränderungen verknüpft.

⇒ Die gesuchte Funktion/Größe ist also durch Gleichungen beschrieben, in denen sowohl die gesuchte Funktion als auch der Differentialquotient bzw. die Ableitung der Funktion auftauchen. Solche Gleichungen nennt man Differentialgleichungen.



**Beispiel 5.1: Elektrischer Schaltkreis**

Gegeben sei ein elektrischer Schaltkreis aus Spule $L$ und Ohm'schem Widerstand $R$ und einer Spannungsquelle $u_0(t)$ (siehe obige Skizze).
Gesucht ist der Strom $i(t)$ als Funktion der Zeit $t$ für eine beliebige gegebene Spannungsquelle $u_0(t)$.

**Lösungsansatz:** Aus der Physik (oder der Vorlesung Signale und Systeme) kennen wir die Beziehung zwischen Strom und Spannung an den einzelnen elektrischen Elementen:

- **Ohm'sches Gesetz:** Die am Widerstand $R$ abfallende Spannung $u_R(t) = R \cdot i(t)$ ist proportional zum Strom $i(t)$.
- **Induktionsgesetz:** Die in der Spule $L$ induzierte Spannung $u_L(t) = L \cdot i'(t)$ ist proportional zur Ableitung $i'(t)$ (d.h. zeitl. Änderung) des Stroms $i(t)$.

Aus der Maschenregel $u_0(t) = u_L(t) + u_R(t)$ folgt also

157
$$u_0(t) = L \cdot i'(t) + R \cdot i(t) \qquad (16) \quad | \qquad (5.1)$$

oder äquivalent

158
$$i'(t) = -\frac{R}{L} i(t) + \frac{1}{L} u_0(t) \ . \qquad (17) \qquad (5.2)$$

Dies ist eine (gewöhnliche) Differentialgleichung in der gesuchten Funktion $i(t)$. Funktionen $i(t)$ die diese Gleichung erfüllen heißen Lösungen der Differentialgleichung.Im allgemeinen haben Differentialgleichungen unendlich viele Lösungen $i(t)$. Durch Festlegen eines Wertes $i(0)$ (z.B. $i(0-) = 0$ falls zur Zeit $t = 0$ die Spannungsquelle erst eingeschaltet wird) erhält man in der Regel eine eindeutige Lösung.

Die folgende Definition legt allgemein fest was wir unter einer gewöhnlichen Differentialgleichung verstehen:

**Definition 5.2: Gewöhnliche Differentialgleichung**

Gegeben ist eine Funktion $f : \mathbb{R}^2 \to \mathbb{R}$, ein Intervall $[t_a, t_b]$ und ein Anfangswert $y_a$. Gesucht ist dazu eine Funktion $y : [t_a, t_b] \to \mathbb{R}$ mit:

159
$$y'(t) = f(t, y(t)) \quad \text{für alle } t \in [t_a, t_b] \quad \text{und} \qquad (18) \qquad (5.3)$$
$$y(t_a) = y_a \qquad (19) \qquad (5.4)$$

(5.3) bezeichnet man als gewöhnliche Differentialgleichung.
(5.3) zusammen mit (5.4) bezeichnet man als Anfangswertproblem.

**Bemerkungen:**

- Das Anfangswertproblem besteht also darin, eine Lösung $y(t)$ der gewöhnlichen Differentialgleichung (5.3) zu finden, die an der Stelle $t = t_a$ den vorgegebenen Wert $y_a$ annimmt.
- Statt (5.3) schreibt man häufig auch kurz $y' = f(t, y)$ .
- Entsprechend gibt es auch Systeme von Differentialgleichungen, dabei sind die unbekannten Größen Vektoren, deren Komponenten Funktionen sind. Wir beschränken uns hier auf den eindimensionalen Fall (alle Verfahren lassen sich aber entsprechend auf den mehrdimensionalen Fall erweitern).

Die einfachste Differentialgleichung ist die sogenannte homogene lineare Differentialgleichung 1.Ordnung, $y' = cy$. Der folgende Satz bestimmt alle Lösungen dieser Differentialgleichung:

**Satz 5.3: Homogene lineare Differentialgleichung 1.Ordnung**

Alle Lösungen der homogenen linearen Differentialgleichung 1.Ordnung

$$y'(t) = c \cdot y(t)$$

sind für $a \in \mathbb{R}$ gegeben durch

$$y(t) = ae^{ct} \ . \qquad (20) \qquad (5.5)$$

160

Die eindeutige Lösung des entsprechenden Anfangswertproblems mit $y(0) = y_0$ erhält man damit für $a = y_0$.

**Beweis:**

- Offensichtlich ist $y(t) = ae^{ct}$ eine Lösung der Differentialgleichung, denn

$$y'(t) = (ae^{ct})' = cae^{ct} = cy(t) \ .$$

161

- Wir müssen noch zeigen, dass es keine weiteren Lösungen (bis auf die multiplikative Konstante $a$) gibt:
  Angenommen $\tilde{y}(t)$ sei eine weitere Lösung. Dann gilt für $h(t) := \frac{\tilde{y}(t)}{y(t)}$ mit der Quotientenregel

$$h'(t) = \frac{\tilde{y}'y - \tilde{y}y'}{y^2} = \frac{c\tilde{y}(t) \cdot ae^{ct} - \tilde{y}(t) \cdot cae^{ct}}{(y(t))^2} = \frac{\tilde{y}(t)(cae^{ct} - cae^{ct})}{(y(t))^2} = 0 \ .$$

162

  D.h. $h(t) = h_0$ ist konstant. Deshalb gilt auch $\tilde{y}(t) = h_0 y(t) = a' e^{ct}$ für $a' \in \mathbb{R}$.

- Offensichtlich gilt für das entsprechende Anfangswertproblem $y' = cy$ mit $y(0) = y_0$ wegen (5.5), dass $y(t) = ae^{ct}$ mit $a = y(0) = y_0$.

□

**Beispiel 5.4: Homogenes lineares Anfangswertproblem 1.Ordnung**

Wir wollen unsere Resultate des vorigen Satzes auf den elektrischen RL-Schaltkreis von Bsp. 5.1 mit $R = 2\Omega$ und $L = 4\text{H}$ anwenden. Dafür nehmen wir weiter an, dass die Spannungsquelle $u_0(t) = 0$ ausgeschaltet ist und dass zur Zeit $t = 0$ ein Strom von $i(t) = i_0 := 6\text{A}$ fließe. D.h. nach (5.2) suchen wir eine Lösung $i(t)$ für das Anfangswertproblem

163
$$i'(t) = -\frac{R}{L} i(t) \quad \text{mit } i(0) = i_0 \ .$$

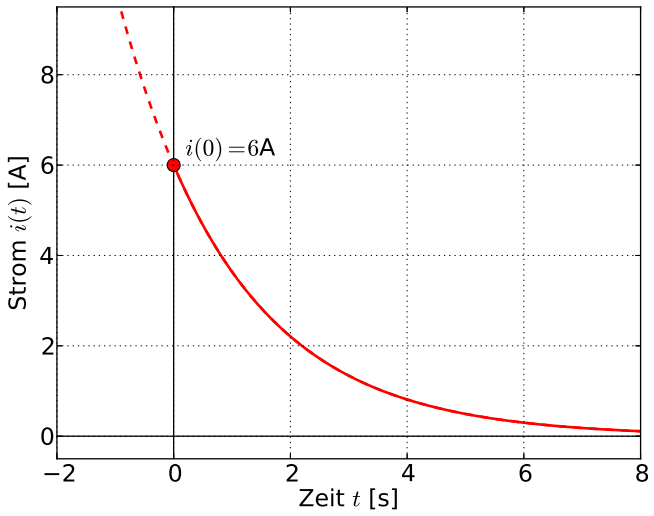
Nach Satz 5.3 erhält man also mit $c = -R/L$ und $a = i_0$ die Lösung

164
$$i(t) = i_0 \cdot e^{-\frac{R}{L} \cdot t} \ .$$

D.h. mit $i_0 = 6A$ und $R/L = 2\Omega/4\text{H} = 1/(2s)$ erhält man also die eindeutige Lösung

165
$$i(t) = i_0 \cdot e^{-\frac{R}{L} \cdot t} = 6A \cdot e^{-\frac{1}{2s} t},$$

d.h. ein exponentielles Abklingen $i(t) \to 0$ für $t \to 0$ (siehe folgende Skizze).



In obigem Beispiel konnten wir die Differentialgleichung (5.2) analytisch lösen, da die Spannungsquelle $u_0(t) = 0$ einen sehr einfachen zeitlichen Verlauf hatte. Dies ist im allgemeinen (für beliebige $u_0(t)$) nicht mehr möglich. Bevor wir numerische Verfahren zur Lösung solcher Differentialgleichungen vorstellen, wollen wir noch diskutieren wie man das Lösen allgemeiner Differentialgleichungen mit Hilfe von Richtungsfeldern veranschaulichen kann:

**Definition 5.5: Richtungsfeld-Darstellung von Differentialgleichungen**

Gegeben sei eine Differentialgleichung $y' = f(t, y)$. Wenn der Graph einer Lösung $y(t)$ durch den Punkt $(t, y(t))$ läuft, dann muss er dort die Steigung

$$y'(t) = f(t, y(t)) \tag{166}$$

haben. Man erhält dann das Richtungsfeld oder Vektorfeld der Differenzialgleichung durch eine Abbildung $\vec{r} : \mathbb{R}^2 \to \mathbb{R}^2$ die jedem Punkt der $(t, y)$-Ebene einen Vektor-Pfeil $\vec{r}$ mit

$$\vec{r}(t, y) = \begin{pmatrix} 1 \\ f(t, y) \end{pmatrix} \tag{167}$$

zuordnet.

**Bemerkungen:**

- In üblichen Darstellungen zeichnet man die Vektoren nur auf einem Gitter der $(t, y)$-Ebene mit Abstand $\Delta t$ und $\Delta y$.
- Zusätzlich kann man die Vektoren mit Faktor $\Delta t$ skalieren.
- Man kann dann eine Näherung der Lösungskurve eines Anfangswertproblems graphisch bestimmen, indem man vom Anfangswert ausgehend dem Richtungsfeld folgt. Dies entspricht der Grundidee vieler numerischer Lösungsverfahren (z.B. sogar exakt dem Euler-Verfahren, das wir nachher diskutieren werden).

**Beispiel 5.6: Richtungsfeld**

Bestimmen Sie das Richtungsfeld der Differentialgleichung

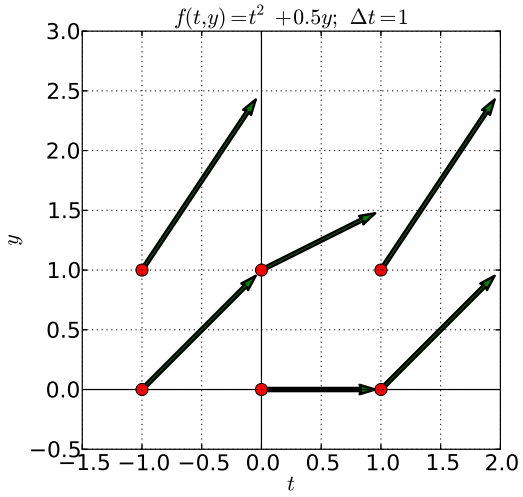
$$y'(t) = t^2 + 0.5y(t)$$

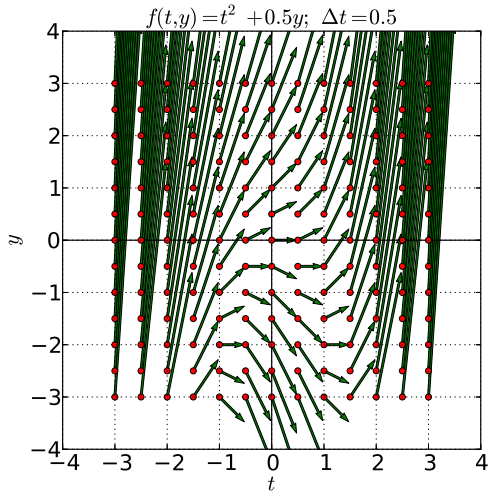
für $\Delta t = \Delta y = 1$ und $0 \leq y \leq 1$ und $-1 \leq t \leq 1$.
Lösung:

168

$$\begin{aligned}
\vec{r}(-1,0) &= \begin{pmatrix} 1 \\ (-1)^2 + 0.5 \cdot 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
\vec{r}(0,0) &= \begin{pmatrix} 1 \\ 0^2 + 0.5 \cdot 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\
\vec{r}(1,0) &= \begin{pmatrix} 1 \\ 1^2 + 0.5 \cdot 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
\vec{r}(-1,1) &= \begin{pmatrix} 1 \\ (-1)^2 + 0.5 \cdot 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1.5 \end{pmatrix} \\
\vec{r}(0,1) &= \begin{pmatrix} 1 \\ 0^2 + 0.5 \cdot 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} \\
\vec{r}(1,1) &= \begin{pmatrix} 1 \\ 1^2 + 0.5 \cdot 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1.5 \end{pmatrix}
\end{aligned}$$

Die 6 Richtungsvektoren des Richtungsfelds sind in der untenstehenden Figur dargestellt (linke Seite). Außerdem ist auf der rechten Seite das Richtungsfeld in feinerer Auflösung dargestellt.



**Beispiel 5.7: Vektorfeld eines RL-Netzwerkes**

Gegeben sei das elektrische Netzwerk der Figur von Seite 73, welches nach Bsp. 5.1 durch die Differentialgleichung (5.2)

$$i'(t) = -\frac{R}{L}i(t) + \frac{1}{L}u_0(t) \ .$$

beschrieben wird. Bestimmen Sie für $R = 2\Omega$ und $L = 4\text{H}$ das Vektorfeld für

a) $u_0(t) = 10\text{V}$.

b) $u_0(t) = 10\text{V} \cdot \sin(2\pi 50\frac{t}{s})$

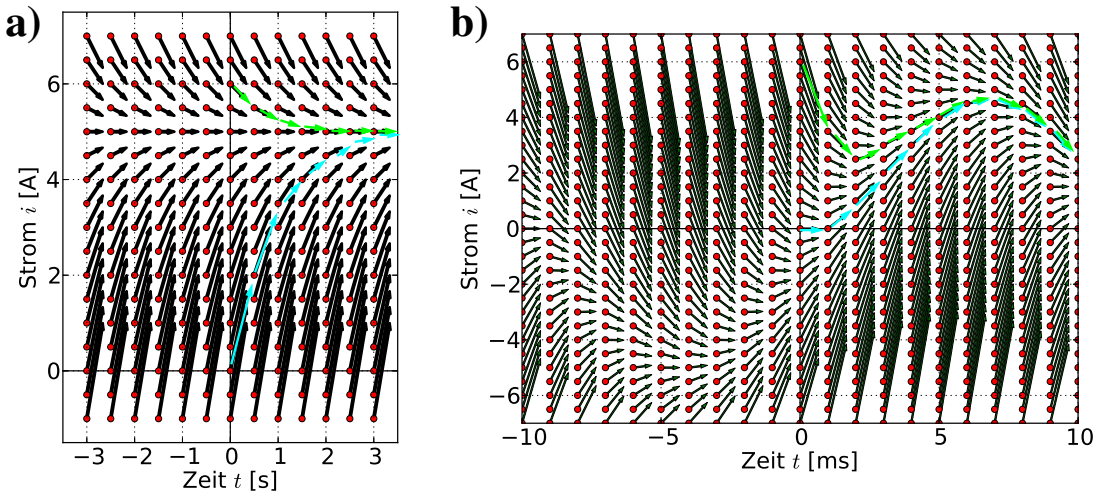
und skizzieren Sie jeweils die Lösung für das Anfangswertproblem mit $i(0) = 6A$.
Lösungen:

169

a) $\vec{r}(t,i) = \begin{pmatrix} 1 \\ -\frac{R}{L}i(t) + \frac{1}{L}u_0(t) \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{2s}i + 2.5\frac{\text{A}}{\text{s}} \end{pmatrix}$.

b) $\vec{r}(t,i) = \begin{pmatrix} 1 \\ -\frac{R}{L}i(t) + \frac{1}{L}u_0(t) \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{2s}i + 2.5\frac{\text{A}}{\text{s}} \cdot \sin(2\pi 50\frac{t}{s}) \end{pmatrix}$.

Die folgende Figur zeigt die beiden Vektorfelder der Beispiele und skizziert jeweils zwei Trajektorien (d.h. Funktionsverläufe) für Anfangswerte $i(0) = 0$ (türkis) und $i(0) = 6A$ (grün).



Im folgenden behandeln wir Verfahren zur Lösung von Anfangswertproblemen.

## 5.2 Das Euler-Verfahren

Beim Euler-Verfahren approximiert man in der Differentialgleichung

$$y'(t) \;=\; f(t, y(t))$$

die Ableitung $y'(t) := \lim_{\Delta t \to 0} \frac{y(t+\Delta t)}{\Delta t}$ durch den Differenzenquotienten

170
$$\frac{y(t+\Delta t) - y(t)}{\Delta t} \approx y'(t) = f(t, y(t)) \; .$$

für eine hinreichend kleine Zeitdifferenz $\Delta t$ . Auflösen nach $y(t + \Delta t)$ ergibt die äquivalente Näherung

171
$$y(t + \Delta t) \approx y(t) + f(t, y(t)) \cdot \Delta t \; .$$

Falls wir also den Wert $y(t_0)$ zur Zeit $t = t_0$ kennen – wie es etwa bei einem Anfangswertproblem mit $y(t_0) = y_0$ für $t_0 = a$ der Fall ist – so können wir damit also $y(t_1)$ zur Zeit $t_1 := t_0 + \Delta t$ berechnen,

172
$$y(t_1) \approx y(t_0) + f(t_0, y(t_0)) \cdot \Delta t \; .$$

Ausgehend von dieser Näherung kann man nach demselben Prinzip eine Näherung für $y(t_2)$ zur Zeit $t_2 := t_1 + \Delta t = t_0 + 2 \cdot \Delta t$ berechnen, und so fort. Daraus resultiert mit $h := \Delta t$ das folgende Euler-Verfahren:

**Definition 5.8: Euler-Verfahren**

Das Euler-Verfahren approxomiert die Lösung des Anfangswertproblems $y' = f(t, y)$, $t \in [a, b]$, $y(a) = y_a$ durch folgendes Vorgehen:

- Sei $(a =: t_0, t_1, t_2, \ldots, t_N := b)$ eine Zerlegung des Intervals $[a, b]$ in $N$ Teilintervalle mit $t_n = t_0 + n \cdot h$ für $n = 0, 1, 2, \ldots, N$ und Schrittweite $h := \frac{b-a}{N}$.
- Dann erhält man Näherungen $y_n \approx y(t_n)$ für die gesuchte Funktion $y(t)$ an den Stützstellen $t_n$ der Zerlegung durch die folgende Rekursion:

173
$$\begin{aligned} y_0 &:= y_a \quad \text{und} \\ y_{n+1} &:= y_n + h \cdot f(t_n, y_n) \quad \text{für } n = 0, 1, 2, \ldots, N-1 \end{aligned}$$

  sodass $y_N \approx y(t_N) = y(b)$ ist.

**Bemerkungen:**

- Das Verfahren entspricht genau dem graphischen Bestimmen der Lösungskurve in Richtungsfelder wie im vorigen Abschnitt (siehe z.B. die Illustrationen auf Seite 79): D.h. ausgehend vom Punkt $(t_n, y_n)$ kommt man zum Folgepunkt durch

$$\begin{pmatrix} t_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} t_n \\ y_n \end{pmatrix} + h \cdot \begin{pmatrix} 1 \\ f(t_n, y_n) \end{pmatrix} \tag{174}$$

  wobei der rechte Summand genau dem (mit $\Delta t := h$ skalierten) Vektor des Richtungsfelds entspricht.

- Das Lösen einer Differentialgleichung bzw. eines Anfangswertproblems bezeichnet man auch als Integrieren der Differentialgleichung $y'(t) = f(t, y)$, denn $y$ ist eine Stammfunktion von $y'$ und nach dem Hauptsatz der Differential- und Integralrechnung (siehe Mathe I) gilt

$$y(b) = y(a) + \int_a^b y'(t)dt \tag{175}$$

  und tatsächlich tut das Euler-Vefahren also nichts anderes als numerisch zu integrieren (vgl. Kapitel 3.3).

**Beispiel 5.9: Euler-Verfahren**

Berechnen Sie für das Anfangswertproblem

$$y'(t) = t^2 + 0.5y(t) \quad \text{mit} \quad y(0) = 0$$

eine Näherung für $y(2)$ mit Hilfe des Euler-Verfahrens und Schrittweite $h = 0.5$ (vgl. Bsp. 5.6).
Lösung:

$$\begin{aligned} y(0) &= y_0 := 0 \\ y(0.5) &\approx y_1 := y_0 + h \cdot (0^2 + 0.5y_0) = 0 \\ y(1) &\approx y_2 := y_1 + h \cdot ((0.5)^2 + 0.5y_1) = (0.5)^3 = \frac{1}{8} = 0.125 \\ y(1.5) &\approx y_3 := y_2 + h \cdot (1^2 + 0.5y_2) = \frac{1}{8} + 0.5(1 + \frac{1}{16}) \\ &= \frac{4 + 16 + 1}{32} = \frac{21}{32} \approx 0.656 \\ y(2) &\approx y_4 := y_3 + h \cdot ((3/2)^2 + 0.5y_3) \approx 0.656 + 0.5(\frac{9}{4} + 0.5 \cdot 0.656) \approx 1.945 \end{aligned} \tag{176}$$

## 5.3 Analyse der Fehlerentwicklung beim Euler-Verfahren

Wir betrachten wieder das Anfangswertproblem $y' = f(t, y)$ mit $y(t_0) = y_0$.

- Der im ersten Schritt des Euler-Verfahrens berechnete Wert $y_1$ weicht vom Wert der exakten Lösung $y(t_1)$ ab.
- Im zweiten Schritt geht man vom Anfangswert $y_1$ im Punkt $t_1$ aus:
  - Sei $z(t)$ die exakte Lösung dieses Anfangswertproblems $z' = f(t, z)$ mit $z(t_1) = y_1$.
  - Ein Schritt im Euler-Verfahren liefert dann

177

$$z_1 = y_1 + h \cdot f(t_1, y_1) = y_2$$

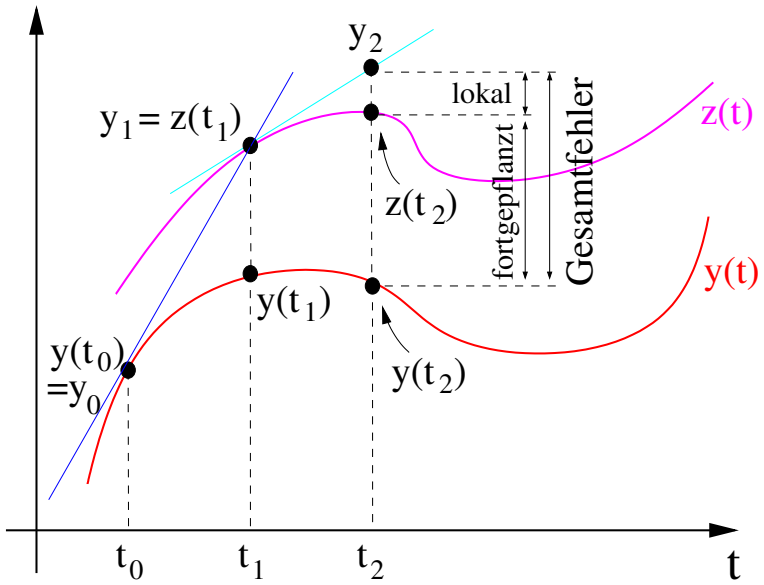
  - Hierbei weicht $z_1 = y_2$ wieder vom Wert der exakten Lösung $z(t_2)$ ab.

Nach zwei Schritten setzt sich der Gesamtfehler von $y_2$ zur exakten Lösung $y(t_2)$ des ursprünglichen Anfangswertproblems wie folgt zusammen,

178

$$y(t_2) - y_2 \;=\; (y(t_2) - z(t_2)) + (z(t_2) - y_2),$$

also aus einem lokalen Fehler $z(t_2) - y_2$ (aus dem letzten Schritt; rechter Summand) und dem aus den vorigen Schritten fortgepflanzten Fehlern $y(t_2) - z(t_2)$ (linker Summand), siehe folgende Skizze:



Wir wollen im folgenden diese beiden Fehler analysieren:
Nach dem Mittelwertsatz (siehe Mathe I) gilt $z(t_2) = z(t_1 + h) = z(t_1) + hz'(\xi) = y_1 + hz'(\xi)$ für ein $\xi \in [t_1, t_1 + h]$, also ist der lokale Fehler in einem Schritt des Euler-verfahrens

$$\phi(t_1, h) := z(t_2) - y_2 = y_1 + hz'(\xi) - (y_1 + hz'(t_1)) = h \cdot (z'(\xi) - z'(t_1)) \tag{179}$$

Da ebenfalls nach dem Mittelwertsatz $z'(\xi) = z'(t_1) + (\xi - t_1) \cdot z''(\eta)$ für ein $\eta \in [t_1, \xi]$ gilt, folgt

$$|z'(\xi) - z'(t_1)| = |(\xi - t_1) \cdot z''(\eta)| \leq h \cdot |z''(\eta)| \tag{180}$$

und damit ist der lokale Fehler beschränkt durch

$$|\phi(t_1, h)| \leq h^2 |z''(\eta)| \leq C \cdot h^2 \tag{181}$$

falls die Krümmung $|z''(t)| = |\frac{d}{dt} f(t, z(t))| \overset{!}{<} C$ für relevante $t$ und $z$ durch eine Konstante $C$ beschränkt ist. In diesem Fall verschwindet der lokale Fehler mit $o(h^2)$ für $h \to 0$.
Allerdings: Je kleiner $h$ ist, desto mehr Schritte $N = (b - a)/h$ sind notwendig um $y(b)$ durch Integration über das Intervall $t \in [a, b]$ zu berechnen. D.h. die Summe über alle $N$ lokalen Fehler ist also

$$N|\phi(t_1, h)| \leq NCh^2 = (b - a)Ch \tag{182}$$

D.h. obwohl der lokale Fehler in einem Schritt mit $o(h^2)$ gegen Null strebt, tut dies die Summe aller lokalen Fehler nur mit $o(h)$. Dies motivert die folgende Definition des lokalen Fehlers:

**Definition 5.10: Lokaler Fehler**

Sei $z$ die exakte Lösung des Anfangswertproblems $z' = f(t, z)$, $z(t_n) = y_n$.
Sei dann $y_{n+1}$ der mit einem numerischen Verfahren mit der Schrittweite $h$ berechnete Näherungswert für $y(t_n + h)$.
Dann ist der lokale Fehler $\phi(t_n, h)$ definiert durch

183
$$\phi(t_n, h) := z(t_n + h) - y_{n+1}$$

Man sagt, das Verfahren hat die Konsistenzordnung $p$, falls gilt:

184
$$|\phi(t_n, h)| \leq C \cdot h^{p+1}$$

für genügend kleine $h$ und eine Konstante $C < 0$, die vom Verfahren und von der Differenzialgleichung abhängt.

D.h. unsere obige Herleitung zeigt, dass das Euler-Verfahren Konsistenzordnung 1 hat.

Bisher haben wir nur die lokalen Fehler berücksichtigt. Lokale Fehler pflanzen sich aber in jedem weiteren Schritt fort und summieren sich zum globalen Fehler auf:

**Definition 5.11: Globaler Fehler**

Sei $y$ die exakte Lösung des Anfangswertproblems $y' = f(t, y)$, $y(t_0) = y_0$ und $y_n$ die mit $n$ Schritten der Schrittweite $h$ des Verfahrens berechnete Näherung an der Stelle $t_n = t_0 + n \cdot h$.

- Dann wird der Gesamtfehler $y(t_n) - y_n$ als globaler Fehler bezeichnet.
- Man sagt, das Verfahren hat die Konvergenzordnung $p$, falls gilt:

  185
  $$|y(t_n) - y_n| \leq C \cdot h^p$$

  für genügend kleine $h$ und eine Konstante $C > 0$, die vom Verfahren und von der Differentialgleichung abhängt.

**Satz 5.12: Konsistenz- und Konvergenzordnung des Euler-Verfahrens**

Das Euler-Verfahren besitzt Konsistenzordnung 1 und Konvergenzordnung 1.

**Beweis:** Die Konsistenzordnung 1 haben wir schon in obiger Herleitung des lokalen Fehlers gezeigt. Der Beweis der Konvergenzordnung 1 funktioniert ähnlich, ist aber etwas technischer. □

## 5.4 Weitere Verfahren

Eine Variante des Euler-Verfahrens führt einen Euler-Schritt mit halber Schrittweite aus, um an dem erhaltenen Punkt die Steigung des Richtungsfeldes auszuwerten:

$$y := y_n + \frac{h}{2} \cdot f(t_n, y_n) \qquad \text{(Euler-Schritt mit } \tfrac{h}{2}\text{)}$$
$$y_{n+1} := y_n + h \cdot f(t_n + \frac{h}{2}, y) \ . \tag{186}$$

Dies ist die sogenannte Mittelpunktsregel:

**Definition 5.13: Mittelpunktsregel**

Das Verfahren

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} \cdot k_1\right) \\ y_{n+1} &= y_n + h \cdot k_2 \end{aligned} \tag{187}$$

heißt Mittelpunktsregel oder modifiziertes Euler-Verfahren. Es besitzt die Konvergenzordnung $p = 2$.

Eine weitere Verbesserung des Euler-Verfahrens ist, einen Euler-Schritt mit Schrittweite $h$ auszuführen und an dem so erhaltenen Punkt die Steigung des Richtungsvektors auszuwerten.
Den arithmetischen Mittelwert dieser Steigung mit der Steigung im Ausgangspunkt verwendet man dann als Steigung um einen Schritt vom Ausgangspunkt auszuführen.

Dies führt auf das Verfahren von Heun:

**Definition 5.14: Verfahren von Heun**

Das Verfahren

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + h, y_n + h \cdot k_1\right) \\ y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2) \end{aligned} \tag{188}$$

heißt Verfahren von Heun und besitzt die Konvergenzordnung $p = 2$.

## 5.5 Gewöhnliche Differentialgleichungen höherer Ordnung

**Beispiel aus der Physik:** Wir betrachten eine beschleunigte Bewegung eines Körpers mit Beschleunigung $a(t)$, Geschwindigkeit $v(t)$ und Weg bzw. Höhe $h(t)$. Aus der Physik wissen wir, dass die Geschwindigkeit die Ableitung (also zeitliche Änderung) des Weges und die Beschleunigung die Ableitung der Geschwindigkeit ist:

189
$$a(t) = \dot{v}(t) = \frac{dv}{dt} = \frac{d^2h}{dt^2} = \ddot{h}(t) = h^{(2)}(t)$$
$$v(t) = \dot{h}(t) = \frac{dh}{dt} = h^{(1)}(t) = h'(t)$$

Die Beziehung zwischen Beschleunigung $a$ und Höhe $h$ ergibt also eine Differentialgleichung höherer Ordnung:

190
$$a(t) = h^{(2)}(t) \qquad (= h''(t)) \ .$$

Allgemein hat eine Differentialgleichung höherer Ordnung die Form

191
$$y^{(j)}(t) = f\left(t, y(t), y^{(1)}(t), \ldots, y^{(j-1)}(t)\right) \ .$$

Wir zeigen im folgenden, wie man eine Differentialgleichung höherer Ordnung in ein gleichwertiges System von Differentialgleichungen erster Ordnung überführt:

Dazu führen wir den Vektor

192
$$u(t) = \left(u_1(t) \ldots u_j(t)\right)$$

ein und definieren die höheren Ableitungen als eigenständige Funktionen:

193
$$u_1(t) := y(t), \quad u_2(t) := y'(t), \quad u_3(t) := y^{(2)}(t), \ldots, u_j(t) := y^{(j-1)}(t)$$

Daraus folgt das Gleichungssystem

194
$$u'(t) = \begin{pmatrix} u_1'(t) \\ u_2'(t) \\ \vdots \\ u_j'(t) \end{pmatrix} = \begin{pmatrix} y'(t) \\ y^{(2)}(t) \\ \vdots \\ y^{(j)}(t) \end{pmatrix} = \begin{pmatrix} u_2(t) \\ u_3(t) \\ \vdots \\ f\left(t, u_1(t), u_2(t), \ldots, u_j(t)\right) \end{pmatrix}$$

Dieses System entspricht (bis auf die mehrdimensionale Vektordarstellung) genau der Form von Def. 5.2. Man kann dieses System ähnlich wie in Kapiteln 5.2 und 5.4 beschrieben z.B. mit dem Euler-Verfahren (komponentenweise) lösen.

# Stichwortverzeichnis