Skript zur Vorlesung

Numerisches Praktikum

Wintersemester 2005/2006

Universität Konstanz
Dr. Eberhard Luik

private Mitschrift

Stand: 17. Februar 2006
www.meidert.net/uni

Achtung:
Dies ist kein offizielles Skript, sondern nur eine private Mitschrift. Ich kann daher keine Gewähr für die Richtigkeit und Vollständigkeit übernehmen. Vor allem können die Nummerierungen zum Teil von den in den Vorlesungen verwendeten abweichen. Falls jemand einen Fehler entdeckt, so möge er/sie mir bitte eine eMail schicken - vielen Dank!

Frieder Meidert (uni@meidert.net)

Inhaltsverzeichnis

1. Numerisches Rechnen

a. Zahlen und ihre Darstellung

Sei $x \in \mathbb{R} \setminus \{0\}, g \in \mathbb{N}, g \geq 2$.

$$x = \sigma \cdot g^N \cdot \sum_{\nu=1}^{\infty} x_\nu \cdot g^{-\nu} \tag{1}$$

mit $\sigma \in \{1, -1\}$ (Vorzeichen), $N \in \mathbb{Z}$, $x_\nu \in \{0, 1, \ldots, g-1\}$

Zusatzforderung:

- $x_1 \neq 0$
- zu jedem $\nu \in \mathbb{N}$ gibt es ein $n > \nu$ mit $x_n \neq g - 1$.

Damit wird die Darstellung 1 eindeutig (normalisierte Darstellung).

Schreibweise $\sigma \cdot 0, x_1, x_2, x_3 \ldots g^N$ (Gleitkommazahl)

Beispiele:
$g = 10$: Dezimalsystem: Ziffern $0, \ldots, 9$
$g = 2$: Dualsystem: Ziffern $0, 1$
$g = 16$: Hexadezimalsystem: Ziffern $0, 1, \ldots, 9, A, B, C, D, E, F$

z.B. $x = 11,1875$
$$\begin{cases} \textit{dezimal} & 0,111875 \cdot 10^2 \\ \textit{dual} & 0,10110011 \cdot 2^4 \\ \textit{hexadezimal} & 0,B3 \cdot 16^1 = 16 \cdot \left(\frac{11}{16} + \frac{3}{256}\right) \end{cases}$$

$$11,1875 = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 + 0 \cdot \tfrac{1}{2} + 0 \cdot \tfrac{1}{4} + 1 \cdot \tfrac{1}{8} + 1 \cdot \tfrac{1}{16} = 16 \cdot \left(1 \cdot \tfrac{1}{2} + 0 \cdot \tfrac{1}{4} \cdot \ldots\right)$$

Computer: $g = 2$
Darstellbar sind dann die 0 und Zahlen der Form $\sigma \cdot 2^N \cdot \sum_{\nu=1}^{t} x_\nu 2^{-\nu}$ mit $\sigma \in \{1, -1\}, N^- \leq N^+, x_{nu} \in \{0, 1\}, x_1 = 1$ und einem festen t.
z.B. 64 Bit: 1 Bit: Vorzeichen, 11 Bit Hochzahl, 52 Bits: Mantisse $(x_1, \ldots, x_{52})$ (IEEE-Standard)

Definition:
Die Menge M der Zahlen der Form (2) heißt **Maschinenzahlen** (endliche Menge!)

Beispiel: nicht von der Form (2): $0,1$

Ist x eine Maschinenzahl $\neq 0$, so gilt:
$g^{N^- - 1} \leq |x| < g^{N^+}$

$y \in \mathbb{R}$ mit $|y| \geq g^{N^+}$: Exponentenüberlauf
$y \in \mathbb{R}$ mit $|y| < g^{N^- - 1}$: wird (in der Regel) durch Null ersetzt.
Beispiel: $\sum_{n=1}^{\infty} \frac{1}{n}$ konvergiert im PC!

b. Operationen mit Gleitpunktzahlen

Sei $x \in \mathbb{R}$ mit $|y| < g^{N^+}, x = \sigma \cdot g^N \cdot \sum_{\nu=1}^{\infty} \cdot g^{-\nu}$

$$rd(x) := \begin{cases} \sigma \cdot g^N \cdot \sum_{\nu=1}^{t} x_\nu g^{-\nu} & , \; falls \; x_{t+1} < \frac{g}{2} \; abrunden \\ \sigma \cdot \gamma^N \cdot \left(\sum_{\nu=1}^{t} x_\nu g^{-\nu} + g^{-t} \right) & , \; falls \; x_{t+1} \geq \frac{g}{2} \; aufrunden \end{cases}$$

Sei $\overline{x}$ ein Näherungswert.

Definition:

(a) $x - \overline{x}$ heißt **absoluter Fehler**,

(b) $\frac{x-\overline{x}}{x}$ heißt **relativer Fehler** (für $x \neq 0$).

Satz:

(a) $|rd(x) - x| \leq 0,5 g^{N-t}$,

(b) $\left|\frac{rd(x)-x}{x}\right| \leq 0,5 g^{-t+1}$ (2^{-t}) im Zweiersystem

Definition:
Die Zahl $\tau := 0,5 \cdot g^{-t+1}$ heißt **relative Rechengenauigkeit**.

$1 + \frac{1}{2}$
$1 + \frac{1}{2^2}$

$\vdots$
$1 + \frac{1}{2^{52}} > 1$
$1 + \frac{1}{2^{-53}} = 1$ (2^{-53} kann noch exakt dargestellt werden)
$\Rightarrow 2^{-52} \widetilde{=}$ relative Rechengenauigkeit

$g^{N^-} < 2^{-53}$

Setze $\varepsilon := \frac{rd(x)-x}{x} \Rightarrow rd(x) = x(1+\varepsilon)$ mit $|\varepsilon| \leq \tau$.

Verknüpfung von Gleitkommazahlen:
$\diamond \in \{+, \cdot, -, /\}$
$x, y \in M$: dann ist im Allgemeinen $x \diamond y \notin M$
$\Rightarrow$ Runden: $rd(x \diamond y) = (x \diamond y)(1+\varepsilon_1)$ mit $|\varepsilon| \leq \tau$.

Beispiel: :
$g := 19, t := 3$
$x = 0,123 \cdot 10^6$ (Maschinenzahl)
$y = 0,456 \cdot 10^2$
$x + y = 0,1230456 \cdot 10^6 \notin M$
$rd(x+y) = 0,123 \cdot 10^6 = x$

$x, y \in \mathbb{R}$:
$rd(x) = x(1+\varepsilon_1)$ mit $|\varepsilon_1| \leq \tau$
$rd(y) = y(1+\varepsilon_2)$ mit $|\varepsilon_2| \leq \tau$

$rd(x) + rd(y) = x(1+\varepsilon_1) + y(1+\varepsilon_2) = x + \varepsilon_1 x + y + \varepsilon_2 y$
$rd(x) + rd(y) - (x+y) = \varepsilon_1 x + \varepsilon_2 y$ (absoluter Fehler)
$\frac{rd(x)+rd(y)-(x+y)}{(x+y)} = \frac{\varepsilon_1 x}{x+y} + \frac{\varepsilon_2 y}{x+y}$ (relativer Fehler)

Fall $\text{sign}(x) = \text{sign}(y)$:
$\left|\frac{rd(x)+rd(y)-(x+y)}{(x+y)}\right| \leq \left|\frac{\varepsilon_1 x}{x+y}\right| + \left|\frac{\varepsilon_2 y}{x+y}\right| = |\varepsilon_1| \underbrace{\left|\frac{x}{x+y}\right|}_{<1} + |\varepsilon_2| \underbrace{\left|\frac{y}{x+y}\right|}_{<1} \leq |\varepsilon_1| + |\varepsilon_2| \leq 2\tau$, also unproblematisch!
Fall $x \approx -y \Rightarrow$ großer relativer Fehler.

Achtung: Problematisch ist die Subtraktion zweier ungefähr gleicher Zahlen. (siehe Aufgabe I, 1a)

$rd(rd(x) \diamond rd(y)) = x \diamond y + F$.
$rd(rd(x)+rd(y)) = (x(1+\varepsilon_1)+y(1+\varepsilon_2))(1+\varepsilon_3) = \ldots \leq x+y+(|x|+|y|)\cdot 2\tau(|x|+|y|)\cdot\tau^2 =$
$x + y + F$ mit $|F| \leq (|x| + |y|)(\tau^2 + 2\tau)$.

Achtung: Im Computer kann man nicht beliebig klammern:

$$(x + y) + z \neq x + (y + z) \text{ (wegen der Rundung)}$$

c. Algorithmen

Algorithmus: Beschreibung einer Methode zur Lösung einer gegebenen Aufgabenstellung

- Bestimmtheit
- Endlichkeit (nach endlich vielen Schritten am Ziel)
- Allgemeingültigkeit

Auswahlkriterien unter Algorithmen: Rechenzeit, Stabilität.

Definition:
Sei A ein Algorithmus. Die Abbildung $T_A : \mathbb{N} \to \mathbb{N}$, welche jeder Anzahl von Eingabedaten die Anzahl der Rechenoperationen zuordnet, heißt **Komplexität von** A.

Beispiel:

1) $T_A(n) = \frac{1}{2}n(n+3) = \frac{1}{2}n^2 + \frac{3}{2}n = \sigma(n^2)$ (für $n \to \infty$)

2) $p(t) = a_0 + a_1 t + \ldots + a_n t^n$, $\xi \in \mathbb{R}$, Berechne $p(\xi)$.
   A_1: Eingabe $n, (a_0, \ldots, a_n), \xi$
   $p := a_0$.
   Für $i := 1, \ldots, n$:
   $b = a_i$
   Für $j = 1, \ldots, i$:
   $b = b \cdot \xi$
   {Ende der j-Schleife}
   $p = p + b$
   {Ende der i-Schleife}

Ausgabe p.

n Additionen, $(1 + 2 + \ldots + n) = \frac{n(n+1)}{2}$ Multiplikationen

$\Rightarrow T_{A_1}(n) = \frac{1}{2}n^2 + \frac{3}{2}n = \sigma(n^2)$.

d. Das Hornerschema

$p(t) = a_0^{(0)} + a_1^{(0)}t + \ldots + a_n^{(0)}t^n; \xi \in \mathbb{R}$; gesucht: $p(\xi)$
Setze $a_n^{(n)} = a_n^{(0)}$
A_2:
Für $i = 1, \ldots, n$:
$a_{n-1}^{(1)} := a_{n-i}^{(0)} + a_{n+1-i}^{(1)} \cdot \xi$.
Behauptung: $a_0^{(1)} = p(\xi)$ (vgl. Übungsaufgaben)
n Additionen, n Multiplikationen
$T_{A_2}(n) = 2n = \sigma(n)$.

2. Interpolation

a. Problemstellung

Gegeben seien $(t_i, s_i) \in \mathbb{R}^2$, $i = 0, \ldots, m$, $t_0, \ldots, t_m$ paarweise verschieden.
Gesucht ist ein $p(t) := \sum_{j=0}^{m} a_i t^i$ mit $p(t_i) = s_i$ $(i = 0, \ldots, m)$ (1)

Interpolationsaufgabe nach Lagrange:
Die (t_i, s_i) heißen **Stützpaare**.

Anwendung:
$f \in C([a, b]; \mathbb{R})$, $a \leq t_0 < t_1 < \ldots < t_m \leq b$, $s_i := f(t_i)$
Gesucht: Interpolationspolynom.

Fall $m = 1$: klar (Gerade)
Fall $m = 2$: (Parabel)

b. Existenz und Eindeutigkeit

Satz:
Die Interpolationsaufgabe nach Lagrange besitzt genau eine Lösung.

Beweis:

(i) Eindeutigkeit:
Seien p und q Lösungen der Interpolationsaufgabe nach Lagrange.
Bezeichnung: Π_m sei der Raum der Polynome vom Grad $\leq m$. D.h. $p, q \in \Pi_m$, und erfüllen (1).
Setze $r(t) := p(t) - q(t)$. Dann gilt $r \in \Pi_m$. Es ist $r(t_i) = p(t_i) - q(t_i) = 0$ für $i = 0, \ldots, m \Rightarrow r$ hat mindestens $m + 1$ paarweise verschiedene Nullstellen $\Rightarrow r \equiv 0 \Rightarrow p = q$.

(ii) Existenz:
Wir setzen $l_i(t) := \prod_{j=0, j \neq i}^{m} \left(\frac{t - t_j}{t_i - t_j} \right)$ für $i = 1, \ldots, m$, wohldefiniert, da Nenner stets $\neq 0$.
Es gilt $l_i \in \Pi_m$ für $i = 1, \ldots, m$

$$l_i(t_k) = \begin{cases} 0 & , i \neq k \\ 1 & , i = k \end{cases}$$

Die l_i heißen **Lagrange-Grundpolynome**.

$$\boxed{p(t) := \sum_{i=0}^{m} s_i l_i(t)} \; ; p \in \Pi_m \tag{2}$$

Es gilt $p(t_k) = \sum_{i=0}^{m} s_i l_i(t_k) = s_k l_k(t_k) = s_k$ für $k = 0, \ldots, m$. □

(2) heißt **Interpolationsformel nach Lagrange**.

Bemerkung:
Das Interpolationspolynom ist abhängig von der Stützstellenwahl.

Gegeben ein ξ, $p(\xi) =?$.

c. Der Neville-Algorithmus

Seien $(t_i, s_i) \in \mathbb{R}^2$ mit t_i paarweise verschieden für $i = 0, \ldots, m$. $p(t)$ sei das Interpolationspolynom nach Lagrange.
Gesucht $p(\xi)$ für ein gegebenes $\xi \in \mathbb{R}$.
$p_{m-1}(t)$ sei das Interpolationspolynom zu den Stützpaaren $(t_0, s_0), \ldots, (t_{m-1}, s_{m-1})$
$q_{m-1}(t)$ sei das Interpolationspolynom zu den Stützpaaren $(t_1, s_1), \ldots, (t_m, s_m)$

Lemma:
Es gilt:

$$p(t) = \frac{(t-t_m)p_{m-1}(t)-(t-t_0)q_{m-1}(t)}{t_0-t_m} =: r(t)$$

Beweis:
$r \in \Pi_m$ klar.
$r(t_i) = \frac{(t_i-t_m)p_{m-1}(t_i)-(t_i-t_0)\cdot q_{m-1}(t_i)}{t_0-t_m} = \frac{(t_i-t_m)s_i-(t_i-t_0)s_i}{t_0-t_m} = s_i$ für $1 \leq i \leq m-1$,
$r(t_0) = \frac{(t_0-t_m)s_0-0}{t_0-t_m} = s_0$,
$r(t_m) = \frac{-(t_m-t_0)s_m}{t_0-t_m} = s_m$.

$\Rightarrow r(t) = p(t)$ □

Bezeichnung:
$p_{i_0,i_1,\ldots,i_k} \in \Pi_k$ sei das Interpolationspolynom zu den Stützpaaren $(t_{i_0}, s_{i_0}), (t_{i_1}, s_{i_1}), \ldots, (t_{i_k}, s_{i_k})$.

d. Der Interpolationsfehler

$f : [a,b] \to \mathbb{R}; a \le t_0 < t_1 < \ldots < t_m \le b$ $(s_i = f(t_i))$
$p(t)$ sei das Interpolationspolynom zu f.

Satz:
Sei $f \in C^{m+1}[a,b]$, sei $z \in [a,b]$; $\alpha := \min\{z, t_0\}, \beta := \max\{z, t_m\}$.
Es gibt ein $\xi \in (a,b)$ mit

$$f(z) - p(z) = \frac{f^{(m+1)}(\xi)}{(m+1)!} \underbrace{\prod_{i=0}^{m} (z - t_i)}_{=:w(z)}.$$

Beweis:
Setze $\Phi(t) := f(t) - p(t) - \frac{w(t)}{w(z)}(f(z) - p(z))$
$\Phi^{(m+1)}(t) = f^{(m+1)}(t) - 0 - \frac{(m+1)!}{w(z)}(f(z) - p(z))$

Für $z \in \{t_0, \ldots, t_m\}$ klar.
Sei deshalb $z \notin \{t_0, \ldots, t_m\}$

$\Phi(t) := f(t) - p(t) - \frac{w(t)}{w(z)}(f(z) - p(z))$
$\Phi^{(m+1)}(t) = f^{(m+1)}(t) - \frac{(m+1)!}{w(z)}(f(z) - p(z))$
Sei ξ mit $\Phi^{(m+1)}(\xi) = 0$.
$0 = f^{(m+1)}(\xi) - \frac{(m+1)!}{w(z)}(f(z) - p(z)) \Rightarrow$ Behauptung.

$$\Phi(t_k) = \underbrace{f(t_k)}_{=s_i} - \underbrace{p(t_k)}_{=s_i} - \overbrace{\frac{w(t_k)}{w(z)}}^{=0}(f(z) - p(z)) = 0 \text{ für } k = 0, \ldots, m.$$

$\Phi(z) = 0$
Φ besitzt in $[\alpha, \beta]$ mindestens $m + 2$ Nullstellen.

Φ' besitzt in (α, β) mindestens $m+1$ Nullstellen.
$\vdots$
$\Phi^{(m+1)}$ besitzt in (α, β) mindestens eine Nullstelle. □

Fehlerschranken:
zu $g \in C[a,b]$ sei $\|g\| := \max\{(g(t)) : t \in [a,b]\}$.
$|f(z) - p(z)| \leq \frac{\|f^{(m+1)}\|}{(m+1)!}|w(z)|$
$\|f - p\| \leq \frac{\|f^{(m+1)}\|}{(m+1)!}\|w\|$
$\|w\|$ ist abhängig von der Stützstellenwahl $t_0, \ldots, t_m$

→ günstige Stützstellenwahl??
(d.h. so, dass $\|w\|$ minimal wird)

$w(t) = t^{m+1} + a_m t^m + \ldots + a_0$ $Q_n := \{p \in \Pi_n | p(t) = t^n + a_{n-1} + \ldots + a_0\}$.

e. Tschebyscheff-Polynome

Zu jedem $n \in \mathbb{N}$ definiert man

$$T_n : [-1,1] \to \mathbb{R},\ T_n(t) := \cos(n \cdot \arccos(t))$$

Lemma:
Für $n \geq 1$ gilt:

$$T_{n+1}(t) = 2tT_n(t) - T_{n-1}(t)$$

$T_{n+1}(t) = \cos((n+1) \cdot \arccos(t)) =$
$= \cos(n \cdot \arccos(t)) \cdot \cos(\arccos(t)) - \sin(n \cdot \arccos(t)) \cdot \sin(\arccos(t))$
$T_{n-1}(t) = \cos((n-1) \cdot \arccos(t)) =$
$= \cos(n \cdot \arccos(t)) \cdot \cos(\arccos(t)) + \sin(n \cdot \arccos(t)) \cdot \sin(\arccos(t))$
$T_{n+1} + T_{n-1} = 2 \cdot \cos(n \cdot \arccos(t)) \cdot t = 2tT_n(t)$ □

Satz:
Damit sind wegen $T_0 = 1$ und $T_1 = t$ alle T_n, $n \in \mathbb{N}$ Polynome vom Grad n.
$T_n(t) = 2^{n-1}t^n + a_{n-1}t^{n-1} + \ldots + a_0$.

Beweis:
Induktion mit vorangegangenem Lemma.

Bezeichnung:
Die T_n heißen **Tschebyscheff-Polynome 1. Art**.

$T_0(t) = 1$, $T_1(t) = t$, $T_2(t) = 2t^2 - 1$, $T_3(t) = 4t^3 - 3t$.

Eigenschaften:

(1) Symmetrie: $T_n(-t) = (-1)^n T_n(t)$,

(2) $T_n(1) = 1, T_n(-1) = (-1)^n, |T_n(t)| \leq 1 \ \forall\ t \in [-1,1] \Rightarrow \|T_n\|_{[-1,1]} = 1$,

(3) Nullstellen: T_n hat n einfache Nullstellen in $[-1,1]$:
$t_i = \cos\left(\frac{2i+1}{2n} \cdot \pi\right)$ für $i = 0, \ldots, n-1$,

(4) Extremstellen: T_n hat in $[-1,1]$ genau $(n+1)$ Extremstellen:
$z_i := \cos\left(\frac{i}{n}\pi\right)$ für $i = 0, 1, \ldots, n$
$T_n(z_i) = (-1)^i$ für $i = 0, \ldots, n$,

(5) Minimalitätseigenschaft: sei $\widehat{T}_n := \frac{T_n}{2^{n-1}} \Rightarrow \widehat{T}_n(t) = t^n + c_{n-1}t^{n-1} + \ldots$, d.h. $\widehat{T}_n \in \mathbb{Q}_n$

Satz:
Unter allen Polynomen aus Q_n hat $\widehat{T}_n$ die minimale Norm.

Beweisidee:
Sei $q \in Q_n$ mit $\|q\| < \|\widehat{T}_n\|$
$r(t) := \widehat{T}_n(t) - q(t) \Rightarrow r \in \Pi_{n-1}$
$r(z_0) = \widehat{T}_n(z_0) - q(z_0) > 0$
$r(z_1) = \widehat{T}_n(z_1) - q(z_1) < 0$
$r(z_2) = \widehat{T}_n(z_2) - q(z_2) > 0$
$\vdots$

$\Rightarrow r$ hat Grad $\leq n-1$ mit mindestens n Nullstellen, da $r \neq 0$ (da $\|q\| < \|\widehat{T}_n\|$) ist dies ein Widerspruch.

Wähle bei der Interpolation die Nullstellen $t_0, \ldots, t_m$ von $T_{n+1}(t)$.
Dann gilt (bezüglich $[-1,1]$):

$\|f - p\|_{[-1,1]} \leq \frac{\|f^{(m+1)}\|_{(-1,1)}}{(m+1)!} \cdot \frac{1}{2^m} \cdot c.$

f. Konvergenzfragen

Sei $f \in C([a,b])$; zu $m \in \mathbb{N}$ seien $a \leq t_{m_0} < t_{m_1} < \ldots < t_{m_m} \leq b$.
Weiter sei $p_m(t)$ das Interpolationspolynom zu f bezüglich $t_{m_0}, \ldots, t_{m_m}$.

Frage:

(1) Gilt gleichmäßige Konvergenz? Also $\|p_m - f\| \to 0$ für $m \to \infty$?

(2) Gilt punktweise Konvergenz? Also $p_m(x) \to f(x)$ für alle $x \in [a,b]$.?

Wir ordnen die Stützstellen in einer Matrix an:

$$S = \begin{pmatrix} t_{00} & & & \\ & t_{11} & & \\ \vdots & & \ddots & \\ t_{m0} & \ldots & \ldots & t_{mm} \\ \vdots & & & \end{pmatrix}$$ - Stützstellenmatrix

Satz: (Faber)
Zu jeder Stützstellenmatrix S gibt es ein $f \in C[a,b]$, so dass die Folge $(p_m)_{m\in\mathbb{N}}$ der Interpolationspolynome nicht gleichmäßig konvergiert.

Satz: (Marcinkiewicz)
Zu jedem $f \in C[a,b]$ gibt es eine Stützstellenmatrix S, so dass die Folge $(p_m)_{m\in\mathbb{N}}$ der Interpolationspolynome gleichmäßig gegen f konvergiert.

Aber: S ist im Allgemeinen nicht bekannt!

Beispiel:
$f(x) = \exp(x)$ im Intervall $[a,b]$.
Sei $p_m(x)$ das Interpolationspolynom zu f bezüglich $a \leq t_{m0} < \ldots < t_{mm} \leq b$.
Behauptung: $\|p_m - f\| \to 0$ für $m \to \infty$.

Beweis:
Für den Interpolationsfehler gilt $\|p_m - f\|_{[a,b]} \leq \frac{\|f^{(m+1)}_{[a,b]}}{(m+1)!}\|w\|_{[a,b]}$.

$|w(t)| := \prod_{i=0}^{m} |(t - t_i)| \leq (b-a)^{m+1} \Rightarrow \|w\|_{[a,b]} \leq (b-a)^{m+1}$
$\|f^{(m+1)}\|_{[a,b]} = \|\exp\|_{[a,b]} = e^b.$

Somit $\|p_m - f\|_{[a,b]} \leq \frac{e^b(b-a)^{m+1}}{(m+1)!} \to 0$ für $m \to \infty$ □

g. Weitere Interpolationsarten

Hermite Interpolation:
Gegeben seien $(t_i, s_i^{(0)}, s_i^{(1)}) \in \mathbb{R}^3$, t_i paarweise verschieden.
Gesucht ist ein $p \in \Pi_{2_m+1}$ mit $(i = 0, \ldots, m)$

- $p(t_i) = s_i^{(0)}$ für $i = 0, \ldots, m$;
- $p'(t_i) = s_i^{(1)}$ für $i = 0, \ldots, m$.

Satz:
Es gibt genau ein solches Polynom.

Beweis:
Eindeutigkeit: ähnlich wie bei Lagrange.
Existenz:

$$p(t) = \sum_{k=0}^{m} s_k^{(0)} l_k^2(t)[1 - 2l_k'(t_k)(t - t_k)] + \sum_{k=0}^{m} s_k^{(1)} l_k^2(t)(t - t_k)$$

Verallgemeinerung:
auch höhere Ableitungen → verallgemeinrtes Hermite-Interpolationspolynom.

Rationale Interpolation:
$R_{m,n}(t) := \frac{a_0 + a_1 t + \ldots + a_m t^m}{b_0 + b_1 t + \ldots + b_n t^n}$

Gegeben seien $m + n + 1$ Paare $(t_i, s_i) \in \mathbb{R}^2$, t_i paarweise verschieden.
Gesucht ist eine rationale Funktion $R_{m,n}$ mit

$$R_{m,n}(t_i) = s_i \text{ für } i = 0, \ldots, m + n$$

Achtung: Existenz nicht immer gesichert!

Trigonometrische Interpolation:
Eine Funktion der Form

$$s_m(t) := a_0 + \sum_{i=1}^{m} a_k \cos(k \cdot t) + \sum_{k=1}^{m} b_k \sin(k \cdot t)$$

heißt trigonometrisches Polynom vom Grad m

Gegeben seien $2m + 1$ Paare $(t_i, s_i) \in \mathbb{R}^2$ mit $-\pi \leq t_0 < \ldots < t_m < \pi$.
Gesucht ist ein $S_m(t)$ mit

$$S_m(t_i) = s_i \text{ für } i = 0, \ldots, 2m$$

Es gibt genau ein solche S_m.

Spline-Interpolation:
Sei $a_0 = t_0 < t_1 \ldots < t_m = b$

Definition:
Unter einer (kubischen) **Splinefunktion** versteht man eine Funktion $h : [a, b] \to \mathbb{R}$ mit folgenden Eigenschaften:

1) $h \in C^2[a, b]$;
2) auf jedem Teil-Intervall $[t_i, t_{i+1}]$ gilt $g|_{[t_i, t_{i+1}]} \in \Pi_3$.

Sei nun $(t_i, s_i) \in \mathbb{R}$ für $i = 0, \ldots, m$
Gesucht ist eine kubische Splinefunktion mit $h(t_i) = s_i$ für $i = 0, \ldots, m$.

Es gibt eine Lösung, aber keine eindeutige Lösung.

Zusatzforderung:
$h^{(2)}(t_0) = h^{(2)}(t_m) = 0$.

3. Nullstellenbestimmung

a. Bisektionsverfahren

Sei $f \in C[a,b]$. Gesucht: $\xi \in [a,b] = I_0$ mit $f(\xi) = 0$.

Es gelte $f(a) \cdot f(b) < 0$.
Zwischenwertsatz $\Rightarrow$ es gibt ein $\xi \in (a,b)$ mit $f(\xi) = 0$.
Bilde $s = \frac{a+b}{2}$.
Dann gilt genau einer der folgenden 3 Fälle:

(1) $f(s) = 0 \Rightarrow s = \xi$;

(2) $f(a) \cdot f(s) < 0 \Rightarrow \xi \in (a,s) =: I_1$;

(3) $f(b) \cdot f(s) < 0 \Rightarrow \xi \in (s,b) =: I_1$.

Fortgesetzte Intervall-Halbierung: I_n
Sei $u_n \in I_n$: dann gilt $|u_n - \xi| \leq \frac{b-a}{2^n} \to 0$ für $n \to \infty$.

$\Rightarrow$ Eindeutigkeit (**natürlicher kubischer Spline**)

b. Newton-Verfahren

Herleitung:
$f \in C^2[a,b]$, sei $\xi \in [a,b]$ mit $f(\xi) = 0$ und $f'(t) \neq 0$ in $[a,b]$.
Sei $x_0 \in [a,b]$ ein Näherungswert an ξ.
Ziel: Berechnung eines besseren Näherungswertes x_1.

Taylor:

$$0 = f(\xi) = f(x_0) + f'(x_0)(\xi - x_0) + \underbrace{\frac{f^{(2)}(\eta)}{2!}(\xi - x_0)^2}_{\text{vernachlässigbar}}$$

$\Rightarrow 0 \approx f(x_0) + f'(x_0)(\xi - x_0)$
$\Rightarrow 0 = f(x_0) + f'(x_0)(x_1 - x_0)$
$\Rightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$

Wiederhole diesen Vorgang:

$$\boxed{x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \text{ für } i = 0, 1, \ldots}$$

Diese Rechenvorschrift heißt **Newton-Verfahren**, x_0 heißt **Startwert**.

Fragen:

- durchführbar? (also $x_i \in [a, b]$ für alle i?)
- Konvergenz gegen ξ?
- Konvergenzgeschwindigkeit?
- Wahl des Startwertes?

Iterationsverfahren:
$\Phi : [a, b] \to [a, b]$ stetig.
Zu einem Startwert $x_0 \in [a, b]$ berechnet man

$$x_{i+1} = \Phi(x_i) \ (i = 0, 1, \ldots)$$

(Φ heißt dann **Iterationsfunktion**)

Ein $\xi \in [a, b]$ heißt Fixpunkt von Φ, falls $\Phi(\xi) = \xi$ gilt.

Frage:
$\lim_{i \to \infty} x_i = \xi$?
$\to$ Banachscher Fixpunktsatz.

Newton-Verfahren: $\Phi(t) = t - \frac{f(t)}{f'(t)}$

c. Konvergenz des Newton-Verfahrens

Definition:

(1) Konvergiert die Folge $(x_i)_{i \in \mathbb{N}}$ für jeden Startwert $x_0 \in [a, b]$, so heißt das Iterationsverfahren **global konvergent**.

(2) Konvergiert die Folge $(x_i)_{i\in\mathbb{N}}$ nur für alle Startwerte $x_0 \in U$ (U eine Umgebung um ξ), so heißt das Verfahren **lokal konvergent**.

Definition:
Sei U eine Umgebung um ξ. Gilt für alle Startwerte $x_0 \in U$ für die Iterationsfolge eine Ungleichung der folgenden Form:

$$|x_{i+1} - \xi| \leq C|x_i - \xi|^p$$

($p \geq 1$, C eine Konstante - falls $p = 1$, so gelte $C < 1$ - , unabhängig von x_0)
so nennt man das Verfahren ein Verfahren der Konvergenzordnung p.

$p = 1$: lineare Konvergenz
$p = 2$: quadratische Konvergenz

Satz:
Sei $f \in C^2[a,b]$; $\xi \in [a,b]$ mit $f(\xi) = 0$ und $f'(t) \neq 0$ für alle $t \in [a,b]$.
Das Newton-Verfahren liefert für jeden Startwert $x_0 \in [a,b]$ die Ungleichung

$$|x_1 - \xi| \leq C|x_0 - \xi|^2$$

mit einer von x_0 unabhängigen Konstanten $c > 0$.

Beweis:
$0 = f(\xi) = f(x_0) + f'(x_0)(\xi - x_0) + \frac{f^{(2)}(\eta)}{2!}(\xi - x_0)^2$
$\Rightarrow \xi - \underbrace{\left(x_0 + \frac{f(x_0)}{f'(x_0)}\right)}_{=x_1} = -\frac{f^{(2)}(\eta)}{2!f'(x_0)}(\xi - x_0)^2.$
$\Rightarrow |\xi - x_1| = \frac{|f^{(2)}(\eta)}{2|f'(x_0)|}|\xi - x_0|^2$

$C_1 := \max\{|f^{(2)}(t)| : t \in [a,b]\} < \infty$
$C_2 := \min\{|f'(t)| : t \in [a,b]\} > 0$

$$|\xi - x_1| \leq \underbrace{\frac{C_1}{2C_2}}_{=:C} |\xi - x_0|^2 = C|\xi - x_0|^2$$ □

Satz:
Sei $f \in C^2[a,b]$; $\xi \in (a,b)$ mit $f(\xi) = 0$ und $f'(\xi) \neq 0$. Dann gilt:

(1) Das Newton-Verfahren besitzt Konvergenzordnung 2

(2) Es gibt eine Umgebung U von ξ so, dass für jeden Startwert $x_0 \in U$ das Newton-Verfahren gegen ξ konvergiert.

Beweis:
Sei C wie oben, sei $\varepsilon := \min\{\frac{1}{2C}, |\xi - a|, |\xi - b|\} > 0$
$U := (\xi - \varepsilon, \xi + \varepsilon)$
Sei $x_0 \in U \Rightarrow |x_1 - \xi| \leq C|\underbrace{x_0 - \xi}_{<\varepsilon}|^2 < C \cdot \varepsilon^2 \leq C \cdot \frac{1}{2C} \cdot \varepsilon = \frac{\varepsilon}{2}$

$\Rightarrow x_1 \in U \Rightarrow x_i \in U$ für $i = 1, 2, \ldots$
$\Rightarrow$ Konvergenzordnung 2.

Es gilt

$$|x_{i+1} - \xi| \leq C|x_i - \xi|^2 < \left(\tfrac{1}{2}\right)^{i+1} \cdot \varepsilon \to 0 \text{ für } i \to \infty.$$

Sei $f : [a,b] \to \mathbb{R}$ mit

(V1) es gibt ein $\xi \in [a,b]$ mit $f(\xi) = 0$,

(V2) $f \in C^1[a,b]$ und $f'(t) \neq 0$ für alle $t \in [a,b]$,

(V3) $f'(s)(t-s) + f(s) \leq f(t)$ für alle $s, t \in [a,b]$,

(V4) Für jeden Startwert $x_0 \in [a,b]$ gilt $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$.

Lemma:
Sei $f \in C^2[a,b]$ mit $f^{(2)}(t) \geq 0$ für alle $t \in [a,b]$. Dann folgt daraus (V3).

Beweis:
$$f(t) = f(s) + f'(s)(t-s) + \underbrace{\frac{f^{(2)}(\eta)}{2}\underbrace{(t-s)^2}_{\geq 0}}_{\geq 0} > f(s) + f'(s)(t-s) \qquad \square$$

SATZ: (monotone Konvergenz)
Seien (V1) - (V4) erfüllt und $x_0 \in [a,b]$ ein beliebiger Startwert.
Dann gilt:

(i) $\xi \leq x_{i+1} \leq x_i$ für $i = 1,2,3,\ldots$ und $f'(t) > 0$, $t \in [a,b]$,

(ii) $x_i \leq x_{i+1} \leq \xi$ für $i = 1,2,3,\ldots$ und $f'(t) < 0$, $t \in [a,b]$,

(iii) $\lim\limits_{i\to\infty} x_i = \xi$.

BEWEIS:
Es gibt genau eine Nullstelle.

(i) Sei $f'(t) > 0$ in $[a,b]$.
$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \Leftrightarrow 0 = f'(x_0)(x_1 - x_0) + f(x_0) \leq f(x_1)$.

Falls $x_1 = \xi \Rightarrow x_i = \xi$ für $i = 1,2,\ldots$
Sei $x_1 \neq \xi$. Nachdem Mittelwertsatz folgt:
$$\frac{\overbrace{f(x_1)}^{>0} - \overbrace{f(\xi)}^{=0}}{x_1 - \xi} = \underbrace{f'(\eta)}_{>0} \Rightarrow x_1 - \xi > 0 \Rightarrow x_1 > \xi.$$
Induktionsanfang: ✓

Es gelte $\xi \leq x_n \leq x_{n-1} \leq \ldots \leq x_1$ mit $f(x_n) \geq 0$
$0 = f'(x_n)(x_{n+1} - x_n) + f(x_n) \leq f(x_{n+1})$ (nach (V3))
$\Rightarrow$ mit dem Mittelwertsatz $\frac{f(x_{n+1}) + f(\xi)}{x_{n+1} - \xi} = f''(\eta) \Rightarrow$ (wie oben) $\xi \leq x_{n+1}$.
$$x_{n+1} = x_n - \underbrace{\frac{f(x_n)}{f'(x_n)}}_{>0} \leq x_n.$$
Induktionsschritt: ✓

(ii) Die Folge $(x_i)_{i\in\mathbb{N}}$ ist monoton fallend und beschränkt und damit konvergent.
$\lim\limits_{i\to\infty} x_i =: \gamma$ $\Phi(t) = t - \frac{f(t)}{f'(t)}$ stetig.
$\Phi(\gamma) = \Phi(\lim\limits_{i\to\infty} x_i) = \lim\limits_{i\to\infty} \Phi(x_i) = \lim\limits_{i\to\infty} x_{i+1} = \gamma$.
Also ist γ eine Nullstelle von $f \Rightarrow \gamma = \xi$ (da f streng monoton wachsend).

ANWENDUNG:
Berechnung von $\sqrt{a}$ für $a > 1$

$f(t) = t^2 - a$ $\qquad 1 < \sqrt{a} < a$
$f'(t) = 2t$ $\qquad f'(t) > 0$ im Intervall $[1, a]$
$f''(t) = 2$ $\qquad f''(t) > 0$ im Intervall $[1, a]$

$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = x_0 - \frac{x_0^2 - a}{2x_0} = \frac{x_0}{2} + \frac{a}{2x_0}$
$\leq \frac{a}{2} + \frac{a}{2} = a.$
bzw. $\geq \frac{1}{2} + \frac{1}{2} = 1.$

$\Rightarrow x_1 \in [1, a]$ für jeden Startwert $x_0 \in [1, a]$.
$\Rightarrow$ (V1) - (V4) sind erfüllt. Für jedes $x_0 \in [1, a]$ konvergiert also das Newton-Verfahren gegen $\sqrt{a}$.

d. Das Sekanten-Verfahren

Idee:
Ersetze im Newton-Verfahren die Tangente durch eine Sekante:

Seien x_0, x_1 zwei Startwerte:
$'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$
$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}) \frac{f(x_i)x_{i-1} - f(x_{i-1})x_i}{f(x_i) - f(x_{i-1})}.$

Bemerkung:
Vorteil:
es muss nur $f(x)$ ausgewertet werden und nicht $f'(x) \Rightarrow$ schneller

Nachteil:
Konvergenzordnung: $p = \frac{1+\sqrt{5}}{2} \approx 1,6\ldots$

regula falsi Verfahren:
Seien $x_0, y_0 \in [a, b]$ zwei Startwerte mit $f(x_0) \cdot f(y_0) < 0$.
$s := \frac{f(x_0)y_0 - f(y_0)x_0}{f(x_0) - f(y_0)}$ (Sekantenschritt)

1. $f(s) = 0$ ✓
2. $f(x_0) \cdot f(s) < 0 \Rightarrow x_1 = x_0, y_1 = s$ (Bisektionsschritt)
3. $f(y_0) \cdot f(s) < 0 \Rightarrow x_1 = s, y_1 = y_0$

e. Anwendung aus dem Bereich der Finanzmathematik

Effektivzins von Sparverträgen (prämienbegünstigte Sparvertäge):
Ein Sparer zahlt zu Beginn eines Jahres einen festen Betrag R ein.
Verzinsung: p%. Dauer n Jahre.
Nach n Jahren bekommt der Sparer das Kapital ausbezahlt und eine Prämie von 14% auf die Sparbeträge.

Frage:
Effektivzins: welcher Zinssatz wäre anzusetzen, damit der gleiche Auszahlbetrag zustande kommt, wenn keine Prämie bezahlt wird?

Hier hat man

für die Spareinlage zu Beginn des ersten Jahres:	$R(\overbrace{1+\frac{p}{100}}^{=:s})^n = Rs^n$
für die Spareinlage zu Beginn des zweiten Jahres:	$R(1+\frac{p}{100})^{n-1} = Rs^{n-1}$
$\vdots$	$\vdots$
für die Spareinlage zu Beginn des n-ten Jahres:	Rs
zusammen	$R(s^n + s^{n-1} + \ldots + s) + 0,14 \cdot R$

Keine Prämie, dafür aber q% Zins. $(r = 1 + \frac{q}{100})$

$$K_2 = R(r^n + \ldots + r)$$

Wie ist r zu wählen, damit $K_1 = K_2$ gilt?
O.B.d.A. $R = 1$.

$$r^n + r^{n-1} + \ldots + r = \underbrace{s^n + s^{n-1} + \ldots + s + 0,14 \cdot n}_{=:c>0}$$

$f(t) = t^n + \ldots + t - c = 0$	(positive Nullstellen für $t \in (0, \infty)$)
$f'(t) = nt^{n-1} + \ldots + 1 > 0$	(V2) ✓
$f''(t) = n(n-1)t^{n-2} + \ldots + 2 > 0$	(V3) ✓

$f(0) = -c < 0$
$f(c) = c^n + \ldots + c - c = c^n + \ldots + c^2 > 0$

$\Rightarrow$ (nach dem Zwischenwertsatz) gibt es eine Nullstelle r in $(0, c)$ (V1) ✓

$f'(x_0)(r - x_0) + f(x_0) \leq f(r) = 0$
$\Rightarrow r \leq x_0 - \frac{f(x_0)}{f'(x_0)} = x_1 \Rightarrow x_1 \in (0, \infty)$ (V4) ✓

Also ist der Satz über die monotone Konvergenz beim Newton-Verfahren erfüllt. Für jeden Startwert $x_0 \in (0, \infty)$ konvergiert das Newton-Verfahren gegen r.

Wahl des Startwertes: s.

4. Lineare Gleichungssysteme

a. Einleitung

$$Ax = b: \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}.$$

Wir setzen eine eindeutige Lösung voraus, d.h. $A \neq 0$.

- **direkte Verfahren:** liefern nach endlich vielen Schritten die exakte Lösung.
  Aber: Rundungsfehler!
  ca. n^3 Operationen.
- **indirekte Verfahren:** iterative Verfahren: Formelfehler und Rundungsfehler.
  ca. $k \cdot n^2$ Operationen $\Rightarrow$ insbesondere für Gleichungssysteme mit großes n interessant.

b. Gestaffelte Systeme

(1) A obere Dreiecksmatrix:
$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ & \ddots & \vdots \\ 0 & & a_{nn} \end{pmatrix}$ mit $a_{ii} \neq 0$ für $i = 1, \dots, n$.

$x_n = \frac{b_n}{a_{nn}}$,
$x_{n-1} = \frac{b_{n-1} - a_{n-1,n} x_n}{a_{n-1,n-1}}$,
$\vdots$
$x_k = \frac{b_k - a_{k,k+1} x_{k+1} - \dots - a_{k,n} x_n}{a_{kk}}$ für $k = n, \dots, 1$.

„Rückwärtsauflösen“

(2) A eine untere Dreiecksmatrix:
$\begin{pmatrix} a_{11} & & \\ \vdots & \ddots & \\ a_{n1} & \dots & a_{nn} \end{pmatrix}$ mit $a_{ii} \neq 0$ für $i = 1, \dots, n$.

$x_1 = \frac{b_1}{a_{11}}$,
$\vdots$
$x_k = \frac{b_k - a_{k1}x_1 - \ldots - a_{k,k-1}x_{k-1}}{a_{kk}}$ für $k = 1, \ldots, n$.

„Vorwärtsauflösen“

c. Die LR-Zerlegung

Hier versucht man eine Zerlegung der Form $A = L \cdot R$ mit

$$L = \begin{pmatrix} 1 & & 0 \\ \vdots & \ddots & \\ l_{n1} & \ldots & 1 \end{pmatrix}, R = \begin{pmatrix} r_{11} & \ldots & r_{1n} \\ & \ddots & \vdots \\ 0 & & r_{nn} \end{pmatrix}.$$

$$Ax = b \Leftrightarrow L \cdot \underbrace{(Rx)}_{y} = b.$$

1. Löse das lineare Gleichungssystem $Ly = b$ durch Vorwärtsauflösen.
2. Löse das lineare Gleichungssystem $Rx = y$ durch Rückwärtsauflösen.

Satz: Ist A regulär, so kann man durch vorherige Zeilenvertauschungen erreichen, dass die LR-Zerlegung existiert, also:

$$P \cdot A = L \cdot R$$

mit einer Permutationsmatrix P.

$Ax = b \Leftrightarrow PAx = Pb$

Berechnung der LR-Zerlegung (Verfahren von Crout)

$$\underbrace{\begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nn} \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1 & & \\ \vdots & \ddots & \\ l_{n1} & \ldots & 1 \end{pmatrix}}_{L} \cdot \underbrace{\begin{pmatrix} r_1 & \ldots & r_{1n} \\ & \ddots & \vdots \\ 0 & & r_{nn} \end{pmatrix}}_{R}$$

$\Rightarrow$

1. $a_{1k} = r_{1k}$ für $k = 1, \ldots, n$.

2. $a_{j1} = l_{j1} \cdot r_{11} \Rightarrow l_{j1} = \frac{a_{j1}}{r_{11}}$ für $j = 2, \ldots, n$.
3. usw.

Allgemein:

$$r_{ik} = a_{jk} - l_{j1}r_{1k} - \ldots - l_{j,j-1}r_{j-1,k} \text{ für } k = j, \ldots, n$$
$$l_{ij} = \frac{a_{ij} - l_{i1}r_{1j} - \ldots - l_{i,j-1}r_{j-1,j}}{r_{jj}} \text{ für } i = j+1, \ldots, n$$

Rechenaufwand:

- für LR-Zerlegung benötigt man $\approx \frac{n^3}{3}$ Punktoperationen,
- ein gestaffeltes Gleichungssystem benötigt man $\approx \frac{n^2}{2}$ Punktoperationen.

Anmerkung:
Man vermeidet die Berechnung von A^{-1}, denn man benötigt n^3 Punktoperationen und das Ergebnis ist instabil (d.h. starke Verfälschung durch z.B. Rundungsfehler).

d. Die Cholesky-Zerlegung

$A \in \mathbb{R}^{n \times n}$ heißt positiv definit, falls gilt

(1) A ist symmetrisch

(2) $x^T Ax > 0$ für alle $x \in \mathbb{R}^n \setminus \{0\}$.

Satz:
Sei A positiv definit. Dann existiert eine Zerlegung der Form

$$A = L \cdot L^T$$

mit einer eindeutig bestimmten unteren Dreiecksmatrix $L = \begin{pmatrix} l_{11} & & 0 \\ \vdots & \ddots & \\ l_{n1} & \ldots & l_{nn} \end{pmatrix}$ und $l_{ii} > 0$ für $i = 1, \ldots, n$.

Diese Zerlegung heißt **Cholesky-Zerlegung**.

Beweis:
vollständige Induktion nach n:
$n = 1$: $A = (a_{11})$ positiv definit $\Rightarrow a_{11} > 0 \Rightarrow L = (\sqrt{a_{11}})$; $LL^T = A$ ✓

Behauptung sei richtig für $A_{n-1} \in \mathbb{R}^{n-1\times n-1}$
$A_{n-1} = L_{n-1} \cdot L_{n-1}^T$
Sei nun $A \in \mathbb{R}^{n\times n}$ positiv definit.

$$A = \left(\begin{array}{c|c} A_{n-1} & b \\ \hline b^T & a_{nn} \end{array}\right) \Rightarrow A_{n-1} \text{ ist positiv definit und } A_{n-1} = L_{n-1} \cdot L_{n-1}^T.$$

Ansatz:

$$L = \left(\begin{array}{c|c} L_{n-1} & 0 \\ \hline c^T & \alpha \end{array}\right)$$

$A = L \cdot L^T$

$$\Rightarrow L \cdot L^T = \left(\begin{array}{c|c} L_{n-1} & 0 \\ \hline c^T & \alpha \end{array}\right) \cdot \left(\begin{array}{c|c} L_{n-1}^T & c \\ \hline 0 & \alpha \end{array}\right)$$

$$= \left(\begin{array}{c|c} L_{n-1} \cdot L_{n-1}^T & L_{n-1} \cdot c \\ \hline c^T \cdot L_{n-1}^T & c^T \cdot c\alpha^2 \end{array}\right) = \left(\begin{array}{c|c} A_{n-1} & b \\ \hline b^T & a_{nn} \end{array}\right)$$

$\Rightarrow L_{n-1} \cdot c = b$. Da L_{n-1} vollen Rang hat, ist sie invertierbar und es folgt $L_{n-1}^{-1} b$.

$c^T c + \alpha^2 = a_{nn} \Rightarrow \alpha^2 = a_{nn} - c^T c$

$$0 < \det A = \det L \cdot \det L^T = (\det L)^2 = \underbrace{(\det L_{n-1})^2}_{>0} \cdot \alpha^2 \Rightarrow \alpha^2 > 0.$$

$\Rightarrow \alpha = \sqrt{a_{nn} - c^T c}$.

Berechnung der Cholesky-Zerlegung:
Spaltenweises Vorgehen liefert für $j = 1, \ldots, n$:

$$l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2}$$

$$l_{ij} = \frac{1}{l_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) \text{ für } i = j+1, \ldots, n.$$

Rechenaufwand:

- für Cholesky-Zerlegung benötigt man $\approx \frac{n^3}{6}$ Punktoperationen.

e. Die QR-Zerlegung von A

Hier zerlegt man A in der Form $A = Q \cdot R$ mit einer orthogonalen Matrix Q und einer oberen Dreiecksmatrix R.

$Ax = b \Leftrightarrow QRx = b \Rightarrow Rx = Q^T b$ ($\rightarrow$ Rückwärtsauflösung)

Satz:
Die QR-Zerlegung existiert für jede Matrix $A \in \mathbb{R}^{n\times}$.

Householder-Matrizen:
$h^T = (0, \ldots, 0, h_k, \ldots, h_n) \in \mathbb{R}^n$ mit $\|h^T\|_2 = 1$

$$h \cdot h^T = \begin{pmatrix} 0 & \ldots & & 0 \\ \vdots & h_k^2 & \ldots & h_k h_n \\ & \vdots & \ddots & \vdots \\ 0 & h_n h_k & \ldots & h_n^2 \end{pmatrix}$$

Definition:
Eine Matrix der Form $H = I - 2h \cdot h^T$ heißt **Householder-Matrix**.

Eigenschaften:

(a) $H^T = H$;

(b) H ist eine orthogonale Matrix:
$H^T H = (I - 2hh^T)^T (I - 2hh^T) = (I - 2hh^T)(I - 2hh^T) = I - 4hh^T + 4h \underbrace{h^T h}_{=1} h^T = I.$

Anmerkung:
Seien Q_1, Q_2 orthogonale Matrizen, dann folgt $Q_1 Q_2$ ist ebenfalls eine orthogonale Matrix.

Konstruktion der QR-Zerlegung:

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}, \begin{pmatrix} a_{11} \\ \vdots \\ a_{n1} \end{pmatrix} =: a_1.$$

1. Schritt:

Es gibt eine Householder-Matrix H_1 mit $H_1 a_1 = \sigma e_1$ $\qquad e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^n$

$$\sigma^2 = \sigma e_1^T \cdot \sigma e_1 = (H_1 a_1)^T H_1 a_1 = a_1^T \underbrace{H^T H}_{=I} a_1 = a_1^T a_1 = \|a_1\|_2^2 \Rightarrow \sigma = \pm \|a_1\|_2$$

$$\sigma e_1 = H_1 a_1 = (I - 2h_1 h_1^T) a_1 = a_1 - 2h_1 \underbrace{h_1^T a_1}_{\in \mathbb{R}} = a_1 - 2(h_1^T a_1) h_1$$

$2(h_1^T a_1) h_1 = a_1 - \sigma e_1$.
Wegen $\|h_1\|_2 = 1$ folgt $h_1 = \frac{a_1 - \sigma e_1}{\|a_1 - \sigma e_1\|_2}$.

Aus Stabilitätsgründen setzt man $\sigma = -\text{sign}(a_{11}) \|a_1\|_2$.
$\Rightarrow h_1 = \frac{a_1 + \text{sign}(a_{11}) \|a_1\|_2 \cdot e_1}{\|a_1 + \text{sign}(a_{11}) \|a_1\|_2 \cdot e_1\|}$.

Nun gilt mit h_1: $H_1 \cdot A = \begin{pmatrix} \sigma & \dots \\ 0 & \\ \vdots & \widetilde{A}_1 \\ 0 & \end{pmatrix}$

2. Schritt:
Es gibt eine Householder-Matrix $\widetilde{H}_2$ ($(n-1) \times (n-1)$-Matrix) mit $\widetilde{H}_2 a_2 = \gamma \widetilde{e_1}$

$\widetilde{e_1} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{n-1}$.

Setze $H_2 := \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \widetilde{H}_2 \end{array}\right)$; H_2 ist eine $n \times n$-Householder-Matrix; $h_2 = (0, x, \ldots, x)$.

Nun gilt $H_2 \cdot H_1 \cdot A = \left(\begin{array}{cc|c} \sigma & & \\ & \gamma & \\ \hline & & \widetilde{A}_2 \\ & & \end{array}\right)$.

$\vdots$

$n-1$. *Schritt:*

$$\Rightarrow \underbrace{H_{n-1}H_{n-2}\ldots H_2H_1}_{=:Q^T} \cdot A = \begin{pmatrix} x & \ldots & x \\ & \ddots & \vdots \\ 0 & & x \end{pmatrix}$$

$$\Rightarrow Q = H_1^T H_2^T \ldots H_{n-1}^T = H_1 H_2 \ldots H_{n-1}.$$

Rechenaufwand:

- für QR-Zerlegung benötigt man $\approx \frac{2n^3}{3}$ Punktoperationen, dafür ist der Algorithmus stabil.

Die QR-Zerlegung bzw. die LR-Zerlegung ist besonders günstig, wenn mehrere Gleichungssysteme mit der selben Koeffizientenmatrix zu lösen sind.

f. Die Gauß-Elimination

$Ax = b$.

$$(A, b) = \begin{pmatrix} a_{11} & \ldots & a_{1n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \ldots & a_{nn} & b_n \end{pmatrix} \rightarrow \text{elementare Zeilenumformungen} \rightarrow \begin{pmatrix} \gamma_{11} & \ldots & \gamma_{1n} & c_1 \\ & \ddots & \vdots & \vdots \\ 0 & & \gamma_{nn} & c_n \end{pmatrix}$$

$$R := \begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1n} \\ & \ddots & \vdots \\ 0 & & \gamma_{nn} \end{pmatrix}$$

$Ax = b \Leftrightarrow Rx = c \rightarrow$ Rückwärtsauflösen.

Elementare Zeilenumformungen:

(1) Vertauschen von Zeilen;

(2) Addition der Vielfachen einer Zeile zu einer anderen Zeile.

Es sei $a_{11} \neq 0$. Mit den Operationen
$l_{i1} := \frac{a_{i1}}{a_{11}}$ für $i = 2, \ldots, n$
$a_{ik}^{(1)} := a_{ik} - l_{i1} a_{1k}$ für $i, k = 2, \ldots, n$
$b_i^{(1)} = b_i - l_{i1} b_1$

$$\Rightarrow \begin{pmatrix} a_{11} & \ldots & \ldots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \ldots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \ldots & a_{nn}^{(1)} & b_n^{(1)} \end{pmatrix}$$

Ein Eliminationsschritt:
$$\Rightarrow x_1 = \frac{1}{a_{11}} \left(b_1 - \sum_{k=2}^{n} a_{1k} x_k \right)$$

Nach $n - 1$ solcher Schritte erhalte:
$$\begin{pmatrix} a_{11} & \ldots & \ldots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \ldots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & a_{nn}^{(n-1)} & b_n^{(n-1)} \end{pmatrix}$$

Definition:
Die Einträge $a_{11}, a_{22}^{(1)}, \ldots, a_{nn}^{(n-1)}$ heißen **Pivot-Elemente**.

Satz:
Ist A regulär, so kann durch Zeilentausch erreicht werden, dass alle Pivot-Elemente von Null verschieden sind (vgl. Übungsaufgabe).

Rechenaufwand:

- für die Gauß-Elimination benötigt man $\approx \frac{n^3}{3}$ Punktoperationen.

Pivot-Strategien

- BETRAGSMAXIMALE SPALTENPIVOTWAHL:
  betragsgrößtes Element: $\max\limits_{i \geq k} |a_{ik}^{(k-1)}| =: |a_{pk}^{(k-1)}|$.
  Vertausche vor dem k-ten Eliminationsschritt die k-te und p-te Zeile.

g. Matrizentechnische Deutung der Gauß-Elimination

$$G_1 = \begin{pmatrix} 1 & & & 0 \\ -l_{21} & 1 & & \\ \vdots & & \ddots & \\ -l_{n1} & 0 & & 1 \end{pmatrix}.$$

Der 1. Eliminationsschritt bedeutet dann $G_1 Ax = G_1 b$.

$$G_2 = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & -l_{32} & 1 & & \\ & \vdots & & \ddots & \\ & -l_{n2} & & & 1 \end{pmatrix}.$$

Der 2. Eliminationsschritt bedeutet dann $G_2 G_1 Ax = G_2 G_1 b$.

$\vdots$

Nach dem $n-1$-sten Eliminationsschritt:
$G_{n-1} \cdot \ldots \cdot G_1 \cdot A = R$.

GAUSS-JORDAN-ELIMINATION

$$A = \begin{pmatrix} a_{11} & \ldots & a_{1k} & \ldots & a_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \ldots & a_{ik} & \ldots & a_{im} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nk} & \ldots & a_{nm} \end{pmatrix} \rightarrow JA = \begin{pmatrix} a_{11}^{(1)} & \ldots & 0 & \ldots & a_{1m}^{(1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1}^{(1)} & \ldots & 1 & \ldots & a_{im}^{(1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n1}^{(1)} & \ldots & 0 & \ldots & a_{nm}^{(1)} \end{pmatrix}$$

$$J = \begin{pmatrix} 1 & & -l_{1i} & & \\ & \ddots & \vdots & & \\ & & l_{ii} & & \\ & & \vdots & \ddots & \\ & & -l_{ni} & & 1 \end{pmatrix} \text{ mit } l_{ii} = \tfrac{1}{a_{ik}}, l_{ij} = \tfrac{a_{jk}}{a_{ik}}, i \neq j.$$

Sei $A \in \mathbb{R}^{n\times n}$ invertierbar. Berechne A^{-1}:

$$\left(\begin{array}{ccc|ccc} a_{11} & \dots & a_{1n} & 1 & & 0 \\ \vdots & \ddots & \vdots & & \ddots & \\ a_{n1} & \dots & a_{nn} & 0 & & 1 \end{array}\right) \to \left(\begin{array}{cccc|ccc} 1 & x & \dots & x & x & 0 & 0 \\ & \vdots & \ddots & \vdots & \vdots & \ddots & \\ 0 & x & \dots & x & x & 0 & 1 \end{array}\right)$$

$$\to \dots \to \left(\begin{array}{ccc|ccc} 1 & & 0 & x & \dots & 0 \\ & \ddots & & \vdots & \ddots & \vdots \\ 0 & & 1 & x & \dots & x \end{array}\right)$$

5. Das lineare Ausgleichsproblem

a. Problemstellung

Gegeben: Messergebnisse (z.B. $(t_i, s_i), i = 1, \ldots, m$);
Theoriefunktion (z.B. $G(t, \underbrace{\alpha_1, \ldots, \alpha_n}_{=\alpha}) = G(t, \alpha)$); $m > n$.

Aufgabe: Bestimme die Parameter $\alpha_1, \ldots, \alpha_n$ anhand der Messergebnisse.

$G(t_i, \alpha_1, \ldots, \alpha_n) = s_i$ für $i = 1, \ldots, m$ ist optimal, aber im Allgemeinen nicht möglich, da $m > n$.

aber: $G(t_i, \alpha_1, \ldots, \alpha_n) \approx s_i$ für $i = 1, \ldots, m$ kann „möglichst gut" erreicht werden.
Bestimme also $\alpha_1, \ldots, \alpha_n$ so, dass dies erreicht wird,
d.h. $\underbrace{\sum_{i=1}^{n} (G(t_i, \alpha_1, \ldots, \alpha_n) - s_i)^2}_{\textit{Fehlerquadratsumme}}$ minimieren.

Definition:
Dies ist ein allgemeines Ausgleichsproblem.

Beispiel:
$G(t, a, b, c) = \frac{a}{1+\exp(b-ct)}$ (logistische Kurve, nicht linear)

Definition:
Ein Ausgleichsproblem heißt **linear**, falls

$$G(t, \alpha_1, \ldots, \alpha_n) = \sum_{i=1}^{n} \alpha_i g_i(t)$$

mit gewissen Grundfunktionen $g_i(t)$.

Beispiel:
$g_i(t) = t^{i-1}$. (Polynomausgleich)
$\sum_{i=1}^{m} \left(\sum_{j=1}^{n} \alpha_j g_j(t_i) - s_i \right)^2$.

Setze $a_{ij} := g_j(t_i)$ für $j = 1, \ldots, n; i = 1, \ldots, m$.

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}.$$

Setze $b = \begin{pmatrix} s_1 \\ \dots \\ s_n \end{pmatrix}$.

Dann gilt $\sum_{i=1}^{m} \left(\sum_{j=1}^{n} \alpha_j g_j(t_i) - s_i \right)^2 = ||A\alpha - b||_2^2$.

Beim Polynomausgleich gilt:

$$A = \begin{pmatrix} 1 & t_1 & \dots & t_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & \dots & t_m^{n-1} \end{pmatrix}.$$

b. Die Normalgleichungen

$A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ gegeben. Gesucht ist ein $x \in \mathbb{R}^n$ mit $||Ax - b||_2^2 \to \min$.

$F : \mathbb{R}^n \to \mathbb{R}^+$, $F(x) = ||Ax - b||_2^2 = (Ax - b)^T \cdot (Ax - b) = x^T A^T Ax - 2(A^T b)^T x + b^T b$.
$\Rightarrow F(x)$ ist differenzierbar.

$\nabla F(x) = 2A^T Ax - 2A^T b$
Notwendige Bedingung für ein lokales Minimum: $\nabla F = 0$.
$\Rightarrow A^T Ax = A^T b$ (**Normalgleichungen**)

Satz:
Jede Lösung der Normalgleichungen ist ein globales Minimum von F.

Beweis:
Sei $\overline{x}$ eine Lösung der Normalgleichungen: $A^T A\overline{x} = A^T b$.
Dann gilt für jedes $x \in \mathbb{R}^n$:
$F(x) = ||Ax - b||_2^2 = \langle Ax - b, Ax - b \rangle = \langle A\overline{x} - b + A(x - \overline{x}), A\overline{x} - b + A(x - \overline{x}) \rangle =$
$= \underbrace{\langle A\overline{x} - b, A\overline{x} - b \rangle}_{F(\overline{x})} + 2 \langle A\overline{x} - b, A(x - \overline{x}) - b \rangle + \langle A(x - \overline{x}), A(x - \overline{x}) \rangle =$

$= F(\overline{x}) + 2 \underbrace{\left\langle A^T A\overline{x} - A^T b, x - \overline{x} \right\rangle}_{=0} + \underbrace{||A(x - \overline{x})||_2^2}_{\geq 0} =$ (da $\langle Ax, y \rangle = \langle x, A^T y \rangle$)

$\geq F(\overline{x})$.

$\Rightarrow \overline{x}$ ist ein globales Minimum.

□

Korollar:
A hab den Rang n. Dann besitzt F genau ein globales Minimnum.

Beweis:
$x^T A^T A x = (Ax)^T (Ax) = \|Ax\|_2^2 \geq 0$.
$\|Ax\|_2^2 = 0 \Leftrightarrow Ax = 0 \Rightarrow x = 0$.
$\Rightarrow A^T A$ positiv definit.

□

Bemerkung:
Man kann die Cholesky-Zerlegung verwenden.

6. Lineare Optimierung

a. Problemstellung

Maximum bzw. Minimum einer Funktion unter einer Nebenbedingung. Maximum bzw. Minimum einer linearen Funktion unter einer linearen Nebenbedingung → lineare Optimierung.

Beispiel:
Ein Landwirt besitzt 80 ha Ackerfläche und baut Gerste und Weizen an. Der Arbeitseinsatz pro ha Gerste beträgt 6 Stunden, pro ha Weizen 10 Stunden. Der Landwirt hat maximal 560 Stunden Zeit. Die Kosten für pro ha Gerste betragen 30 € , pro ha Weizen 100 €; es stehen maximal 4700 € zur Verfügung.
Der Gewinn beträgt pro ha Gerste 40 € und pro ha Weizen 50 €.

Frage: wie ist die Ackerfläche aufzuteilen, um einen möglichst hohen Gewinn zu erzielen?

Sei x_1 die Fläche für Gerste und x_2 die Fläche für Weizen.
$\Rightarrow x_1 + x_2 \leq 80$ (Anbaufläche)
$6x_1 + 10x_2 \leq 560$ (Arbeit)
$30x_1 + 100x_2 \leq 4700$ (Kapital)
$x_1 \geq 0, x_2 \geq 0.$

Gewinn: $g(x_1, x_2) = 40x_1 + 50x_2 \rightarrow \max.$

Matrixschreibweise:
$P = \begin{pmatrix} 1 & 1 \\ 6 & 10 \\ 30 & 100 \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, b = \begin{pmatrix} 80 \\ 560 \\ 4700 \end{pmatrix}, q = \begin{pmatrix} 40 \\ 50 \end{pmatrix}.$
$\Rightarrow g(x) := q^T x \rightarrow \max$ unter den Nebenbedingungen:
$Px \leq b, x \geq 0.$ $x \leq y :\Leftrightarrow x_i \leq y_i, i = 1, \ldots, n.$

Allgemeiner Fall:
$q \in \mathbb{R}^n, b \in \mathbb{R}^m, P \in \mathbb{R}^{m \times n}; x \in \mathbb{R}^n.$
$g(x) = q^T x \rightarrow \max$ unter $Px \leq b, x \geq 0.$ (Normalform 1)

Diese Aufgabenstellung heißt ein **lineares Optimierungsproblem** (**lineares Programm**). $g(x) = q^T x$ heißt **Zielfunktion**.

Die Menge $K = \{x \in \mathbb{R}^n : Px \leq b \text{ und } x \leq 0\}$ heißt **zulässiger Bereich**.

b. Grafische Lösung

siehe Mathe-Heft Klasse 9 ;-).

$\Rightarrow$ das Maximum liegt in einer Ecke!

c. Normalformen

NORMALFORM 1: $g(x) = q^T x \to \max$ unter $Px \leq b, x \geq 0$. (siehe oben)

NORMALFORM 2: $\max g(x) = q^T x$ unter $Px + s = b, x \geq 0, s \geq 0$. $s = \begin{pmatrix} s_1 \\ \vdots \\ s_m \end{pmatrix}$

s heißt **Schlupfvariable**.

NORMALFORM 3: $A \in \mathbb{R}^{m \times (m+n)}$ mit $rk(A) = m$; $b \in \mathbb{R}^m, c \in \mathbb{R}^{m+n}$
$\max g(z) = c^T z$ unter $Az = b, z \geq 0$.
$A = (P, I); c = (q, 0 \ldots 0)^T, z = (x_1, \ldots, x_n, s_1, \ldots, s_m)^T$.

ANMERKUNG:

(1) $\min g(x)$ ist äquivalent zu $\max -g(x)$;

(2) Eine Nebenbedingung der Form

$$a_{k1}x_1 + \ldots + a_{kn}x_n \geq b_k$$

wird zu

$$-a_{k1}x_1 - \ldots - x_{kn}x_n \leq -b_k;$$

(3) untergliegt eine Komponente x_k keiner Vorzeichen-Beschränkung, dann $x_k = x_{k1} - x_{k2}$ mit $x_{k1} \geq 0$ und $x_{k2} \geq 0$.

Alle linearen Optimierungsprobleme lassen sich also auf die oben genannten Normalformen zurückführen.

d. Charakterisierung des zulässigen Bereichs

$K = \{x \in \mathbb{R}^n : Px \leq b \text{ und } x \geq 0\}$ (Normalform 1)
$K = \{x \in \mathbb{R}^{n+m} : Az = b \text{ und } z \geq 0\}$ (Normalform 2)

Konvexe Mengen:

Definition:
Eine Menge $M \subset \mathbb{R}^k$ heißt **konvex**, falls für alle $x_1, x_2 \in M$ und alle $0 \leq \lambda \leq 1$ der Punkt $x_\lambda := \lambda x_1 + (1-\lambda)x_2 \in M$ gilt.

Eigenschaften:

(1) Der Durchschnitt beliebig vieler konvexer Mengen ist konvex.

(2) Seien $x_1, \ldots, x_r \in M \Rightarrow \sum_{i=1}^{r} \mu_i x_i \in M$ für alle $\mu_i \geq 0$ mit $\sum_{i=1}^{r} \mu_i = 1$. (konvexe Linearkombination)

(3) Seien $y_1, \ldots, y_r \in \mathbb{R}^k$. Dann ist die Menge $K = K(y_1, \ldots, y_r) := \{x \in \mathbb{R}^k : x = \sum_{i=1}^{r} \mu_i y_i, \mu_i \geq 0 \text{ mit } \sum_{i=1}^{r} \mu_i = 1\}$ konvex. (**konvexe Hülle** der Punkte $y_1, \ldots, y_r$)

Definition:
Sei M eine konvexe Menge. Ein $x \in M$ heißt **Ecke** (oder **Extrempunkt**), wenn $x = \mu x_1 + (1-\mu)x_2$ mit $0 < \mu < 1$, $x_1, x_2 \in M$, folgt, dass $x = x_1 = x_2$.

Hyperebenen und Halbräume

Definition:
Sei $\alpha^T = (\alpha_1, \ldots, \alpha_k) \in \mathbb{R}^k \setminus \{0\}$ und $\gamma \in \mathbb{R}$. Die Menge $H := \{x \in \mathbb{R}^k : \alpha^T x = \gamma\}$ heißt **Hyperebene**.

Eine Hyperebene zerteilt den $\mathbb{R}^k$ in zwei Halbräume

$$H^+ := \{x \in \mathbb{R}^k : \alpha^T x \geq \gamma\}$$
$$H^- := \{x \in \mathbb{R}^k : A^T x \leq \gamma\}.$$

Lemma:
H, H^-, H^+ sind konvex und abgeschlossen.

Zulässiger Bereich K:

Jede Nebenbedingung liefert eine Hyperebene bzw. einen Halbraum.
K ist endlicher Durchschnitt von Hyperebenen bzw. Halbräumen. $\Rightarrow$

Satz:
K ist konvex und abgeschlossen.

Anmerkung:
Sei $K \neq \emptyset$ und K beschränkt, dann gibt es eine optimale Lösung.

e. Ecken des zulässigen Bereichs

(Normalform 3: $\max g(z) = c^T z$ unter $Az = b, z \geq 0$)

$A = \left(a^{(1)}, a^{(2)}, \ldots, a^{(m+n)}\right)$.
Sei $x \in K$, dann definiere $I(x) := \{i \in \{1, \ldots, m+n\} : x_i > 0\}$.

Satz: (Charakterisierungssatz)
Äquivalent sind (für $x \in K$)

(1) x ist eine Ecke von K;

(2) Die Spalten $a^{(i)}$ mit $i \in I(x)$ sind linear unabhängig.

Beweis:
(i) $\Rightarrow$ (ii):
sei x eine Ecke. Ohne Einschränkung der Allgemeinheit sei $I(x) = \{1, \ldots, r\}$.
$b = Ax = \sum_{i=1}^{m+n} x_i a^{(i)} = \sum_{i=1}^{r} x_i a^{(i)}$.
Annahme: $a^{(1)}, \ldots a^{(r)}$ sind linear abhängig: dann gibt es ein $(\alpha_1, \ldots, \alpha_r) \neq (0, \ldots, 0)$
mit $\sum_{i=1}^{r} \alpha_i a^{(i)} = 0$. (*)
Wegen $x_i > 0$ für $i = 1, \ldots, r$ gibt es ein $\varepsilon > 0$ mit $x_j \pm \varepsilon\alpha_j \geq 0$ für $j = 1, \ldots, r$.
$\overline{\alpha} = (\alpha_1, \ldots, \alpha_r, 0, \ldots, 0)$
$y_+ = (x_1 + \varepsilon\alpha_1, \ldots, x_r + \varepsilon\alpha_r, 0, \ldots, 0) \geq 0$

$y_- = (x_1 - \varepsilon\alpha_1, \ldots, x_r - \varepsilon\alpha_r, 0, \ldots, 0) \geq 0$
Es gilt $y_+ \neq y_-$; $x = \frac{1}{2}y_+ + (1 - \frac{1}{2})y_-$.

$$Ay_\pm = Ax \pm \varepsilon A\widetilde{\alpha} = Ax + \varepsilon \underbrace{\sum_{i=1}^{r} \alpha_i a^{(i)}}_{=0\text{(nach (*))}} = b$$

$\Rightarrow y_+, y_- \in K \Rightarrow x$ ist keine Ecke.

(ii) $\Rightarrow$ (i):
zu $x \in K$ seien $a^{(1)}, \ldots, a^{(r)}$ linear unabhängig.
Sei $x = \lambda y + (1 - y)z$, $0 < \lambda < 1$, $y, z \in K$.
$\Rightarrow I(x) = I(y) \cup I(z)$.

$$Ay = Az = b \Rightarrow 0 = Ay - Az = A(y - z) = \sum_{j=1}^{m+n} (y_j - z_j)a^{(j)} = \sum_{j=1}^{r} (y_j - z_j)a^{(j)}$$

$\Rightarrow (y_j - z_j) = 0$ für alle $j \in \{1, \ldots r\}$ (da $a^{(1)}, \ldots, a^{(r)}$ linear unabhängig)
$\Rightarrow y = z \Rightarrow x$ ist eine Ecke.

□

Satz: (Existenzsatz)
Der zulässige Bereich $K \subset \mathbb{R}^{m+n}, K \neq \emptyset$, hat Ecken.

Beweis:
$I := \{|I(z)| : z \in K\} \subset \{0, \ldots, n + m\}$
$\gamma = \min I$. Sei $x \in K$ mit $|I(x)| = \gamma$.
$\gamma = 0$: $\Rightarrow x = (0, \ldots, 0)$ ist eine Ecke.
$\gamma > 0$: Ohne Einschränkung sei $I(x) = \{1, \ldots, \gamma\}$.
Betrachte die Spalten $a^{(1)}, \ldots, a^{(\gamma)}$. Annahme: diese Spalten sind linear abhängig.
$\Rightarrow \sum_{j=1}^{\gamma} \alpha_j a^{(j)} = 0$ mit $(\alpha_1, \ldots, \alpha_r) \neq (0, \ldots, 0)$.

$$\lambda : \min\left\{\frac{x_j}{|\alpha_j|} : \alpha_j \neq 0, 1 \leq j \leq r\right\} = \frac{x_k}{|\alpha_k|} \text{ für ein } k. \qquad \text{(O.E. } |\alpha_k| = \alpha_k\text{)}$$

Setze $\overline{x} = (x_1 - \lambda\alpha_1, x_2 - \lambda\alpha_2, \ldots, x_g - \lambda\alpha_\gamma, 0, \ldots, 0) \geq 0$.

$$A\overline{x} = Ax - \lambda \underbrace{\sum_{j=1}^{\gamma} \alpha_j a^{(j)}}_{=0} = Ax = b.$$

$\Rightarrow \overline{x} \in K$ und $\overline{x_k} = 0$.
$\Rightarrow |I(\overline{x})| < \gamma \rightarrow$ Widerspruch.
Also $a^{(1)}, \ldots, a^{(\gamma)}$ sind linear unabhängig. $\Rightarrow x$ ist eine Ecke.

□

Satz: (Darstellungssatz)
Sei $K \neq \emptyset$ und beschränkt.
Zu jedem $x \in K$ gibt es Ecken gibt es Ecken $z^{(1)}, \ldots, z^{(l)}$ mit

$$x = \sum_{j=1}^{l} \lambda_j z^{(j)}; \lambda_j \geq 0, \sum_{j=1}^{l} \lambda_j = 1.$$

f. Basislösungen

Definition:
Sei $A \in \mathbb{R}^{m \times (n+m)}$ mit rk $(A) = m$ und $B(a^{(i_1)}, \ldots, a^{(i_m)})$ eine Teil-Matrix mit Rang $B = m$.
Ein $x \in \mathbb{R}^{m+n}, x \geq 0$ heißt **Basispunkt** zu B, falls gilt:

$$x_j = 0 \text{ für } j \notin \{i_1, \ldots, i_m\}$$

$$\sum_{j=1}^{m} x_{i_j} a^{(i_j)} = b$$

Die **Komponenten** $x_{i_1}, \ldots, x_{i_m}$ heißen **Basisvariable**.
Ein $x \in \mathbb{R}^{m+n}$ heißt **Basispunkt** (bzw. **Basislösung**), wenn es eine Teil-Matrix B von A gibt, so dass x Basispunkt zu B ist.

Satz: (Äquivalenzsatz)
Sei rk $(A) = m$ und $x \in \mathbb{R}^{m+n}$. Dann sind äquivalent:

(1) x ist eine Ecke von K;

(2) x ist eine Basislösung.

Beweis:
(1) $\Rightarrow$ (2):
Sei x eine Ecke mit $I(x) = \{i_1, \ldots, i_p\}$.
Nach dem Charakterisierungssatz sind $a^{(i_1)}, \ldots, a^{(i_p)}$ linear unabhängig.
$(p \leq m)$.

$$B = \left(a^{(i_1)}, \ldots, a^{(i_p)}, b^{(i_{p+1})}, \ldots, b^{(i_m)}\right).$$

(ergänze $a^{(i_1)}, \ldots, a^{(i_p)}$ linear unabhängig zu $a^{(i_1)}, \ldots, a^{(i_p)}, b^{(i_{p+1})}, \ldots, b^{(i_m)}$)

(2) $\Rightarrow$ (1):
Sei x ein Basispunkt zu $B = \left(a^{(i_1)}, \ldots, a^{(i_m)}\right) \Rightarrow a^{(i_1)}, \ldots, a_{(i_m)}$ sind linear unabhängig.
$I(x) \subset \{i_1, \ldots, i_m\} \Rightarrow a^{(j)}$, $j \in I(x)$ sind linear unabhängig $\Rightarrow x$ ist eine Ecke.

□

Korollar:
Es gibt maximal $\binom{n+m}{m}$ Ecken.

g. Lösung der linearen Optimierungsaufgabe

Satz: (Hauptsatz)
Sei $K \neq \emptyset$ und beschränkt. Dann nimmt die Zielfunktion ihr Maximum in einer Ecke an.

Beweis:
K ist kompakt; die stetige Funktion $g(z)$ nimmt also ihr Maximum an in einem $\overline{z} \in K$.

$\Rightarrow \overline{z} = \sum_{j=1}^{l} \lambda_j x^{(j)}$, wobei $x^{(j)}$ Ecken sind.

$$c^T x^{(k)} := \max\{c^T x^{(j)} : j = 1, \ldots, l\} \leq \max\{c^T z : z \in K\} = c^T \overline{z} = \sum_{j=1}^{l} \lambda_j \underbrace{c^T x^{(j)}}_{\leq c^T x^{(k)}} \leq c^T x^{(k)}$$

$\Rightarrow c^T x^{(k)} = c^T \overline{z}$.

□

Ziel: Systematisches Berechnen von Ecken. $\Rightarrow$ **Simplexverfahren**.

h. Der Simplex-Algorithmus

$g(x) := q^T x + \alpha^{(0)} \rightarrow \max$ unter den Nebenbedingungen
$Px + s = b$
$x \geq 0$
$s \geq 0$ (Normalform 2)

Zusatzforderung: $b \geq 0$

$$\left(\begin{array}{c|c|c} P & I_m & b \\ \hline q^T & 0^T & \alpha^{(0)} \end{array}\right) = \left(\begin{array}{ccc|ccc|c} p_{11} & \dots & p_{1n} & 1 & & 0 & b_1 \\ \vdots & \ddots & \vdots & & \ddots & & \vdots \\ p_{m1} & \dots & p_{mn} & 0 & & 1 & b_m \\ \hline q_1 & \dots & q_n & 0 & \dots & 0 & \alpha^{(0)} \end{array}\right) = H^{(0)}$$

Eine Basislösung ist $z^{(0)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ b_1 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ s_1 \\ \vdots \\ s_m \end{pmatrix}$

Suche nun nach einer neuen Basislösung (Ecke) mit höherem Wert der Zielfunktion.
Gauß-Jordan-Schritt auf $H^{(0)}$ liefert eine neue Basislösung aus $H^{(1)}$. Gauß-Jordan-Schritt muss so erfolgen, dass $b^{(1)} \geq 0$ (sonst ist die abgelesene Lösung nicht im zulässigen Bereich).
Wahl des Pivots: $p_{rs} > 0$; $q_s > 0$.

$b_i^{(1)} = b_i^{(0)} - \frac{p_{is}}{p_{rs}} b_r \geq 0$
$\Rightarrow b_i^{(0)} p_{rs} \geq p_{is} b_r$
$\frac{b_i^{(0)}}{p_{is}} \geq \frac{b_r}{p_{rs}}$

$$H^{(i)} = \left(\begin{array}{ccccc|c} a_{11}^{(i)} & \dots & a_{1n}^{(i)} & \dots & a_{1,n+m}^{(i)} & b_1^{(i)} \\ \vdots & & \ddots & & \vdots & \vdots \\ a_{m1}^{(i)} & \dots & a_{mn}^{(i)} & \dots & a_{m+n,mn}^{(i)} & b_m^{(i)} \\ \hline c_1^{(i)} & \dots & a_{mn}^{(i)} & \dots & c_{m+n}^{(i)} & -\alpha^{(i)} \end{array}\right) \text{ mit Basislösung } z^{(i)}.$$

Satz: (Optimalitätsprüfung)
Gilt $c_j^{(i)} \leq 0$ für $j = 1, \dots, m+n$, so gilt $z^{(i)}$ eine optimale Lösung.

Satz: (Nichtlösbarkeit)
Gibt es ein $c_k^{(i)} < 0$ mit $a_{jk}^{(i)} \leq 0$ für $j = 1, \dots, m$, dann gibt es keine Lösung (also kein Maximum).

Satz:
Gibt es ein $c_k^{(i)} > 0$ und ein $a_{rk}^{(i)} > 0$, so findet man eine Basislösung $z^{(i+1)}$ mit

$g(z^{(i+1)}) \geq g(z^{(i)})$.

i. Zweiphasenmethode

Falls $b \geq 0$ nicht erfüllt ist, dann ist $z^{(0)} = \begin{pmatrix} 0 \\ b \end{pmatrix}$ keine zulässige Lösung!

$\Rightarrow$ Anlaufrechnung, um eine zulässige Basislösung zu bekommen.

Phase 1:
Bestimme eine zulässige Basislösung.

OBdA sei $b_i < 0$ für $i = 1, \ldots, p$ und $b_i \geq 0$ für $i = p+1, \ldots, m$.
Führe zusätzliche Variablen $\xi_1, \ldots, \xi_p$ ein.
$-\sum_{k=1}^{n} p_{ik} x_k - s_i + \xi_i = -b_i$ für $i = 1, \ldots, p$
$x \geq 0, s \geq 0, \xi \geq 0$.

$$\left(\begin{array}{ccc|cccc|ccc|c}
-p_{11} & \ldots & -p_{1n} & -1 & & & 0 & 1 & & 0 & -b_1 \\
 & & & & \ddots & & & & \ddots & & \vdots \\
 & & & & & -1 & & 0 & & 1 & -b_p \\
 & & & & & & 1 & 0 & \ldots & 0 & b_{p+1} \\
 & & & & & & & \vdots & & \vdots & \vdots \\
p_{m1} & \ldots & p_{mn} & 0 & & & 1 & 0 & \ldots & 0 & b_m \\
\hline
q_1 & \ldots & q_m & 0 & & & 0 & 0 & \ldots & 0 & -\alpha
\end{array}\right)$$

Basislösung ist $z^{(0)} = (x_1, \ldots, x_m, s_1, \ldots, s_p, s_{p+1}, \ldots, s_m, \xi_1, \ldots, \xi_p) = (0, \ldots, 0, 0, \ldots, 0, b_{p+1}, \ldots, b_m, -b_1,$

Zielfunktion:
$h(\xi) := -\sum_{i=1}^{p} \xi_i \to \max.$

(x, s, ξ) ist eine optimale Lösung des Hilfsproblems $\Leftrightarrow \xi = (0, \ldots, 0)$
(x, s, ξ) ist eine optimale Lösung des Hilfsproblems $\Rightarrow (x, s)$ ist eine zulässige Lösung des Ausgangsproblems.

Phase 2:
Führe mit (x, s) das Simplex-Verfahren durch.

7. Numerische Lösung von Anfangswertaufgaben

a. Einleitung und Beispiele

$\dot{x} = F(t, x); x(t_0) = y^{(0)}$

$$\begin{bmatrix} \dot{x}_1 = f_1(t, x_1, \ldots, x_n); x_1(t_0) = y_1^{(0)} \\ \vdots \\ \dot{x}_n = f_n(t, x_1, \ldots, x_n); x_n(t_0) = y_n^{(0)} \end{bmatrix}$$

Wir setzen die eindeutige Lösbarkeit voraus.

Beispiele:

(1) Biologie: Wachstum
Wachstum ist proportional zum Umfang der Population.
$x(t)$: Umfang der Population zum Zeitpunkt t.
$\dot{x}(t) = \lambda x(t)$ für $\lambda > 0$.

$\dot{x} = F(x)$.

Wachstum wird durch die Umwelt begrenzt.
$\dot{x} = Rx\left(1 - \frac{x}{K}\right), R > 0, K > 0$ (Verhulstgleichung)

(2) Physik / Chemie: Radioaktiver Zerfall
$\dot{x}(t) = k \cdot x(t)$ für $k > 0$.

(3) Physik:

(a) Fallschirmspringer
$v(t)$: Fallgeschwindigkeit zum Zeitpunkt t
t_1: Reißleine wird gezogen; t_2: Fallschirm vollständig offen.
$\dot{v}(t) = g - \frac{k(t) \cdot v(t)^2}{m}$;

$$k(t) = \begin{cases} k_1(t) & \text{, für } 0 \leq t \leq t_1 \\ k_1(t) + \frac{(t-t_1)(k_2-k_1)}{t_2-t_1} & \text{, für } t_1 \leq \qquad k(1) < k_2 \\ k_2(t) & \text{, für } t_2 \leq t \end{cases}$$

$\dot{x} = F(t, x)$.

(b) Pendel (ohne Reibung)
$\ddot{x} = -\lambda \sin(x)$

(4) Räuber-Beute-Modell:
$y(t)$: Räuber
$z(t)$: Beute
$\dot{y} = \alpha \cdot z \cdot y - \beta \cdot y$
$\dot{z} = -\gamma \cdot z \cdot y + \delta \cdot z$ (Differentialgleichungssystem)

Es reicht, Algorithmen für autonome Anfangswertaufgaben 1. Ordnung zu entwickeln:

$$\dot{x} = F(x), x(t_0) = y^{(0)}$$

denn

(1) nicht autonome Differentialgleichungen ⇒ autonome Differentialgleichungen:

$$\dot{x} = F(t, x), x(t_0) = y^{(0)}$$
$$z := \begin{pmatrix} t \\ x \end{pmatrix} \Rightarrow \dot{z} = \begin{pmatrix} \dot{t} \\ \dot{x} \end{pmatrix} = \begin{pmatrix} 1 \\ \dot{x} \end{pmatrix} = \begin{pmatrix} 1 \\ F(z) \end{pmatrix} = G(z) \text{ (autonom)}$$

(2) Differentialgleichungen höherer Ordnung ⇒ Differentialgleichung 1. Ordnung:

$$\ddot{x} = F(x) \rightarrow z_1 := x, z_2 := \dot{x}$$
$$\Rightarrow \dot{z}_1 = \dot{x} = z_2, \dot{z}_2 = \ddot{x} = F(x) = F(z_1).$$

$$\Rightarrow \dot{z} = G(z) = \begin{pmatrix} z_2 \\ F(x) \end{pmatrix}.$$

b. Einteilung der Näherungsverfahren

$\dot{x} = F(x), x(t_0) = y^{(0)}$
Es sei $x(t)$ die exakte Lösung.

Kontinuierliche Verfahren liefern eine (mindestens stetige) Funktion.

- Picard-Iteration:
  $\varphi_0(t) = y^{(0)}$
  $\varphi_{n+1}(t) = y^{(0)} + \int_{t_0}^{t} F(\varphi_n(s))\, ds.$

Diskrete Verfahren liefern in Gitterpunkten $t_0 < t_1 < t_2 < \ldots < t_k < \ldots$ Näherungswerte $y^{(k)} \approx x(t_k)$.

(1) Einschritt-Verfahren: $y^{(k+1)}$ wird aus $y^{(k)}$ berechnet.

(2) Mehrschritt-Verfahren: $y^{(k+1)}$ wird berechnet aus $y^{(k)}, y^{(k-1)}, \ldots, y^{(k-l+1)}$ (l-Schritt-Verfahren).
Man benötigt hier l Startwerte.

c. Einschritt-Verfahren

Gegeben sei $h > 0$ (Schrittweite)
$t_k := t_0 + kh$.

Ein Einschritt-Verfahren basiert auf einer Verfahrensfunktion $V(h, x) \in C(\mathbb{R}^{n+1}; \mathbb{R})$.
$y^{(0)}$ erhält man aus der Anfangsbedingung.
$y^{(k+1)} = y^{(k)} + h \cdot V(h, y^{(k)})$.

Das Euler-Cauchy-Verfahren
Hier wählt man als Verfahrensfunktion:

$$V(h, x) = F(x)$$

$\rightarrow \boxed{y^{(k+1)} = y^{(k)} + hF(y^{(k)})}$

$\dot{x} = F(t, x)$, $\dot{z} = G(z)$.
Ohne Rückführung auf autonome Differentialgleichung:
$y^{(k+1)} = y^{(k)} + hF(t_k, y^{(k)})$.

Matlab:
function z = ecv(funk, y, h)
z = zeros(size(y))

K = feval(funk, y) (liefert funk an der Stelle y)
z = y + h.*K

Explizite Runge-Kutta-Verfahren
werden dargestellt durch eine Verfahrensmatrix:

$$\left(\begin{array}{ccccc} \beta_{21} & & & & \\ \beta_{31} & \beta_{32} & & & \\ \vdots & & \ddots & & \\ \beta_{s1} & & & \beta_{s,s-1} & \\ \hline \gamma_1 & \cdots & \cdots & \gamma_{s-1} & \gamma_s \end{array}\right) \text{ mit } \beta_{ij} \in \mathbb{R}, \gamma_i \in \mathbb{R}.$$

$K^{(1)}(h,x) = F(x)$
$K^{(2)}(h,x) = F(x + h \cdot \beta_{21} K^{(1)}(h,x))$
$\vdots$
$K^{(j)} = F\left(x + h \cdot \sum_{i=1}^{j-1} \beta_{ij} K^{(i)}(h,x)\right)$ für $j = 1, \ldots, s$

$V(h,x) = \sum_{j=1}^{s} \gamma_j K^{(j)}(h,x)$
$y^{(k+1)} = y^{(k)} + hV(h, y^{(k)})$

explizites **Runge-Kutta-Verfahren** der Stufe s.

Beispiele:

(1) $\left(\overline{1}\right)$: **Euler-Cauchy-Verfahren** ($s = 1$).

(2) $s = 2$:

- $\begin{pmatrix} \frac{1}{2} & \\ 0 & 1 \end{pmatrix}$: **Halbschritt-Verfahren**.
- $\begin{pmatrix} 1 & \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$: **zweistufiges Verfahren von Heun**.

  $K^{(1)}(h, y^{(k)}) = F(y^{(k)})$
  $K^{(2)}(h, y^{(k)}) = F\left(y^{(k)} + hK^{(1)}(h, y^{(k)})\right)$
  $\Rightarrow y^{(k+1)} = y^{(k)} + \frac{h}{2}\left(K^{(1)}(h, y^{(k)}) + K^{(2)}(j, y^{(k)})\right)$

(3) $s = 3$

- $\begin{pmatrix} \frac{1}{2} & & \\ -1 & 2 & \\ \frac{1}{4} & 0 & \frac{3}{4} \end{pmatrix}$: **Verfahren von Kutta**.

(4) $s = 4$:

- $\begin{pmatrix} \frac{1}{2} & & & \\ 0 & \frac{1}{2} & & \\ 0 & 0 & 1 & \\ \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{pmatrix}$: klassisches **Runge-Kutta-Verfahren**.

Bemerkung:
Bei allen guten Verfahren gilt $\sum_{i=1}^{s} \gamma_j = 1$.

Implizite Runge-Kutta-Verfahren
werden dargestellt durch eine Verfahrensmatrix:
$\left(\begin{array}{ccc} \beta_{11} & \dots & \beta_{1s} \\ \vdots & \ddots & \vdots \\ \beta_{s1} & \dots & \beta_{ss} \\ \hline \gamma_1 & \dots & \gamma_s \end{array}\right)$ mit $\beta_{ij} \in \mathbb{R}, \gamma_i \in \mathbb{R}, \beta_{ij} \neq 0$ für mindestens ein $j \geq i$.

$K^{(j)}(h,x) = F(x + h \cdot \sum_{i=1}^{s} \beta_{ji} K^{(i)}(h,x))$ für $j = 1, \dots, s$.

Beispiel:
$\left(\frac{1}{1}\right)$ $(s = 1)$

$K^{(1)} = F(\underbrace{y^{(k)} + hK^{(1)}}_{y^{(k+1)}})$

$y^{(k+1)} = y^{(k)} + h \cdot K^{(1)} = y^{(k)} + hF(y^{(k+1)})$
implizites Euler-Verfahren (Rückwärts-Euler)

d. Konsistenz und Konvergenz

Diskretisierungsfehler:
$\dot{x} = F(x), x(t_0) = y^{(0)}$.

$$\Delta(h, t_0, y^{(0)}) := \begin{cases} \frac{x(t_0+h)-x(t_0)}{h} & \text{, falls } h \neq 0; \\ F(y^{(0)}) & \text{, falls } h = 0. \end{cases}$$

Δ heißt **wahre Zuwachsfunktion**.

Einschritt-Verfahren: $y^{(1)} = y^{(0)} + hV(h, y^{(0)})$.

Definition:
Die Funktion $\tau(h, t_0, y^{(0)}) := \Delta(h, t_0, y^{(0)}) - V(h, y^{(0)})$ heißt **lokaler Diskretisierungsfehler**.

$x(t_1) - y^{(1)} = x(t_0 + h) - y^{(1)} = x(t_0 + h) - y^{(0)} - hV(h, y^{(0)}) = h\left(\frac{x(t_0+h)-x(t_0)}{h} - V(h, y^{(0)})\right) = h \cdot \tau(h, t_0, y^{(0)})$.

$x(t_1) - y^{(1)}$: Schrittfehler = Schrittweite · Diskretisierungsfehler.

Bezeichnung:
B_N sei die Menge der Funktionen $F : \mathbb{R}^n \to \mathbb{R}^n$ mit

- F ist stetig und beschränkt;
- es existieren alle partiellen Ableitungen der Ordnung N von F; diese sind stetig und beschränkt.

Definition:
Das von der Verfahrensfunktion $V(h, x)$ definierte Verfahren heißt **konsistent**, falls gilt

$$\lim_{h \to 0} \tau(h, t_0, y^{(0)}) = 0$$

für jeden Anfangswert (t_0, y_0) und jedes $F \in B_1$.
Das von $V(h, x)$ erzeugte Verfahren hat die **Konsistenzordnung** p, falls gilt

$$\tau(h, t_0, y^{(0)}) = O(h^p) \text{ für } h \to 0$$

für jeden Anfangswert $(t_0, y^{(0)})$ und jedes $F \in B_p$

Aus Konsistenz folgt also $V(0, y^{(0)}) = \lim_{h \to 0} V(h, y^{(0)}) = F(y^{(0)}) = V(0, y^{(0)})$

Runge-Kutta-Verfahren:
$V(h, x) = \sum_{i=1}^{s} \gamma_i K^{(i)}(h, x)$

Konsistenz: $F(x) = V(0,x) = \sum_{i=1}^{s} \gamma_i K^{(i)}(0,x) = \underbrace{\left(\sum_{i=1}^{s} \gamma_i\right)}_{\text{muss 1 sein}} F(x)$

Fazit:
Runge-Kutta-Verfahren konsistent $\Rightarrow \sum_{i=1}^{s} \gamma_i = 1$.

Beispiel: Konsistenzordnung des Euler-Cauchy-Verfahrens (im skalaren Fall):
$\dot{x} = f(x), x(t_0) = y^{(0)}$ (Taylorentwicklung)

$x(t_0 + h) = x(t_0) + \dot{x}(t_0) \cdot h + \ddot{x}(\xi)\frac{h^2}{2}$
$\dot{x}(t_0) = f(x(t_0)) = f(y^{(0)})$
$\ddot{x}(\xi) = f'(x(\xi)) \cdot \dot{x}(\xi) = f'(x(\xi)) \cdot f(x(\xi))$

$\Rightarrow \Delta(h, t_0, y^{(0)}) = \frac{x(t_0+h)-x(t_0)}{h} = \dot{x}(t_0) + \ddot{x}(\xi) \cdot \frac{h}{2} = f(y^{(0)}) + \frac{h}{2} f'(x(\xi)) \cdot f(x(\xi))$

$V(h, y^{(0)}) = f(y^{(0)})$
$\Rightarrow \tau(h, t_0, y^{(0)}) = \frac{h}{2} \cdot f'(x(\xi)) \cdot f(x(\xi)) = O(h)$ (wegen $f \in B_1$)

$\Rightarrow$ Konsistenzordnung 1.

Konvergenz:
$\dot{x} = F(x), x(t_0) = y^{(0)}$.
Sei $x(t)$ die exakte Lösung. Wir berechnen mit einem Einschrittverfahren Näherungen in Gitterpunkten $t_i := t_0 + i \cdot h$ ($h > 0$ Schrittweite), $y_h^{(i)}$
Intervall: $[t_0, b]$.
Sei $R = R(h)$ der Index, welcher $t_{R-1} < b \leq t_R$ erfüllt.

$e_h^{(i)} := x(t_i) - y_h^{(i)}$ für $i = 0, 1, \ldots, R$.
$E(h) := \max\{\|e_h^{(i)}\| : i = 0, \ldots, R\}$ (maximaler Fehler in $[t_0, t_R]$)

Definition:

(1) Das Einschrittverfahren heißt **konvergent**, falls gilt

$$\lim_{h \to 0} E(h) = 0$$

für jedes $F \in B_1$, jeden Anfangswert $(t_0, y^{(0)})$ und jedes Intervall $[t_0, b]$.

(2) Das Einschrittverfahren hat die Konvergenzordnung p ($p \in \mathbb{N}_1$), falls

$$E(h) = O(h^p)$$

für jedes $F \in B_p$, jeden Anfangswert $(t_0, y^{(0)})$ und jedes Intervall $[t_0, b]$.

Anmerkung:
Ein konsistentes explizites Runge-Kutta-Verfahren ist auch konvergent.

Formelfehler: $E(h)$: h klein $\Rightarrow$ $E(h)$ klein.
Rundungsfehler: $R(h)$: h klein $\Rightarrow$ $R(h)$ groß.
Optimales h??

e. Schrittweitensteuerung

$\underbrace{t_k, y^k}_{\text{sei akzeptiert}} \rightarrow t_{k+1}, y^{(k+1)}$

Formelfehler = Fortpflanzungsfehler (Fehler, der schon in $t_k, y^{(k)}$ enthalten ist) + Schrittfehler.

Idee: Kontrolliere den Schrittfehler.

Betrachte die Anfangswertaufgaben $\dot{x} = F(x)$, $x(t_k) = y^{(k)}$.
Exakte Lösung sei $u(t)$. $y_h^{(k+1)} = y^{(k)} + hV(h, y^{(k)})$.
Das Einschrittverfahren habe die Konsistenzordnung p.
$u(t_h + h) - y_h^{(k+1)} = d \cdot h^{p+1} + O(h^{p+2})$.
Wenn h klein ist, dann kann $O(h^{p+2})$ vernachlässigt werden. Man nennt deshalb $d \cdot h^{p+1}$ das **Fehlerhauptglied**.

Berechne zusätzlich zwei Schritte mit Schrittweite $\frac{h}{2}$:
$z^{(k+1)} = y^{(k)} + \frac{h}{2} V\left(\frac{h}{2}, y^{(k)}\right)$
$z_{\frac{h}{2}}^{(k+1)} = z^{(k+1)} + \frac{h}{2} V\left(\frac{h}{2}, z^{(k+1)}\right)$

Satz:
Das Einschrittverfahren besitze die Konsistenzordnung p und $f \in B_{p+1}$. Dann gilt

$$u(t_k + h) - y_h^{(k+1)} = \frac{y_{\frac{h}{2}}^{(k+1)} - y_h^{(k+1)}}{1 - 2^{-p}} + O(h^{p+2}).$$

Korollar:
Setzt man

$$y^{(k+1)} := y_h^{(k+1)} + \frac{y_{\frac{h}{2}}^{(k+1)} - y_h^{(k+1)}}{1-2^{-p}} = \frac{2^p y_{\frac{h}{2}}^{(k+1)} - y_h^{(k)}}{2^{p-1}},$$

so gilt: $u(t_k + h) - y^{(k+1)} = O(h^{p+2})$ (Konsistenzordnung wird um 1 größer).

Praktische Durchführung der Schrittweitensteuerung am Beispiel des klassischen Runge-Kutta-Verfahrens (also Konsistenzordnung 4) für nichtautonome Anfangswertaufgaben (skalar)

$$\text{sei gegeben } \dot{x} = f(t,x), x(t_0) = y^{(0)}$$

Sei $h > 0$ eine gegebene Schrittweite, $0 < \varepsilon_1 < \varepsilon_2$ (Genauigkeitsschranken). Sei $y^{(k)}$ eine akzeptierte Näherung zum Zeitpunkt t_k.

(1) Berechne mit dem aktuellen h die Näherungen $y_h^{(k+1)}$ und $y_{\frac{h}{2}}^{(k+1)}$.

(2) Berechne die Fehlerschätzung $E := \frac{16}{15} \left\| y_{\frac{h}{2}}^{(k+1)} - y_h^{(k+1)} \right\|$

(3) Gilt $E > \varepsilon_2$, so setze $h_{neu} := \frac{h}{2}$ und gehe zu (1).

(4) Ist $E \leq \varepsilon_2$, so setze gemäß dem obigen Korollar

$$y^{(k+1)} = \frac{16 y_{\frac{h}{2}}^{(k+1)} - y_h^{(k+1)}}{15} \text{ und } t_{k+1} := t_k + h.$$

$y^{(k+1)}$ wird zum Zeitpunkt t_k akzeptiert.

(5) Gilt $E < \varepsilon_1$, so setze $h = 2h$.

f. Stabilität

Test-Differentialgleichung: $\dot{x} = \lambda x, x(0) = 0, \lambda < 0$.
Die exakte Lösung: $x(t) = a \cdot \exp(\lambda t); \lim_{t\to\infty} x(t) = 0$.

Löse diese Anfangswertaufgabe mit einem Einschrittverfahren: Gitter $t_k = t_0 + k \cdot h$, Näherungswert $y^{(k)}$.

FORDERUNG: $\lim_{k\to\infty} y^{(k)}$ muss 0 sein (sonst ist das Verfahren schlecht oder die Schrittweite zu groß)

Z.B. Euler-Cauchy-Verfahren:
$y^{(k)} = y^{(k-1)} + h \cdot \lambda y^{(k-1)} = (1 + \lambda h)y^{(k-1)} = (1 + \lambda h)^2 y^{(k-2)} = \ldots = (1 + \lambda h)^k y^{(0)}$.
Forderung: $(1 + \lambda h)^k y^{(0)} \to 0$ für $k \to \infty$.
$\Rightarrow |1 + \lambda h| < 1$ oder $\lambda h \in (-2, 0)$. $\to$ Einschränkung an die Schrittweite.
Z.B. $\lambda = -10 \Rightarrow h < 0,2$.

Man setzt $W(z) := 1 + z$.
$\Rightarrow y^{(k)} = W(\lambda h)^k \cdot a$.
$W(z)$ heißt **Stabilitätsfunktion** des Euler-Cauchy-Verfahrens.

Allgemein wird jedem Runge-Kutta-Verfahren $\beta = \left(\begin{array}{ccc} \beta_{11} & \ldots & \beta_{1s} \\ \vdots & \ddots & \vdots \\ \beta_{s1} & \ldots & \beta_{ss} \\ \hline \gamma_1 & \ldots & \gamma_s \end{array}\right)$ eine Stabilitätsfunktion $W_\beta(z)$ zugeordnet.
Dies ist die Funktion, welche $y^{(k)} = W_\beta(\lambda h)^k \cdot y^{(0)}$ (also $y^{(k)} = W_\beta(\lambda h) \cdot y^{(k-1)}$) erfüllt.

FORDERUNG:
$\lim_{k\to\infty} y^{(k)} = 0 \Rightarrow |W_\beta(\lambda h)| < 1$.

DEFINITION:
Die Menge $S_\beta := \{z \in \mathbb{R} : |W_\beta(z) < 1\}$ heißt **Stabilitätsbereich** von β.

NACHTEIL der expliziten Runge-Kutta-Verfahren: der Stabilitätsbereich ist relativ klein, d.h. es gibt eine Einschränkung an die Schrittweite h.

IMPLIZITES EULER-VERFAHREN:
$y^{(k)} = y^{(k-1)} + hf(y^{(k)}) = y^{(k-1)} + h\lambda y^{(k)}$.
$\Rightarrow y^{(k)} = \frac{1}{1-h\lambda} y^{(k-1)}$.

$\Rightarrow$ Stabilitätsfunktion $W_\beta(z) = \frac{1}{1-z}$.
$|W_\beta(z)| < 1$ f+r alle $z \in (-\infty, 0)$.
Also gibt es keine Einschränkung an die Schrittweite h.

8. Numerische Integration

a. Einleitung

Sei $f \in C[a,b]$. Dann existiert $\int_a^b f(x)\,dx$.

Ist $F(x)$ eine Stammfunktion zu $f(x)$, so gilt $\int_a^b f(x)\,dx = F(b) - F(a)$.

Z.B. günstig, falls f ein Polynom ist.

Problematisch:

- Stammfunktion ist nicht explizit bekannt (z.B. $f(x) = e^{-x^2}$),
- Stammfunktion wird kompliziert (z.B. bei rationalen Funktionen),
- f ist nicht explizit bekannt, sondern nur $f(x_i$ für $i = 0, \ldots, m$ (z.B. bei Messergebnissen).

Deshalb braucht man Näherungsmethoden:

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{m} w_i f(x_i) \qquad \text{(\textbf{Quadratursumme})}$$

Die x_i heißen **Stützstellen**, die w_i heißen **Gewichte**.

Formelfehler (bzw. Quadraturfehler):

$$R[f] := \int_a^b f(x)\,dx - \sum_{i=0}^{m} w_i f(x_i)$$

$R : C[a,b] \to \mathbb{R}$ ist ein lineares Funktional (also $R[\alpha f + \beta g] = \alpha R[f] + \beta R[g]$).

Eine Quadratursumme (Quadraturformel) heißt **exakt** für ein $f \in C[a,b]$, falls $R[f] = 0$ gilt. Eine Quadraturformel hat den **Exaktheitsgrad** $n \in \mathbb{N}$, falls gilt

$$R[p] = 0 \text{ für alle } p \in \Pi_n \text{ und}$$
$$R[q] \neq 0 \text{ für ein } q \in \Pi_{n+1}$$

Äquivalent dazu:

$$R[1] = R[x] = \ldots = R[x^n] = 0 \text{ und } R[x^{n+1}] \neq 0$$

Günstige Wahl einer Quadratursumme:

- *interpolatorische Quadraturformel:* Integrand f wird durch ein Polynom p approximiert (z.B. Interpolation):

$$\int_a^b f(x)\,dx \approx \int_a^b p(x)\,dx$$

- *zusammengesetzte Quadraturformel:* das Intervall $[a, b]$ wird in Teilintervalle zerlegt, auf jedem Teilintervall verwendet man eine eigene Quadratursumme
- *Romberg-Verfahren:* durch geeignete Kombination von zwei Quadratursummen erhält man eine neue (vielleicht bessere) Quadraturformel.

Affine Transformationen:
Wie erhält man aus einer Formel für ein Intervall $[a, b]$ eine Formel für ein Intervall $[c, d]$?

$\int_a^b f(x)\,dx \approx \sum_{i=0}^{m} w_i f(x_i)$.
Sei nun $g \in C[c, d]$; gesucht ist $\int_c^d g(s)\,ds$.
$\varphi : [a, b] \to [c, d]$, $\varphi(t) = \alpha t + \beta$ mit $\varphi(a) = c$ und $\varphi(b) = d$. $\Rightarrow \varphi(t) = \frac{d-c}{b-a}t + \frac{bc-ad}{b-a}$.

$f := g \circ \varphi \in C[a, b] \Rightarrow \int_c^d g(s)\,ds = \int_a^b g(\varphi(t)) \cdot \varphi'(t)\,dt = \alpha \int_a^b g(\alpha t + \beta)\,dt$
$\approx \alpha \cdot \sum_{i=0}^{m} w_i g(\alpha x_i + \beta) = \sum_{i=0}^{m} \overline{w_i} g(y_i)$ mit $\overline{w_i} := \alpha \cdot w_i$.

b. Interpolatorische Quadraturformeln

Sei $a \leq t_0 < t_1 < \ldots < t_m \leq b$ und $f \in C[a, b]$. Bilde das Lagrange-Interpolationspolynom $p(t)$ zu diesen Stützstellen $p(t) = \sum_{i=0}^{m} l_i f(t_i)$.

$$\int_a^b f(t)\,dt \approx \int_a^b p(t)\,dt = \int_a^b \left(\sum_{i=0}^m l_i(t) f(t_i)\right) dt = \sum_{i=0}^m \left[f(t_i) \cdot \int_a^b l_i(t)\,dt\right] = \sum_{i=0}^m w_i f(t_i) \text{ mit } w_i = \int_a^b l_i(t)\,dt.$$

Satz:
Eine interpolatorische Quadraturformel (mit $m+1$ Stützstellen) hat mindestens den Exaktheitsgrad m.

Lemma:
Für eine interpolatorische Quadraturformel gilt:

$$\sum_{i=0}^m w_i = b - a.$$

Beispiele:

- *Mittelpunktsregel* ($m = 0$, $t_0 = \frac{a+b}{2}$):
  $\int_a^b f(t)\,dt = (b-a) \cdot f(\frac{a+b}{2}) + R[f]$ mit $R[f] = \frac{(b-a)^3}{24} f^{(2)}(\xi)$ mit einem $\xi \in (a,b)$.
  Man kann zeigen, dass der Exaktheitsgrad $M = 1$ ist.
- *Trapezregel* ($m = 1, t_0 = a, t_1 = b$):
  $\int_a^b f(t)\,dt = \frac{b-a}{2}(f(a) + f(b)) + R[f]$ mit $R[f] = -\frac{(b-a)^3}{12} f^{(2)}(\eta)$.
  Wieder lässt sich zeigen, dass der Exaktheitsgrad $M = 1$ ist.
- *Keplersche Fassregel* ($m = 2, t_0 = a, t_1 = \frac{a+b}{2}, t_2 = b$):
  $\int_a^b f(t)\,dt = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) + R[f]$.
- *Newton-Cotes-Formel* (m beliebig, $t_i = a + i \cdot \frac{b-a}{m}$ (äquidistant))

c. Zusammengesetzte Quadraturformeln

Zerlege das Intervall $[a,b]$ in N Teilintervalle. $a = x_0 < x_1 < \ldots < x_N = b$.
$\int_a^b f(x)\,dx = \sum_{i=1}^N \int_{x_{i-1}}^{x_i} f(x)\,dx$.

Verwende auf jedem Teilintervall eine einfache Quadraturformel (z.B. Trapezregel, Keplersche Fassregel, ...).

Beispiel (zusammengesetzte Trapezregel):
$\int_{x_{i-1}}^{x_i} f(x)\,dx = \frac{x_i - x_{i-1}}{2}(f(x_{i-1}) + f(x_i)) + R_i[f]$.
Fall: alle Teilintervalle gleich: $x_i = a + i \cdot \frac{b-a}{N}$.
$\Rightarrow \frac{x_i - x_{i-1}}{2}(f(x_{i-1}) + f(x_i)) + R_i[f] = \frac{b-a}{2N}(f(x_{i-1}) + f(x_i)) + R_i[f]$ mit $R_i[f] = -\frac{(b-a)^3}{12N^3} f^{(2)}(\xi_i)$.
Also: $\int_a^b f(x)\,dx = \frac{b-a}{2N} \sum_{i=1}^{N} (f(x_{i-1}) + f(x_i)) + R[f] = \frac{b-a}{N}\left(\frac{1}{2}f(a) + \frac{1}{2}f(b) + \sum_{i=1}^{N-1} f(x_i)\right) + R[f]$.

Bei der zusammengesetzten Trapezregel gilt, falls $f \in C^2[a,b]$: $R[f] = -\frac{(b-a)^3}{12N^2} f^{(2)}(\xi)$ mit $\xi \in (a,b)$

Fehlerschranke:
$|R[f]| \leq \frac{(b-a)^3}{12N^2} \|f^{(2)}\|_{[a,b]} \to 0$ für $N \to \infty$.
$R[f] = \sum_{i=1}^{N} R_i[f] = \sum_{i=1}^{N} -\frac{(b-a)^3}{12N^3} f^{(2)}(\xi_i) = -\frac{(b-a)^3}{12N^3} \cdot \sum_{i=1}^{N} f^{(2)}(\xi_i) = \frac{1}{N} \sum_{i=1}^{N} f^{(2)}(\xi_i) = f^{(2)}(\eta)$.

Fehlerabschätzung (zusammengesetzte Mittelpunktsregel):
$R[f] = \frac{(b-a)^3}{24N^2} f^{(2)}(\xi)$ mit $\xi \in (a,b)$, falls $f \in C^2[a,b]$.

Anwendung:
Sei $f \in C^2[a,b]$ und $\varepsilon > 0$. Bestimme einen Näherungswert J für $I = \int_a^b f(x)\,dx$ mit $|J - I| < \varepsilon$.
Wähle dazu N so, dass $\frac{(b-a)^3}{12N^2} \|f^2\| < \varepsilon$ gilt. Berechne dann diesen Näherungswert mit Hilfe der zusammengesetzten Trapezregel mit N Teilintervallen.

Zusammengesetzte Keplersche Fassregel (Simpsonregel):
Teilintervalle: $R_i[f] = -\frac{(b-a)^5}{2880N^4} f^{(4)}(\xi_i)$, falls $f \in C^4[a,b]$.

Gesamt:
$R[f] = -\frac{(b-a)^5}{2880N^4} f^{(4)}(\xi)$.

d. Das Prinzip der Extrapolation

Sei $\varphi : [-a, a] \to \mathbb{R}$ mit $\varphi(h) = \varphi(-h)$ für alle $h \in [-a, a]$.
Seien $0 < h_m < \ldots < h_1 < h_0$ gegeben und seien $\varphi(h_0), \ldots, \varphi_{h_m}$ bekannt.
Gesucht: $\varphi(0)$

Idee 1: bestimme das Interpolationspolynom $\overline{p}$ zu den Stützpaaren $(h_i, \varphi(h_i))$.
$\to \overline{p}(0) \approx \varphi(0)$.

Idee 2: bestimme das Interpolationspolynom $\overline{q}$ zu den Stützpaaren $(\pm h_i, \varphi(h_i))$.
$\to \overline{q}(0) \approx \varphi(0)$.

Idee 3: Bilde das Interpolationspolynom zu den Stützpaaren $(h_i^2, \varphi(h_i))$ für $i = 0, \ldots, m$.
$\Rightarrow p(0) \approx \varphi(0)$ (beste Idee)
Durchführung mit Neville-Schema.

Sei nun $\varphi(h) = \underbrace{\tau_0}_{=\varphi(0)} + \tau_1 h^2 + \tau_2 h^4 + \tau_3 h^6 + \ldots$.

Sei $h_0 > h_1$, $h_1 = \alpha \cdot h_0$.
$p_0 = \tau_0 + \tau_1 h_0^2 + \tau_2 h_0^4 + \ldots$
$p_1 = \tau_0 + \tau_0 h_1^2 + \tau_2 h_1^4 + \ldots$

$p_{0,1} = \frac{(0-h_1^2)p_0 - (0-h_0^2)p_1}{h_0^2 - h_1^2} = \frac{h_0^2(\tau_0+\tau_1 h_1^2+\tau_2 h_1^4+\ldots) - h_1^2(\tau_0+\tau_1 h_0^2+\tau_2 h_0^4+\ldots)}{h_0^2 - h_1^2} = \tau_0 - \tau_2 h_0^2 h_1^2 - \ldots = \tau_0 + \overline{\tau}_2 h_0^4 + \overline{\tau}_3 h_0^6 + \ldots$ mit $\overline{\tau}_2 = -\tau_2 \alpha^2$.

Fazit:
$p_0 - \tau_0 = \tau_1 h_0^2 + \tau_2 h_0^4 + \ldots$
$p_{01} - \tau_0 = \overline{\tau}_2 h_0^4 + \overline{\tau}_3 h_0^6 + \ldots$
$p_{012} - \tau_0 = \widetilde{\tau}_3 h_0^6 + \ldots$.

e. Das Romberg-Verfahren

$T(h) = \frac{h}{2} \cdot (f(a) + f(b) + 2 \sum_{i=1}^{N-1} f(x_i))$ für $h = \frac{b-a}{N}$ (zusammengesetzte Trapezregel mit N Teilintervallen)

Beachte: $T(h)$ ist nur für gewisse h definiert.

Satz (Summenformel von Euler-MacLaurin):

$$T(h) = \underbrace{\int_a^b f(t)\,dt}_{\tau_0} + \tau_1 h^2 + \tau_2 h^4 + \tau_3 h^6 + \ldots.$$

Neville-Schema:

	Romberg-Tableau			
h_0^2	$p_0 = T(h_0) = T_0^0$	T_1^0	$\ldots$	T_i^0
h_1^2	$p_0 = T_0^1$	T_1^1	$\ldots$	
h_2^2	$p_0 = T_0^2$	T_2^1		
$\vdots$	$\vdots$	$\vdots$		
h_i^2	$p_0 = T_0^i$			

T_0^0: Trapezregel
T_0^1: Trapezregel mit zwei Teilintervallen
T_0^2: Trapezregel mit 4 Teilintervallen
T_0^i: Trapezregel mit 2^i Teilintervallen

$$\boxed{T_j^k = \frac{4^j \cdot T_{j-1}^{k+1} - T_{j-1}^k}{4^j - 1}}$$

$T_j^k = p_{k\ldots k+j}(0)$; $h_{k+j}^2 = \frac{h_k^2}{4^j}$

Praktische Durchführung des Romberg-Verfahrens:
$M(h_i)$: zusammengesetzte Mittelpunktsregel mit 2^i Teilintervallen.

Lemma:
$T(h_{i+1}) = \frac{1}{2}(T(h_i) + M(h_i))$.

Numerisches Verfahren:
Romberg-Tableau wird zeilenweise aufgebaut:

- maximale Zeilenzahl N vorgeben, so dass Algorithmus auch abbricht, falls das Romberg-Verfahren nicht konvergiert.

- T_0^0 wird berechnet ($T_0^0 = T(h_0)$)
- i-te Zeile: berechne $M(h_{i-1})$
  $T_0^i = T(h_i) = \frac{1}{2}(T(h_{i-1}) + M(h_{i-1}) = \frac{1}{2}(T_0^{i-1} + M(h_{i-1})$
- für die restlichen Einträge in der i-ten Zeile berechne T_j^k für $j + k = i$: mit der Rekursionsformel.
- Abbruchbedingung: sei $\varepsilon > 0$ als Genauigkeitsschranke gegeben.
  $|T_{i-1}^1 - T_{i-1}^0| < \varepsilon \Rightarrow$ akzeptiere T_i^0
  $|T_{i-1}^1 - T_{i-1}^0| \geq \varepsilon \Rightarrow i = i + 1$, gehe zum dritten Punkt, falls $i \leq N$.

f. Gauß-Quadraturformeln

Maximaler Exaktheitsgrad bei $m + 1$ Stützstellen $t_0 < t_1 < \ldots < t_m$?

Bekannt: interpolatorische Quadraturformel $\Rightarrow$ Exaktheitsgrad $\geq m$.

Lemma:
Eine Quadraturformel mit $m + 1$ Stützstellen hat höchstens den Exaktheitsgrad $2m + 1$.

Beweis:
$q(t) := \prod_{i=0}^{m}(t - t_i)^2; q \in \Pi_{2_m+2}$
$\Rightarrow q(t) \geq 0$ und $\int_a^b q(t)\,dt < 0$.
$\sum_{j=0}^{m} w_j \underbrace{q(t_j)}_{=0} = 0$
$\Rightarrow R[q] \neq 0$.

□

Legendre-Polynome
$C[a, b]$; $f, g \in C[a, b]$:
$\langle f, g\rangle := \int_{-1}^{1} f(t)g(t)\,dt$ definiert ein Skalarprodukt auf $C[-1, 1]$.
Zwei Polynome p, q heißen orthogonal, falls $\langle p, q\rangle = 0$.

Satz:
Es gibt eine Folge $(P_n)_{n\in\mathbb{N}}$ orthogonaler Polynome mit positivem Hauptkoeffizienten und $\deg P_n = n$ (d.h. $\langle P_i, P_k\rangle = 0$ für $i \neq k$).
(ohne Beweis - Beweis z.B. mit Basis von $C[a,b]$ und Gram-Schmidt)

Anmerkung:
$\langle r, P_k\rangle = 0$ für alle $r \in \Pi_{j-1}$

Diese Polynome P_n heißen Legendre-Polynome.

Für die Legendre-Polynome gilt

$$P_{n+1}(t) = \tfrac{2n+1}{n+1} t \cdot P_n(t) - \tfrac{n}{n+1} P_{n-1}(t)$$

und $P_0(t) = 1$; $P_1(t) = t$.

(Bemerkung: mit dem Skalarprodukt $\int_{-1}^{1} \frac{1}{\sqrt{1-t^2}} f(t)g(t)\,dt$ erhält man die Tschebyscheff-Polynome)

Satz:
P_n hat in $(-1,1)$ n einfache Nullstellen.

Beweis:
Annahme: P_n hat nur $l < n$ Nullstellen $-1 < t_1 < \ldots < t_l < 1$ ungerader Ordnung.
$q(t) = \prod_{j=1}^{l}(t - t_j) \Rightarrow \deg q < n$.

$P_n(t) \cdot q(t) \geq 0 \Rightarrow \int_{-1}^{1} P_n(t)q(t)\,dt > 0$

Andererseits gilt $\langle q, P_n\rangle = 0$, da $\deg q < \deg P_n$. $\langle q, P_n\rangle = 0 = \int_{-1}^{1} P_n(t)q(t)$.

$\Rightarrow$ Widerspruch; es gibt also mindestens n Nullstellen ungerader Ordnung, also n einfach Nullstellen in $(-1,1)$.

□

Verwende nun die Nullstellen $-1 < t_0 < \ldots < t_m < 1$ von P_{m+1} und bilde damit die interpolatorische Quadraturformel.

Definition:
Diese Formeln heißen **Gauß-Quadraturformeln**.

Satz:
Die Gauß-Quadraturformel mit $m+1$ Stützstellen hat den Exaktheitsgrad $M = 2m+1$.

Beweis:
Sei $q \in \Pi_{2m+1}$.
$\Rightarrow q = r \cdot P_{m+1} + s$ mit $r, s \in \Pi_m$
$\Rightarrow R[q] = R[\gamma \cdot P_{m+1}] + R[s]$ mit $R[s] = 0$, da interpolatorische Quadraturformel und $s \in \Pi_m$.

$$R[r \cdot P_{m+1}] = \underbrace{\int_{-1}^{1} r(t) P_{m+1}(t)\, dt}_{=\langle r, P_{m+1}\rangle = 0, \text{ da } r \in \Pi_m} - \underbrace{\sum_{i=0}^{m} w_i r(t_i) \cdot \underbrace{P_{m+1}(t_i)}_{=0}}_{=0} = 0.$$

□

Lemma:
Bei Gauß-Quadraturformeln sind alle Gewichte positiv (vgl. Übungsaufgabe auf Blatt 11).

Beispiel:
$m = 2$: (vgl. Übungsaufgabe 19 (a)).

9. Eigenwertaufgaben

a. Eigenwerte und Eigenvektoren

Sei $A \in \mathbb{R}^{n\times n}$. Ein $\lambda \in \mathbb{C}$ heißt Eigenwert von A, falls ein $x \in \mathbb{C}^n \setminus \{0\}$ gibt mit $Ax = \lambda x$. x heißt dann ein Eigenvektor.
Eigenwerte sind genau die Nullstellen vom charakteristischen Polynom $p(\lambda) := \det(\lambda I - A)$. In der Numerik vermeidet man allerdings diesen Weg (weil instabil!, vgl. Abschnitt 12).

Sei Q eine invertierbare Matrix. Dann sind A und $B := QAQ^{-1}$ ähnlich und haben damit dieselben Eigenwerte. *(Eigenvektoren: x ist Eigenvektor zu $A \Rightarrow y := Qx$ ist Eigenvektor zu B)*

$$B = \begin{pmatrix} b_{11} & \dots & * \\ & \ddots & \vdots \\ 0 & & b_{nn} \end{pmatrix} \Rightarrow b_{11}, \dots, b_{nn} \text{ sind die Eigenwerte.}$$

In der Numerik verwendet man Ähnlichkeitstransformationen, insbesondere orthogonale Ähnlichkeitstransformationen (da dann keine Probleme mit den Inversen Matrizen auftreten). Sei Q eine orthogonale Matrix, dann gilt $Q^{-1} = Q^T$, also $B = QAQ^T$.

b. Beispiel aus der Physik: lineare Kette

N Massen M, die durch $N + 1$ Federn linear verbunden sind; die beiden äußeren Federn werden fest eingespannt.

c_i = Federkonstante der i-ten Feder
l_0 = Länge der Federn (alle gleich lang)
L = Endpunkt-Anfangspunkt
$x_i(t)$: Ort der i-ten Masse zum Zeitpunkt t
$\dot{x}_i(t)$: Geschwindigkeit der i-ten Masse zum Zeitpunkt t
$\ddot{x}_i(t)$: Beschleunigung der i-ten Masse zum Zeitpunkt t
$F_i(t)$: Federkraft der i-ten Feder zum Zeitpunkt t
$K_i(t)$: Kraft, die auf die i-te Kugel wirkt zum Zeitpunkt t

Bewegungsgleichung (für die i-te Kugel): $M \cdot \ddot{x}_i(t) = K_i(t)$
$K_i(t) = F_i(t) - F_{i-1}(t)$
Hook'sches Gesetz: $F_i(t) = c_i \cdot (x_{i+1}(t) - x_i(t) - l_0)$

$\Rightarrow \ddot{x}_i(t) = \frac{1}{M} K_i(t) = \frac{1}{M} [c_i (x_{i+1}(t) - x_i(t) - l_0) - c_{i-1} (x_i(t) - x_{i-1}(t) - l_0)]$
$= \frac{1}{M} [c_{i-1} x_{i-1}(t) - (c_i + c_{i-1}) x_i(t) + c_i x_{i+1}(t) + (c_{i-1} - c_i) l_0]$ für $i = 1, \ldots, N$

$$x(t) = \begin{pmatrix} x_0(t) \\ \vdots \\ x_N(t) \end{pmatrix}; A = \frac{1}{M} \begin{pmatrix} c_0 + c_1 & -c_1 & & 0 \\ -c_1 & c_1 + c_2 & \ddots & \\ & \ddots & \ddots & -c_{N-1} \\ 0 & & -c_{N-1} & c_{n-1} + c_N \end{pmatrix};$$

$$b = \frac{1}{M} \begin{pmatrix} (c_0 - c_1) l_0 \\ \vdots \\ (c_{N-2} - c_{N-1}) l_0 \\ (c_{N-1} - c_N) l_0 + c_N L \end{pmatrix} \Rightarrow \boxed{\ddot{x}(t) = b - Ax(t)}$$

Transformation auf ein homogenes System:
Ruhelage: $0 = b - Ax$: sei $\overline{x}$ die Lösung von $Ax = b$.
$\Rightarrow z(t) = \overline{x}$ (spezielle Lösung)
Sei $x(t)$ eine Lösung von $\ddot{x}(t) = b - Ax(t)$
Setze $y(t) := x(t) - \overline{x}$
$\Rightarrow \ddot{y}(t) = \ddot{8}x) = b - Ax(t) = A\overline{x} - Ax(t) = -A(x(t) - \overline{x}) = -Ay(t)$.

Lösungen des homogenen Systems:
Falls alle $c_i > 0$, so ist A positiv definit (vgl. Übungsaufgabe 9c); das ist hier der Fall, da c_i Federkonstanten sind.
$\Rightarrow A$ hat lauter positive Eigenwerte $0 < \lambda_1 \leq \ldots \leq \lambda_n$.
Seien $v^{(1)}, \ldots, v^{(n)}$ zugehörige (linear unabhängige) Eigenvektoren.

Behauptung:
Alle Lösungen von $\dot{y} = -Ay$ sind von der Form
$$y(t) = \sum_{j=1}^{N} \left(\alpha_j \cos(\sqrt{\lambda_j} \cdot t) \cdot v^{(j)} + \beta_j \sin(\sqrt{\lambda_j} \cdot t) \cdot v^{(j)} \right).$$
Beweis siehe Übungsblatt.

Die Lösungen des inhomogenen Systems sind dann
$$x(t) = \sum_{j=1}^{N} \left(\alpha_j \cos(\sqrt{\lambda_j} \cdot t) \cdot v^{(j)} + \beta_j \sin(\sqrt{\lambda_j} \cdot t) \cdot v^{(j)} \right) + \overline{x} \text{ mit } \alpha_j, \beta_j \in \mathbb{R}.$$

Anfangsbedingungen:
$x_i(0)$ für $i = 1, \ldots, N$ und
$\dot{x}_i(0)$ für $i = 1, \ldots, N$ legen α_j und β_j eindeutig fest.

Numerische Probleme:

- lineares Gleichungssystem mit A positiv definit (z.B. Cholesky-Zerlegung);
- Berechnung der Eigenwerte und Eigenvektoren von A;
- zwei lineare Gleichungssysteme zur Berechnung der α_j bzw. der β_j.

c. Lokalisierung der Eigenwerte

Sei $A = \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nn} \end{pmatrix} \in \mathbb{R}^{n \times n}$

$K_i := \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{k=1, k \neq i}^{n} |a_{ik}|\}$ (Kreis in der komplexen Ebene um a_{ii} mit Radius $\sum_{k=1, k \neq 1}^{n} |a_{ik}|$)

Definition:
Diese K_i heißen Gerschgorin-Kreise.

Satz: (von Gerschgorin)
$\bigcup_{j=1}^{K_j}$ enthält alle Eigenwerte von A.

(Beweis vgl. Übungsaufgabe)

Beispiel:
$A = \begin{pmatrix} 1 & 0,1 & -0,1 \\ 0 & 2 & 0,4 \\ -0,2 & 0 & 3 \end{pmatrix}$

d. Transformation auf obere Hessenberg-Gestalt

Definition:
Eine $(n \times n)$-Matrix der Gestalt $\begin{pmatrix} * & \dots & \dots & * \\ * & \ddots & & \vdots \\ & \ddots & \ddots & \vdots \\ 0 & & * & * \end{pmatrix}$ heißt obere **Hessenberg-Matrix**.

Satz:
Sei $A \in \mathbb{R}^{n\times n}$. Dann gibt es eine orthogonale Matrix Q, so dass $B = QAQ^T$ eine obere Hessenberg-Matrix ist.

Q ist Produkt von $n-2$ Householder-Matrizen (siehe QR-Zerlegung, 4.e.)

Beweisidee:
$A = \left(\begin{array}{c|c} a_{11} & b \\ \hline c & A \end{array}\right)$; es gibt eine $(n-1)\times(n-1)$ Householder-Matrix $\widetilde{H}_1$ mit $\widetilde{H}_1 c = be_1$ $(e_1 \in \mathbb{R}^{n-1})$.
Setze $H_1 := \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \widetilde{H}_1 \end{array}\right)$; diese ist ebenfalls eine Householder-Matrix.
Dann gilt $H_1 A = \left(\begin{array}{c|c} a_{11} & b \\ \hline \sigma & \\ & \overline{A}_1 \\ 0 & \end{array}\right)$.

$$H_1 A H_1^T = H_1 A H_1 = \left(\begin{array}{c|c} a_{11} & b \\ \hline \sigma & \\ & \overline{A}_1 \\ 0 & \end{array}\right) \cdot \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \widetilde{H}_1 \end{array}\right) = \left(\begin{array}{c|ccc} a_{11} & \multicolumn{3}{c}{b} \\ \hline \sigma & x & \dots & x \\ & \vdots & & \vdots \\ 0 & x & \dots & x \end{array}\right).$$

Korollar:
Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch.
Dann gibt es eine orthogonale Matric Q, so dass $QAQ^T = B$ mit B eine symmetrische Tridiagonalmatrix, also $B = \begin{pmatrix} * & * & & 0 \\ * & \ddots & \ddots & \\ & \ddots & \ddots & * \\ 0 & & * & * \end{pmatrix}$

Beweisidee:

$$H_1 A H_1 = \left(\begin{array}{c|c} a_{11} & c^T \\ \hline \sigma & \\ \vdots & \widetilde{H}_1 A_1 \\ 0 & \end{array}\right)\left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & \\ \vdots & \widetilde{H}_1 \\ 0 & \end{array}\right) = \left(\begin{array}{c|c} a_{11} & \sigma \quad 0 \\ \hline \sigma & \\ \vdots & \overline{A}_1 \\ 0 & \end{array}\right), \text{wegen } c^T \widetilde{H}_1 = c^T \widetilde{H}_1^T =$$

$(\widetilde{H}_1 c)^T = (\sigma e_1)^T = \begin{pmatrix} \sigma & 0 & \dots & 0 \end{pmatrix}$.

e. Das QR-Verfahren

Sei $A \in \mathbb{R}^{n \times n}$.
QR-Zerlegung: $A = QR$.

Verwende diese orthogonale Matrix Q zur Ähnlichkeitstransformation:
$Q^T A Q = Q^T Q R Q = RQ$, d.h. RQ ist ähnlich zu QR. RQ heißt die **QR-Tranformierte** von A.

Satz:
Sei B eine obere Hessenberg-Matrix mit $B = QR$.
Dann ist RQ ebenfalls eine obere Hessenberg-Matrix.

Beweis:
vgl. Übungsaufgabe 23.

Korollar:
Sei B eine symmetrische Tridiagonalmatrix. Dann ist die QR-Transformierte ebenfalls eine symmetrische Tridiagonalmatrix.

Beweis:
$B = QR$, $C = Q^T B Q$, $C^T = (Q^T B Q)^T = Q B^T Q^T = Q^T B Q = C$.

$\Rightarrow C$ ist symmetrisch und C ist eine obere Hessenberg-Matrix (Satz).
$\Rightarrow C$ ist eine symmetrische Tridiagonalmatrix.

QR-Algorithmus
$A \in \mathbb{R}^{n \times n}$; $A^{(1)} := A$. Für $k = 1, 2, 3, \dots$:
$A^{(k)} = Q^{(k)} \cdot R^{(k)}$ (QR-Zerlegung)
$A^{(k+1)} := R^{(k)} Q^{(k)}$ (QR-Transformierte)

In der Praxis:

1. Stufe Transformation auf Hessenberg-Gestalt B;

2. Stufe QR-Algorithmus mit $A^{(1)} = B$ (spart Rechenarbeit).

Konvergenz:

1. Alle Eigenwerte betragsmäßig verschieden:
   $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$
   $$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & * & \cdots & * \\ a_{21}^{(k)} & \ddots & & \vdots \\ & \ddots & \ddots & * \\ 0 & & a_{n,n-1}^{(k)} & a_{nn}^{(k)} \end{pmatrix}.$$
   Konvergenz falls $a_{21}^{(k)}, \ldots, a_{n,n-1}^{(k)} \to 0$.

   Es gilt $a_{j+1,j}^{(k)} \leq c \left|\frac{\lambda_{j+1}}{\lambda_j}\right|^k$ für $j = 1, \ldots, n-1$.

   Wegen $|\lambda_{j+1}| < |\lambda_j|$ gilt $\left|\frac{\lambda_{j+1}}{\lambda_j}\right|^k \to 0$ für $k \to \infty$.
   $\Rightarrow a_{j+1,j}^{(k)} \to 0$ für $k \to \infty$ und $j = 1, \ldots, n-1$.

   Aber Konvergenz kann langsam sein!
2. $\to$ Möglichkeiten zur Konvergenzbeschleunigung!
3. Reduktion.
4. Komplexe Eigenwerte.

f. Das Verfahren von Hyman

Sei $B = \begin{pmatrix} b_{11} & \cdots & \cdots & b_{1n} \\ b_{21} & \ddots & & \vdots \\ & \ddots & \ddots & \vdots \\ 0 & & b_{n,n-1} & b_{nn} \end{pmatrix}$ eine obere Hessenberg-Matrix.

Sei $p(\lambda) := \det(B - \lambda I)$ das charakteristische Polynom.

Ziel: Sei $\mu \in \mathbb{R}$, berechne $p(\mu)$ (ohne $p(\lambda)$ explizit auszurechnen).

Sei zusätzlich $b_{21} \neq 0, \ldots, b_{n,n-1} \neq 0$.

Anmerkung: Eine Matrix mit diesen Eigenschaften heißt unzerlegbar.

Sei $\mu \in \mathbb{R}$. Betrachte das lineare Gleichungssystem
$(B - \mu I)x = \alpha \cdot e_1$ (α unbekannt)

$$
\begin{aligned}
&(b_{11} - \mu)x_1 + b_{12}x_2 + \ldots + b_{1n}x_n = \alpha \\
&b_{21}x_1 + (b_{22} - \mu)x_2 + \ldots + b_{2n}x_n = 0 \\
&\vdots \quad \ddots \quad \vdots \\
&b_{n-1,n-2}x_{n-2} + (b_{n-1,n-1} - \mu)x_{n-1} + b_{n-1,n}x_n = 0 \\
&b_{n,n-1}x_{n-1} + (b_{nn} - \mu)x_n = 0
\end{aligned}
$$

Anmerkung:
Sei x ein Eigenvektor zur oberen Hessenberg-Matrix B. Dann gilt für $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$: $x_n \neq 0$.
(denn aus $(B - \lambda I)x = 0$ folgt dann $x_n = x_{n-1} = \ldots = x_1 = 0$)

Setze also $x_n = 1$.
$\Rightarrow x_{n-1} = \frac{-(b_{nn}-\mu)}{b_{n,n-1}} = x_{n-1}(\mu)$.
$\Rightarrow x_{n-2} = \ldots = x_{n-2}(\mu)$.
$\vdots \; \vdots$
$\Rightarrow x_1 = \ldots = x_1(\mu)$
$\Rightarrow \alpha = \ldots = \alpha(\mu)$

Lemma:
Es gilt $\alpha(\mu) = \frac{(-1)^{n+1}}{b_{21}b_{32}\cdot\ldots\cdot b_{n,n-1}} p(\mu)$.

Beweis:
Cramersche Regel für $(B - \mu I)x = \alpha e_1$ ergibt:

$$
x_n = \frac{\det C}{\det(B-\lambda I)} \text{ mit } C = \left(\begin{array}{ccc|c} b_{11} - \mu & \ldots & b_{1,n-1} & \alpha \\ b_{21} & \ddots & & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & b_{n,n-1} & 0 \end{array}\right).
$$

$\det C = (-1)^{n+1}\alpha \cdot b_{21}b_{32} \cdot \ldots \cdot b_{n,n-1}$ (Entwicklungssatz Laplace)

$\Rightarrow 1 = x_n = \frac{\det C}{p(\mu)} = \frac{(-1)^{n+1} \cdot \alpha \cdot b_{21} \cdot \ldots \cdot b_{n,n-1}}{p(\mu)}.$
$\Rightarrow \alpha(\mu) = \frac{(-1)^{n+1}}{b_{21} \cdot \ldots \cdot b_{n,n-1}} \cdot p(\mu).$

□

Nun Nullstellenverfahren, z.B. Bisektionsverfahren.

$\alpha(\mu)$ ist differenzierbar (nach μ). Berechnung von α':
$(B - \mu I)x = \alpha \cdot e_1$ (α unbekannt)

$$
\begin{aligned}
&(b_{11} - \mu)x_1(\mu) + b_{12}x_2(\mu) + \ldots + b_{1n}x_n(\mu) = \alpha(\mu)\\
&b_{21}x_1(\mu) + (b_{22} - \mu)x_2(\mu) + \ldots + b_{2n}x_n(\mu) = 0\\
&\vdots \quad \ddots \quad \vdots\\
&b_{n-1,n-2}x_{n-2}(\mu) + (b_{n-1,n-1} - \mu)x_{n-1}(\mu) + b_{n-1,n}x_n(\mu) = 0\\
&b_{n,n-1}x_{n-1}(\mu) + (b_{nn} - \mu)x_n(\mu) = 0
\end{aligned}
$$

$\Rightarrow$

$$
\begin{aligned}
&(b_{11} - \mu)x_1'(\mu) - x_1(\mu) + b_{12}x_2'(\mu) + \ldots + b_{1n}x_n'(\mu) = \alpha(\mu)\\
&b_{21}x_1'(\mu) + (b_{22} - \mu)x_2'(\mu) - x_2(\mu) + \ldots + b_{2n}x_n'(\mu) = 0\\
&\vdots \quad \ddots \quad \vdots\\
&b_{n,n-1}x_{n-1}'(\mu) - \underbrace{x_n(\mu)}_{=1} = 0
\end{aligned}
$$

Unbekannt: $x_i'(\mu)$ und $\alpha'(\mu)$. Durch Auflösen erhält man diese.

$\Rightarrow$ Newton-Verfahren ist auch anwendbar.

g. Eigenwerte und Eigenvektoren in Matlab

d = eig(A): d ein Vektor, der die Eigenwerte enthält;
[V,D] = eig(A): D eine Diagonalmatrix, die die Eigenwerte enthält, die Spalten von V enthalten die Eigenvektoren;
[P,H] = hess(A): Hessenberg-Gestalt;
[Q,R] = qr(A): QR-Zerlegung.

h. Vektornormen und Matrixnormen

Vektornormen auf $\mathbb{K}^n$:
$\|\cdot\| : \mathbb{K}^n \to \mathbb{R}$ heißt Vektornorm auf $\mathbb{K}^n$, falls gilt

(1) $\|x\| > 0$ für alle $x \in \mathbb{K}^n \setminus \{0\}$;

(2) $\|\lambda x\| = |\lambda| \|x\|$ für alle $x \in \mathbb{K}^n$ und alle $\lambda \in \mathbb{K}$;

(3) $\|x + y\| \leq \|x\| + \|y\|$ für alle $x, y \in \mathbb{K}^n$.

Beispiele:

- $\|x\|_\infty := \max\{|x_i|_i = 1, \ldots, n\}$;
- $\|x\|_2 := \sqrt{\sum_{i=1}^{n} |x_i|^2}$;
- $\|x\|_1 := \sum_{i=1}^{n} |x_i|$.

Matrixnormen:
$N(\cdot) : \mathbb{K}^{n\times n} \to \mathbb{R}$ heißt eine Matrixnorm, falls gilt:

(1) $N(A) > 0$ für alle $x \in \mathbb{K}^{n\times n} \setminus \{0\}$;

(2) $N(\lambda x) = |\lambda| N(A)$ für alle $x \in \mathbb{K}^{n\times n}$ und alle $\lambda \in \mathbb{K}$;

(3) $N(A + B) \leq N(A) + N(B)$ für alle $A, B \in \mathbb{K}^{n\times n}$.

Eine Matrix heißt **submultiplikativ**, falls gilt:

(1) $N(A \cdot B) \leq N(A) \cdot N(B)$ für alle $A, B \in \mathbb{K}^{n\times n}$.

Die Matrixnorm $N(\cdot)$ und die Vektornorm $\|\cdot\|$ heißen **verträglich**, falls gilt $\|Ax\| \leq N(A) \cdot \|x\|$ für alle $A \in \mathbb{K}^{n\times n}$ und alle $x \in \mathbb{K}^n$.

Beispiele:

$$A = \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nn} \end{pmatrix}$$

- $N_E(A) := \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}|^2}$: **Euklidische Norm** oder **Schur-Norm**;
- $N_Z(A) := \max\{\sum_{k=1}^{n}|a_{ik}|| : i = 1,\ldots,n\}$: Zeilensummennorm;
- $N_S(A) := \max\{\sum_{i=1}^{n}|a_{ik}|| : k = 1,\ldots,n\}$: Spaltensummennorm.

Verträglich sind: N_E mit $\|\cdot\|_2$, N_Z mit $\|\cdot\|_\infty$, N_S mit $\|\cdot\|_1$.

Anmerkung:
Zu jeder Vektornorm $\|\cdot\|$ wird durch

$$\text{lub}(A) := \max\{\|Ax\| : x \in \mathbb{K}^n \text{ mit } \|x\| = 1\}$$

eine *submultiplikative*, *verträgliche* Matrixnorm definiert.

Anmerkung:

(1) *lub* bedeutet **l**east **u**pper **b**ound.

(2) *lub*(A) ist die kleinste Konstante c, welche $\|Ax\| \leq c \cdot \|x\|$ erfüllt.

(3) *lub*(A) heißt auch die von $\|\cdot\|$ induzierte Matrixnorm.

Beispiele:

- $\|x\|_\infty \Rightarrow \text{lub}_\infty(A) = N_Z(A)$;
- $\|x\|_1 \Rightarrow \text{lub}_1(A) = N_S(A)$;
- $\|x\|_2 \Rightarrow \text{lub}_2(A) = \sqrt{\rho(A \cdot A^T)} \neq N_E(A)$.

Definition:
Sei $A \in \mathbb{K}^{n\times n}$. Dann heißt $\sigma(A) := \{\lambda \in \mathbb{C} : \lambda \text{ ist ein Eigenwert von } A\}$ das **Spektrum** von A.

$\rho(A) := \max\{|\lambda| : \lambda \in \sigma(A)\}$ heißt **Spektralradius** von A.

Lemma:
Es gilt $\text{lub}(A) \geq \rho(A)$.

Beweis:
Sei $\lambda \in \sigma(A)$ und v ein zugehöriger Eigenvektor mit $\|v\| = 1$. Dann gilt $\|Av\| = \|\lambda v\| = |\lambda|\|v\| = |\lambda| \leq \rho(A)$ (weil $\rho = \max\{|\lambda|\}$).
$\Rightarrow \text{lub}(A) \geq \rho(A)$.

□

Lemma:
Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch. Dann gilt $\text{lub}_2(A) = \rho(A)$.

Beweis:
Es gibt eine orthogonale Matrix Q mit $Q^TAQ = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} =: D$

Sei $x \in \mathbb{R}^n$ und $\|x\|_2 = 1 = \|x\|_2^2$. Setze $y = Qx$ ($\Rightarrow x = Qy$).
$\|y\|_2^2 = \|Q^Ty\|_2^2 = (x^TQ)Q^Tx = x^Tx = \|x\|_2^2$.

$\|Ax\|_2^2 = \|AQy\|_2^2 = (y^TQ^TA^T)AQy = y^TQ^TA^TQQ^TAQy = y^TD^TDy = \|Dy\|_2^2 = \sum_{i=1}^{n}(\lambda_i y_i)^2 = \sum_{i=1}^{n}\lambda_i^2 y_i^2 \leq \rho(A)\cdot\sum_{i=1}^{n} y_i^2 = \rho(A)^2\cdot\|y\|_2^2 = \rho(A)^2$.

$\Rightarrow \|Ax\|_2 \leq \rho(A) \Rightarrow \text{lub}_2(A) \leq \rho(A)$.
Mit vorangegangenem Lemma gilt also $\text{lub}_2(A) = \rho(A)$.

□

Korollar:
$A = \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{pmatrix} \Rightarrow \text{lub}_2(A) = \max\{|a_{ii}| : i = 1,\ldots,n\}$.

Sei A invertierbar, so heißt die Größe

$$\text{cond}(A) := \text{lub}(A)\cdot\text{lub}(A^{-1})$$

die **Konditionszahl** einer Matrix *(es gilt immer cond*$(A) \geq 1$*)*.

i. Störempfindlichkeit bei Eigenwertaufgaben

Lemma:
Seien $A, B \in \mathbb{R}^{n\times n}$. Sei $\lambda \in C$ ein Eigenwert von A, aber *kein* Eigenwert von B. Dann gilt:

$$1 \leq \text{lub}((\lambda I - B)^{-1}(A - B)) \leq \text{lub}((\lambda I - B)^{-1})\text{lub}(A - B)$$

Beweis:
Sei v ein Eigenvektor zu λ mit $\|v\| = 1$.

$(A - B)v = (\lambda I - B)v \Rightarrow (\lambda I - B)^{-1}(A - B)v = v.$
$\|(\lambda I - B)^{-1}(A - B)v\| = \|v\| = 1$

$\Rightarrow \text{lub}((\lambda I - B)^{-1}(A - B)) \geq 1.$

□

Satz:
Sei $B \in \mathbb{R}^{n\times n}$ diagonalisierbar mit Eigenwerten $\lambda_1(B), \ldots, \lambda_n(B)$ *(d.h. $B = PDP^{-1}$ mit $D = diag(\lambda_1(B), \ldots, \lambda_n(B))$)*.
Sei $A \in \mathbb{R}^{n\times n}$ eine beliebige Matrix, so gibt es zu jedem Eigenwert λ von A einen Eigenwert $\lambda_j(B)$

$$|\lambda - \lambda_j(B)| \leq \text{cond}_2(P) \cdot \text{lub}_2(A - B)$$

Beweis:
Sei λ kein Eigenwert von B. Dann gilt $\text{lub}_2((\lambda I - B)^{-1}) = \text{lub}_2((\lambda PP^{-1} - PDP^{-1})^{-1}) = \text{lub}_2((P(\lambda I - D)P^{-1})^{-1}) = \text{lub}_2(P(\lambda I - D)^{-1}P^{-1}) \leq \text{lub}_2(P)\cdot\text{lub}_2((\lambda I - D)^{-1})\cdot\text{lub}_2(P^{-1}) = \text{cond}_2(P)\cdot\text{lub}_2((\lambda I - D)^{-1}) = \text{cond}_2(P)\cdot\max\left\{\left|\frac{1}{\lambda - \lambda_i(B)}\right| : i = 1, \ldots, n\right\} = \text{cond}_2(P)\cdot\frac{1}{|\lambda - \lambda_j(B)|}$
(für ein bestimmtes $j \in \{1, \ldots, n\}$).

Nach vorangegangenem Lemma gilt $1 \leq \text{cond}_2(P)\frac{1}{|\lambda - \lambda_j(B)|}\text{lub}_2(A - B)$.

□

Also:
$\text{lub}_2(P)$ klein $\Rightarrow$ kleine Störung in B bedeutet, dass die Eigenwerte nahe beieinander liegen.
$\text{lub}_2(P)$ groß $\Rightarrow$ kleine Störung in B kann total verschiedene Eigenwerte zur Folge haben.

Lemma:
Ist Q eine orthogonale Matrix, so gilt

(1) $\text{lub}_2(Q) = 1$;

(2) $\text{cond}_2(Q) = 1$.

Beweis:
$\text{lub}_2(Q) = \max\{||Qx||_2 : x \in \mathbb{K}^n, ||x||_2 = 1\}$.

Sei $||x||_2 = 1$.
$\Rightarrow ||Qx||_2^2 = (Qx)^T(Qx) = x^TQ^TQx = x^Tx = ||x||_2^2 = 1 \Rightarrow ||Qx||_2 = 1$.
$\Rightarrow \text{lub}_2(Q) = 1$.

$\text{cond}_2(Q) = \text{lub}_2(Q) \cdot \text{lub}_2(Q^{-1}) = \text{lub}_2(Q) \cdot \text{lub}_2(Q^T)$.
Q^T ist ebenfalls orthogonal $\Rightarrow \text{lub}_2(Q^T) = 1$.
$\Rightarrow \text{cond}_2(Q) = 1$.

□

Korollar:
Ist $B \in \mathbb{R}^{n\times n}$ symmetrisch und $A \in \mathbb{R}^{n\times n}$ beliebig, so gibt es zu jedem Eigenwert λ von A einen Eigenwert $\lambda_j(B)$ von B mit $|\lambda - \lambda_j(B)| \leq \text{lub}_2(A - B)$.

Beweis:
Es gibt eine orthogonale Ähnlichkeitstransformation.
Rest folgt mit obigem Lemma aus obigem Satz.

□

Bei symmetrischen Matrizen ist die Eigenwert-Aufgabe also störunempfindlich.

Störempfindlichkeit bei Ähnlichkeitstransformationen

$B = P^{-1}AP$ und Störung $A + \Delta A$.

Ähnlichkeitstransformation:
$$P^{-1}(A + \Delta A)P = \underbrace{P^{-1}AP}_{=B} + \underbrace{P^{-1}\Delta AP}_{=\Delta B}$$

$\Rightarrow \text{lub}_2(\Delta B) = \text{lub}_2(P^{-1}\Delta AP) \leq \text{lub}_2(P^{-1})\text{lub}_2(P)\text{lub}_2(\Delta A) = \text{cond}_2(P) = \text{lub}_2(\Delta A)$

Beim QR-Verfahren orthogonale Ähnlichkeitstransformation (Q ist orthogonal):
$\Rightarrow \text{lub}_2(\Delta B) \leq \text{lub}_2(\Delta A)$

Fazit: QR-Verfahren ist störunempfindlich.

10. Iterative Verfahren

a. Iteration

Sei $D \subset \mathbb{R}^n$ und $\Phi : D \to D$. Ein $\xi \in D$ heißt Fixpunkt von Φ, falls $\Phi(\xi) = \xi$ gilt.

Fixpunkte werden iterativ berechnet. Startwert $x^{(0)} \in D$ berechne dann $x^{(i+1)} = \Phi(x^{(i)})$.

Frage: $\lim_{i\to\infty} x^{(i)} = \xi$??

Φ stetig: wenn Konvergenz $\Rightarrow \lim_{i\to\infty} x^{(i)}$ ist Fixpunkt.

Newton-Verfahren: $x^{(i+1)} = x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})}$: $\Phi(x) = x - \frac{f(x)}{f'(x)}$.

Definition:
Sei (X, d) ein metrischer Raum. Eine Abbildung $\Phi : X \to X$ heißt kontrahierend, falls es ein $0 \leq L < 1$ gibt mit

$$d(\Phi(s), \Phi(t)) \leq L \cdot d(s, t) \text{ für alle } s, t \in X.$$

Kontraktionssatz (Banachscher Fixpunktsatz):
Sei (X, d) vollständig und $\Phi : X \to X$ kontrahierend. Dann gilt

(1) Es gibt genau einen Fixpunkt $\xi \in X$.

(2) Für jeden beliebigen Startwert $x^{(0)} \in X$ konvergiert die Folge $x^{(i+1)} = \Phi(x^{(i)})$ gegen ξ.

Newton-Verfahren: $X := [a, b], d(s, t) = |s - t|$.

Anmerkung:
Sei $\Phi : [a, b] \to [a, b]$ stetig differenzierbar.
$L := \|\Phi'\|_{[a,b]} = \max\{|\Phi'(s)| : s \in [a, b]\}$.

$\Rightarrow |\Phi(s) - \Phi(t)| \leq L \cdot |s - t|$.
(*Denn aus Mittelwertsatz:* $\frac{|\Phi(s)-\Phi(t)|}{|s-t|} = |\Phi'(\xi)| \leq \|\Phi'\|_{[a,b]}$)

b. Indirekte Verfahren für lineare Gleichungssysteme

$Ax = b \rightarrow$ Fixpunktformen: $x = \Phi(x) = Tx + g$.

Lemma:
Sei $A \in \mathbb{R}^{n\times n}$ regulär. Dann kann man durch Zeilenumformungen erreichen, dass alle Diagonalelemente von Null verschieden sind.

Beweis:
Induktion nach n:

- Sei A regulär. $n = 1$: $A = (a_{11})$ mit $a_{11} \neq 0 \Rightarrow$ Aussage klar.
- $n - 1 \Rightarrow n$: $A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}$

  $\det A = \sum_{i=1}^{n} (-1)^{i+1} a_{i1} \cdot \det A_{i1}$ (Entwicklung nach Laplace, 1. Spalte)

  Es gibt mindestens ein $k \in \{1, \dots, n\}$ mit $a_{k1} \cdot \det A_{k1} \neq 0 \Rightarrow a_{k1} \neq 0$ und $\det A_{k1} \neq 0$.

  Vertausche erste und k-te Zeile $\Rightarrow \widetilde{A} = \left(\begin{array}{c|ccc} a_{k1} & * & \dots & * \\ \hline * & & & \\ \vdots & & \widetilde{A}_{k1} & \\ * & & & \end{array}\right)$

  $\widetilde{A}_{k1}$ entsteht aus A_{k1} durch Zeilenvertauschung. $\Rightarrow \det \widetilde{A}_{k1} \neq 0$.

$\Rightarrow$ Behauptung □

Sei nun $A \in \mathbb{R}^{n\times n}$, A regulär mit $a_{ii} \neq 0$ für $i = 1, \dots, n$.

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{pmatrix} + \begin{pmatrix} 0 & \dots & 0 \\ & \ddots & \vdots \\ a_{n1} & & 0 \end{pmatrix} + \begin{pmatrix} 0 & & a_{1n} \\ \vdots & \ddots & \\ 0 & \dots & 0 \end{pmatrix}$$

$A = L + D + U$

Daraus ergeben sich zwei Verfahren:

1. $Ax = b \Rightarrow (L + D + U)x = b \Rightarrow Dx + (L + U)x = b \Leftrightarrow Dx = -(L + U)x + b \Leftrightarrow x = -D^{-1}(L + U)x + D^{-1}b$.

   Also $x = Tx + g$ mit $T = -D^{-1}(L + U)$ und $g = D^{-1}b$.

   Nun: $x^{(0)} \in \mathbb{R}^n$: $x^{(i+1)} = Tx^{(i)} + g$.

   Dieses Verfahren heißt **Gesamtschrittverfahren** oder **Gauß-Jacobi-Verfahren**.

2. $Ax = b \Leftrightarrow (L + U + D)x = b \Leftrightarrow (L + D)x + Ux = b \Leftrightarrow (L + D)x = -Ux + b \Rightarrow x = -(L + D)^{-1}Ux + (L + D)^{-1}b$.

   Also $x = Tx + g$ mit $T = -(L + D)^{-1}U$ und $g = (L + D)^{-1}b$.

   Nun: $x^{(0)} \in \mathbb{R}^n$: $x^{(i+1)} = Tx^{(i)} + g$.

   Dieses Verfahren heißt **Einzelschrittverfahren** oder **Gauß-Seidel-Verfahren**.

Praktische Durchführung des Einzelschrittverfahrens:
$(L + D)x^{(i+1)} = -Ux^{(i)} + b$. Löse dieses lineare Gleichungssystem (durch Vorwärtsauflösen).

Gesamtschrittverfahren:
$$\begin{pmatrix} x_1(i) \\ \vdots \\ x_n^{(i)} \end{pmatrix} \to \begin{pmatrix} x_1^{(i+1)} \\ \vdots \\ x_n^{(i+1)} \end{pmatrix}$$

Einzelschrittverfahren:
$$\begin{pmatrix} x_1^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix} \to \begin{pmatrix} x_1^{(i+1)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix} \to \begin{pmatrix} x_1^{(i+1)} \\ x_2^{(i+1)} \\ x_3^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix} \dots$$

Zusammenfassung:
$Ax = b \Leftrightarrow x = \Phi(x) = Tx + g$
$A = L + D + U$
$T_G = -D^{-1}(L + U)$ Gesamtschritt
$T_E = -(L + D)^{-1}U$ Einzelschritt

Aufwand:
pro Iterationsschritt: n^2 Punktoperationen $x^{(i+1)} = Tx^{(i)} + g$
gesamt: $k \cdot n^2$ (k Schritte)

direkte Verfahren: $O(n^3)$ (indirekte Verfahren also nur bei großen n sinnvoll)

Konvergenz:
$\Phi(x) = Tx + g$ - wann ist hier Φ kontrahierend?

$\|\Phi(x) - \Phi(y)\| \leq L\|x - y\|$ mit $L < 1$.
$\|Tx + g - (Ty + g)\| = \|T(x - y)\| \leq N(T) \cdot \|x - y\| \leftarrow N(T) < 1$.
($\|\cdot\|$ und $N(\cdot)$ seien verträglich).

Gilt $N(T) < 1$, so erhält man Konvergenz der Iteration $x^{(i+1)} = Tx^{(i)} + g$ für jeden Startwert $x^{(0)} \in \mathbb{K}^n$.

Genau-Dann-Bedingung:
Sei $\rho(T)$ der Spektralradius.

Satz:
$x^{(i+1)} = Tx^{(i)} + g$ ist konvergent (gegen den Fixpunkt) für jeden Startwert $\Leftrightarrow \rho(T) < 1$.

Beweis:

„$\Rightarrow$“ Sei $\overline{x}$ ein Fixpunkt und $x^{(0)} \in \mathbb{K}^n$ ein Startwert.
$x^{(i)} - \overline{x} = Tx^{(i-1)} - T\overline{x} = T(x^{(i-1)} - \overline{x}) = T^2(x^{(i-2)} - \overline{x}) = \ldots = T^i(x^{(0)} - \overline{x})$.

$\Rightarrow$ Sei λ ein Eigenwert von T und y ein zugehöriger Eigenvektor. Wähle $x^{(0)} := y + \overline{x}$
$\rightarrow x^{(i)} = Tx^{(i-1)} + g$
Nach Voraussetzung gilt $\lim\limits_{i\to\infty}(x^{(i)} - \overline{x}) = 0$.

$\Rightarrow \lim\limits_{i\to\infty}(T^i(x^{(0)} - \overline{x})) = \lim\limits_{i\to\infty}(T^i(x^{(0)} - \overline{x})) = \lim\limits_{i\to\infty}(T^i y) = \lim\limits_{i\to\infty}(\lambda^i y) = 0$
$\Rightarrow |\lambda| < 1 \Rightarrow \rho(T) < 1$.

„$\Leftarrow$“ Sei $\rho(T) < 1$. Dann gibt es eine Vektornorm $\|\cdot\|$, so dass für die zugehörige lub-Norm gilt $\rho(T) \leq \text{lub}(T) < 1$. $\Rightarrow$ Behauptung.

□

Hinreichende Bedingung:

1. für Gesamtschrittverfahren: $T = T_G = -D^{-1}(L + U)$.
$$\begin{pmatrix} \frac{1}{a_{11}} & & \\ & \cdots & \\ & & \frac{1}{a_{nn}} \end{pmatrix} \begin{pmatrix} 0 & & a_{1n} \\ & \ddots & \\ a_{n1} & & 0 \end{pmatrix}$$

   Zeilensummennorm $N_Z(T_G) < 1$

   $N_Z(T_G) = \max\{\frac{1}{|a_{ii}|} \cdot \sum_{k=1,k\neq i}^{n} |a_{ik}| : i = 1, \ldots, n\} < 1.$

   $\sum_{k=1,k\neq i}^{n} |a_{ik}| < |a_{ii}|$ für jedes $i = 1, \ldots, n$ (Zeilensummenkriterium)

   Entsprechend liefert die Spaltensummennorm $N_S(T_G)$ die Bedingung
   $\sum_{k=1,k\neq j} |a_{kj}|a_{kj}| \leq |a_{jj}|$ (Spaltensummenkriterium)

2. für Einzelschrittverfahren:

   Satz:
   Zeilensummen- und Spaltensummenkriterium sind auch hinreichend für die Konvergenz des Einzelschritt-Verfahrens.

   Beweis:
   für Zeilensummenkriterium:
   Zeige $N_Z(T_E) \leq N_Z(T_G$ (< 1 nach Voraussetzung)
   $N_Z(\cdot) = \text{lub}_\infty(\cdot)$
   Sei $y \in \mathbb{K}^n$ und $z := T_E y$

   Wir zeigen durch vollständige Induktion $|z_k| \leq \frac{1}{\|a_{kk}\|} \| \sum_{j=1, j\neq k}^{n} |a_{kj}| \, \|y\|_\infty \leq N_Z(T_G)\|y\|_\infty$.

   $(L + D)z = -Uy$ (Vorwärtsauflösen)
   $z_k = -\frac{1}{a_{kk}} \cdot \left(\sum_{j=1}^{k-1} a_{kj} z_j + \sum_{j=k+1}^{n} a_{kj} y_j \right)$

   Induktionsanfang: $|z_1| < \frac{1}{|a_{11}|} \cdot \sum_{j=2}^{n} |a_{kj}| \underbrace{|y_j|}_{\|y\|_\infty} \leq \left(\frac{1}{|a_{11}|} \cdot \sum_{j=2}^{n} |a_{1j}| \right) \|y\|_\infty \leq N_Z(T_G)\|y\|_\infty$

Induktionsschluss: $|z_k| \leq \frac{1}{|a_{kk}|}\left(\sum_{j=1}^{k-1}|a_{kj}|\,|z_j| + \sum_{j=k+1}^{n}|a_{kj}|\,|y_j|\right)$

$$\leq \frac{1}{|a_{kk}|}\left(\left(\sum_{j=1}^{k-1}|a_{kj}|\right)\cdot \underbrace{N_Z(T_G)}_{<1,\,n.V.}\|y\|_\infty + \left(\sum_{j=k+1}^{n}|a_{kj}|\right)\|y\|_\infty\right)$$

$$\leq \frac{1}{|a_{kk}|}\left(\sum_{j=1,j\neq k}^{n}|a_{kj}|\right)\cdot\|y\|_\infty \leq N_Z(T_G)\|y\|_\infty$$

$\Rightarrow \|T_E y\|_\infty = \|z\|_\infty \leq N_Z(T_G)\|y\|_\infty$
$\Rightarrow \mathrm{lub}_\infty(T_E) = N_Z(T_E) \leq N_Z(T_G)$.

Analog für Spaltensummenkriterium.

□

Konvergenzgeschwindigkeit
hängt ab vom Spektralradius.
Methoden zur Konvergenzbeschleunigung

- Relaxationsverfahren „Verschieben der Eigenwerte“- hängt davon ab, wie die Eigenwerte verteilt sind

c. Nichtlineare Gleichungssysteme, mehrdimensionales Newtonverfahren

Sei $U \subset \mathbb{R}^n$ und $F : U \to \mathbb{R}^n$ hinreichend glatt.
Gesucht ist ein $x \in U$ mit $F(x) = 0$, d.h.
$f_1(x_1, \ldots, x_n) = 0$
$\vdots$
$f_n(x_1, \ldots, x_n)$

Sei $DF(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \ldots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \ldots & \frac{\partial f_n}{\partial x_n}(x) \end{pmatrix}$ die Funktionalmatrix.

Taylor-Entwicklung:
$F(x) = F(y) + DF(y)\cdot(x-y) + Rest$

$\Rightarrow 0 = F(y) + DF(y)(x - y) \Rightarrow x = y - DF(y)^{-1} \cdot F(y)$, falls $\det(DF(y)) \neq 0$.

Wähle nun ein $x^{(0)} \in U$ und bilde

$$\boxed{x^{(i)} = x^{(i-1)} - [DF(x^{(i-1)})]^{-1} \cdot F(x^{(i-1)})}$$

(mehrdimensionales Newton-Verfahren).

Praktische Durchführung:
$DF(x^{(i-1)}) \cdot (x^{(i)} - x^{(i-1)}) = -F(x^{(i-1)})$ (gesucht: $x^{(i)}$; sei $v := x^{(i)} - x^{(i-1)}$)

Löse also das lineare Gleichungssystem $DF(x^{(i-1)}) \cdot v = F(x^{(i-1)})$ und setze dann $x^{(i)} = v + x^{(i-1)}$

Abbruchbedingungen:
$\|F(x^{(i)})\|_\infty < \varepsilon$ und $\|x^{(i)} - x^{(i)}\|_\infty < \varepsilon$.

Matlab:
$Ax = b$ wird mit dem Befehl „x = b \ A;“ gelöst.

In jedem Schritt muss $A_k = DF(y^{(k)})$ berechnet werden; das sind n^2 Funktionsauswertungen. Deshalb gibt es einfachere Formen:

Vereinfachte Form
Wähle für A_k eine feste Matrix, z.B. $A_k := A_0 = DF(y^{(0)})$ mit $y^{(0)}$ ist Startwert.

Löse dann $A_0 \cdot v = F(y^{(k)})$ z.B. mit LR-Zerlegung (die Zerlegung muss nur einmal gemacht werden).

Diese Methode heißt **Parallel-Verfahren**.

Veranschaulichung im Fall $n = 1$:
$y^{(k+1)} = y^{(k)} - \frac{f(y^{(k)})}{f'(y^{(0)})}$

Konvergenz des Newton-Verfahrens:

Satz: *(lokale Konvergenz)*
Sei $U \subset \mathbb{R}^n$ offen und $F : U \to \mathbb{R}^n$ zweimal stetig differenzierbar. Sei $\overline{u} \in U$ mit $F(\overline{u}) = 0$ und $DF(\overline{u})$ invertierbar.
Dann gilt:

(1) Es gibt eine Kugel $K_\rho(\overline{u}) := \{x \in U : \|x - \overline{u}\|_2 \leq \rho\}$ mit $F(x) \neq 0$ für alle $x \in K_\rho(\overline{u}) \setminus \{\overline{u}\}$ *(d.h. $\overline{u}$ ist eine isolierte Nulltstelle)*.

(2) Für jeden Startwert $y^{(0)} \in K_\rho(\overline{u})$ konvergiert das Newton-Verfahren gegen $\overline{u}$.

(3) es gilt quadratische Konvergenz.

In der Praxis nicht nützlich, weil die Nullstelle und ρ nicht bekannt sind.

11. Minimierung

a. Einleitung

Gesucht ist ein (globales oder lokales) Minimum von

$$h : \mathbb{R}^n \supset D \to \mathbb{R}.$$

Gesucht ist also ein $\overline{z} \in \mathbb{R}^n$ mit

- $h(\overline{z}) \leq h(x)$ für alle $x \in D$ (globales Minimum)
- $h(\overline{z}) \leq h(x)$ für alle $x \in U, U \subset D$ (lokales Minimum)

Wir beschränken uns auf die Suche der lokalen Minima.

b. Nichlinearer Ausgleich

$h(z) = h(z_1, \ldots, z_n) = \sum_{j=1}^{m} (s_j - G(t_j, z_1, \ldots, z_n))^2$ (vergleiche Abschnitt 5.a.)
(t_j, s_j) für $j = 1, \ldots, m$, waren die Messwerte.

Beispiele:

(1) Physik: Messungen (t_i, s_i) mit $i = 0, \ldots, 20$.
$G(t, \alpha_1, \alpha_2, \alpha_3, \alpha_4) = \alpha_1 \cdot \sin(\alpha_2 t + \alpha_3) + \alpha_4$.

(2) Biologie: Enzymaktionen $X \longrightarrow_{Enzym} P$ (X: Substrat, P: Produkt)
Reaktionsgeschwindigkeit v hängt von der Konzentration des Substrats x ab.

$v(x) = \frac{\mu \cdot x}{K+x}$

Messungen (x_i, v_i) mit $i = 1, \ldots, m$.
$G(x_i, \mu, K) := \frac{\mu \cdot x}{K+x}$.

Sei $h(x) = h(x_1, \ldots, x_n)$ hinreichend glatt. Notwendige Bedingung für ein lokales Minimum $\overline{z}$:

$$\left.\begin{array}{c} \frac{\partial}{\partial x_1} h(\overline{z}_1, \ldots, \overline{z}_n) = 0 \\ \vdots \\ \frac{\partial}{\partial x_n} h(\overline{z}_1, \ldots, \overline{z}_n) = 0 \end{array}\right\} \Rightarrow \operatorname{grad} h(\overline{z}) = 0.$$

Hinreichende Bedingung:
Hess $h(\overline{z})$ positiv definit.

c. Höhenlinien

$n = 2$; $h(z_1, z_2)$; $h : \mathbb{R}^2 \to \mathbb{R}$.
Zu $c \in \mathbb{R}$ heißt $H_c := \{(z_1, z_2) \in \mathbb{R}^2 : h(z_1, z_2) = c\}$ **Höhenlinie** zum Niveau c.

Zeichnen von Höhenlinien durch Lösen von Anfangswertaufgaben:
$\dot{z}_1 = h_{z_2}(z_1, z_2), z_1(0) = \alpha$
$\dot{z}_2 = -h_{z_1}(z_1, z_2), z_2(0) = \beta$

Für die Lösung $(z_1(t), z_2(t))$ dieser Anfangswertaufgabe gilt

$$(z_1(t), z_2(t)) \in H_c \text{ mit } c = h(\alpha, \beta)$$

(vgl. Übungsaufgabe 27).

d. Abstiegsverfahren

$h : D \subset \mathbb{R}^n \to \mathbb{R}$: gesucht ist ein lokales Minimum.
h hinreichend glatt, differenzierbar, ...
Notwendig: $\operatorname{grad}(h(\overline{z})) = 0$.

Sei $z \in D$ gegeben. Gibt es ein $z^* \in D$ mit $h(z^*) - h(z)$? Gibt es einen Vektor $p \in \mathbb{R}^n$ mit $h(z + tp) < h(z)$ für alle $t \in (0, t_0]$?

Dann heißt p eine **Abstiegsrichtung**.

$$z + tp = \begin{pmatrix} z_1 + tp_1 \\ \vdots \\ z_n + tp_n \end{pmatrix}.$$

Sei $I \supset (0, t_0]$. Setze $\varphi : I \to \mathbb{R} : \varphi(t) := h(z + tp)$ (bei festen z, p).

$$\psi : I \to \mathbb{R}^n \colon \psi(t) = \begin{pmatrix} z_1 + tp_1 \\ \vdots \\ z_n + tp_n \end{pmatrix} \Rightarrow \psi'(t) = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix}.$$

$h : \mathbb{R}^n \to \mathbb{R}$.
$V(t) = h(\psi(t)) \Rightarrow \varphi$ ist differenzierbar (als Verknüpfung von Funktionen) und es gilt $\varphi'(t) = \operatorname{grad} h(\psi(t)) \cdot \psi'(t)$

$$= \begin{pmatrix} h_{z_1}(\psi(t)) & \dots & h_{z_n}(\psi(t)) \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} = \sum_{i=1}^{n} h_{z_i}(\psi(t)) \cdot p_i = \sum_{i=1}^{n} h_{z_i}(z + tp) \cdot p_i.$$

$\varphi'(0) = \sum_{i=1}^{n} h_{z_i}(z) \cdot p_i$. $\varphi'(0) < 0 \Rightarrow p$ ist eine Abstiegsrichtung.
Gilt $\operatorname{grad} h(z) \neq 0$, so ist $p := -\operatorname{grad} h(z)$ eine Abstiegsrichtung, denn dann gilt $\varphi'(0) = -\sum_{i=1}^{n} h_{z_i}(z)^2 < 0$.

Basisalgorithmus:
Gesucht ist ein $\overline{z} \in D \subset \mathbb{R}^n$ mit $\operatorname{grad} h(\overline{z}) = 0$.

1) Sei $z^{(0)} \in D$ gegeben; $k = 0$

2) Berechne $\operatorname{grad} h(z^{(k)})$; falls $\operatorname{grad} h(z^{(k)}) = 0$, beende diese Schleife. Ansonsten wähle eine Abstiegsrichtung $p^{(k)}$ und eine Schrittweite $t_k > 0$ mit $h(z^{(k)} + t_k \cdot p^{(k)}) < h(z^{(k)})$.

3) Setze dann $z^{(k+1)} := z^{(k)} + t_k p^{(k)}$; $k = k + 1$ und gehe zu 2).

Ein Abstiegsverfahren besteht aus einer Richtungs- und aus einer Schrittweitenstrategie.

Richtungsstrategie:
z.B. $p^{(k)} = -\operatorname{grad} h(z^{(k)})$ (negativer Gradient)
→ **Gradientenverfahren**

Schrittweitenstrategie:
Betrachte $\varphi(t) := h(z^{(k)} + tp^{(k)})$ ($\varphi : (0, t_0) \to \mathbb{R}$, t_0).
t_k als erstes Minimum von $\varphi(t)$.
→ **exakte Schrittweite**.

Konvergenz (für das Gradientenverfahren):

Satz:
Sei $h \in \mathbb{C}^2(D;\mathbb{R})$, D offen und $z^{(0)} \in D$ der Startwert.
Die Niveaumenge $L_0 := \{x \in D : h(x) \leq h(z^{(0)})\}$ sei kompakt und besitze genau ein $\overline{z} \in L_0$ mit $\operatorname{grad} h(\overline{z}) = 0$.
Dann konvergiert das Gradientenverfahren (mit Startwert $z^{(0)}$) gegen $\overline{z}$.
(vgl. J.Werner - Numerische Mathematik, Band 2).

e. Differentialgleichungsmethode

Idee: Nullstellen des Gradienten als stationäre Punkte einer Differentialgleichung.

$\dot{x} = F(x)$: sei $\overline{y}$ eine Nullstelle von F, dann ist $x(t) = \overline{y}$ eine Lösung der Differentialgleichung ($\rightarrow$ stationäre Punkte).

$$\dot{z} = -\operatorname{grad} h(z)^T,\ x(0) = z^{(0)}, \text{ also } \begin{array}{rcl} \dot{z}_1 &=& -h_{z_1}(z) \\ \vdots & & \vdots \\ \dot{z}_n &=& -h_{z_n}(z) \end{array} \qquad (*)$$

Sei $\Psi(t)$ die Lösung dieser Anfangswertaufgabe.
$\varphi(t) = h(\Psi(t))$ ist differenzierbar und es gilt

$$\varphi'(t) = \operatorname{grad}(h(\Psi(t)) \cdot \varphi'(t)) = -\operatorname{grad} h(\Psi(t)) \cdot \operatorname{grad} h(\Psi(t)) = \sum_{i=1}^{n} h_{z_i}(\Psi(t))^2 \leq 0.$$

$\Rightarrow \varphi(t)$ ist monoton fallend.

Numerisches Verfahren:

(1) Wähle Startwert $z^{(0)} \in D$; feste Schrittweite σ; $k := 1$.

(2) Berechne mit einem Näherungsverfahren (für Differentialgleichungen; am Besten implizite Verfahren) Näherungswerte der Anfangswertaufgabe (*) zum Zeitpunkt $k \cdot \sigma$.

$$z^{(k)} \approx \Psi(k \cdot \sigma)$$

(3) Gilt $\|\operatorname{grad} h(z^{(k)})\| < \varepsilon$ (für $\varepsilon > 0$ gegeben), so beende das Verfahren und akzeptiere $z^{(k)}$ als Näherungswert für $\overline{z}$.
Ansonsten setze $k = k + 1$ und gehe zu (2).

12. Stabilität und Störungsfragen

a. Einleitung

Im ganzen Kapitel bedeutet „klein“ nicht klein, sondern „betragsmäßig klein“.

Stabilität
Sei $F : \mathbb{R}^n \to \mathbb{R}^n$. Gesucht ist ein $\overline{x}$ mit $F(\overline{x}) = 0$. Bekannt ist ein y mit $F(y) \approx 0$.
Frage: Ist y ein guter Näherungswert an $\overline{x}$?

Bezeichnung: Die Größe $F(y)$ heißt **Defekt**.

Gute Stabilität: kleiner Defekt $\Rightarrow$ gute Näherung.
Schlechte Stabilität: kleiner Defekt $\not\Rightarrow$ gute Näherung.

Störempfindlichkeit
F ist nicht exakt bekannt, sondern nur eine Approximation $\overline{F}$. Gesucht ist ein $\overline{x}$ mit $F(\overline{x}) = 0$. Bekannt ist ein y mit $\overline{F}(y) = 0$.
Frage: Folgt aus $F - \overline{F}$ klein auch $\overline{x} - y$ klein.

Bezeichnung: **Störunempfindlichkeit** bedeutet, dass aus einer kleiner Störung folgt, dass y ein guter Näherungswert an $\overline{x}$ ist.

Die Störung wird durch einen p-dimensionalen Parameter-Vektor modelliert:

$$G : \mathbb{R}^{n+p} \to \mathbb{R}^n, (x, \mu) \mapsto G(x, \mu)$$

z.B. $p(x) = F(x)$ ein Polynom vom Grad n:
$G(x, \mu) = G(x, \mu_0, \ldots, \mu_n) = p(x) + \sum_{i=0}^{n} \mu_i x^i$.

Das ungestörte Problem ist ein Ausgangsproblem für ein bestimmtes $\overline{\mu}$. Dann gilt $G(x, \overline{\mu}) = F(x)$.
Ausgangsproblem: $F(\overline{x}) = 0$ (hier $\overline{\mu} = 0$).
Es gibt ein $\overline{\mu}$ mit $G(\overline{x}, \overline{\mu}) = F(\overline{x}) = 0$
Berechnet ist $G(y, \eta) = 0$; Störung $\overline{\mu} - \eta$.
Störunempfindlich: $\overline{\mu} - \eta$ klein $\Rightarrow$ $y - \overline{x}$ klein.

b. Eine allgemeine Fehlerdarstellung

Gegeben sei eine stetig differenzierbare Funktion $G : \mathbb{R}^{n+p} \to \mathbb{R}^n, (u, v) \mapsto G(u, v)$.

$$G(u,v) = \begin{pmatrix} g_1(u_1, \ldots, u_n, v_1, \ldots, v_n) \\ \vdots \\ g_n(u_1, \ldots, u_n, v_1, \ldots, v_n) \end{pmatrix}$$

Für jede Komponente g_j gilt nach dem Mittelwertsatz:
$g_j(x, \mu) - g_j(y, \eta)$
$= D_u g_j(y + t_j(x-y), \eta + t_j(\mu - \eta))(x - y) + D_v g_j(y + t_j(x-y), \eta + t_j(\mu - \eta))(\mu - \eta)$
$= \sum_{i=1}^{n} \frac{\partial}{\partial u_i} g_j(y + t_j(x-y), \eta + t_j(\mu-\eta))(x_i - y_i) + \sum_{i=1}^{p} \frac{\partial}{\partial v_i} g_j(y + t_j(x-y), \eta + t_j(\mu-\eta))(\mu_i - \eta_i)$
mit $t_j \in (0, 1)$, D_u, D_v die partiellen Ableitungen nach u bzw. v.

Kompakt:
$G(x, \mu) - G(y, \eta) = D_u G \cdot (x - y) + D_v G \cdot (\mu - \eta)$.

$D_u G$ hat die Komponente ($n \times n$-Matrix)
$\frac{\partial}{\partial u_j} g_i(\ldots)$ für $i = 1, \ldots, n; j = 1, \ldots, n$.
$D_v G$ hat die Komponente ($n \times p$-Matrix)
$\frac{\partial}{\partial v_j} g_i(\ldots)$ für $i = 1, \ldots, n; j = 1, \ldots, p$.

$p = 0$:
$G(x) - G(y) = DG \cdot (x - y)$.

Stabilitätsproblem:
$F(\overline{x}) = 0$, Näherungswert y.
$F(\overline{x}) - F(y) = DF \cdot (\overline{x} - y)$.

Problem: Zwischenstellen in DF sind nicht bekannt.
Deshalb wird $DF \approx DF(\overline{x})$ gesetzt.

$$\Rightarrow \underbrace{F(\overline{x})}_{=0} - F(y) \approx DF(\overline{x}) \cdot (\overline{x} - y)$$

Sei nun $DF(\overline{x})$ invertierbar. Dann folgt $\overline{x} - y \approx -[DF(\overline{x})]^{-1} \cdot F(y)$.

Fazit: Norm$([DF(\overline{x})]^{-1})$ klein $\Rightarrow$ gute Stabilität.

Störungsproblem:
$0 = G(\overline{x}, \overline{\mu}) - G(y, \eta) \approx D_u G(\overline{x}, \overline{\mu}) \cdot (\overline{x} - y) + D_v G(\overline{x}, \overline{\mu}) \cdot (\overline{\mu} - \eta)$.

Sei $D_u G(\overline{x}, \overline{\mu})$ invertierbar.
$(\overline{x} - y) \approx -[D_u G(\overline{x}, \overline{\mu})]^{-1} D_v G(\overline{x}, \overline{\mu}) \cdot (\overline{\mu} - \eta)$.

Fazit: Norm($[D_u G(\overline{x}, \overline{\mu})]^{-1} D_v G(\overline{x}, \overline{\mu})$) klein $\Rightarrow$ störungsunempfindlich.

c. Stabilität und Störunempfindlichkeit einer Nullstelle

Sei $f : \mathbb{R} \to \mathbb{R}$ stetig und differenzierbar, sei $\overline{x}$ eine Nullstelle mit $f'(\overline{x}) \neq 0$:
$y - \overline{x} \approx \frac{1}{f'(\overline{x})} f(y)$.

Z.B. $|f'(\overline{x}) \approx 10^k$, $|f(y)| = 10^d$.

$\Rightarrow |y - \overline{x}| \approx 10^{d-k}$.

Beachte:
Das Stabilitätsverhalten in den einzelnen Nullstellen kann sehr untersschiedlich sein.

Störungsproblem
$p(\overline{x}) = 0$

Berechnet: $f(\overline{x}, \overline{\mu}) = 0$, $f(y, \eta) = 0$.
Annahme: $f(x, \mu) = p(x) + \mu q(x)$ mit $p, q \in C^1[a, b]$.

$\overline{\mu} = 0 \Rightarrow$ ungestörtes Problem.
$f_x(x, \mu) = p'(x) + \mu q'(x) \Rightarrow f_x(\overline{x}, \overline{\mu}) = f_x(\overline{x}, 0) = p'(\overline{x})$
$f_\mu(x, \mu) = q(x) \Rightarrow f_\mu(\overline{x}, 0) = q(\overline{x})$.

$\Rightarrow \overline{x} - y \approx \frac{q(\overline{x})}{p'(\overline{x})} \cdot \eta$

Definition:
Doe Größte $\left|\frac{q(\overline{x})}{p'(\overline{x})}\right|$ heißt **Konditionszahl** der Nullstelle $\overline{x}$.

Fazit:

- kleine Konditionszahl ⇒ störunempfindlich;
- große Konditionszahl ⇒ störempfindlich.

Beispiel: *(vgl. Buch von Schwarz)*
$p(x) = \prod_{k=1}^{12}(x-k) = x^{12} \pm \ldots - 6926634x^7 \pm \ldots (= p_{12}x^{12} + \ldots + p_1 x + p_0)$

Störung nur im j-ten Koeffizienten: $\mu \cdot q(x) = \mu \cdot x^j$.

Konditionszahlen:
$\frac{q(k)}{p'(k)} = \frac{|p_j| \cdot k^j}{(k-1)!(12-k)!} = c_{k,j}$.

(z.B. Konditionszahlen bei $j = 7$ relativ groß, bei $j = 1$ relativ klein)

Störung bei $j = 7$: $p_7^* := p_7 + \eta p_7 = 6926634,001 \Rightarrow \eta \approx 1,44 \cdot 10^{-10}$.

$k = 9$: $9 - y_9 \approx c_{9,7} \cdot \eta \Rightarrow y_9 \approx 8,980\ldots$

d. Störempfindlichkeit linearer Gleichungssysteme

$Ax = b$, $A \in \mathbb{R}^{n\times n}$, $\det A \neq 0$
exakte Lösung: $\overline{x}$: $A\overline{x} = b$.

Störung in der rechten Seite:

$$F(x,\mu) = Ax - b - \mu, \text{ mit } \mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix} \in \mathbb{R}^n.$$

$D_x F(\overline{x}, 0) = A$, $D_\mu F(x,\mu) = -I \Rightarrow D_\mu F(\overline{x}, 0) = -I$.

$\Rightarrow \overline{x} - y \approx -[D_x F(\overline{x}, 0)]^{-1} D_\mu(\overline{x}, 0) \cdot \eta = -A^{-1}\eta$.
Hier gilt sogar: $\overline{x} - y = -A^{-1}\eta$, da $D_x F$ und $D_\mu F$ konstant sind (und dadurch die

Zwischenwerte nicht abweichen); vgl. 12.b, Störungsproblem.

$\Rightarrow \|\overline{x} - y\| \leq \text{lub}(A^{-1})\eta$ (absoluter Fehler)
$A\overline{x} = b \Rightarrow \|b\| \leq \text{lub}(A)\|\overline{x}\| \Rightarrow \frac{1}{\|\overline{x}\|} \leq \text{lub}(A) \cdot \frac{1}{\|b\|}$

$$\Rightarrow \boxed{\frac{\|\overline{x}-y\|}{\|\overline{x}\|} \leq \underbrace{\text{lub}(A^{-1}) \cdot \text{lub}(A)}_{\text{cond}(A)} \cdot \frac{\|\eta\|}{\|b\|}} \quad \text{(relativer Fehler)}$$