Numerik I

WS 04/05 gelesen von Dr. Andreas Dedner

Vorwort

Das ist meine Mitschrift zur Vorlesung “Numerik I” gelesen von Dr. Andreas Dedner an der Universität Freiburg im Wintersemester 2004/2005.

Es besteht keine Garantie auf Richtigkeit und/oder Vollständigkeit. Ich möchte mich bei den Leuten bedanken, die mir dabei geholfen haben, besonders bei meinem besten Freund Sascha Fröschl, weil er jede Menge Fehler findet.

Falls ihr Fehler (jeder Art) findet, oder ein Kommentar oder sonts noch was habt, dann schreibt bitte an pabloy@pcpool.mathematik.uni-freiburg.de. Ich werde versuchen eine Subversion-Repository-Version zu erstellen, so dass jeder mitschreiben kann. Der LaTeX-Quellcode findet ihr unter http://pcpool.mathematik.uni-freiburg.de/~pabloy/numi/. Der Code ist frei verfügbar und kann jeder Zeit heruntergeladen werden und verändert werden und anschließend eine Kopie weiter veröffentlichen. Die einzige Bediengung ist, dass man den Source Code weitergibt.

Pablo Yanez Trujillo

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Grundlagen

1.1 Normierte Räume

Im folgenden Abschnitt werden wir die nötigen Definitionen kennenlernen, die wir brauchen. Wir verwenden folgende Notation für einen Körper $\mathbb{K}$: $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$, V Vektorraum.

Definition 1.1 (Die Norm)
Eine Abbildung $\|\cdot\| : V \longrightarrow \mathbb{R}$ *heißt* **Norm**, *falls*

(i) $\|v\| > 0 \quad \forall v \in V \backslash \{0\}$

(ii) $\|\lambda v\| = |\lambda| \, \|v\| \quad \forall \lambda \in \mathbb{K}, \ \forall v \in V$

(iii) $\|v + w\| \leq \|v\| + \|w\| \quad \forall v, w \in V$

Beispiel 1.2
Sei $V = \mathbb{R}^n, v = (v_1, \ldots, v_n) \in \mathbb{R}^n$

$\|v\|_\infty := \max_{1 \leq i \leq n} |v_i| \,, \quad \|v\|_1 := \sum_{i=1}^{n} |v_i|,$

$\|v\|_2 = \left(\sum_{i=1}^{n} |v_i|^2 \right)^{\frac{1}{2}}, \quad \|v\|_p := \left(\sum_{i=0}^{n} |v_i|^p \right)^{\frac{1}{p}} \quad (1 \leq p < \infty)$

Beispiel 1.3
Sei $V = C^0(I), I = [a, b] \subset \mathbb{R}$
$\|v\|_\infty := \sup \{|v(x)| \mid x \in I\}$

$\|v\|_p := \left(\int_a^b |v(x)|^p \, \mathrm{d}x \right)^{\frac{1}{p}}$

Definition 1.4 (Normierter Raum)
Ein Vektorraum V *zusammen mit einer Norm* $\|\cdot\|$, *geschrieben* $(V, \|\cdot\|)$, *heißt* **normierter Raum**

Definition 1.5
Eine Folge $(u_n)_{n\in\mathbb{N}} \subset V$ *konvergiert gegen* $u \in V \quad :\Longleftrightarrow$
$\forall \varepsilon > 0\ \exists N\ \forall\ n > N : \|u_n - u\| < \varepsilon$

Eine Folge $(u_n)_{n\in\mathbb{N}} \subset V$ *heißt* **Cauchy Folge** $:\Longleftrightarrow$
$\forall\ \varepsilon > 0\ \exists\ N\ \forall\ m, l > N : \|u_n - u_m\| < \varepsilon$

$(V, \|\cdot\|)$ *ist ein* **Banachraum**, *falls alle Cauchy-Folgen konvergieren.*

Beispiel 1.6
$(\mathbb{R}^n, \|\cdot\|)$ ist ein Banachraum für alle $\|\cdot\|$,
$\left(C^0(I), \|\cdot\|_\infty\right)$ ist ein Banachraum, $\left(C^0(I), \|\cdot\|_p\right)$ ist dagegen nicht vollständig

Satz 1.7
Sei $\dim V < \infty$, $\|\cdot\|_1$ und $\|\cdot\|_2$ zwei Normen. Dann existieren $m, M \in \mathbb{R} : m\|v\|_1 \leq \|v\|_2 \leq M\|v\|_1\ \forall\ v \in V$, d.h. $\|\cdot\|_1$ und $\|\cdot\|_2$ sind **äquivalente Normen**.

Definition 1.8 (Skalarprodukt)
$\langle\cdot,\cdot\rangle : V \times V \longrightarrow \mathbb{C}$ *ein* **Skalarprodukt**, *falls*

(i) $\forall\ v \in V\backslash\{0\} :\ \langle v, v\rangle \geq 0$

(ii) $\forall\ u, v \in V\ \langle u, v\rangle = \langle v, u\rangle$

(iii) $\forall\ u, v, w \in V \forall\ \alpha \in \mathbb{K} :$
$\langle \alpha u, v\rangle = \alpha\,\langle u, v\rangle$
$\langle u + v, w\rangle = \langle u, w\rangle + \langle v, w\rangle$

Folgerung: $\langle u, \alpha v\rangle = \overline{\alpha}\,\langle u, v\rangle\ ,\ \langle u, v + w\rangle = \langle u, v\rangle + \langle u, w\rangle$

Satz 1.9
Sei $\langle\cdot,\cdot\rangle$ ein Skalarprodukt, dann wird durch $\|v\| := \sqrt{\langle v, v\rangle}$ eine Norm induziert

Definition 1.10
Ein Vektorraum mit Skalarprodukt heißt **Prähilbertraum** *, falls* V *mit der induzierten Norm* ***nicht*** *vollständig ist, sonst bezeichnet* V *einen* **Hilbertraum**

Beispiel 1.11 (Cauchy-Schwarz-Ungleichung)
$\forall\ u, v \in V :\ |\langle u, v\rangle| \leq \sqrt{\langle u, u\rangle\,\langle v, v\rangle}$, Gleichheit $\Longleftrightarrow$ u, v linear abhängig

Beispiel 1.12
Sei $V = \mathbb{R}^n,\ \langle u, v\rangle := \sum_{i=1}^{n} u_i v_i$ ist ein Skalarprodukt und induziert die **euklidische Norm**
$\|v\|_2 := \left(\sum_{i=1}^{n} |v_i|^2\right)^{\frac{1}{2}}$

1.2 Operatoren

Definition 1.13
U, V normierte Vektorräume, $D \subseteq U$. Wir bezeichnen eine Abbildung $T : D \longrightarrow V$ als **Operator**. *Dabei gilt:*

(i) T heißt **stetig** *in $u \in D$ $:\Longleftrightarrow$*
$\forall\, \varepsilon > 0\ \exists\, \delta > 0\ \forall\, v \in D : \|u - v\|_U < \delta \implies \|T(u) - T(v)\|_V < \varepsilon$

(ii) T heißt stetig in D $:\Longleftrightarrow$
T ist stetig für alle $u \in D$

(iii) T heißt **Lipschitz-stetig** $:\Longleftrightarrow$
es existiert ein $L > 0\ \forall\, u, v \in D :\ \|T(u) - T(v)\|_V < L \|u - v\|_U$

Bemerkung 1.14
Es ist leicht zu sehen, dass aus **(iii)** **(ii)** folgt und aus **(ii)** folgt **(i)**

Definition 1.15
T heißt **linearer Operator** *(oder einfach linear), falls $\forall\, u, v \in V, \alpha \in \mathbb{K}$:*

(i) $T(u + v) = T(u) + T(v)$

(ii) $T(\alpha v) = \alpha T(v)$

Bemerkung 1.16
Ist T linear, so schreibt man häufig Tu statt $T(u)$

Beispiel 1.17
$V = W = \mathbb{R}^n, A \in \mathbb{R}^{n\times n} :\ Tu = Au$ ist ein linearer Operator
$V = C^0(I), W = \mathbb{R} :\ Tu := \int_a^b u(x)\mathrm{d}x$ ist ein linearer Operator

Definition 1.18
$T : U \longrightarrow V$ sei ein Operator. T heißt **beschränkt**, *falls es ein $C > 0$ gibt, so dass* $\forall\, u \in U :\ \|T(u)\|_V \leq C \|u\|_U$

Satz 1.19
Für einen linearen Operator $T : U \longrightarrow V$ sind äquivalent:

(i) T ist beschränkt

(ii) T ist Lipschitz-stetig

(iii) T ist stetig in 0

Beweis: Siehe Übungsblatt 1

Bemerkung 1.20

(i) $\dim U < \infty, \dim V < \infty$, dann sind alle linearen Operatoren beschränkt und damit stetig.

(ii) Auf unendlich-dimensionalen Vektorräumen existieren auch unbeschränkte lineare Operatoren

(iii) Die Aussage von Satz 1.19 (Seite 3) ist nur richtig für lineare Operatoren.
Bsp: $T : \mathbb{R} \longrightarrow \mathbb{R},\ x \longmapsto x^2$: es existiert keine Konstante C mit $\left|x^2\right| < C\,|x|\ \forall\, x \in \mathbb{R}$

Definition 1.21
Mit $\mathrm{B}(U,V)$ *bezeichnen wir den Raum der beschränkten linearen Operatoren.* $\mathrm{B}(U,V)$ *ist ein Vektorraum. Durch*

$$\|T\|_{U,V} := \sup_{u \in U \setminus \{0\}} \frac{\|Tu\|_V}{\|u\|_U}$$

wird eine Norm auf $\mathrm{B}(U,V)$ *definiert. Diese wird als die durch* $\|\cdot\|_U, \|\cdot\|_V$ **induzierte Operatornorm** *bezeichnet.*

Folgerung 1.22

(i) $\|T\|_{U,V} = \sup\limits_{\substack{u \in U \\ \|u\|_U = 1}} \|Tu\|_V$ (wegen Linearität von T)

(ii) $\|Tu\| \leq \|T\|\,\|u\|$, und $\|T\|_{U,V}$ ist die kleinste Konstante mit dieser Eigenschaft für alle $u \in U$ (folgt aus der Definition)

(iii) $\|id\|_{U,V} = 1$, dabei ist $id \in \mathrm{B}(U,V),\ u \longmapsto u$

Beispiel 1.23
$U, V = \mathbb{R}^n$, dann entspricht $\mathrm{B}(U,V)$ dem Raum der $n \times n$ Matrizen. Daher wird die Operatornorm auch häufig Matrixnorm genannt.

Satz 1.24
Die induzierte (Matrix-) Operatornorm ist submultiplikativ, d.h. $\|A \circ B\| \leq \|A\| \cdot \|B\|$

Beweis: (gilt nur für die induzierte Matrixnorm)
$\|(A \circ B)x\| = \|A(B(x))\| \overset{1.22ii}{\leq} \|A\|\,\|Bx\| \overset{1.22ii}{\leq} \|A\|\,\|B\|\,\|x\|$
Sei $x \neq 0 \implies \frac{\|ABx\|}{\|x\|} \leq \|A\|\,\|B\|$ □

Bemerkung 1.25
Die induzierten Operatornormen ergeben nicht alle Normen auf $\mathrm{B}(U,V)$. Sei etwa $A \in \mathbb{R}^{n \times n}, A = (a_{ij})$, dann wird durch $\|A\| = \sup\limits_{1 \leq i,j \leq n} |a_{ij}|$ eine Norm definiert, die nicht induziert ist.

Beispiel 1.26
Die durch $\|\cdot\|_1$ und $\|\cdot\|_\infty$ induzierte Operatornormen werden in den Übungen behandelt.

Sei $A : (\mathbb{R}^n, \|\cdot\|_2) \longrightarrow (\mathbb{R}^n, \|\cdot\|_2)$, dann gilt $\|A\|_{2,2} = \sqrt{\lambda_{\max}(A^*A)}$, wobei $\lambda_{\max}(B)$ für $B \in \mathbb{R}^{n\times n}$ den betragsmäßig größten **Eigenwert (EW)** bezeichnet. Sei $A = (a_{ij})$, dann ist $A^* = \overline{A}^\top$. Diese Norm wird als **Spektralnorm** bezeichnet. Ist $A \in \mathbb{R}^{n\times n}$, dann ist $\overline{A}^\top = A^\top$.

Beweis: **Bemerkungen:** $(A^*A)^* = A^*A \implies A^*A$ ist hermitesch $\implies$ alle EW sind reell.
Auch gilt: $x^*(AA^*)x = (Ax)^* Ax = \langle Ax, Ax\rangle \geq 0$

$\implies A^*A$ positiv definit $\implies$ alle EW sind positiv

Da A^*A hermitesch ist, existiert ein $U \in \mathbb{C}^{n\times n}$ mit $U^*U = id$ (d.h. U ist unitär) und

$$U^*(A^*A)U = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} =: D \ (*)$$

Sei U_i die i-te Spalte von U, d.h. $U = (U_1, \ldots, U_n)$ und $\|U_i\|_2 = 1 \ \forall i \in \{1, \ldots, n\}$, dann ist
$A^*AU = UD$, da $U^{-1} = U^* \implies A^*AU_i = \lambda_i u_i$, d.h. u_i sind Eigenvektoren (EV) von A^*A
Ebenfalls gilt $u_i^* A^* A u_i = \lambda_i$ wegen $(*)$.

Sei $x \in \mathbb{C}^n mit \|x\|_2 = 1$, d.h. $1 = \|x\|_2^2 = (x, x) = x^*x$
Setze $y = U^*x$, d.h. $x = Uy$, da $U^*U = id$. Es gilt:

$$\begin{aligned} \|Ax\|_2^2 &= \langle Ax, Ax\rangle = \langle x, A^*Ax\rangle = \langle x, UDU^*x\rangle \text{ wegen } (*) \\ &= \langle x, UDy\rangle = \langle U^*x, Dy\rangle = \langle y, Dy\rangle \\ &= \sum_{i=1}^n \overline{y_i}\lambda_i y_i \leq \max_{1\leq i\leq n} \lambda_i \sum_{i=1}^n y_i^2 \text{ da alle } \lambda_i > 0 \\ &= \lambda_{\max}(A^*A)\,\|y\|_2^2 = \lambda_{\max}(A^*A), \text{ da } \|y\|_2 = \|U^*x\| = \|x\| = 1, \text{ da } U^* \text{ unitär ist} \end{aligned}$$

Also $\|Ax\|_2 \leq \sqrt{\lambda_{\max}(A^*A)} \ \forall\, x$ mit $\|x\|_2 = 1 \implies \|A\|_{2,2} \leq \sqrt{\lambda_{\max}(A^*A)}$

Sei λ_i der größte EW, u_i der zugehörige EV mit $\|u_i\|_2 = 1$. Da $\|A\|_{2,2} = \sup_{\|x\|_2=1} \|Ax\|_2$ folgt $\|A\|_{2,2} \geq \|Au_i\|_2$

$$\begin{aligned} \implies \|A\|_{2,2}^2 &\geq \|Au_i\|_2^2 \\ &= \langle Au_i, Au_i\rangle \\ &= \langle u_i, A^*Au_i\rangle \\ &= \langle u_i, \lambda_i u_i\rangle \\ &= \lambda_i \langle u_i, u_i\rangle \\ &= \lambda_i \|u_i\|_2^2 = \lambda_i = \lambda_{\max}(A^*A) \implies \text{Behauptung} \quad \square \end{aligned}$$

1.3 Banachscher Fixpunktsatz

Definition 1.27
Sei $D \subset X$, X normierter Verktorraum, Y normierter Vektorraum. Dann heißt ein Operator $T : D \longrightarrow Y$ eine **Kontraktion**, *falls T Lipschitz-stetig mit Lipschitz-Konstante $0 < L < 1$ ist, d.h. $\forall\, u, v \in D : \|T(u) - T(v)\|_Y \leq L\,\|u - v\|_X$*

Definition 1.28
Sei $T : D \longrightarrow D$ ein Operator. Dann heißt $\overline{u} \in D$ **Fixpunkt** *von T in D, falls $T(\overline{u}) = \overline{u}$*

Satz 1.29 (Banachscher Fixpunktsatz)
Sei X ein Banachraum, $D \subseteq X$ abgeschlossen, $T : D \longrightarrow D$ eine Kontraktion. Dann gilt:

(i) T hat genau einen Fixpunkt $\overline{u} \in D$

(ii) Sei $u_o \in D$ beliebig und $u_{k+1} := T(u_k), k = 0, 1, \ldots \implies u_k \longrightarrow \overline{u}$

(iii) $\|\overline{u} - u_k\| \leq L \|\overline{u} - u_{k-1}\|$ $(k \geq 1)$, d.h. der Fehler nimmt monoton ab

(iv) $\|\overline{u} - u_k\| \leq \frac{L^k}{1-L} \|T(u_0) - u_0\|$ $(k \geq 1)$, **a-priori Abschätzung**

(v) $\|\overline{u} - u_k\| \leq \frac{L}{1-L} \|u_k - u_{k-1}\|$ $(k \geq 1)$, **a-posteriori Abschätzung**

Beweis: Siehe Übungen

Bemerkung 1.30
Satz 1.29 (iv) (Seite 6) gibt eine a-priori Schranke, die man nutzen kann, um ein Index k_0 zu bestimmen mit $\|\overline{u} - u_{k_0}\| \leq TOL$ für eine gegebene Toleranz $TOL > 0$

Sei TOL gegeben, O.B.d.A $TOL < 1$

$$\begin{aligned} \|u_{k_0} - \overline{u}\| \quad &\leq \quad \tfrac{L^{k_0}}{1-L} \|T(u_0) - u_0\| \leq TOL \\ &\iff \quad L^{k_0} \leq (1-L) \tfrac{TOL}{\|T(u_0) - u_0\|} \\ &\iff \quad k_0 \log L \leq \log(1-L) + \log TOL - \log\left(\|T(u_0) - u_0\|\right) \\ &\iff \quad k_0 \geq \tfrac{\log(1-L) + \log TOL - log(\|T(u_0) - u_0\|)}{\log L}, \text{ da } 0 < L < 1 \text{ und daher } \log L < 0 \end{aligned}$$

Meistens ist dies eine Überschätzung des Aufwands.

Satz 1.29 (v) (Seite 6) kann als Abbruchkriterium während der Iteration benutzt werden, d.h. man bricht ab, falls $\frac{L}{1-L} \|u_k - u_{k-1}\| < TOL$

1.4 Taylorreihe

Definition 1.31
Sei $I = (a.b)$ ein offenes Intervall. Dann bezeichnen wir mit $C^0(I)$ den Raum der **stetigen Funktionen auf I** *. Mit $C^m(I) := \left\{f : I \longrightarrow \mathbb{R} \mid f, f', f'', \ldots, f^{(m)} \text{ ex. und sind stetig}\right\}$ bezeichnen wir den Raum der* $\boldsymbol{m}$ **Mal stetig differenzierbaren Funktionen**.

Kurzschreibweise: $C^m(a, b)$ statt $C^m((a, b))$
$C^\infty(I) := \bigcap_{m \in \mathbb{N}} C^m(I) \implies C^\infty(I) \subset \ldots \subset C^m(I) \subset \ldots \subset C^0(I)$

Satz 1.32 (Taylorreihe mit Lagrange Restglied)
Sei $f \in C^{m+1}(a, b)$ $x_0 \in (a, b)$ fest. Dann existiert für jedes $x \in (a, b)$ ein ξ zwischen x_0 und x mit

$$f(x) = \sum_{k=0}^{m} \frac{1}{k!} f^{(k)}(x_0)(x - x_0)^k + R_m(x)$$

wobei $R_m(x) := \frac{1}{(m+1)!} f^{(m+1)}(\xi)(x - x_0)^{m+1}$

Satz 1.33 (Taylorreihe mit Integralrestterm)
Sei $f \in C^{m+1}(a,b),\ x_0 \in (a,b)$ fest. Dann gilt für jedes $x \in (a,b)$

$$f(x) = \sum_{k=0}^{m} \frac{1}{k!} f^{(k)}(x_0)(x-x_0)^k + R_m(x)$$

wobei $R_m(x) := \frac{1}{m!} \int_{x_0}^{x} f^{(m+1)}(t)(x-t)^m \mathrm{d}t$

Beweis: Für beide Sätze siehe Analysis I

Folgerung 1.34 (häufig verwendete Form)
Sei $f \in C^{m+1}(x_0 - h_0, x_0 + h_0),\ x_0 \in \mathbb{R}, h_0 > 0$. Sei $|h| \leq h_0$ dann

$$f(x_0+h) = f(x_0) + \sum_{k=1}^{m} \frac{f^{(k)}(x_0)}{k!} h^k + \omega_m(h) h^m$$

mit $\omega_m : (-h_0, h_0) \longrightarrow \mathbb{R}$ mit $\lim\limits_{h \longrightarrow 0} \omega_m(h) = 0$

Beweis: Wende Satz 1.32 (Seite 6) an mit $x = x_0 + h$, d.h. es existiert ein ξ mit $|\xi| < |h|$ und

$$\begin{aligned} f(x_0+h) - \sum_{k=0}^{m-1} \frac{f^{(k)}(x_0)}{k!} h^k &= \frac{f^{(m)}(\xi)}{m!} h^m \\ &= \frac{f^{(m)}(x_0)}{m!} h^m - \frac{f^{(m)}(x_0)}{m!} h^m + \frac{f^{(m)}(\xi)}{m!} h^m \\ &= \frac{f^{(m)}(x_0)}{m!} h^m + \omega_m(h) h^m \end{aligned}$$

mit $\omega_m(h) = \frac{f^{(m)}(\xi) - f^{(m)}(x_0)}{m!}$. Da $|\xi| < |h|$ und $f^{(m)}$ stetig $\implies \lim\limits_{h \longrightarrow 0} \frac{f^{(m)}(\xi) - f^{(m)}(x_0)}{m!} = 0$ □

Definition 1.35
Die Funktion $f \in C^1(x_0 - h_0, x_0 + h_0)$ ist in **erster Näherung** *gleich $f(x_0) + f'(x_0)h$ in einer Umgebung um x_0, d.h. es existiert ein $\overline{\omega} : (-h_0, h_0) \longrightarrow \mathbb{R}$ mit $\frac{|\overline{\omega}(h)|}{|h|} \longrightarrow 0$ und $f(x_0+h) = f(x_0) + f'(x_0)h + \overline{\omega}(h)$*

Notation: $f(x_0+h) \stackrel{\bullet}{=} f(x_0) + f'(x_0)h$

Definition 1.36 (Landau Symbole)
Seien $f, g : \mathbb{R} \longrightarrow \mathbb{R}$. Dann schreiben wir:

(i) $g(t) = O(h(t))$ für $t \longrightarrow 0 \iff$ es eine Konstante $C > 0$ und ein $\delta > 0$ gibt, so dass $|g(t)| \leq C\,|h(t)|\ \forall\ |t| < \delta$

(ii) $g(t) = o(h(t))$ für $t \longrightarrow 0 \iff$ es ein $\delta > 0$ und ein $c : (0, \delta) \longrightarrow \mathbb{R}$ gibt, so dass $|g(t)| \leq c\,(|t|)\,|h(t)|\ \forall\ |t| < \delta$ und $c(t) \longrightarrow 0$ für $t \longrightarrow 0$

Beispiel: $f \in C^1(\mathbb{R})$, dann ist $f(x) - (f(x_0) + f'(x)(x-x_0)) = o\left(|x-x_0|^2\right)$ wegen Folgerung 1.34 (Seite 7) mit $h = x - x_0$ und $m = 1$.

Ist $f \in C^1(\mathbb{R})$, dann ist $f(x) - (f(x_0) + f'(x_0)(x - x_0)) = O\left(|x - x_0|^2\right)$ wegen Satz 1.32 (Seite 6), da f'' beschränkt in einer Umgebung von x_0, d.h. $|f''(\xi)| < C$.

1.5 Fehleranalyse: Approximationsfehler

Problem: Ein Stahlseil der Länge $L = 1$ sei an seinen Endpunkten so befestigt, dass es (fast) straff gespannt erscheint. Nun soll die Ausleknung des Seils berechnet werden, wenn sich in der Mitte des Seils ein Seiltänzer befindet.

1. **Modellfehler:** Wir gehen davon aus, dass sich das Seil als Graph einer Funktion $y : (0,1) \longrightarrow \mathbb{R}$ beschreiben lässt, welche die sogenannte **potentielle Gesamtenergie**:

$$E(y) = \frac{c}{2} \int_0^1 \frac{y'(t)^2}{\sqrt{1 + y'(t)}} \mathrm{d}t - \int_0^1 f(t) y(t) \mathrm{d}t$$

   minimiert.

   Dabei ist c eine Materialkonstante und f die Belastungsdichte.

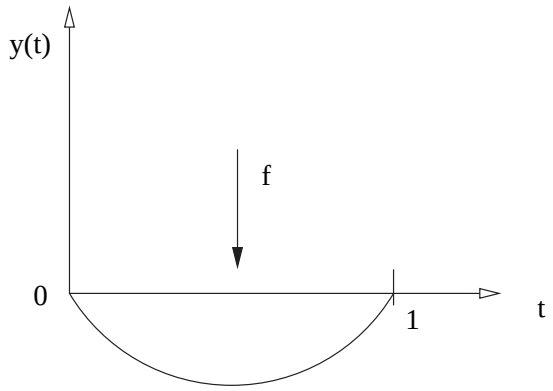


Abbildung 1.1: Modellfehler

2. Zur Vereinfachung (Abb 1.1) nehmen wir an, dass $|y'(t)| \ll 1$. Dann können wir das Funktional E vereinfachen zu:

$$\overline{E}(y) = \frac{c}{2} \int_0^1 y'(t)^2 \mathrm{d}t - \int_0^1 f(t) y(t) \mathrm{d}t$$

   Dabei sind eine Reihe von Effekten vernachlässigt worden, diese führen zu Modellfehlern, diese spielen in dieser Vorlesung keine weitere Rolle. Wir nehmen an, dass (2) das zu lössende Problem sei: Als notwendige und hinreichende Bedingung für die Minimierung von (2) erhält man durch Variation

$$\left(\frac{d}{d\alpha} \overline{E}\left(y + \alpha\varphi\right)|_{k=0} = 0 \ \forall \text{ “zulässige“ } \varphi \right)$$

   die $Dgl: \ -cy''(t) = f(t), t \in (0,1), y(0) = y(1) = 0$

3. **Datenfehler:** c ist eine Materialkonstante, die vom Material des Seils abhängt (aber auch von Temperatur und Luftfeutigkeit). Der Wert für c kann nur durch Experimente bestimmt werden, und das ist zwangsläufig fehlerbehaftet. Daher muss sichergestellt werden, dass sowohl y als auch das numerische Verfahren nicht sensitiv vom konkreten Wert für c abhängen.

Einschub: Dgl: $u'(t) = (c - u(t))^2,\ u(0) = 1\ c > 0$, durch Substitution

$$v(t) = \frac{1}{c - u(t)} \implies v'(t) = \frac{u'(t)}{(c - u(t))^2} = 1 \implies u(t) = \frac{1 + tc(c-1)}{1 - t(c-1)}$$

.

Verhalten für c

$c = 1:\ u'(t) = 0 \implies u \equiv 1$

$c > 1:\ u' > 0$, d.h. u monoton wachsend und $\lim_{t \longrightarrow \infty} u(t) = c$

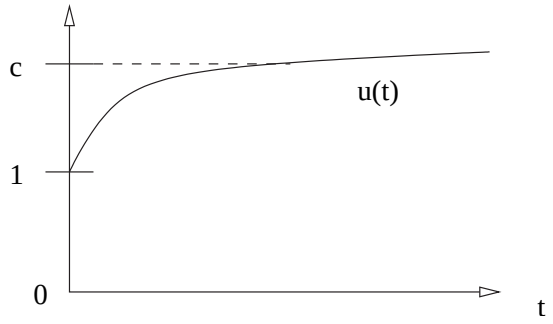


Abbildung 1.2: Datenfehler

$c < 1:\ u' > 0, \lim_{t \longrightarrow t_0} u(t) = \infty$ für $t_0 = \frac{1}{1-c} > 0$

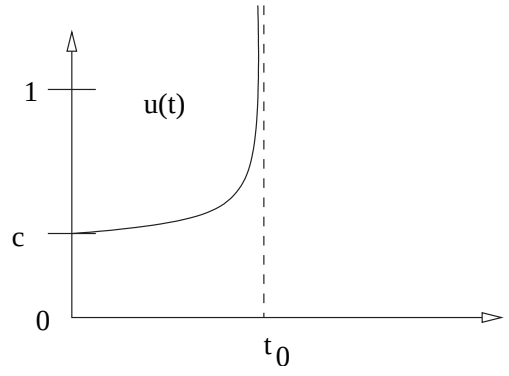


Abbildung 1.3: Datenfehler

Durch Messfehler oder auch Approximationsfehler kann leicht $y(t) > c$ sein, dann sieht man ein "blow-up" verhalten.

4. **Fehleranalyse**: Die Ableitungen müssen durch etwas Berechenbares ersetzt werden, auch können wir $y(t)$ nicht für alle t bestimmen. Sei $N \in \mathbb{N}, x_i := ih, i = 0, \ldots, N+1$ mit $h = \frac{1}{N+1}$, ersetze
$y''(x_i) \sim \frac{1}{h^2}\left(y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\right)$.
Setze: $f_i \equiv f(x_i)$ und $y_i \sim y(t_i)$, dann ist eine Finite-Diffenrenzen Approximation von

$-\frac{c}{h^2}\left(y_{i+1} - 2y_i + y_{i-1}\right) = f_i,\ i = 0, \ldots, N+1$

$-cy''(t) = f(t),\ t \in (0,1),\ y(0) = y(1) = 0.$

Setze $A = \begin{pmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{N\times N}$

$F = (f_i)_{i=1}^N \in \mathbb{R}^N$

Diskretes Problem: finde $y_h \in \mathbb{R}^N$

$$\frac{c}{h^2} A y_h = F$$

Zur Lösung verwenden wir die Identität

$$Dy_h = Dy_h - Ay_h + \frac{h^2}{c}F,\ D = diag(2) = \begin{pmatrix} 2 & & 0 \\ & \ddots & \\ 0 & & 2 \end{pmatrix}$$

y_h^0 sei ein Startwert (z.B: $y_h^0 = 0$)

$Dy_h{}^{n+1} := Dy_h - Ay_h^n + \frac{h^2}{c}F$ bzw $y_h^{n+1} = y_h^n - D^{-1}\left(Ay_h^n + \frac{h^2}{c}F\right)$
Es muss gezeigt werden, dass $y_h{}^n \to y_h$ für $n \to \infty$.

5. **Abbruchfehler:** Wir können nur bis zu einem endlichen $n_0 \in \mathbb{N}$ rechnen. Das heißt die Lösung eines Problems wird $y_h^{n_0}$ sein. Das heißt ein Fehler: $\|y_h^{n_0} - y_h\|$
6. **Rundungsfehler:** Auf einem Rechner kann nur eine endliche Teilmenge von $\mathbb{R}$ bearbeitet werden. Daher wird nicht $y_h^{n_0}$ berechnet, sondern die Approximation in dieser endlichen Menge.

Definition 1.37 (Gleitkommazahl)
Eine Gleitkommazahl zur Basis $b \in \mathbb{N}$ ist eine Zahl $a \in \mathbb{R}$ der Form

$$a = \pm\left[m_1 b^{-1} + \ldots + m_r b^{-r}\right] b^{\pm\left[e_{s-1}b^{s-1}+\ldots+e_0 b^0\right]} (*)$$

Man schreibt $\pm a = 0, m_1 \ldots m_r b^{\pm E}$ mit $E = [e_{s-1}b^{s-1} + \ldots + e_0 b^0]$ und $m_i \in \{0, \ldots, b-1\}, E \in \mathbb{N}, r, s \in \mathbb{N}$ abhängig von der Rechnerarchitektur.

Bemerkung:

1. Diese Darstellung ermöglicht die gleichzeitige Speicherung sehr unterschiedlich großer Zahlen, wie etwa die Lichtgeschwindigkeit $c \approx 0.29998 \cdot 10^{11}$ oder Elektronenruhemasse $m_0 \approx 0.911 \cdot 10^{-29}$
2. Als Normierung nimmt man für $a \neq 0$ an, dass $m_1 \neq 0$
3. Für Computer ist $b = 2$ üblich, für Menschen $b = 10$

Definition 1.38 (Maschinenzahlen)
Zu geg. (b, r, s) sei $A = A(b, r, s)$ die Menge der $a \in \mathbb{R}$ mit einer Darstellung $(*)$

$A(b, r, s)$ ist endlich mit größtem und kleinstem positiven Element $a_{\max} = (1 - b^{-r}) \cdot b^{b^s - 1}$, $a_{\min} = b^{-b^s}$

Zur Speicherung einer Zahl $a \in D = [-a_{\max}, -a_{\min}] \cup [a_{\min}, a_{\max}]$ wird eine Rundungsfunktion $rd : D \longrightarrow A$ mit $|a - rd(a)| = \min_{\overline{a} \in A} |\overline{a} - a|$ definiert.

$rd(a)$ wird gespeichert als:[1]

$\pm$	m_1	$\cdots$	m_r	$\pm$	e_0	$\cdots$	e_{s-1}

$rd(a) = 0, \underbrace{m_1, \ldots, m_r}_{\text{Mantisse } M}, b^{\pm E}$ mit E als Exponent.

Die heutigen PC benutzen 52 Bits für die Mantisse und 11 Bits für den Exponent; die $\pm$ werden mit 1 (negativ) und 0 (positiv) dargestellt.

Für $a \in (-a_{\min}, a_{max})$ wird in der Regel $rd(a) = 0$ ("underflow")

Für $|a| > a_{\max}$ wird von "overflow" geredet. Viele Compiler setzen $a = NaN$ (not a number) und die Rechnung muss abgebrochen werden.

Satz 1.39 (Rundungsfehler)
Der absolute Rundungsfehler, der durch Rundung verursacht wird, kann abgeschätzt werden durch

$$|a - rd(a)| \leq \frac{1}{2} b^{-r} \cdot b^E$$

wobei E der Exponent von a ist (in der $(*)$ Darstellung) und für den relativen Rundungsfehler gilt für $a \neq 0$

$$\frac{|rd(a) - a|}{|a|} \leq \frac{1}{2} b^{-r+1}$$

Die Zahl $eps := \frac{1}{2} b^{-r+1}$ heißt Maschinenzahl.

Beweis: $rd(a)$ weicht maximal eine halbe Einheit in der letzten Mantissenstelle von a ab.
Also $|a - rd(a)| \leq \frac{1}{2} b^{-r} b^E$.

Aufgrund der Normalisierung: $m_1 \neq 0 \implies |a| \geq b^{-1} b^E \implies$

$$\frac{|rd(a) - a|}{|a|} \leq \frac{\frac{1}{2} b^{-r} b^E}{b^{-1} b^E} = \frac{1}{2} b^{-r+1}$$

Setze $\varepsilon := \frac{rd(a)-a}{a} \implies |\varepsilon| \leq eps \implies rd(a) = \varepsilon a + a = a(1+\varepsilon)$ □

Definition 1.40 (Maschinenoperation)
Die Grundoperation $\star \in \{+, -, \times, /\}$ wird ersetzt durch $\circledast$. In der Regel gilt:

$$a \circledast b = rd(a \star b) = (a \star b)(1 + \varepsilon)$$

mit $|\varepsilon| \leq eps$

Bemerkung: Die Verknüpfungen $\circledast$ erfüllen **nicht** das Assoziativ- bzw. Distributivgesetz

Beispiel 1.41
Berechne das Integral $I_k := \int_0^1 \frac{x^k}{x+5} \mathrm{d}x$

(A) Es gilt

$$I_0 = \ln(6) - \ln(5)$$

und

$$I_k + 5I_{k-1} = \frac{1}{k} \ (k \geq 1) \text{ da}$$

[1] Jedes Kästchen entspricht einem Bit

$$\int_0^1 \frac{x^k}{x+5} + 5\frac{x^{k-1}-1}{x+5} = \int_0^1 x^{k-1}\mathrm{d}x = \frac{1}{k}$$

Bei einer Berechnung mit nur 3 Dezimalstellen ($r = 3, b = 10$) ergibt sich:

$$\begin{aligned} I_0 &= 0.182 \cdot 10^0 \\ I_1 &= 0.900 \cdot 10^{-1} \\ I_2 &= 0.500 \cdot 10^{-1} \\ I_3 &= 0.833 \cdot 10^{-1} \\ I_4 &= -0.166 \cdot 10^0 \end{aligned}$$

Die Berechnung ist fehlerhaft. Offensichtlich sind die I_k monoton fallend, da $I_k \searrow 0$ ($k \longrightarrow \infty$), aber es gibt widersprüchliche Ergebnisse (siehe I_3). Auf einem Standard PC ergab: $I_{21} = -0.158 \cdot 10^{-1}$ und $I_{39} = 8.960 \cdot 10^{10}$.

Dies ist ein Beispiel für **Fehlerfortpflanzung**, da der Fehler in I_{k-1} mit 5 multipliziert wird, um I_k zu berechnen.

(B) $I_{k-1} = \frac{1}{5}\left(\frac{1}{k} - I_k\right)$, $I_9 = I_{10}$

$$\begin{aligned} I_4 &= 0.343 \cdot 10^{-1} \\ I_3 &= 0.431 \cdot 10^{-1} \\ I_2 &= 0.500 \cdot 10^{-1} \\ I_1 &= 0.884 \cdot 10^{-1} \\ I_0 &= 0.182 \cdot 10^0 \end{aligned}$$

Hier tritt die **Fehlerdämpfung** auf.

Beispiel 1.42
Zu lösen ist das LGS

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.86419999 \\ 0.14400001 \end{pmatrix} =: b$$

Die exakte Lösung ist $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.9911 \\ -0.4870 \end{pmatrix}$

Durch Messfehler oder auch Rundung erhalten wir eine rechte Seite

$$\overline{b} = \begin{pmatrix} 0.8642 \\ 0.1440 \end{pmatrix}$$

Dann ist die Lösung $\begin{pmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \end{pmatrix}$ mit ca. 100% Abweichung.

Das bedeutet, dass kleine Änderungen der Eingabedaten zu großen Änderungen der Lösungen führen können.

Definition
Eine numerische Aufgabe (z.B. effizientes Lösen eines LGS oder Integrals) heißt **gut konditioniert**, *falls kleine Änderungen der Eingabedaten zu kleinen Änderungen der Lösung führen; sonst heißt das Problem* **schlecht konditioniert**.

Präzisieren wir: Was ist eine numerische Aufgabe? Was heißt klein?

$$A := \begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix}$$

sollte schlecht konditioniert sein.
Im folgenden 2 Ansätze:

1. Für einfache Probleme
2. Für etwas komplexere Probleme

Definition 1.43
Sei $f : U \longrightarrow \mathbb{R}^n$ mit $U \subset \mathbb{R}^m$ und sei $x_0 \in U$ vorgegeben. Dann versteht man unter der Aufagbe (f, x_0) die effektive Berechnung von f an der Stelle x_0. Dabei sind x_0 die Eingabe(daten).

Beispiel: $Ax = b, \quad (f, b)$ mit $f(b) = A^{-1}b$

Beispiel 1.44
Sei $x_0 = (x_1, \ldots, x_m)$ und $x_0 + \Delta_x \in U$ eine Störung der Eingabedaten mit $\|\Delta x\| \ll 1$. Falls $f : U \longrightarrow \mathbb{R}$ (d.h. $n = 1$) einmal stetig differenzierbar; so ist der Ergebnisfehler $\Delta f(x_0) = f(x_0) - f(x_0 + \Delta x)$ in erster Nährung gleich

$$\sum_{j=1}^{m} \frac{\partial f}{\partial x_j}(x_0)\Delta x_j = \bigtriangledown f(x_0)\Delta x$$

Für den relativen Fehler gilt in erster Nährung:

$$\frac{\Delta f(x_0)}{f(x_0)} \stackrel{\bullet}{=} \sum_{j=1}^{m} \left(\frac{\partial f}{\partial x_j}(x_0) \frac{x_j}{f(x_0)} \right) \frac{\Delta x_j}{x_j}$$

Definition 1.45 (Konditionszahlen I)
Wir nennen den Faktor $k_j := \frac{\partial f}{\partial x_j}(x_0)\frac{x_j}{f(x_0)}$ **(relative) Konditionszahl**.

Beweis: (Beweis von Satz 1.44)

Wie in der Folgerung 1.34 (Seite 7) kann man hier den Satz von Taylor anwenden:

$$f(x_0 + \Delta x) = f(x_0) + \bigtriangledown f(x_0)\Delta_x + \overline{\omega}\left(\|\Delta x\|\right)$$

mit $\overline{\omega}\left(\|\Delta x\|\right) = o\left(\|\Delta x\|\right) \implies$ Behauptung für den absoluten Fehler. Für den relativen Fehler ist klar. □

Bemerkung: k_j beschreibt, wie der relative Fehler in den Eingabedaten x_j verstärkt bzw. abgeschwächt wird.

Definition 1.46
Wir nennen das Problem (f, x_0) **gut konditioniert**, *falls alle k_j $(j = 1, \ldots, m)$ klein sind, sonst* **schlecht konditioniert**

Beispiel 1.47 (Arithmetische Operationen)

(i) $f(x_1, x_2) = x_1 x_2,\ k_1 = \frac{\partial f}{\partial x_1}(x_1, x_2)\frac{x_1}{f(x_1,x_2)} = \frac{x_2 x_1}{x_1 x_2} = 1$

Analog für k_2 ergibt sich ebenfalls 1 $\implies$ Multiplikation ist gut konditioniert.

(ii) Division ist gut konditioniert

(iii) Addition $f(x_1, x_2) = x_1 + x_2$

$$k_j = 1\frac{x_j}{x_1 + x_2} = \frac{x_j}{x_1 + x_2}$$

k_j wird beliebig groß, wenn $x_1 x_2 < 0$ und x_1 und x_2 betragsmäßig gleich groß sind. Das heißt, in diesem Fall ist die Addition schlecht konditioniert, ansonsten ist sie gut konditioniert.

(iv) Subtraktion ist schlecht konditioniert, falls $x_1 x_2 > 0$ und x_1 und x_2 betragsmäßig gleich groß sind.

Beispiel: $(n = 3) x = 0.9995 \quad y = 0.9984 \quad rd(x) = 0.1 \cdot 10^1 \quad rd(y) = 0.998 \cdot 10^0$ Dann gilt für $\circledast = -$

$$x \circledast y = rd(1 - 0.998) - rd(0.2 \cdot 10^{-2}) = 0.2 \cdot 10^{-2}$$

Der absolute Fehler beträgt $x \circledast y - (x - y) = 0.0001$
Der relative Fehler beträgt $\frac{x \circledast y - (x-y)}{(x-y)} = 0.82$

Das Problem wird als Auslöschung bezeichnet.

Bei komplexeren Problemen (etwa $n > 1$) betrachten wir einen anderen Ansatz:

Definition 1.48
Das Problem (f, x_0) ist **wohlgestellt** *in*

$$B_\delta(x_0) := \{x \in U \mid \|x - x_0\| < \delta\}$$

falls es eine Konstante $L_{abs} \geq 0$ gibt, mit $\|f(x) - f(x_0)\| \leq L_{abs} \|x - x_0\|$ $(*)$ für alle $B_\delta(x_0)$. Gibt es keine solche Konstante, so heißt das Problem **schlecht gestellt**.

Sei $L_{abs}(\delta)$ die kleinste Zahl mit der Eigenschaft $(*)$.
Analog: $L_{rel}(\delta)$ als kleinste Zahl mit

$$\frac{\|f(x) - f(x_0)\|}{\|f(x_0)\|} \leq L_{rel}(\delta)\frac{\|x - x_0\|}{\|x_0\|}$$

Definition 1.49 (Konditionszahlen II)
$K_{abs} := \lim_{\delta \searrow 0} L_{abs}(\delta)$ **absolute Konditionszahl** $K_{rel} := \lim_{\delta \searrow 0} L_{rel}(\delta)$ **relative Konditionszahl**

Bemerkung: Falls f differenzierbar, so gilt

$$K_{rel} = \|f'(x_0)\| \frac{\|x_0\|}{\|f(x_0)\|}$$

Beachte: $f'(x_0)$ ist eine Matrix und $|f'(x_0)|$ eine Matrixnorm. K_{rel} hängt von der Wahl der Normen ab.

Beispiel 1.50 (Konditionierung eines LGS)
Zu lösen ist $Ax = b$, d.h. $f(b) = A^{-1}b$ und $f'(x) = A^{-1}$
$\implies K_{abs} = \left\|A^{-1}\right\|$, $K_{rel} = \left\|A^{-1}\right\| \frac{\|b\|}{\|A^{-1}b\|} = \left\|A^{-1}\right\| \frac{\|Ax\|}{\|x\|}$
$\leq \frac{\left\|A^{-1}\right\| \cdot \|A\| \cdot \|x\|}{\|x\|} = \left\|A^{-1}\right\| \cdot \|A\| =: cond(A)$

Beachte, dass es existiert ein $x \in \mathbb{R}^m$ mit $\|Ax\| = \|A\| \, \|x\|$, d.h. $cond(A)$ ist eine gute Abschätzung für die Konditionierung vom Problem (f, b)

Mit A wie in Beispil 1.42 gilt: $cond(A) = \left\|A^{-1}\right\| \|A\| \approx 10^9$, da $A^{-1} = \frac{1}{\det A} A$ und $\det A = \frac{1}{100.000.000}$ (schlechte Konditionierung)

Kapitel 2

Lineare Gleichungssysteme

Wir werden in diesem Kapitel Probleme der Form

$$Ax = b$$

betrachten, wobei $A \in \mathbb{R}^{n \times n}$ und $x, b \in \mathbb{R}^n$. Es gibt im Wesentlichen 2 Klassen von Verfahren

1. Direktes Verfahren
2. Iteratives Verfahren

Aus der Schule (und den Lineare Algebra-Vorlesungen) ist uns das direkte Verfahren am meisten bekannt, das Gaußsche Eliminationsverfahren. Für kleine Gleichungssysteme eignet sich dieses Verfahren, jedoch wenn man mehr als 1000 Zeilen und Spalten in einer Matrix hat, dann ist dieses Verfahren sehr ineffizient, weil das Verfahren eine Laufzeit von $n!$ hat. Aus diesem Grund werden wir andere Verfahren kennenlernen, mit denen man schneller ans Ziel kommen kann.

Wir werden Probleme solcher Art behandeln:

(A) Geg: $A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n$
Ges: $x \in \mathbb{R}^n$ mit $Ax = b$ (falls es eine Lösung existiert)

(B) Geg: $A \in \mathbb{R}^{n \times n}, b_1, \ldots, b_l \in \mathbb{R}^n$
Ges: $x_i \in \mathbb{R}^n$ mit $Ax_i = b_i$ $(i = 1, \ldots, l)$ (falls es eine Lösung existiert)

(C) Geg: $A \in \mathbb{R}^{n \times n}$
Ges: A^{-1} (falls es eine Lösung existiert)

Es sind äquivalent

(i) $\exists!\ x \in \mathbb{R}^n :\ Ax = b$

(ii) $Ax = 0 \iff x = 0$

(iii) $\det(A) \neq 0$

(iv) 0 ist kein Eigenwert von A

(v) A ist regulär, d.h. $\exists\ B \in \mathbb{R}^{n \times n}$ mit $AB = BA = E_n$. Dabei ist $B = A^{-1}$ und $x = A^{-1}b$ ist die eindeutige Lösung von $Ax = b$

Es gilt: Können wir (C) lösen, dann auch (A), (B), da $x = A^{-1}b$
Es gilt: Können wir (B) lösen, dann auch (C)
$A(x_1, \ldots, x_n), x_i$ Spaltenvektor ist Lösung von $A\lambda_i = e_i$ (e_i ist der i. Eigenvektor. (A) $\implies$ (B)
Alle Probleme sind äquivalent, aber es existieren Verfahren, die besonders geeignet für eines dieser Probleme sind.

Verfahren

1. Direkte Verfahren liefern die exakte Lösung x nach endlich vielen Schritten (bis auf Rundungsfehler). Beispiele dafür sind der *Gaußalgorithmus* mit Aufwand $O(n^3)$ und die *Cramersche Regel* mit Aufwand $O(n!)$. Der minimale theoretische Aufwand liegt bei $O(n^2)$, aber es existiert kein Verfahren mit dieser Komplexität.

   Der Vorteil ist, dass A^{-1} mitbestimmt wird und somit ist der Aufwand für (A), (B), (C) ungefähr gleich.

   Der Nachteil ist, dass während der Berechnung keine Näherung möglich ist, d.h. das Resultat steht erst nach Abarbeitung des Algorithmus, also erst nach n Schritten, fest. Je nach Anwendung sind diese Verfahren viel zu aufwändig und deshalb besonders ungeeignet für Problem (A), wenn n sehr groß ist.

2. Iterative Verfahren liefern nach endlich vielen Schritten eine beliebig genaue Approximation der Lösung (bis auf Rundungsfehler).

   Der Vorteil liegt daran, dass man in der Lage ist, die Lösung so genau zu bestimmen, wie es nötig ist. Häufig hat man bereits eine brauchbare Lösung nach $k \ll n$ Schritten

Satz 2.1 (Störungssatz für lineare Gleichungssysteme)
Sei $A \in \mathbb{R}^{n\times n}$ regulär und $\|\cdot\|$ die induzierte Matrixnorm. Sei $\Delta A \in \mathbb{R}^{n\times n}$ gegeben mit $\|\Delta A\| < \frac{1}{\|A^{-1}\|}$ und sei $b \in \mathbb{R}^n$ und $\Delta b \in \mathbb{R}^n$. Dann gilt $A + \Delta A$ regulär und

$$\frac{\|x - \overline{x}\|}{\|x\|} \leq \frac{cond(A)}{1 - cond(A)\frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right)$$

wobei $Ax = b$ und $\overline{x}$ ist die Lösung des Problems $(A + \Delta A)\,\overline{x} = b + \Delta b$

Bemerkung: Kein Einfluss von Rundungsfehlern
Beweis: Siehe Übungsblatt

Bemerkung: $cond(A)$ ist der entscheidende Verstärkungsfaktor für den relativen Fehler.

Definition

$$cond(A) = \|A\| \left\|A^{-1}\right\|$$

2.1 Direkte Verfahren

Idee: Hat A eine einfache Gestalt, so lässt sich x leicht bestimmen.

Beispiel 2.2
Sei $A \in \mathbb{R}^{n\times n}$ eine obere $\triangle$-Matrix, d.h. $a_{ij} = 0$ für $i > j$, also

$$A = \begin{pmatrix} * & \cdots & * \\ & \ddots & \vdots \\ 0 & & * \end{pmatrix}$$

Es gilt $\det(A) = \prod_{i=1}^{n} a_{ii}$, d.h. A ist regulär $\iff$ $a_{ii} \neq 0\ \forall\ i \in \{1, \ldots, n\}$. Ist A regulär, dann ist $Ax = b$ lösbar. $b_i = \sum_{u=1}^{n} a_{iu}x_u = \sum_{u=i}^{n} a_{iu}x_u$.

$i = n : x_i = \frac{b_n}{a_{nn}}$,
$i < n : x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{u=i+1}^{n} a_{iu} x_n \right)$

Frage: Kann eine beliebige reguläre Matrix A so umgeformt werden, dass sie obere $\triangle$-Gestalt hat? D.h gesucht ist $\widetilde{A} \in \mathbb{R}^{n \times n}$ mit oberer $\triangle$-Gestalt, $\widetilde{b} \in \mathbb{R}^n$, so dass $\widetilde{A} \in \mathbb{R}^{n \times n} x = \widetilde{b} \in \mathbb{R}^n$ dieselbe Lösung hat wie $Ax = b$.

(a) **Gaußalgorithmus**

$$(A, b) = \left(A^{(0)}, b^{(0)}\right) = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right)$$

$$\downarrow \text{ mit } A^{(0)}x = b^{(0)} \iff A^{(1)}x = b^{(1)}$$

$$\left(A^{(1)}, b^{(1)}\right) = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{(1)}^{nn} & b_n^{(1)} \end{array}\right)$$

$$\downarrow$$

$$\left(A^{(p-1)}, b^{(p-1)}\right) = \left(\begin{array}{ccccccc|c} a_{11} & \cdots & \cdots & \cdots & \cdots & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \cdots & \cdots & \cdots & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & 0 & \ddots & \cdots & \cdots & \cdots & a_{in}^{(i-1)} & b_i^{(i-1)} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & a_{pp}^{(p-1)} & \cdots & a_{pn}^{(p-1)} & b_p^{(p-1)} \\ \vdots & \vdots & \vdots & \vdots & a_{(p+1)(p+1)}^{(p-1)} & \cdots & a_{(p+1)n}^{(p-1)} & b_{(p+1)}^{(p-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & a_{n(p+1)}^{(p-1)} & \cdots & a_{nn}^{(p-1)} & b_n^{(p-1)} \end{array}\right)$$

$$\downarrow$$

$\left(A^{(n-1)}, b^{(n-1)}\right)$, $A^{(n-1)}$ ist eine obere $\triangle$-Matrix und $Ax = b \iff A^{(n-1)}x = b^{(n-1)}$

Um $\left(A^{(1)}, b^{(1)}\right)$ zu berechnen, wird die i. Zeile $(i = 2, \ldots, n)$ mit $\frac{a_{i1}^{(0)}}{a_{11}^{(0)}} \cdot 1.$ Zeile aufaddiert.

Dies geht so lange $a_{11} \neq 0$ ist, sonst müssen Zeilen vertauscht werden. Es gilt: Weder das Vertauschen von Zeilen noch der Eliminationsschritt verändern die Lösung. Kann in einem Schritt $a_{pp}^{(p-1)} \neq 0$ nicht erreicht werden, nachdem man sämtliche Zeilen vertauscht hat, so heißt das gerade, dass A singulär ist. Das Zeilevertauschen wird als **Pivotisierung** bezeichnet. I.A. wird die Zeile ausgesucht mit

$$\left|a_{kp}^{(p-1)}\right| = \max_{p \leq i \leq n} \left|a_{ip}^{(p-1)}\right|$$

und wird als **Teilpivotisierung/ Spaltenpivotisierung** bezeichnet.

Beispiel 2.3
Sei $A = \begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix}$ und $Ax = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \implies x = \begin{pmatrix} \frac{1}{1-\varepsilon} \\ \frac{1-2\varepsilon}{1-\varepsilon} \end{pmatrix} \approx \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ für $\varepsilon \ll 1$

Ohne Pivotisierung: $\left(A^{(1)}, b^{(1)}\right) = \left(\begin{array}{cc|c} \varepsilon & 1 & 1 \\ 0 & 1-\frac{1}{\varepsilon} & 2-\frac{1}{\varepsilon} \end{array}\right)$

$x_2 = \frac{2-\varepsilon^{-1}}{1-\varepsilon^{-1}} \approx 1$ und $x_1 = (1-x_2)\varepsilon^{-1} \approx 0$, da auf einem Computer $2-\varepsilon^{-1} = -\varepsilon^{-1}, 1-\varepsilon^{-1} = -\varepsilon^{-1}$ berechnet werden.

Mit Pivotisierung: Nach dem Vertasuchen von Zeilen erhalten wir

$$\left(\begin{array}{cc|c} 1 & 1 & 2 \\ \varepsilon & 1 & 1 \end{array}\right)$$

und schließlich nach der Elimination

$$\left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1-\varepsilon & 1-2\varepsilon \end{array}\right)$$

$\implies$ $x_2 = \frac{1-2\varepsilon}{1-\varepsilon} \approx 1$ und $x_1 = 2 - x_2 \approx 1$, da auf einem Computer $1 - 2\varepsilon = 1$ für sehr kleines ε.

Das äquivalente Problem

$$\begin{pmatrix} 1 & \varepsilon^{-1} \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \varepsilon^{-1} \\ 2 \end{pmatrix}$$

kann durch Spaltenpivotisierung nicht gelöst werden. Dafür muss **total pivoting** benutzt werden, d.h. sind die Matrixeinträge sehr unterschiedlich groß, so müssen auch die Spalten vertauscht werden. Das Vertauschen der Spalten ist jedoch umständlich und wird selten angewandt. Man vertauscht die Spalten nur dann, wenn man keine andere Wahl hat.

Algorithmus 2.4 (Gaußverfahren)
Setze $q_i = i \quad (i = 1, \ldots, n)$
$p = 1, \ldots, n-1$

Wähle $j \in \{p, \ldots, n\}$ mit $\left|a_{q_j p}\right| = \max\limits_{k=p,\ldots,n} |a_{q_k p}|$

Vertausche $q_j \longleftrightarrow q_p$ [Spaltenpivotisierung]

Falls $a_{q_p p} = 0 \implies$ Abbruch

$l = \frac{a_{q_k p}}{a_{a_q p}}$ [Multiplikationsfaktor]

$a_{q_k p} = l$ [Speichere l statt eine 0]

Für $j = p+1, \ldots, n$
$a_{q_k j} = a_{q_k j} - l \cdot a_{q_p j}$ [Matrix $A^{(0)}$]

$b_{qk} = b_{qk} - l \cdot b_{q_p}$ [Vektor $b^{(p)}$]

Lösen: Es sei $x_n = b_{q_n}/a_{q_n n}$. Für $k = n-1, \ldots, 1 : x_k = \left(b_{q_k} - \sum\limits_{i=k+1}^{n} a_{q_k i} x_i\right) \Big/ a_{q_k k}$

Bemerkung: Der Aufwand des Algorithmus liegt bei $\frac{1}{3}n^3 + O(n^2)$

(i) Anstelle der Nullen wird der Multiplikationsfaktor l gespeichert

(ii) Die Matrix A und der Vektor b werden überschrieben. Es ist deshalb ratsam, eine Kopie der Vektoren zu machen

(iii) Die Pivotisierung wird als Vektor gespeichert, und die Zeilenvertauschung im Speicher wird nicht durchgeführt.

Formale Berechnung

i. Schritt des Gaußalgorithmus:

1. Zeilenvertauschen $i \longleftrightarrow k,\ i \leq k$
2. Eliminierung $a_{ji} = 0,\ j > i$

Zu 1. sei

$$P_{ik} := \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & 0 & \cdots & 1 & & \leftarrow & i \\ & & & \vdots & \ddots & \vdots & & & \\ & & & 1 & \cdots & 0 & & \leftarrow & j \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{pmatrix}$$

Lemma 2.5

(i) $B = P_{ik}A$ entspricht der Matrix nach der Vertauschung der i. und k. Zeile

(ii) $B = AP_{ik}$ entspricht der Matrix nach der Vertauschung der i. und k. Spalte

(iii) $P_{ik}^2 = E_n$, d.h. $P_{ik}^{-1} = P_{ik}$

Beweis: Durch nachrechnen

Definition 2.6
Eine Matrix $P \in \mathbb{R}^{n\times n}$ heißt **Permutationsmatrix**, *falls P durch Zeilenvertauschungen aus der Einheitsmatrix E_n entsteht.*

Bemerkung: P_{ik} ist eine Permutationsmatrix. Sei P eine Permutationsmatrix $\implies P^2 = E_n$

Zu 2. sei

$$L_i := \left(\begin{array}{ccc|c|cc} 1 & & & 0 & & 0 \\ & \ddots & & \vdots & & \\ & & & 1 & \leftarrow & l_{(i+1),i} \\ & & & \vdots & \ddots & \\ 0 & & & 0 & & 1 \end{array}\right)$$

mit $l_{ji} := \frac{a_{ji}}{a_{ii}},\ j = i+1, \ldots, n$

Lemma 2.7

(i) $B = L_i A = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$, wobei $b_i \in \mathbb{R}^n$ Zeilenvktoren sind

$\implies b_j = a_j \ (j = 1, \ldots, i)$

$b_j = a_j + l_{ji} a_i \ (j = i+1, \ldots, n)$

(ii) $L_i^{-1} = 2E_n - L_i$, also

$$L_i^{-1} := \left(\begin{array}{ccc|c|c} 1 & & & 0 & & 0 \\ & \ddots & & \vdots & & \\ & & & 1 & \leftarrow & l_{i,(i+1)} \\ & & & \vdots & \ddots & \\ 0 & & & 0 & & 1 \end{array}\right)$$

d.h. $B = L_i^{-1} A = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$

$\implies b_j = a_j, \ (j = 1, \ldots, i)$

$b_j = a_j - l_{ji} a_i, \ (j = i+1, \ldots, n)$

Folgerung 2.8
Die Transformation von A auf obere $\triangle$-Gestalt kann geschrieben werden als

$$R = L_{n-1}^{-1} P_{n-1} \cdots \underbrace{L_1^{-1} P_1}_{\text{Pivotisierung } P_{ij},\ i<j} A$$

wobei R eine $\triangle$-Matrix ist.

Satz 2.9 (LR-Zerlegung)
Sei $A \in \mathbb{R}^{n \times n}$ eine reguläre Matrix. Dann gilt:

- Es existiert eine Permutationsmatrix P, eine untere $\triangle$-Matrix L mit Diagonalelementen 1 und eine obere $\triangle$-Matrix R mit
$$PA = LR$$
- Es gilt: $A = LR = MS$, wobei L, M untere $\triangle$-Matrizen mit Diagonalelementen 1 sind, und R, S obere $\triangle$-Matrizen sind $\implies L = M, \ R = S$

Bemerkung: Ist $PA = LR$ gegeben, so kann man

$$Ax = b$$

lösen, indem man

1. $Lz = Pb$ lösen
2. $Rx = z$ lösen

$$\begin{aligned} Ax &= b \iff PAx = Pb \\ &\iff LRx = Pb \\ &\iff LRx = Lz \\ &\iff Rx = z \end{aligned}$$

$\Longrightarrow$ 1. und 2. leicht zu lösen

I.A. wird L in den frei verwendenden Stellen von A gepseichert. Also

$$A^{n-1} := \begin{pmatrix} 1 & & & & R \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ L & & & & 1 \end{pmatrix}$$

L und R benötigen n^2 Speicherstellen (zusammen). Nicht nur R wird erzeugt sondern auch L

Es gilt $P^{-1} = P$

Beweis: (von 2.9)
Nach dem Gaußalgorithmus gilt: $R = L_{n-1}^{-1} P_{n-1}, \dots L_1^{-1} P_1 A$

wobei R obere $\triangle$-Matrix ist $\Longrightarrow$ $P_1 L_1 \dots P_{n-1} L_{n-1} R = A$

Setze: $P = P_{n-1} \dots P_1$, so ist P eine Permutationsmatrix und $L := P^{-1} P_1 L_1 \dots P_{n-1} L_{n-1}$

$\Longrightarrow$ $P^{-1} LR = A$ $\Longrightarrow$ $LR = PA$

z.z ist: L untere $\triangle$-Matrix und

$$\begin{aligned} L &= P^{-1} P_1 L_1 \dots P_{n-1} L_{n-1} = P_{n-1} \dots \underbrace{P_1 R}_{=E_n} L_1 P_2 \dots P_{n-1} L_{n-1} \\ &= P_{n-1} \dots P_2 L_1 P_2 \dots P_{n-1} L_{n-1} \end{aligned}$$

Bemerkung:

$$\widetilde{L}_p := L_1 \dots L_p = \begin{pmatrix} 1 & & & & & & 0 \\ * & \ddots & & & & & \\ \vdots & \ddots & \ddots & & & & \\ \vdots & & * & \ddots & & & \\ \vdots & & \vdots & 0 & \ddots & & \\ \vdots & & \vdots & \vdots & \ddots & \ddots & \\ * & \cdots & * & 0 & \cdots & 0 & 1 \end{pmatrix}$$

$\Longrightarrow$ $P_q \widetilde{L}_p P_q$ (mit $q > p$) hat dieselbe Gestalt, da $P_q = P_{qk},\ q < k$

Also
$$\begin{aligned} L &= P_{n-1}\dots P_3 L_2 L_1 L_2 P_3 \dots P_{n-1} L_{n-1} \\ &= P_{n-1}\dots P_3 \widetilde{L}_1 L_2 P_3 \dots P_{n-1} L_{n-1} \\ &= P_{n-1}\dots P_4 \widetilde{L}_1 \widetilde{L}_2 L_3 P_4 \dots P_{n-1} L_{n-1} \\ &\vdots \\ &= \widetilde{L}_1 \dots \widetilde{L}_n L_{n-1} \end{aligned}$$
und ist untere $\triangle$-Matrix

Zu 2.
$LR = MS$. Es gilt: L^{-1} hat ebenfalls untere $\triangle$-Gestalt (betrachte $L^{-1}L = E_n$). Analog:

S^{-1} ist eine obere $\triangle$-Matrix

$\Longrightarrow$ RS^{-1} obere $\triangle$-Gestalt
$\Longrightarrow$ RS^{-1} obere $\triangle$-Gestalt, $L^{-1}M$ ist untere $\triangle$-Matrix
Also gilt $LR = MS \implies RS^{-1} = L^{-1}M = E_n \implies L = S \wedge L = M$, wobei R obere $\triangle$-Matrix ist und L eine untere $\triangle$-Matrix ist. $\square$

Weiter Anwendungen der LR-Zerlegung

1. Determinantenberechnung einer Matrix A. Hat A obere/untere $\triangle$-Gestalt, dann gilt
$$\det(A) = \prod_{i=1}^{n} a_{ii}$$
und
$$R = L_{n-1}^{-1} P_{n-1} \dots L_1^{-1} P_1 A$$
$$\implies \det R = \det\left(L_{n-1}^{-1} P_{n-1} \dots L_1^{-1} P_1 A\right) = \det(L_{n-1}^{-1}) \det(P_{n-1}) \dots \det(L_1^{-1}) \det(P_1) \det(A)$$
Es gilt $\det\left(L_i^{-1}\right) = 1$ und
$$\det(P_i) = \det(P_{ik}) = \begin{cases} 1 & : \quad i = k \\ -1 & : \quad i \neq k \end{cases}$$
$\Longrightarrow$
$$\det(A) = \begin{cases} \det(R) & : \text{ gerade Anzahl von Zeilenvektoren} \\ -\det(R) & : \text{ ungerade Anzahl von Zeilenvektoren} \end{cases}$$
$$= \begin{cases} \prod_{i=1}^{n} r_{ii} & : \text{ gerade Anzahl von Zeilenvektoren} \\ -\prod_{i=1}^{n} r_{ii} & : \text{ ungerade Anzahl von Zeilenvektoren} \end{cases}$$

2. Bestimmung von $\text{Rang}(A) = \sharp$ der linear unabhängigen Zeilenvektoren bei einer nicht unbedingt quadratischen Matrix.

   Ist im p. Schritt $a_{pp}^{(p)} = 0$, so müssen Zeilen und Splaten vertauscht werden. Ist dies nicht möglich
$$\implies A^{(p)} = \left(\begin{array}{c|c} * & * \\ \hline 0 & 0 \end{array}\right)$$

Die ersten p. Spaltenvektoren sind linear unabhängig, aber alle weiteren Spaltenverktoren sind linear abhängig $\implies$ $\text{Rang}\left(A^{(0)}\right) = \text{Rang}(A) = p$

Beispiel: Aufgrund von Rundungsfehlern kann dieses Verfahren das falsche Ergebnis liefern:

Für $\varepsilon \neq 0 \implies \det(A_\varepsilon) = \varepsilon$

$$A_\varepsilon = \begin{pmatrix} \frac{1}{\varepsilon} & 1 & 0 \\ -1 & 1 & \varepsilon \\ 0 & 1 & \varepsilon \end{pmatrix}$$

$\implies \text{Rang}(A_\varepsilon) = 3$

Nach Gaußalgorithmus: $A_\varepsilon \longrightarrow A_\varepsilon^{(1)} = \begin{pmatrix} \frac{1}{\varepsilon} & 1 & 0 \\ 0 & 1+\varepsilon & \varepsilon \\ 0 & 1 & \varepsilon \end{pmatrix}$

$$\implies A_\varepsilon^{(2)} = \begin{pmatrix} \frac{1}{\varepsilon} & 1 & 0 \\ 0 & 1+\varepsilon & \varepsilon \\ 0 & 0 & \varepsilon - \frac{\varepsilon}{1+\varepsilon} \end{pmatrix}$$

Und $\varepsilon - \frac{\varepsilon}{1+\varepsilon} = \frac{\varepsilon^2}{1+\varepsilon} \neq 0$

Der Gaußalgorithmus wurde hier nicht abgebrochen.

Ist $\varepsilon < eps$ (Maschinengenauigkeit)

$\implies 1 + \varepsilon = 1$ (auf einem Computer, wegen Rundungsfehlern, siehe Seite 11)

$$\varepsilon = b^{-30} = Mb^0$$

Dann gilt

$$\widetilde{A}_\varepsilon^1 = \begin{pmatrix} \frac{1}{\varepsilon} & 1 & 0 \\ 0 & 1 & \varepsilon \\ 0 & 1 & \varepsilon \end{pmatrix}$$

wegen Rundungen, also

$$\widetilde{A}_\varepsilon^2 = \begin{pmatrix} \frac{1}{\varepsilon} & 1 & 0 \\ 0 & 1 & \varepsilon \\ 0 & 0 & 0 \end{pmatrix}$$

$\implies \text{Rang}(\widetilde{A}_\varepsilon^2) = 2 \neq \text{Rang}(A)$

3. Berechnung der Umkehrmatrix A^{-1} einer Matrix A

a) 1. Ansatz: Sei e_i der i. Einheitsvektor. Löse $Ax^{(i)} = e_i$ für $i = 1, \ldots, n \implies A^{-1} = (x_1, \ldots, x_n)$ mittels LR-Zerlegung mit $Lz^{(i)} = Pe_i$ und $Rx^{(i)} = z^{(i)}$

2. Ansatz:

$$\left(\begin{array}{c|ccc} & 1 & & 0 \\ A & & \ddots & \\ & 0 & & 1 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|ccc} * & \cdots & * & * & & 0 \\ & \ddots & \vdots & \vdots & \ddots & \\ 0 & & * & * & \cdots & * \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} a_1 & & 0 & \\ & \ddots & & * \\ 0 & & a_n & \end{array}\right)$$

Vorwärtselimination Rückwärtselimination

b) **Gauß-Jordan Verfahren**

Die Idee des Verfahren ist folgende: Ist $a_{pq} \neq 0$, so kann die Gleichung nach x_q aufgelöst werden, wobei $x_q = -\frac{a_{p1}}{a_{pq}}x_1 - \ldots - \frac{a_{pq-1}}{a_{pq}}b_q - \frac{a_{pq+1}}{a_{pq}}x_{q+1} - \ldots - \frac{a_{pn}}{a_{pq}}x_n$.

Durch Einsetzung von x_q in die anderen Gleichungen ($j \neq p$)

$$\sum_{k=1}^{q-1}\left[a_{jk} - \frac{a_{jq}a_{pk}}{a_{pq}}\right]x_k + \frac{a_{iq}}{a_{pq}}b_q + \sum_{k=q+1}^{n}\left[a_{jk} - \frac{a_{jq}a_{jk}}{a_{pq}}\right]x_k = b_j$$

D.h. $\widetilde{A}\begin{pmatrix} x_1 \\ \vdots \\ b_q \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ x_q \\ \vdots \\ b_n \end{pmatrix}$

Kann dieser Schritt n-mal durchgeführt werden

$$\widetilde{A}\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \implies \widetilde{A} = A^{-1}$$

Das ist ein Algorithmus ohne Pivotisierung, d.h. $a_{ii} \neq 0$

c) **Cholesky Verfahren**

Sei $A \in \mathbb{R}^{n\times n}$ eine symmetrische positive definite Matrix. Es existiert ein $L \in \mathbb{R}^{n\times n}$ mit unterer $\triangle$-Gestalt, so dass gilt

$$A = LL^T$$

Idee: Simultane Elimination der Zeieln/Spalten

$$\begin{pmatrix} x_0 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \widetilde{A} & \\ 0 & & & \end{pmatrix} \longrightarrow \begin{pmatrix} * & & 0 \\ & \ddots & \\ 0 & & * \end{pmatrix}$$

Dieser Algorithmus hat einen halben Aufwand im Vergeleich mit dem Gauß-Algorithmus.

Beispiel: Sei A eine Tridiagonalmatrix

$$A = \begin{pmatrix} \alpha_1 & \gamma_1 & & 0 \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \gamma_{n-1} \\ 0 & & \beta_{n-1} & \alpha_n \end{pmatrix}$$

A kann zuerst mittels LR-Zerlegung zerlegt werden und die Zerlegung kann in Abhängigkeit von α, β, γ hingeschrieben werden.

2.2 Lösung überbestimmter LGS (Ausgleichungsrechnung)

Problem: Gegeben sind m Messdaten (zum Beispiel Zeit und Konzentration) $(x_1, y_1), \ldots, (x_m, y_m)$ und Funktionen $p_1, \ldots, p_n,\ (n, m \in \mathbb{N},\ n \leq n)$

Gesucht: Linear Kombination $u(x) = \sum_{i=1}^{n} u_i p_i(x)$, welche die mittlere Abweichung minimier, also:

$$\triangle_2 := \left(\sum_{i=1}^{m} (u(x_i) - y_i)^2 \right)^{\frac{1}{2}} = \inf_{u_1,\ldots,u_n \in \mathbb{R}} \left(\sum_{j=1}^{m} \left(\sum_{i=1}^{n} (u_i p_i(x_i)) - y_i \right)^2 \right)^{\frac{1}{2}}$$

Dieses Problem wird als das **Gaußsche Ausgleichsproblem** bezeichnet, wobei u der Lösung der kleinsten Quadrate (least squares) darstellt.

Bemerkung: Das **Tschebyscheffsche Ausgleichsproblem**, bei dem

$$\triangle_\infty := \min_{1 \leq i \leq n} |p(x_i) - y_i|$$

ist deutlich schwerer.

Setze:
$c = (c_1, \ldots, c_n)^\top \in \mathbb{R}^n$ (gesuchte Werte)
$x = (x_1, \ldots, x_m)^\top \in \mathbb{R}^m,\ y = (y_1, \ldots, y_m)^\top \in \mathbb{R}^m$
$A = (a_{ij}) \in \mathbb{R}^{m \times n}$ mit $(a_{ij}) = u_j(x_i)$

Dann ist das Ausgleichsproblem (AGP) äquivalent mit dem Minimieren des Funktionals $F(c) := \|Ac - y\|_2$

Bemerkung: Sind $m = n,\ p_1, \ldots, p_n$ linear unabhängig, $x_1, \ldots, x_m$ paarweise verschieden $\implies$ A ist regulär und $c = A^{-1}y$ ist das gesuchte Minimum.

Satz 2.10
Sei $A \in \mathbb{R}^{m \times n},\ b \in \mathbb{R}^m\ (n \leq m)$ gegeben. Dann existiert mindestens eine Lösung $\overline{x} \in \mathbb{R}^n$ des Ausgleichsproblems $(Ax = b)$. Dies ist äquvalent dazu, dass $\overline{x}$ die **Normgleichung**

$$A^\top A x = A^\top b$$

löst. ist $\text{Rang}(A) = n$ (d.h. maximal) $\implies$ die Lösung ist eindeutig bestimmt, andernfalls ist jede weitere Lösung von der folgenden Gestalt:

$$x = \overline{x} + y$$

mit $y \in \text{Kern}(A)$. In diesem Fall wird i.a. Lösungen X_A gesucht mit minimaler 2-Norm, d.h.

$$\|X_A\| := \inf \left\{ \|x_2\| \;\middle|\; x \text{ Lösung des Ausgleichsproblems} \right\}$$

Lemma 2.11
$A^\top A \in \mathbb{R}^{n \times m}$ (bzw. $AA^\top \in \mathbb{R}^{m \times n}$) erfüllt

(i) $A^\top A$ ist symmetrisch

(ii) $A^\top A$ ist positiv semidefinit und falls $\text{Rang}(A)$ sogar positiv definit.

(iii) $\text{Kern}(A^\top A) = \text{Kern}(A)$

(iv) $\mathbb{R}^m = \text{Bld}(A) \oplus \text{Kern}(A^\top)$

(v) $A^\top A$ und $AA^\top$ haben dieselben (positiven, reellen) Eigenwerte

(vi) $r = \text{Rang}(A) = \text{Rang}(A^\top A) = \text{Rang}(AA^\top) = \text{Rang}(A^\top) = \left|\left\{\lambda > 0 \;\middle|\; \lambda \text{ EW von } A^\top A\right\}\right|$

Beweis: (Lemma 2.11(iv))

Es gilt $\mathbb{R}^m = \text{Bld}(A) \oplus \text{Bld}(A)^\perp$
z.z. ist $\text{Bld}(A)^\perp = \text{Kern}(A^\top)$

sei $y \in \text{Bld}(A)^\perp$, d.h. $\forall\, z \in \text{Bld}(A) : \langle y, z\rangle = 0$
$\iff \forall\, x \in \mathbb{R}^n : \langle y, Ax\rangle = 0 \iff \forall\, x \in \mathbb{R}^n : \left\langle A^\top y, x\right\rangle = 0$
$\iff A^\top y = 0 \iff y \in \text{Kern}(A^\top)$ □

Beweis: (Satz 2.10)

Wir zeigen zunächst die Äquivalenz des Minimierungsproblems mit dem Lösen der Normalengleichung. Sei $\overline{x}$ Lösung von $A^\top Ax = A^\top b$

$$\begin{aligned}
\implies \|b - Ax\|_2^2 &= \|b - A\overline{x} + A(\overline{x} - x)\|_2^2 \\
&= \langle b - A\overline{x} + A(\overline{x} - x), b - A\overline{x} + A(\overline{x} - x)\rangle \\
&= \langle b - A\overline{x}, b - A\overline{x}\rangle + 2\langle b - A\overline{x}, A(\overline{x} - x)\rangle + \langle A(\overline{x} - x), A(\overline{x} - x)\rangle \\
&= \|b - A\overline{x}\|_2^2 + \|A(\overline{x} - x)\|_2^2 + 2\left\langle A^\top(b - A\overline{x}9, \overline{x} - x\right\rangle \\
&\geq \|b - A\overline{x}\|_2^2 \\
\implies \forall\, x \in \mathbb{R}^n : \|b - A\overline{x}\|_2 &\leq \|b - Ax\|_2
\end{aligned}$$

Sei $\overline{x}$ eine Lösung des Minimierungsproblems.

$$\begin{aligned}
0 &= \tfrac{\partial}{\partial x_i}(F(x))^2_{|x=\overline{x}} = \tfrac{\partial}{\partial x_i}\left(\sum_{i=1}^m\left(\sum_{k=1}^n a_{jk}x_k - b_j\right)^2\right)_{|x=\overline{x}} \\
&= \sum_{j=1}^n a_{ji} 2\left(\sum_{k=1}^n a_{jk}\overline{x}_k - b_j\right) = 2\Big(\sum_{i=1}^m a_{ji}\sum_{k=1}^n a_{jk}\overline{x}_k - \underbrace{\sum_{j=1}^m a_{ji}b_j}_{a_{ij}^\top}\Big) \\
&= 2(A^\top X\overline{x} - A^\top b) \\
&\implies A^\top A\overline{x} = A^\top b \implies \overline{x} \text{ Lösung der Normalgleichung}
\end{aligned}$$

Das heißt, die Existenz einer Lösung des AGPs kann gezeigt werden durch das Lösen der Normalgleichung. Zur Lösung der Normalgleichung: $b \in \mathbb{R}^m = \text{Bld}(A) \oplus \text{Kern}(A^\top) \implies b = s + r$ mit $s \in \text{Bld}(A)$, $r \in \text{Kern}(A^\top)$. Zu $s \in \text{Bld}(A)$ existiert ein $\overline{x} \in \mathbb{R}^n$ mit $A\overline{x} = s$

Es gilt: $A^\top A\overline{x} = A^\top s + A^\top r = A^\top(r + s) = A^\top b$, d.h. $\overline{x}$ ist Lösung der Normalgleichung.

Sei $\text{Rang}(A) = n \implies A^\top A$ positiv definit (Lemma 2.11(ii))

$\implies A^\top A$ ist regulär $\implies \overline{x} = (A^\top A)^{-1}A^\top b$ ist eindeutige Lösung der Normalgleichung.

Sei $\text{Rang}(A) < n$. Seien x_1, x_2 Lösungen der Normalgleichung: $b = Ax_1 + (b - Ax_1) \in \text{Bld}(A) \oplus \text{Kern}(A^\top)$, da x_1 Lösung der Normalgleichung.

Da die Zerlegung $\mathbb{R}^m = \text{Bld}(A) \oplus \text{Kern}(A^\top)$ eindeutig $\implies Ax_1 = s = A\overline{x} \implies A(x - \overline{x}) = 0 \implies x - \overline{x} \in \text{Kern}(A)$

Es sei

$$K := \left\{ x \in \mathbb{R}^n \;\middle|\; x \text{ Lösung des AGPs und } \|x\|_2 \leq \|\overline{x}\|_2 \right\}$$

$\Longrightarrow$ K kompakt und da $\|\cdot\|_2$ stetig ist, nimmt sie ihr Minimun auf K an. Sei x_A mit

$$\|x_A\|_2 = \inf \left\{ \|x_2\| \;\middle|\; x \in K \right\} =: \rho$$

Seien $x_1, x_2 \in K$ mit $\|x_1\|_2 = \|x_2\|_2 = \rho$

$$\begin{aligned}
&\Longrightarrow \tfrac{x_1+x_2}{2} \in K \text{ und } \rho \leq \left\| \tfrac{x_1+x_2}{2} \right\|_2 \leq \tfrac{1}{2}\|x_1\|_2 + \tfrac{1}{2}\|x_2\|_2 = \rho \\
&\Longrightarrow \left\| \tfrac{x_1+x_2}{2} \right\|_2 = \rho \\
&\Longrightarrow \rho^2 \quad = \quad \left\| \tfrac{x_1+x_2}{2} \right\|_2^2 = \tfrac{1}{4}\langle x_1 + x_2, x_1 + x_2 \rangle \\
&\qquad\quad = \quad \tfrac{1}{4}\left(\|x_1\|_2^2 + 2\langle x_1, x_2 \rangle + \|x_2\|_2^2 \right) \\
&\qquad\quad = \quad \tfrac{1}{4}\left(\rho^2 + 2\langle x_1, x_2 \rangle + \rho^2 \right) = \tfrac{1}{2}\rho^2 + \tfrac{1}{2}\langle x_1, x_2 \rangle \\
&\qquad\quad \Longrightarrow \quad \langle x_1, x_2 \rangle = \rho^2 \\
&\qquad\quad \Longrightarrow \quad \|x_1 - x_2\|_2^2 = \|x_1\|_2^2 - 2\langle x_1, x_2 \rangle + \|x_2\|_2^2 = 2\rho^2 - 2\rho^2 = 0 \\
&\qquad\quad \Longrightarrow \quad x_1 = x_2 \qquad \square
\end{aligned}$$

Beispiel 2.12

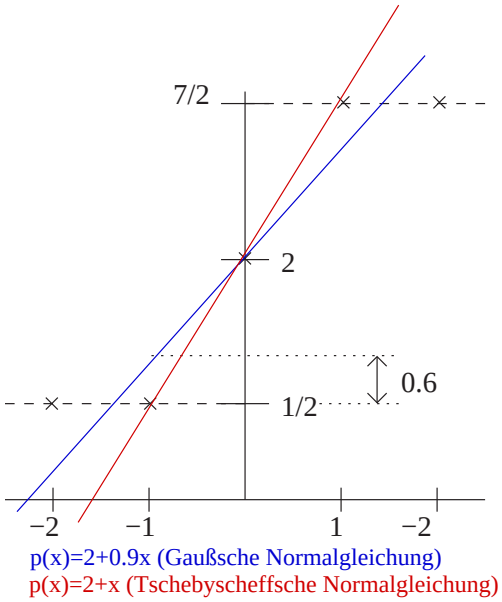


Abbildung 2.1: Ausgleichungsgerade

Gegeben: Messdaten:

x_i	-2	-1	0	1	2
y_i	1/2	1/2	2	7/2	7/2

Gesucht: Ausgleichungsgerade (linear fit[1]) $p(x) = bx + a$ mit

$$\left(\sum_{j=1}^{5} (bx_j + a - y_i)^2 \right)^{\frac{1}{2}} = \min_{(\hat{a},\hat{b}) \in \mathbb{R}^2} \left(\sum_{j=1}^{5} \left(\hat{b}x_j + \hat{a} - y_i \right)^2 \right)^{\frac{1}{2}} \leq \left(\sum_{j=1}^{5} \left(\hat{b}x_j + \hat{a} - y_i \right)^2 \right)^{\frac{1}{2}} \quad \forall\, \hat{b}, \hat{a} \in \mathbb{R}$$

[1] englischer Ausdruck der Ausgleichungsgerade

$\Longrightarrow$ $(a,b)^\top$ löst die Normalgleichung

$$A = \begin{pmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix};\ c = \begin{pmatrix} 1/2 \\ 1/2 \\ 2 \\ 7/2 \\ 7/2 \end{pmatrix}$$

wobei $A^\top A = \begin{pmatrix} 5 & 0 \\ 0 & 10 \end{pmatrix}$ und $A^\top c = \begin{pmatrix} 10 \\ 9 \end{pmatrix}$

$\Longrightarrow$ $\mathrm{Rang}(A) = 2$ $\Longrightarrow$ Normalgleichung eindeutig lösbar. $A^\top A \begin{pmatrix} a \\ b \end{pmatrix} = A^\top c \Longrightarrow \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 2 \\ 0.9 \end{pmatrix}$,

d.h. $p(x) = 2 + 0.9x$

Abweichung (Fehler in 2-Norm): $\triangle_2 = \sqrt{0.9} < 1$, $\triangle_\infty = 0.6$

Die Lösung des Tschebyscheffs-Problems ist gegeben durch $p(x) = 2 + x$ und $\triangle_2 = 1;\ \triangle_\infty = \frac{1}{2}$

Bemerkung: Physikalisch könnte gelten: z.B: $p(x) = \frac{a}{1+bx}$ (d.h. Zusammenhang nicht linear).

Setze $\hat{p}(x) = \frac{1}{p(x)} = \frac{1}{a} + \frac{b}{a}x = \hat{a} + \hat{b}x$, und jetzt kann die lineare Ausgleichungsgerade berechnet werden.

Bemerkung: Das Ausgleichsproblem kann durch Lösen der Normalengleichung (etwa LR-Zelegung) behandelt werden. Allerdings ist dies numerisch nicht unbedingt der beste Zugang, da $cond(A^\top A)$ sehr viel größer als $cond(A)$ ist. Beispiel: gilt $\mathrm{Rang}(A) = n,\ A \in \mathbb{R}^{n\times n}$, dann ist $cond(A^\top A) \sim cond(A)^2$.
Idee: Sei $\mathrm{Rang}(A) = n,\ A \in \mathbb{R}^{m\times n}$

Gegeben: $A = QR,\ R$ obere $\triangle$-Matrix im $\mathbb{R}^{m\times n},\ Q \in \mathbb{R}^{m\times m}$ orthogonale Matrix, d.h. $Q^{-1} = Q^\top$

$$A = Q \left(\begin{array}{ccc} * & & * \\ & \ddots & \\ 0 & & * \\ \hline & 0 & \end{array}\right) = Q \begin{pmatrix} \widetilde{R} \\ \hline 0 \end{pmatrix}$$

wobei $\widetilde{R}$ reguläre obere $\triangle$-Matrix ist.

$$A^\top A = (QR)^\top QR = T^\top Q^\top QR = R^\top R$$
$$A^\top b = R^\top Q^\top b$$

d.h. $A^\top A\overline{x} = A^\top b \iff R^\top R\overline{x} = R^\top Q^\top b$

Setze: $c := Q^\top b = \begin{pmatrix} c_1 \\ \vdots \\ c_n \\ c_{n+1} \\ \vdots \\ c_m \end{pmatrix};\ \widetilde{c} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \in \mathbb{R}^n$

$$R^\top = \Big(\underbrace{\widetilde{R}^\top}_{n} \Big|\ \underbrace{0}_{m-n}\Big) \Longrightarrow R^\top c = \begin{pmatrix} \widetilde{R}^\top \widetilde{c} \\ 0 \end{pmatrix}$$

Es gilt: $\widetilde{R}^\top$ regulär und sei $\overline{x} \in \mathbb{R}^n$ mit $\widetilde{R}\overline{x} = \widetilde{c}$ (leicht zu lösenendes LGS, da $\widetilde{R}$ obere $\triangle$-Matrix).

$$\Longrightarrow \quad R\overline{x} = \begin{pmatrix} \widetilde{c} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \Longrightarrow \quad R^\top R\overline{x} = R^\top c = R^\top Q^\top b$$

$\Longrightarrow$ $\overline{x}$ ist eine Lösung der Normalgleichung.

d) **QR-Zerlegung nach Householder** [2]

Sei $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ mit $\mathrm{Rang}(A) = n$

Ziel: Finde obere $\triangle$-Matrix $R \in \mathbb{R}^{m\times n}$ und eine orthogonale Matrix $Q \in \mathbb{R}^{m\times m}$ mit $A = QR$

Definition 2.13
Seien $u, v \in \mathbb{R}^m$ (Spaltenvektoren)

Dann heißt die Matrix

$$A = uv^\top = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} (v_1, \ldots, v_m)$$

das **dyadische Produkt** *von u, v.*

$A \in \mathbb{R}^{m\times m}$ und $a_{ij} = u_i v_j$ $(1 \leq i, j \leq m)$

Bemerkung: $\langle u, v\rangle = u^\top v = (u_1, \ldots, u_n) \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} \in \mathbb{R}$

Folgerung 2.14
$A = uv^\top$ und $w \in \mathbb{R}^m$

(i) $Aw = \langle v, w\rangle\, u$

(ii) $A^2 = \langle u, v\rangle\, A$

Beweis:

(i) $(Aw)_i = \sum_{k=1}^{m} u_i v_k w_k = \langle v, w\rangle\, u_i$

(ii) $(A^2)_{ij} = (AA)_{ij} = \sum_{k=1}^{m} u_i v_k u_k v_j = \left(\sum_{k=1}^{m} v_k u_k\right) u_i v_j = \langle v, u\rangle\, (A)_{ij}$ $\square$

[2] Das ist die Forsetzung der Verfahren ab Seite 26

Definition 2.15
$v \in \mathbb{R}^m,\ v \neq 0$. Die Matrix $H(v) = \mathbb{I} - 2\frac{vv^\top}{\|v\|_2^2}$ heißt **Householder Matrix**.

Wir setzen $H(0) = \mathbb{I}$

Folgerung 2.16
$v \in \mathbb{R}^m$. Es gilt:

(i) $H(v)$ ist symmetrisch

(ii) $H(v)$ ist orthogonal, d.h. $H(v) = H(v)^\top = H(v)^{-1}$

Beweis:

(i) $H(v)_{ij} = \delta_{ij} - 2\frac{v_i v_j}{\|v\|_2^2}$ (δ_{ij} ist das Kronecker Symbol), also

$$\delta_{ij} - 2\frac{v_i v_j}{\|v\|_2^2} = H(v)^\top v_j$$

(ii) Zu zeigen: $H(v)H(v) = \mathbb{I}$, wegen (i) ist also zu zeigen ist $H(v)^2 = \mathbb{I}$

$$\begin{aligned} H(v)^2 &= \left(\mathbb{I} - 2\tfrac{vv^\top}{\|v\|_2^2}\right)\left(\mathbb{I} - 2\tfrac{vv^\top}{\|v\|_2^2}\right) = \mathbb{I} - 4\tfrac{vv^\top}{\|v\|_2^2} + 4\tfrac{(vv^\top)^2}{\|v\|_2^4} \\ &= \mathbb{I} - \tfrac{4}{\|v\|_2^2}\left(vv^\top - \tfrac{(vv^\top)^2}{\|v\|_2^2}\right) \\ &= \mathbb{I} - \tfrac{4}{\|v\|_2^2}\left(vv^\top - \tfrac{\langle v, v^\top\rangle vv^\top}{\|v\|_2^2}\right) \text{ (wegen 2.14(ii))} \\ &= \mathbb{I} \quad \square \end{aligned}$$

Satz 2.17
Sei $a \in \mathbb{R}^m$, setze $u = a \pm \|a\|_2 e_k \in \mathbb{R}^m$ $(1 \leq k \leq m)$. Dann gilt

$$H(u)a = \mp \|a\|_2 e_k$$

Beweis: Im Fall $u = 0 \implies 0 = \mp \|a\|_2 e_k$ (wegen Definition von u)

$\implies H(u)a = \mathbb{I}a = \mp \|a\|_2 e_k$

Sei nun $u \neq 0$, etwa $u = a - \|a\|_2 e_k$

$\implies H(u) = \mathbb{I} - \frac{uu^\top}{h}$ mit $h = \frac{1}{2}\|u\|_2^2$

$$\begin{aligned} h &= \tfrac{1}{2}\langle a - \|a\|_2 e_k, a - \|a\|_2 e_k\rangle = \tfrac{1}{2}\left(\|a\|_2^2 - 2\|a\|_2\langle a, e_k\rangle + \|a\|_2^2\right) \\ &= \|a\|_2^2 - \|a\|_2 e_k \end{aligned}$$

$$\begin{aligned} H(u)a &= \left(\mathbb{I} - \tfrac{1}{h}uu^\top\right)a = a - \tfrac{1}{h}(uu^\top)a \\ &= a - \tfrac{1}{h}\langle u, a\rangle u \text{ Folgerung 2.14(i)} \\ &= a - \tfrac{1}{h}\langle a - \|a\|_2 e_k, a\rangle u \\ &= a - \tfrac{1}{h}\left(\|a\|_2^2 - \|a\|_2\langle e_k, a\rangle\right)u \\ &= a - u = \|a\|_2 e_k \text{ wegen Definition von } u \quad \square \end{aligned}$$

Verfahren:

$A \in \mathbb{R}^{m\times n},\ A = (a_1, \ldots, a_n) = (a_1^{(0)}, \ldots, a_n^{(0)})$

Schritt 1

$a^{(0)} = (a_1^{(0)}) - \left\|a_1^{(0)}\right\|_2 e_1 \in \mathbb{R}^m,\ Q_1 = H(u^{(0)})$

$$\Longrightarrow\quad Q_1 A = R^{(1)} = \begin{pmatrix} * & \cdots & * \\ 0 & & \\ \vdots & A^{(1)} & \\ 0 & & \end{pmatrix} \text{ mit } A^{(1)} \in \mathbb{R}^{m-1\times n-1} = \left(a_2^{(1)}, \ldots, a_n^{(1)}\right)$$

Schritt 2

$$u^{(1)} = a_2^{(1)} - \left\|a_2^{(1)}\right\| e_1 \in \mathbb{R}^{m-1},\ Q_2 := \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & H(u^{(1)}) & \\ 0 & & & \end{pmatrix}$$

$$\Longrightarrow\quad Q_2 \in \mathbb{R}^{m\times m}, H(u^{(1)}) \in \mathbb{R}^{m-1\times m-1}$$

$$\Longrightarrow\quad Q_2 Q_1 A = Q_2 R^{(1)} = \begin{pmatrix} * & \cdots & \cdots & 0 \\ 0 & * & \cdots & * \\ \vdots & 0 & & \\ \vdots & \vdots & A^{(2)} & \\ 0 & 0 & & \end{pmatrix}. \text{ Nach } n \text{ Schritten}$$

$$Q_n \cdots Q_1 A = R^{(n)} = \left(\begin{array}{ccc} * & \cdots & * \\ & \ddots & \\ 0 & & * \\ \hline & 0 & \end{array}\right) = R$$

$Q = Q_1 \cdots Q_n \quad\Longrightarrow\quad Q$ ist orthogonal, da Q_i orthogonal und $A = QR$, da $QQ_1 \cdots Q_i = \mathbb{I}$ $(Q_i^2 = \mathbb{I})$

$$\begin{aligned} \text{Ausgleichsproblem} &\iff \text{Normalgleichung} \\ &\iff R\overline{x} = \widetilde{c} \text{ mit} \\ &\qquad A = QR = \left(\frac{\widetilde{R}}{0}\right),\ R \text{ reguläre obere } \triangle\text{-Matrix, da Rang}(A) = n \end{aligned}$$

Q, R mit Householder Verfahren:

Es gilt:
$$\begin{aligned} cond_2(A) &= \|A\|_2 \left\|A^{-1}\right\|_2 \\ &= \|QR\|_2 \left\|R^{-1}Q^{-1}\right\|_2 \\ &= \|QR\|_2 \left\|R^{-1}Q^\top\right\|_2 \\ &= \|R\| \cdot \left\|R^{-1}\right\| \end{aligned}$$

$\Longrightarrow$ $cond_2(A) = cond_2(R)$ falls $\text{Rang}(A) = n,\ A \in \mathbb{R}^{m\times n}$

Singulärwertzerlegung von A

Satz 2.18
Sei $A \in \mathbb{R}^{m\times n},\ \text{Rang}(A) = r,\ p = \min\{m, n\}$. Dann existieren orthogonale Matrizen $U = (u_1, \ldots, u_n) \in \mathbb{R}^{m\times n}$ und $V = (v_1, \ldots, v_n) \in \mathbb{R}^{m\times n}$ mit $U^\top AV = \Sigma \in \mathbb{R}^{m\times n}$ mit

$$\Sigma = diag(\sigma_1, \ldots, \sigma_p),$$

wobei $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_p = 0$.

D.h:
$$\Sigma = \left(\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_r & \\ \hline & 0 & & 0 \end{array}\right)$$

wobei $diag(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{r\times r}$.

Definition 2.19
Die Werte $\sigma_1, \ldots, \sigma_r$ heißen **singuläre Werte** *von A. Sie entsprechen gerade den Wurzeln aus den Eigenwerten von $A^\top A$ bzw. $AA^\top$ (Bem: nach 2.11(v) haben $A^\top A$ und $AA^\top$ dieselben positven und reellen Eigenwerte).*

Beweis: (Satz 2.18)

Eindeutigkeit: Sei $U^\top AV = \Sigma$ und U, V orthogonal.

$\Longrightarrow$ $Av_i = \sigma_i u_i$, da $AV = U\Sigma$ und
$A^\top u_i = \sigma_i v_i$, da $A^\top U = V\Sigma$

$\Longrightarrow$ $A^\top Av_i = \sigma_i A^\top u_i = \sigma_i^2 v_i$ $\Longrightarrow$ σ_i^2 ist Eigenwert von $A^\top A$.

Analog: $AA^\top u_i = \sigma_i^2 u_i$ $\Longrightarrow$ σ_i^2 ist Eigenwert von $AA^\top$

Eigenwerte sind eindeutig $\Longrightarrow$ σ_i hängt nur von A ab.

Existenz: Sei $\sigma_1 := \|A\|_2 = \max\limits_{\|x\|_2=1} \|Ax\|_2$ (Bemerkung: $\|A\|_2 = \sqrt{\lambda_{\max}(A^\top A)}$, d.h. σ_1 ist ein guter Kandidat).

$\Longrightarrow$ $x_1 \in \mathbb{R}^n, y_1 \in \mathbb{R}^m$ mit $\|x_1\|_2 = 1,\ \|y_1\|_2 = 1$ und $Ax_1 = \sigma_1 y_1$. Sei $V = (x_1, \ldots, x_n) \in \mathbb{R}^{n\times n}$ eine orthogonale Basis des $\mathbb{R}^n$ (jeder Vektor kann erweitert werden zu einer ONB)

$U_1 = (y_1, \ldots, y_m) \in \mathbb{R}^{m\times m}$ ebenfalls ONB des $\mathbb{R}^n$

$\Longrightarrow$ $A_1 := U_1^\top AV_1 = \left(\begin{array}{c|c} \sigma & w^\top \\ \hline 0 & B \end{array}\right),\ B \in \mathbb{R}^{m-1\times n-1},\ w \in \mathbb{R}^{n-1}$.

Da U_1, V_1 orthogonal, gilt:

$$\|A_1\|_2 = \|A\|_2 = \sigma$$

Desweiteren gilt: $A_1 \begin{pmatrix} \sigma \\ w \end{pmatrix} = \begin{pmatrix} \sigma^2 + w^\top w \\ Bw \end{pmatrix} = \begin{pmatrix} \sigma^2 + \|w\|_2^2 \\ Bw \end{pmatrix}$

$$\begin{aligned} \implies \sigma^2 = \|A_1\|_2^2 &= \left(\max_x \tfrac{\|A_1 x\|}{\|x_2\|}\right)^2 \geq \tfrac{1}{\|\sigma_1 w^\top\|_2^2} \left\| A_1 \begin{pmatrix} \sigma \\ w \end{pmatrix} \right\|_2^2 \\ &= \tfrac{1}{(\sigma^2 + \|w\|_2^2)} \left(\left(\sigma^2 + \|w\|_2^2\right)^2 + \underbrace{\|Bw\|_2^2}_{\geq 0} \right) \\ &\geq \tfrac{1}{\sigma^2 + \|w\|_2^2} \left(\sigma^2 + \|w\|_2^2\right)^2 = \sigma^2 + \|w\|_2^2 \end{aligned}$$

$$\implies \sigma^2 \geq \sigma^2 + \|w\|_2^2 \implies \|w\|_2^2 = 0 \implies w = 0$$

$$\implies U_1^\top A V_1 = \left(\begin{array}{c|c} \sigma & 0 \\ \hline 0 & B \end{array}\right)$$

Den Rest per Induktion $\square$

Lösung des Ausgleichsproblems:

Gesucht: $x \in \mathbb{R}^n$ mit $\|Ax - b\|_2 = \inf_{z \in \mathbb{R}^n} \|Az - b\|_2$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $m \geq n \geq r = \text{Rang}(A)$.

Sei $U^\top A V = \Sigma = diag(\sigma_1, \ldots, \sigma_n)$

$$\begin{aligned} \implies \|Ax - b\|_2^2 &= \langle Ax - b, Ax - b \rangle \overset{\text{da } U^\top \text{ orth.}}{=} \langle U^\top (Ax - b), U^\top (Ax - b) \rangle \\ &= \langle U^\top A V (V^\top x) - U^\top b, U^\top A V (V^\top x) - U^\top b \rangle \\ &\overset{VV^\top = \mathbb{I}}{=} \langle \Sigma V^\top x - U^\top b, \Sigma V^\top x - U^\top b \rangle \\ &= \left\| \Sigma V^\top x - U^\top b \right\|_2^2 \end{aligned}$$

$$\begin{aligned} \implies \|Ax - b\|_2^2 &= \sum_{i=1}^{r} \left(\sigma_i (V^\top x) i - u_i^\top b\right)^2 + \sum_{i=r+1}^{m} \left(u_i^\top b\right)^2 \\ &\geq \sum_{i=r+1}^{m} \left(u_i^\top b\right) \end{aligned}$$

Folgerung 2.20
$x \in \mathbb{R}^n$ ist genau dann Lösung des Ausgleichsproblems, wenn $(V^\top x)_i = \frac{u_i^\top b}{\sigma_i}$

$$\iff V^\top x = \left(\frac{u_1^\top b}{\sigma_1}, \ldots, \frac{u_r^\top b}{\sigma_r}, \alpha_{r+1}, \ldots, \alpha_n \right), \ \alpha_i \text{beliebig}$$

Ist x Lösung des AGP, so ist $\|x\|_2^2 \overset{V^\top \text{ orth.}}{=} \left\| V^\top x \right\|_2^2 = \sum_{i=1}^{r} \left(\frac{u_i^\top b}{\sigma_i}\right)^2 + \sum_{i=r+1}^{n} \alpha_i^2$

$\implies \|x\|_2$ ist minimal $\iff \alpha_{r+1} = \ldots = \alpha_n = 0$. D.h. die eindeutige Lösung des AGP mit minimaler 2-Norm ist gegeben durch

$$x = V \left(\frac{u_i^\top b}{\sigma_1}, \ldots, \frac{u_r^\top b}{\sigma_r}, 0, \ldots, 0 \right)^\top = \sum_{i=1}^{r} \frac{u_i^\top b}{\sigma_i} v_i$$

2.3 Iterative Verfahren

Im Grunde wollen wir lineare Gleichungssysteme lösen, aber nicht mittels Gauß oder andere Verfahren, weil sie viel zu aufwändig sind. Stattdessen werden wir uns so nah wie möglich an die Lösung annähern, und das ist unser Ziel.

Idee: $Ax = b$ als Fixpunktgleichung umschreiben, so dass die Kontraktionsbediengung (vgl. Satz 1.29 auf Seite 5) erfüllt ist.

Ansatz: $A = M - N$ mit M regulär (i.A. M vorgegeben, dann ist $N = M - A$)

$$\begin{array}{rcl} Ax = b & \iff & (M-N)x = b \iff Mx - Nx = b \iff Mx = Nx + b \\ & \iff & x = M^{-1}Nx + M^{-1}b =: Tx + M^{-1}b \end{array}$$

Setze $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n : x \longmapsto Tx + c$ mit $c := M^{-1}b,\ T := M^{-1}N = \mathbb{I} - M^{-1}A$

Dann gilt: x ist die Lösung von $Ax = b \iff x$ ist Fixpunkt von F, d.h. $x = F(x) = Tx + c$

Verfahren: $x^0 \in \mathbb{R}^n$ vorgegeben, $x^{k+1} = F(x^k)$ $(K \geq 0)$, d.h. x^{k+1} ist gegeben durch

$$r^k := b - Ax^k \text{ (Residuum)}$$

und $My^k = r^k,\ x^{k+1} = x^k + y^k$

Satz 2.21
Sei $\|\cdot\|$ eine Norm auf dem $\mathbb{R}^n$ gegeben, so dass für die induzierte Matrixnorm gilt: $q := \|T\| < 1$. Dann gilt $x^k \longrightarrow x$ konvergiert mit $Ax = b$ und $\left\|x - x^k\right\| \leq \frac{q^k}{1-q} \left\|x^1 - x^0\right\|$ bzw. $\left\|x - x^k\right\| \leq \frac{q}{1-q} \left\|x^k - x^{k-1}\right\|$

Beweis: Siehe Satz 1.29 auf Seite 5

$$\begin{array}{rcl} \|F(y_1) - F(y_2)\| & = & \|Ty_1 + c - (Ty_2 + c)\| \\ & = & \|T(y_1 - y_2)\| \leq \|T\| \, \|y_1 - y_2\| = q \, \|y_1 - y_2\| \end{array}$$

Da $q < 1 \implies F$ ist eine Kontraktion □

Problem: Die Kontraktionsbediengung ist abhängig von der Norm, die Konvergenz nicht, da alle Normen auf $\mathbb{R}^n$ äquivalent sind.

Definition 2.22
Sei $B \in \mathbb{R}^{n \times n}$.

(i) $\lambda \in \mathbb{C}$ *Eigenwert von* B, *falls* $\det(B - \lambda\mathbb{I}) = 0$

(ii) $u \in \mathbb{C}^n \backslash \{0\}$ *ist ein Eigenvektor zum Eigenwert* λ, *falls* $Bu = \lambda u$ *gilt.*

$\rho(B) := \max\{|\lambda| \mid \lambda \text{ EW von } B \in \mathbb{C}\}$ *heißt* **Spektralradius**.

Bemerkung: Eine Matrix B hat in $\mathbb{C}$ $\lambda_1, \ldots, \lambda_n$ Eigenwerte (falls Vielfachheit zugelassen wird). Es existiert eine reguläre Matrix $U \in \mathbb{C}^{n \times n}$ mit

$$U^{-1}BU = \begin{pmatrix} \lambda_1 & & r_{ij} \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix},$$

also eine Ähnlichkeitstransformation.

Lemma 2.23
Es gilt für $B \in \mathbb{R}^{n\times n}$

(i) $\rho(B) \le \|B\|$ für jede induzierte Norm

(ii) $\forall\, \varepsilon > 0$ gibt es eine induzierte Norm $\|\cdot\|$ im $\mathbb{C}^{n\times n}$ mit $\|B\| \le \rho(B) + \varepsilon$

Beweis:

(i) Sei λ ein Eigenwert von B in $\mathbb{C}$ und $u \in \mathbb{C}^n\backslash\{0\}$ der zugehörige Eigenvektor, $\|\cdot\|$ eine Norm auf C^n.

$$\begin{aligned} Bu = \lambda u \quad &\Longrightarrow \quad |\lambda|\,\|u\| = \|\lambda u\| = \|Bu\| \le \|B\|\,\|u\| \\ &\Longrightarrow \quad |\lambda| < \|B\| \text{ f.a. EW} \quad \Longrightarrow \quad \rho(B) \le \|B\| \end{aligned}$$

(ii) Sei $U \in \mathbb{C}^{n\times n}$ regulär mit $U^{-1}BU = \begin{pmatrix} \lambda_1 & & r_{ij} \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$

Für $\delta > 0$ sei $D_\delta = diag(\delta^0, \ldots, \delta^{n-1})$.

Im Allg. gilt für $G \in \mathbb{C}^{n\times n}$: $\left(D_\delta^{-1} G D_\delta\right) = g_{ij}\delta^{j-i}$

$$\Longrightarrow \quad (UD_\delta)^{-1} B\,(UD_\delta) = D_\delta^{-1}\left(U^{-1}BU\right) D_\delta = \begin{pmatrix} \lambda_1 & \delta r_{12} & \delta^2 r_{13} & \cdots & \delta^{n-1} r_{1n} \\ 0 & \lambda^2 & \delta r_{23} & \cdots & \delta^{n-2} r_{2n} \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \delta r_{n-1,n} \\ 0 & \cdots & \cdots & \cdots & \lambda^n \end{pmatrix}$$

Sei $\varepsilon > 0$ vorgegeben. Dann existiert ein $\delta > 0$ mit $\sum\limits_{k=i+1}^{n} \delta^{k-i}\,|r_{ik}| \le \varepsilon$ f.a. $i \in \{1, \ldots, n-1\}$.

Setze $\|x\|_\delta = \left\|(UD_\delta)^{-1} x\right\|_\infty$. Diese ist eine Norm auf $\mathbb{C}^n$, da $(UD_\delta)^{-1}$ regulär. Es gilt:

$$\|B\|_\delta = \sup_{x\in\mathbb{C}^n\backslash\{0\}} \frac{\|Bx\|_\delta}{\|x\|_\delta} = \sup_{x\in\mathbb{C}^n\backslash\{0\}} \frac{\left\|(UD_\delta)^{-1} Bx\right\|_\infty}{\left\|(UD_\delta)^{-1} x\right\|_\infty}$$

Mit $w := (UD_\delta)^{-1} x$ bzw. $x = (UD_\delta)\, w$ gilt:

x läuft über $\mathbb{C}^n$, dann läuft w über den ganzen $\mathbb{C}^n$, da $(UD_\delta)^{-1}$ regulär, d.h. $\sup\limits_{x\neq 0} \cdots = \sup\limits_{w\neq 0} \cdots$

$$\begin{aligned} \Longrightarrow \quad \|B\|_\delta &= \sup_{w\neq 0} \frac{\left\|(UD_\delta)^{-1} B (UD_\delta) w\right\|_\infty}{\|w\|_\infty} \\ &\le \left\|(UD_\delta)^{-1} B\,(UD_\delta)\right\|_\infty \\ &= \max_i \left\{ |\lambda_i| + \underbrace{\sum_{k=i+1}^{n} \delta^{k-1} \cdot |r_{ik}|}_{\le \varepsilon} \right\} \text{ (Zeilensummennorm)} \\ &= \max_i |\lambda_i| + \varepsilon = \rho(B) + \varepsilon \end{aligned}$$

□

Satz 2.24
Es ist äquivalent

(i) $T^\nu \longrightarrow 0$ (im $\mathbb{C}^{n\times n}$) für $\nu \longrightarrow \infty$

(ii) Für $u \in \mathbb{C}^n$ fest: $T^\nu u \longrightarrow 0$ (im $\mathbb{C}^n$) für $\nu \longrightarrow \infty$

(iii) $\rho(T) < 1$

(iv) Es existiert eine Norm auf $\mathbb{C}^n$, so dass für die induzierte Matrixnorm gilt

$$\|T\| < 1$$

Beweis:
(i) $\Longrightarrow$ (ii)

$\|T^\nu u\| \leq \|T^\nu\| \, \|u\| \longrightarrow 0$, da $T^\nu \longrightarrow 0$
$\Longrightarrow$ $T^\nu u \longrightarrow 0$ (für $\nu \longrightarrow \infty$)

(ii) $\Longrightarrow$ (iii)

Annahme: $\rho(T) \geq 1$, d.h. es existiert ein EW $\lambda \in \mathbb{C}$ mit $|\lambda| \geq 1$. Sei $u \in \mathbb{C}^n \backslash \{0\}$ der zugehörige EV
$\Longrightarrow$ $T^\nu u = T^{\nu-1}(Tu) = T^{\nu-1}(\lambda u) = \lambda T^{\nu-1} u = \cdots = \lambda^\nu u$

$\Longrightarrow$ $\|T^\nu u\| = \|\lambda^\nu u\| = |\lambda|^\nu \, \|u\| \not\longrightarrow 0$, da $|\lambda| \geq 1$ Widerspruch zu *(ii)*
(iii) $\Longrightarrow$ (iv)

Lemma 2.23

(iv) $\Longrightarrow$ (i)

$\|T^\nu\| \overset{\text{Submultipli.}}{\leq} \|T\|^\nu \longrightarrow 0$, da $\|T\| < 1$ $\square$

Folgerung 2.25
Das Iterationsverfahren konvergiert gdw. $\rho(T) < 1$

Bemerkung: (Ohne Beweis)

Aus den bisher gezeigten Sätzen folgt:

Um eine Dezimalstelle im Fehler zu gewinnen, müssen $K \sim -\frac{\ln 10}{\ln(\rho(T))}$ Schritte durchgeführt werden.

Das heißt, $\rho(T) \sim 1$, denn $-\ln(\rho(T)) \sim 0$ und K ist sehr groß. Somit muss das Ziel bei der Wahl der regulären Matrix M sein:

1. $\rho(T) = \rho(\mathbb{I} - M^{-1}A)$ möglichst klein
2. Gleichungssystem $My^k = r^k$ muss leicht zu lösen sein.

Widersprüchliche Forderungen:
Optimal für Bedingung 1 wäre $M = A$ $\Longrightarrow$ $\rho(T) = 0$, aber dann ist die Bediengung 2 überhaupt nicht erfüllt.

e) **Jacobi Verfahren**[3]Sei A regulär und $a_{ii} \neq 0$ f.a. $i = 1, \ldots, n$

$M := D = diag(a_{11}, \ldots, a_{nn})$
$\Longrightarrow \quad T = \mathbb{I} - D^{-1}A$

Das führt zu folgender Iterationsvorschrift:

Iteration: x^0 Startwert, x^{k+1} Lösung von $M^{(k+1)}x = Nx^k + b$

Es gilt: $A = A_L + A_D + A_R = \begin{pmatrix} 0 & \cdots & 0 \\ & \ddots & \vdots \\ * & & 0 \end{pmatrix} + \begin{pmatrix} * & & 0 \\ & \ddots & \\ 0 & & * \end{pmatrix} + \begin{pmatrix} 0 & & * \\ \vdots & \ddots & \\ 0 & \cdots & 0 \end{pmatrix}$

$x^0 \in \mathbb{R}^n$ vorgegeben.

$$\begin{aligned} x^{k+1} &= (\mathbb{I} - D^{-1}A)x^k + D^{-1}b \\ &= D^{-1}(Dx^k - Ax^k + b) \\ \Longrightarrow \quad x_i^{(k+1)} &= \frac{1}{a_{ii}} \left(b_i - \sum_{k=1, l \neq 1}^{n} a_{il} x_l^k \right), \; i = 1, \ldots, n \end{aligned}$$

Satz 2.26 (Hinreichende Bediengung für die Konvergenz des Jacobi-Verfahren)
Falls entweder:

(a) $\max\limits_i \left(\sum\limits_{k \neq i} \frac{|a_{ik}|}{|a_{ii}|} \right) < 1$ oder

(b) $\max\limits_i \left(\sum\limits_{k \neq i} \frac{|a_{ki}|}{|a_{ii}|} \right) < 1,$

dann konvergiert das Jacobi-Verfahren.

Bemerkung:
(a) heißt **starkes Zeilensummenkriterium**
(b) heißt **starkes Spaltensummenkriterium**

Das nennt sich **Diagonaldominanz**, die Einträge der Diagonalen sind größer als die Summe der Zeilen bzw. Spalten der Matrix.

Beweis: Gelte (a):

$$\begin{aligned} \Longrightarrow \quad \rho(T) \leq \|T\|_\infty &= \left\| \mathbb{I} - D^{-1}A \right\|_\infty \\ &= \max_i \left(\sum_{k=1}^{n} \left| \delta_{ik} - \frac{|a_{ik}|}{|a_{ii}|} \right| \right) \\ &= \max_i \left(\sum_{k=1, k \neq i}^{n} \frac{|a_{ik}|}{|a_{ii}|} \right) < 1 \text{ (wegen (a))} \end{aligned}$$

Gelte (b):

$\rho(T) \leq \|T\|_1 = \max\limits_i \left(\sum\limits_{k \neq i}^{n} \frac{|a_{ki}|}{|a_{ii}|} \right) < 1$
$\Longrightarrow$ Das Iterationsverfahren konvergiert. Das ist eine sehr starke Bediengung.

[3]Das ist die Forsetzung der Verfahren ab Seite 31

Beispiel 2.27
Bei der Diskretisierung $\partial_{xx} u = f$ in §1.5 musste ein LGS $Ax = b$ gelöst werden mit

$$A = \begin{pmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}$$

$\Longrightarrow \sum_{k \neq i} \frac{|a_{ik}|}{|a_{ii}|} = 1$ für $i = 2, \ldots . n-1$ und für $i = 1, n$ gilt $\sum_{k \neq i} \frac{|a_{ik}|}{|a_{ii}|} < 1$

Definition 2.28
Eine Matrix $A = (a_{ik})$ heißt **zerlegbar**, *falls es Teilmengen $N_1, N_2 \subset N = \{1, \ldots . n\}$ gibt mit $N_1 \neq \emptyset$, $N_2 \neq \emptyset$ und $N = N_1 \biguplus N_2$ und $a_{ik} = 0 \; \forall (i,k) \in N_1 \times N_2$*

Lemma 2.29
Für $A \in \mathbb{R}^{n \times n}$ sind äquivalent

(i) A ist zerlegbar

(ii) Der zugehörige gerichtete Graph $G(A)$ ist nicht zusammenhängend .

$G(A) := \big(\text{Knoten } P_1, \ldots, P_n, \text{ gerichtete Kanten } \overline{P_j P_k} \iff a_{jk} \neq 0\big)$

Nicht zusammenhängend: Es existieren Knoten P_j und P_k, die nicht durch einen Kantenzug verbunden sind, d.h es existiert keine Folge $P_0, \ldots, P_n \in \{1, \ldots, N\}$ mit $P_0 = j, P_n = k$ und $a_{p_i p_{i+1}} \neq 0$

(iii) Es existiert eine Permutationsmatrix $P \in \mathbb{R}^{n \times n}$ mit

$$PAP^\top = \begin{pmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{pmatrix}$$

mit $A_{11} \in \mathbb{R}^{p \times p}$, $A_{22} \in \mathbb{R}^{q \times q}$, $A_{21} \in \mathbb{R}^{q \times p}$ und $p + q = n$

Beispiel 2.30

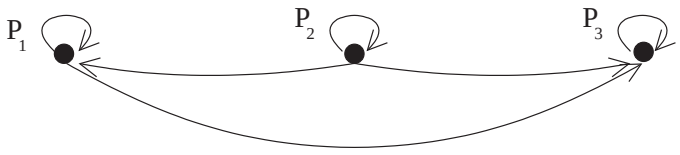


Abbildung 2.2: Graph, Beispiel 2.30

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

(i) $N_1 = \{1, 3\},\ N_2 = \{2\}$ ist geeignete Zerlegung

(ii) $G(A)$ nicht zusammenhängend, da es keine Verbindung zwischen P_1 und P_2 existiert. (Siehe Abbildung 2.2 auf Seite 40)

(iii)

$$P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix};\ PAP^\top = \left(\begin{array}{c|cc} 2 & 0 & 0 \\ \hline 1 & 2 & 2 \\ 2 & 0 & 2 \end{array}\right)$$

Beispiel 2.31
Sei A eine Tridiagonalmatrix ohne Nullen auf der Nebendiagonalen, d.h.

$$A = \begin{pmatrix} * & * & & 0 \\ * & \ddots & \ddots & \\ & \ddots & \ddots & * \\ 0 & & * & * \end{pmatrix}$$

$\Longrightarrow$ A unzerlegbar (Beweis, siehe Übung).

Satz 2.32
Sei $A \in \mathbb{R}^{n\times n}$ unzerlegbar und erfülle das **schwache Zeilensummenkriterium**, d.h.

$$\max_i \left(\sum_{k\neq i} \frac{|a_{ik}|}{|a_{ii}|} \right) \leq 1$$

und es existiert ein $r \in \{1, \ldots, n\}$ mit $\sum\limits_{k\neq r} \frac{|a_{rk}|}{a_{rr}} < 1$. Dann gilt: Das Jacobi-Verfahren kann angewandt werden und konvergiert für alle Startvektoren $x^0 \in \mathbb{R}^n$.

Beweis: Es gelte: $\sum\limits_{k\neq i} |a_{ik}| \leq |a_{ii}|$, z.z $|a_{ii}| > 0$

Da A zerlegbar $\Longrightarrow$ $\sum\limits_{k\neq i} |a_{ik}| > 0$, d.h. $|a_{ii}| > 0$ (sonst wähle $N_1 = \{i\}$ und $N_2 = N \backslash N_1$)

$\Longrightarrow$ Das Jacobi-Verfahren kann angewandt werden. Wie im Beweis von Satz 2.26 $\Longrightarrow$ $\rho(T) \leq 1$ mit $T = M^{-1}N$.

z.z $\rho(T) \neq 1$: Annahme: Es existiert ein Eigenwert $\lambda \in \mathbb{C}$ mit $|\lambda| = 1$. Sei $v \in \mathbb{C}^n$ der zugehörige Eigenvektor mit $\|v\|_\infty = 1$, d.h. $\exists\ s \in \{1, \ldots, n\}$ mit $|v_s| = 1$: Aus $Tv = \lambda v$ folgt für $i = 1, \ldots, n$:

$$\lambda v_i = \sum_{k=1}^{n} t_{ik} v_k = \sum_{k=1}^{n} \left(\delta_{ik} - \frac{a_{ik}}{a_{ii}} \right) v_k = \sum_{k\neq i} \frac{a_{ik}}{a_{ii}} v_k$$

(Vergleiche Satz 2.26) $\Longrightarrow$ $|v_i| = |\lambda|\,|v_i| = |\lambda v_i| \leq \frac{1}{|a_{ii}|} \sum\limits_{i=0}^{n} |a_{ik}|\,|v_k|$ $(*)$. Da $G(A)$ zshg. existiert ein Pfad zwischen P_s und P_r, d.h. $p_0 = s, \ldots, p_k = r$ und $a_{p_i p_{i+1}} \neq 0$ $(i = 0, \ldots, k-1)$ mit $(*)$.

$$|v_r| \leq \frac{1}{|a_{ri}|} \sum_{k\neq r} |a_{rk}|\,|v_k|_\infty \leq \frac{\|v\|_\infty}{|a_i|} \sum_{k\neq i} |a_{rk}| < \|v\|_\infty$$

$$\begin{aligned}
|v_{p_{k-1}}| &\overset{*}{\leq} \left|\tfrac{1}{a_{p_{k-1}p_{k-1}}}\right| \left(\sum_{k\neq p_{k-1}; k\neq p_k} \left|a_{p_{k-1}} b\right| |v_k| + |a_{p_k} P_k| \, |v_{p_n}| \right) \\
&\leq \tfrac{|v|_\infty}{\left|a_{p_{k-1}} P_{n-1}\right|} \left(\sum_{k\neq P_k} \left|a_{p_{k-1}} k\right| \right) \leq \|v\|_\infty \\
&\vdots \\
|v_s| &= |v_{p_0}| < \|v\|_\infty
\end{aligned}$$

ist ein Widerspruch, da es so gewählt wurde, dass $|v_s| = 1 = \|v\|_\infty$.

f) **Einzelschritt Verfahren (ESV) (Gauß-Seidel-Verfahren)**[4]

Es sei $x_i^{(0)}$ gegeben und:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{l<i} a_{il} x_l^{(k)} - \sum_{l>1} a_{il} x_l^{(k)} \right)$$

Sei A regulär, $a_{ii} \neq 0$, $A = A_L + A_D + A_R$. Setze $M = A_L + A_D = \begin{pmatrix} * & & 0 \\ \vdots & \ddots & \\ * & \cdots & * \end{pmatrix}$

$N = M - A = -A_R$. Da $a_{ii} \neq 0 \implies M$ regulär.

Iteration: $x^{(0)} \in \mathbb{R}^n$, $x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b = (A_L + A_D)^{-1}\left(b - A_R x^{(k)}\right)$

Nun kann man man nach $x_i^{(k+1)}$ auflösen.

$$x^{(k+1)} = (A_L + A_D)^{-1}\left(b - A_R x^{(k)}\right)$$

$$\implies x_i^{(k+1)} = \tfrac{1}{a_{ii}} \left(b_i - \sum_{l=1}^{i-1} a_{il} x_l^{(k+1)} - \sum_{l=i+1}^{n} a_{il} x_l^{(k)} \right), \; i = 1, \ldots, n$$

Satz 2.33
Falls $\max_i \left(\sum_{k\neq i} \frac{|a_{ik}|}{|a_{ii}|} \right) < 1$ (starkes Zeilensummenkriterium), dann konvergiert das Einzelschrittverfahren.

Beweis: $T := M^{-1}N$, $q = \max_i \left(\sum_{k\neq i} \frac{|a_{ik}|}{|a_{ii}|} \right)$

Behauptung: $q \leq 1 \implies \|T\|_\infty \leq q \implies \|T\|_\infty \leq q < 1 \implies T$ ist Kontraktion.

$\|T\|_\infty := \sup_{\|x\|_\infty = 1} \|Tx\|_\infty$. Sei $x \in \mathbb{R}^n$ mit $\|x\|_\infty = 1$

$y := Tx$, z.z. $\|y\|_\infty \leq q$, d.h. $|y_i| \leq q$ $(1 \leq i \leq n)$

Durch Induktion: I.A. $y_1 = -\frac{1}{a_{ii}} \left(\sum_{k=2}^{n} a_{1k} x_k \right)$

$$\implies |y_i| \leq \tfrac{1}{|a_{11}|} \sum_{k=2}^{n} |a_{1k}| \underbrace{|x_k|}_{\leq 1} \leq \tfrac{1}{|a_{11}|} \sum_{k\neq 1} |a_{1k}| \leq q$$

[4] Das ist die Forsetzung der Verfahren ab Seite 39

I.S. $y_i = -\frac{1}{a_{ii}} \left(\sum_{k=i+1}^{n} a_{ik} x_k - \sum_{k=1}^{i-1} a_{ik} y_k \right)$

$$\implies |y_i| \leq \frac{1}{|a_{ii}|} \left(\sum_{k=i+1}^{n} \underbrace{|a_{ik}|\,|x_k|}_{\leq 1,\ \text{da } \|x\|_\infty \leq 1} + \sum_{k=1}^{i-1} |a_{ik}|\,|y_k| \right) \leq q, \text{ I.V. } |y_k| \leq q \leq 1 \quad \square$$

Satz 2.34 (Ohne Beweis)
$A \in \mathbb{R}^{n\times n}$ unzerlegbar und erfülle das schwache Zeilensummenkriterium, dann konvergiert das Gauß-Seidel-Verfahren

Satz 2.35
A sei symmetrisch und positiv definit, dann konvergiert das Gauß-Seidel-Verfahren.

Bemerkung: Die analoge Aussage gilt nicht für das Jacobi-Verfahren.

Beweis: $A = L + D + R$, da A symmetrisch gilt: $R = L^\top$ bzw. $R^\top = L$, $M = L + D$, $N = -R$, $T = M^{-1}N = -(L+D)^{-1}R$

Sei $\lambda \neq 0$ ein Eigenwert von T in $\mathbb{C}$ (T ist nicht notwendig symmetrisch, d.h. $\lambda \in \mathbb{C}$). Sei $x \in \mathbb{C}^n$ der zugehörige Eigenvektor mit $\|x\|_2 = 1$: $-(L+D)^{-1}Rx = \lambda x$ bzw. $-Rx = \lambda Dx + \lambda Lx = \lambda Dx + \lambda R^\top x$

$$D = A - L - R = A - R^\top - R \implies -Rx = \lambda(A - R^\top - R)x + \lambda R^\top x = \lambda Ax - \lambda Rx$$

Setze $\alpha := \langle Ax, x\rangle > 0$, da A positiv definit.

$\sigma := \langle Rx, x\rangle = \sigma_1 + i\sigma_2 \in \mathbb{C}$, $\delta := \langle Dx, x\rangle \implies -\sigma = \lambda\alpha - \lambda\sigma = \lambda(\alpha - \sigma)$

Es gilt $\alpha - \sigma \neq 0$, da sonst $\sigma = 0 \implies \alpha = \sigma = 0$ ist Widerspruch zur positiven Definitheit.

$\implies \lambda = -\frac{\sigma}{\alpha - \sigma} \implies |\lambda|^2 = \frac{\sigma\overline{\sigma}}{(\alpha-\sigma)(\alpha-\sigma)} = \frac{\sigma_2^2 + \sigma_2^2}{(\alpha-\sigma_1)^2 + \sigma_2^2}$, da $\alpha \in \mathbb{R}$ $\alpha - \sigma = \alpha - (\alpha_1 + i\sigma_2) = (\alpha - \sigma_1) - i\sigma_2$ da

$$\begin{aligned} \alpha = \langle Ax, x\rangle &= \langle R^\top x, x\rangle + \langle Dx, x\rangle + \langle Rx, x\rangle \\ &= \delta + 2\sigma_1 \implies \delta = \alpha - 2\sigma_1 \end{aligned}$$

$$\begin{aligned} (\alpha - \sigma_1)^2 &= (\delta + \sigma_1)^2 = \delta^2 + 2\delta\sigma_1 + \sigma_1^2 = \delta(\alpha - 2\sigma_1) + 2\delta\sigma_1 + \sigma_1^2 \\ &= \delta\alpha + \sigma_1^2 \geq \mu\delta + \sigma_1^2 \end{aligned}$$

wobei μ der kleinste Eigenwert von A ist, da A positv definit $\implies \mu \in \mathbb{R}$, $\mu > 0$.

$$\delta = \langle Dx, x\rangle = \sum_{i=1}^{n} a_{ii} x_i \overline{x}_i \geq \min_i a_{ii} \|x\|_2^2 = \min_i a_{ii} =: \overline{\delta}$$

$$\implies |\lambda|^2 = \frac{\sigma_1^2 + \sigma_2^2}{(\alpha - \sigma_1)^2 + \sigma_2^2} \leq \frac{\sigma_1^2 + \sigma_2^2}{\mu\overline{\delta} + \sigma_1^2 + \sigma_2^2} < 1$$

da $\mu\overline{\delta} > 1 \implies |\lambda| < 1 \implies \rho(T) < 1 \quad \square$

2.4 Zusammenfassung

Wir haben in diesem Kapitel verschiedene Methoden zur Lösung linearer Gleichungssysteme der Form

$$Ax = b$$

kennengelernt und näher betrachtet,

1. für $A \in \mathbb{R}^{n\times n}$ regulär

2. für $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ (d.h. überstimmt)

Verfahren:

1. Direkte Verfahren oder Direktlöser (LR-, Cholesky, QR-Zerlegung)
2. Iterative Verfahren oder Iterativenlöser (Jacobi-, Gauß-Seidel-Verfahren)

Vor-/Nachteile

- **Direkte Verfahren:** Die Lösung wird bis auf Rundungsfehler exakt berechnet. Sie sind für große Gleichungssysteme sehr langsam (die Anzahl der arithmetischen Operationen liegen in $O(n^3)$); es tritt ein *fill-in* Problem auf, d.h. in Anwendungen ist A häufig dünn besetzt, d.h. pro Zeile sind nur k viele Einträge ungleich Null, wobei k unabhängig von n, und diese Einträge sind unstrukturiert verteilt. Eine typische Zeile sieht dann so aus:

$$\left(\begin{array}{cccccc} * & & & & & 0 \\ 0 & \ddots & & & & \\ \hline * & 0\cdots 0 & * & *0\cdots 0 & & * \\ \hline & & & \ddots & & \\ & & & & & * \end{array}\right)$$

  Bei Zerlegungsverfahren werden Nulleinträge u.a. durch Einträge ungleich Null ersetzt, der Speicheraufwand zum Speichern von A liegt in der Regel bei $O(NK) = O(N)$; nach der Zerlegung wächst der Speicheraufwand bis auf $O(N^2)$

- **Iterative Verfahren:** Die Lösung wird nur nährungsweise bestimmt und die Geschwindigkeit der Konvergenz hängt von $\rho(T)$ (Spektralradius) ab.

  Allerdings ist der zusätzliche Speicheraufwand sehr klein (Gauß-Seidel-Verfahren gar kein zusätzlicher Speicheraufwand). Iterative Verfahren benötigen ein gutes Abbruchkriterium, i.A. gilt für eine beliebige Norm $\left\|Ax^{(k)} - b\right\| < TOL$ (Residuum$< TOL$).

Beschleunigung/Stabilisierung

Das Hauptproblem: Einfluss durch Rundungsfehler und ihre Reduzierung.

Beispiel: Die Pivotisierung bei der LR-Zerlegung. Die Kondition des Problems ist proportional zur Kondition von A: $cond(A) = \|A\| \left\|A^{-1}\right\|$. Durch **Vorkonditionierung** kann versucht werden, die Kondition des Problems zu verkleinern.

Beispiel: Wähle C_1, C_2 reguläre Matrizen. Dann gilt:

$$Ax = b \iff \underbrace{C_1 A C_2}_{:=\tilde{A}} \underbrace{C_2^{-1} x}_{:=\tilde{x}} = \underbrace{C_1 b}_{:=\tilde{b}}$$

mit $\tilde{A} := C_1 A C_2,\ \tilde{x} := C_2^{-1} x,\ \tilde{b} := C_1 b$ und C_1, C_2 so gewählt, dass $cond(\tilde{A}) < cond(A)$ gilt.

Wir haben auch die **Zeilenäquilibrierung**, $C_2 := \mathbb{I}$ und $C_1 = diag(d_1, \ldots, d_n)$ mit $d_i = \left(\sum_{j=1}^{n} |a_{ij}|\right)^{-1}$

$\implies cond_\infty(\tilde{A}) \leq cond_\infty(A)$ [5]

Mit ähnlichen Ansätzen versucht man, $\rho(T)$ klein zu halten.

[5] $cond(A)$ hängt von der gewählten Norm ab, $cond_\infty(A)$ bezieht sich auf die Kondition von A bezüglich der Maximumsnorm

Beschleunigung durch Relaxation:

Ein Iterationsverfahren hat die Gestalt

$$r^k = b - Ax^k,\ My^k = r^k,\ x^{k+1} = x^k + y^k$$

Statt y^k zu korrigieren, wählt man $w > 0$ so und setzt

$$x^{k+1} = x^k + wy^k$$

Das führt auf das **SOR-Verfahren** [6]

Weitere Verfahren

Wenn A symmetrsich, positiv definit, dann ist $Ax = b \iff Q(x) \leq Q(z)\ \forall\ z \in \mathbb{R}^n$ mit

$$Q(z) := \frac{1}{2}\langle Az, z\rangle - \langle b, z\rangle$$

Konstruiere eine Minimalfolge. Sie führt auf die **Methode des steilsten Abstiegs** und auf das **konjugierte Gradienten-Verfahren**, die in der Vorlesung Numerik II betrachtet werden.

[6] **S**uccesive **O**ver **R**elaxation; Das SOR-Verfahren gehört zur Klasse der stationären iterativen Verfahren (von einer [beliebigen] Startlösung schrittweise Näherung an die *echte* Lösung mit immer der selben Verfahrensvorschrift).

Kapitel 3

Nullstellensuche

Problem:

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}^n \text{ gegeben}$$

A: **Gesucht:** ein $x^* \in \mathbb{R}^n$ mit $f(x^*) = 0$

B: **Gesucht:** alle (größtes, kleinstes,usw.) $x^* \in \mathbb{R}^n$ mit $f(x^*) = 0$

Beispiel:
$f(x) = Ax - b,\ A \in \mathbb{R}^{m\times n}$ (siehe Kap. 2)
$f(x) = ax^2 + bx + c$ Alle Nullstellen explizit berechenbar
$f(x) = \cos(x)$ Gesucht $x^* \in [1, 2]$, $x^* = \frac{\pi}{2}$

Wir beschäftigen uns im wesentlichen mit Verfahren zur Lösung vom Problem **A** für $n = 1$ und

$$f : [a, b] \longrightarrow \mathbb{R}$$

glatt.

Wir nehmen an, dass es ein $x^* \in [a, b]$ gibt mit $f(x^*) = 0$, etwa $f \in C^0(a, b)$ und $f(a)f(b) < 0$. Alle Verfahren konstruieren eine Folge $\left(x^{(k)}\right)_{k\in\mathbb{N}}$ mit $x^{(k)} \longrightarrow x^*,\ f(x^*) = 0$. Direkte Verfahren existieren nur in Spezialfällen.

Bemerkung: Selbst bei sehr glatten Funktionen $f \in C^\infty$ kann die Nullstellenbestimmung sehr schlecht konditioniert sein. Ist $f \in C^2(I)$ mit $f(x^*) = 0$ und $g \in C^2(I)$ Störung von f mit $g(x^*) \neq 0$.

$$F(x) := f(x) + \varepsilon g(x),\ |\varepsilon| \ll 1 \text{ mit } F(x^* + \delta) = 0$$

$\implies \delta \doteq -\varepsilon \frac{g(x^*)}{f'(x^*)+\varepsilon g'(x^*)} \sim -\frac{\varepsilon g(x^*)}{f'(x^*)}$ für $|\varepsilon|$ hinreichend klein (ε ist Störung, δ ist Störung der Nullstelle und $\doteq$ Gleichheit in erster Nährung)

$\left|\frac{g(x^*)}{f'(x^*)}\right|$ ist unabhängig von ε und kann sehr groß sein.

Beispiel: (von Wilkinson 60er, the perifidious polynomiel)

$f(x) = \prod_{k=1}^{20}(x - k)$ hat die Nullstellen $x^* = 1, \ldots, 20$ und es gilt

$$f(x) = \prod_{k=1}^{20}(x - k) = x^{20} - 210x^{19} + \ldots$$

Sei $g(x) = x^{20}$, d.h. Störung des höchsten Koeffizients. $F(x) = (1+\varepsilon)x^{20} - 210x^{19} + \ldots,\ F(2+\delta) = 0$

$\Longrightarrow \quad |\delta| = |\varepsilon| \frac{g(20)}{f'(20)} = |\varepsilon| \frac{20^{20}}{q!} \sim |a|\, 10^9$

Verändert man den Koeffizient von x^{19} von -210 zu $-(210 + 2^{-23})$, so verändert sich das Nullstellenpaar $16, 17$ zum konjugierten komplexen Paar $16.73\ldots \pm i2.8112 \in \mathbb{C}\backslash\mathbb{R}$

3.1 Verfahren in einer Raumdimension

a) **Intervallschachtelung**

Voraussetzungen: $f \in C^0(a,b),\ f(a)f(b) < 0,\ a < b$

Verfahren: $a_0 = a, b_0 = b$. Für $n = 0, \ldots, N:\ x^{(n)} = \frac{1}{2}(a_n + b_n)$.

Falls $f(a_n)f(x^{(n)}) = 0$ dann Abbruch.

Falls $f(a_n)f(x^{(n)}) < 0 \quad \Longrightarrow \quad a_{n+1} := a_n;\ b_{n+1} := x^{(n)}$

Falls $f(a_n)f(x^{(n)}) > 0 \quad \Longrightarrow \quad a_{n+1} := x^{(n)};\ b_{n+1} := b_n$

Satz 3.1
Seien $(a_n)_{n\in\mathbb{N}},\ (b_n)_{n\in\mathbb{N}},\ \left(x^{(n)}\right)_{n\in\mathbb{N}}$ durch das Intervallschachtelungsverfahren dfiniert, dann gilt

$$\lim_{n\longrightarrow\infty} a_n = \lim_{n\longrightarrow\infty} b_n = \lim_{n\longrightarrow\infty} x^{(n)} = x^* \text{ mit } f(x^*) = 0$$

Es gilt: $\left|x^{(n)} - x^*\right| \leq 2^{-(n+1)}\,|b-a|$

Beweis: Es gilt $(a_n)_{n\in\mathbb{N}}$ monoton wachsend und $(b_n)_{n\in\mathbb{N}}$ monoton fallend und
$0 < b_n - a_n = 2^{-n}(b-a),\ a_n < b, a < b_n$

$\Longrightarrow \quad (a_n)_{n\in\mathbb{N}},\ (b_n)_{n\in\mathbb{N}}$ konvergieren und $\lim_{n\longrightarrow\infty} a_n = \lim_{n\longrightarrow\infty} b_n =: x^*$

Da $f \in C^0:\ f(x^*) = \lim_{n\longrightarrow\infty} f(a_n) = \lim_{n\longrightarrow\infty} f(b_n)$

Nach Konstruktion: $f(a_n)f(b_n) < 0$

$\Longrightarrow \quad f(x^*)^2 = \lim_{n\longrightarrow\infty} = \lim_{n\longrightarrow\infty} (f(a_n)f(b_n)) \leq 0 \quad \Longrightarrow \quad f(x^*) = 0$

Da $x^* \in \left(x^{(n)}, b_n\right)$ oder $x^* \in \left(a_n, x^{(n)}\right)$

$\Longrightarrow \quad \left|x^{(n)} - x^*\right| \leq \min\left\{\left|x^{(n)} - b_n\right|, \left|x^{(n)} - a_n\right|\right\} = \frac{1}{2}\,|b_n - a_n| \leq 2^{-n-1}(b-a)$ □

Bemerkung: Das Verfahren ist sehr robust aber auch sehr langsam (ca. 3 Schritte, um eine Dezimalzahlstelle Genauigkeit zu gewinnen).

Aufwand: Eine Auswertung von f pro Schritt: Das Verfahren wird in der Regel eingesetzt, um eine Nullstelle grob zu lokalisieren, damit ein effizienteres Verfahren den Rest ausrechnet.

b) **Newton Verfahren**

Idee: Sei $x^{(k)}$ eine gegebene Approximation von x^*, d.h. $h = x^* - x^{(k)} \neq 0$ ist klein. Dann gilt: (Taylorreihe)

$$0 = f(x^*) = f(x^{(k)} + h) = f(x^{(k)}) + f'(x^{(k)})h + \frac{1}{2}f^n(\xi)h^2$$

unter Vernachlässigung des Terms (zweiter Ordung h^2)

Man kann nach h auflösen:

$$h = -\frac{f(x^{(k)})}{f'(x^{(k)})}$$

Daher sollte $x^{(k+1)} = x^k + h = x^k - \frac{f(x^{(k)})}{f'(x^{(k)})}$ eine bessere Approximierung von x^* errechnen.

Verfahren: $x^{(0)}$ gegeben. Setze

$$x^{(k+1)} := x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$$

Aufwand: Eine Auswertung von f und f' pro Schritt, d.h. $f \in C^1$ und f' muss bekannt sein.

Geometrische Interpretation

Sei $l(x)$ die Linearisierung von f an der Stelle $x^{(k)}$, d.h. $l(x^{(k)}) = f(x^{(k)})$, $l'(x^{(k)}) = f'(x^{(k)})$ und $l(x) = ax + b$

$\Longrightarrow$ (Taylor) $l(x) = f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)})$

Statt Nullstellen von f zu suchen, sei $x^{(n+1)}$ Nullstelle von l

$$0 = f(x^{(k)}) + f'(x^{(k)})(x^{(k+1)} - x^{(k)})$$

Daraus das Newton Verfahren.

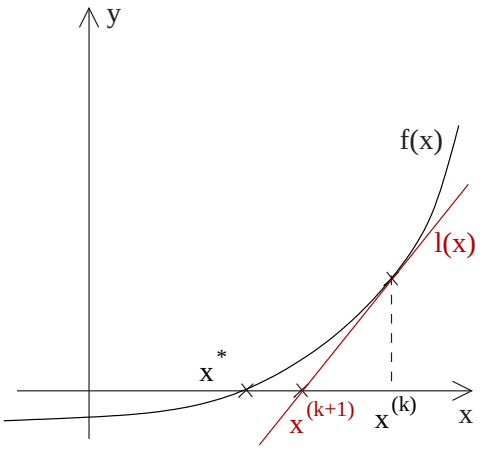


Abbildung 3.1: Newton Verfahren, Beispiel 1

Mit der Anschauung von Abbildung 3.1 ist es klar, wie das Newton Verfahren funktioniert. Allerdings fällt es sofort auf, dass das Newton Verfahren nicht für alle Startwerte $x^{(0)}$ funktioniert.

In Abbildung 3.2 gehen die Tangenten viel zu weit weg voneinander, so dass gilt $\left|x^{(k)}\right| \longrightarrow \infty$

In Abbildung 3.3 ist es überhaupt nicht klar, welche der beiden Nullstellen gefunden wird.

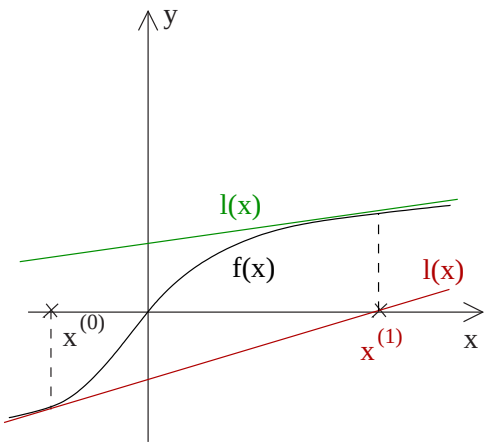


Abbildung 3.2: Newton Verfahren, Beispiel 2

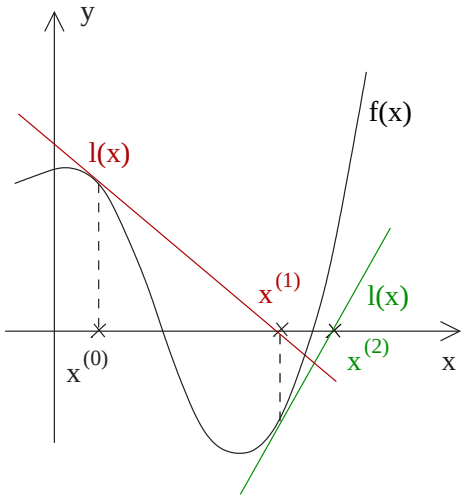


Abbildung 3.3: Newton Verfahren, Beispiel 3

Satz 3.2 (Konvergenz des Newton-Verfahrens)
Sei $f \in C^2(a,b)$ und es es existiere $x^* \in (a,b)$ mit $f(x^*) = 0$. Sei $m := \min\limits_{a \le x \le b} |f'(x)| > 0$ (Hauptanforderung) und $M := \max\limits_{a \le x \le b} |f''(x)|$.

Sei $\rho > 0$ so gewählt, dass $B_\rho := \{x \mid |x - x^*| < \rho\} \subset [a,b]$ und $q := \frac{M}{2m}\rho < 1$. Dann konvergiert das Newton-Verfahren für jeden Startwert $x^{(0)} \in B_\rho$.

Es gilt die a-priori Fehlerschranke:

(a) $\left|x^{(k)} - x^*\right| \le \frac{M}{2m}\left|x^{(k-1)} - x^*\right| \le \frac{2m}{M} q 2^k$

Und für die a-posteriori Schranke

(b) $\left|x^{(k)} - x^*\right| \le \frac{1}{m}\left|f(x^{(k)}\right| \le \frac{M}{2m}\left|x^{(k)} - x^{(k-1)}\right|^2$

Bemerkung: Aus dem Mittelwertsatz folgt
$\left|\frac{f(x)-f(y)}{x-y}\right| = |f'(\xi)| \ge m \;\forall\; x,y \in [a,b];\; x \ne y \implies |x-y| \le \frac{1}{m}|f(x) - f(y)|$

$\Longrightarrow$ x^* ist die einzige Nullstelle in $[a,b]$ und x^* ist **einfache** Nullstelle, d.h. $f(x^*)=0$ und $f'(x)\neq 0$.

Beweis: Mit Hilfe vom Satz 1.33 (Taylor)

(1) $f(x)=f(x)+f'(x)(y-x)+R(y,x)$ mit

$$R(y,x)=\int_x^y f''(\xi)(y-\xi)\mathrm{d}\xi=(y-x)^2\int_0^1 f''(x+s(y-x))(1-s)\mathrm{d}s$$

(2) $|R(y,x)|\leq |y-x|^2 M\int_0^1(a-s)\mathrm{d}s=\frac{M}{2}|y-x|^2$

$\Phi(x):=x-\frac{f(x)}{f'(x)}$ für $x\in B_\rho(x^*)$

$$\begin{aligned}\Longrightarrow\quad |\Phi(x)-x^*| &= \left|(x-x^*)-\tfrac{f(x)}{f'(x)}\right|=\left|-\tfrac{1}{f'(x)}\left[f(x)-(x^*-x)f(x)\right]\right|\\ &\overset{(1)}{=} \left|-\tfrac{1}{f'(x)}R(x^*,x)\right|=\tfrac{1}{|f'(x)|}|R(x^*,x)|\overset{(2)}{\leq}\tfrac{1}{m}|x-x^*|^2\tfrac{M}{2}\end{aligned}$$

$\Longrightarrow$ für $x\in B_\rho(x^*)$:

$$\begin{aligned}|\Phi(x)-x^*| &\leq \tfrac{M}{2m}|x-x^*|^2,\ \text{da } x\in B_\rho(x^*)\\ &\leq \tfrac{M}{2m}\rho^2=q\rho<\rho,\ \text{da } q<1\end{aligned}$$

$\Longrightarrow$ $x^{(k)}\in B_\rho(x^*)$ falls $x^{(0)}\in B_\rho(x^*)$

Sei $\rho^{(k)}=\frac{M}{2m}\left|x^{(k)}-x^*\right|$, so gilt

$$\begin{aligned}\rho^{(k)} &= \tfrac{M}{2m}\left|\Phi\left(x^{(k-1)}\right)-x^*\right|\leq\tfrac{M}{2m}\left(\tfrac{M}{2m}\left|x^{(k-1)}-x^*\right|^2\right)\\ &= \left(\rho^{(k-1)}\right)^2\leq\ldots,\leq\left(\rho^{(0)}\right)^{2^k}\end{aligned}$$

$$\begin{aligned}\Longrightarrow\quad \left|x^{(k)}-x^*\right| &\leq \tfrac{2m}{M}\rho^{(k)}\leq\tfrac{2m}{M}\left(\rho^{(0)}\right)^{2^k}=\tfrac{2m}{M}\left(\tfrac{M}{2m}\left|x^{(0)}-x^*\right|^2\right)^{2^k}\\ \left(x^{(0)}\in B_\rho(x^*)\right) &\leq \tfrac{2m}{M}\left(\tfrac{M}{2m}\right)^{2^k}=\tfrac{2mq^{2^k}}{M}\end{aligned}$$

$\Longrightarrow$ A-priori Abschätzung und $q<1$ und $q^{(2^k)}\longrightarrow 0$ $\Longrightarrow$ $x^{(k)}\longrightarrow x^*$

Für die A-posteriori Abschätzung benutzen wir nochmal (1)

$$\begin{aligned}f(x^{(k)}) &= f(x^{(k-1)})f\left((x^{(k)}-x^{(k-1)}\right)f'(x^{(k-1)})R(x^{(k)},x^{(k-1)})\\ &= R(x^{(k)},k^{(k-1)})\ \text{da } x^{(k)}=x^{(k-1)}-\tfrac{f(x^{(k-1)})}{f'(x^{(k+1)})}\end{aligned}$$

$$\begin{aligned}\Longrightarrow\quad \left|x^{(k)}-x^*\right| &\overset{MWS}{\leq} \left|f(x^{(k)}-f(x^*)\right|=\tfrac{1}{m}\left|f(x^{(k)})\right|,\ \text{da } f(x^*)=0\\ &= \tfrac{1}{M}R(x^{(k)},x^{(k-1)})\leq\tfrac{M}{2}\left|x^{(k)}-x^{(k-1)}\right|\quad\square\end{aligned}$$

Bemerkung: Falls $x^{(0)} \in B_\rho(x^*)$, so konvergiert das Newton-Verfahren sehr schnell. Sei zum Beispiel $q = \frac{1}{2}$, so gilt nach 10 Iterationen $\left|x^{(0)} - x^*\right| \leq \frac{2m}{M} q^{1024} \sim \frac{2m}{M} \omega^{-303}$

beim ISV it dem selben Startwert:
$|b - a| = \rho = q\frac{2m}{M} = \frac{m}{M}$, wobei $q = \frac{1}{2}$ gilt. und so viele Schritte
$\left|x^{(10)} - x^*\right| = 2^{-11}\,|b - 1| = 2^{11}\frac{2m}{M}2^{-10} \sim 10^{-11}\frac{m}{M}$

Offene Fragen

1. Wie kann ρ effektiv bestimmt werden? Wie ist die Lokalisierung von x^*?
2. Kann die Berechnung von f' umgegangen werden
3. Wass passiert falls $f'(x^*) = 0$, d.h x^* mehrfache Nullstelle?
4. Gibt es noch schnellere Verfahren?

Folgerung 3.3
Für $f \in C^2(\mathbb{R})$ existiert für jede einfache Nullstelle x^* eine Umgebung U um den Wert x^*, so dass das Newton-Verfahren für alle $x^{(0)} \in U$ konvergiert und $\left|x^{(k)} - x^*\right| \leq q^{2^k}$ für $q < 1$.

Beweis: z.z.: es existiert eine Umgebung U von x^* mit $m := \min_{x \in U} |f'(x)| > 0$ (dann wähle a, b mit $[a, b] \subset U$, $x^* \in [a, b]$ und wende den Satz 3.2 an, so dass ein U existiert, da $f' \in CC^0(\mathbb{R})$ und $f'(x^*) \neq 0$ □

c) **Kombination von Newton und ISV**[1]

Idee: ISV einsetzen, um ausreichend nah an einer Nullstelle x^* zu kommen, so dass das newton-Verfahren schnell konvergiert.

Verfahren: Voraussetzungen: $a < b$ gegeben mit $f(a)f(b) < 0$

Setze $x := \frac{1}{2}(a + b),\ \widetilde{a} = a,\ \widetilde{b} = b,\ f_0 = f(x),\ f_a = f(a)$

Solange $|f_0| > TOL$

$$\left[\begin{array}{l}
\texttt{Falls } f_a f_0 < 0 \texttt{ dann } \widetilde{b} = x \texttt{ sonst } (\widetilde{a} = x;\ f_a = f_0) \\
x = x - \frac{f_0}{f'(x)} \\
f_1 = f(x) \\
\qquad \texttt{Falls } \ (|f_1| > |f_0| \texttt{ oder } x \notin (a, b)) \texttt{ dann} \\
\qquad\qquad \left[a = \widetilde{a},\ b = \widetilde{b},\ x = \frac{1}{2}(a + b),\ f_1 = f(x)\right] \\
f_0 = f_1
\end{array}\right.$$

[1] Das ist die Fortsetzung der Verfahren ab Seite 48

d) **Sekantenverfahren**

Der Nachteil des Newton-Verfahrens ist die Auswertung von f'.

Idee:

$$f'(x^{(k)}) \sim \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$

Verfahren:

$$x^{(k+1)} = x^{(k)} = f\left(x^{(k)}\right) \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})}$$

Geometrische Interpretation:

Die Sekante an f durch die Punkte $x^{(k)}, x^{(k-1)}$ ist gegeben durch

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

wobei y die Geradengleichung durch die Punkte $\left(x^{(k-1)}, f(x^{(k-1)})\right)$, $\left(x^{(k)}, f(x^{(k)})\right)$ ist.

$$\Longrightarrow \quad \left[y = \frac{f(x^{(k-1)}) - f(x^{(k)})}{x^{(k-1)} - x^{(k)}} \left(x - x^{(k)}\right) + f(x^{(k)})\right]$$

Dann $x^{(k+1)}$ als Nullstelle der Sekante wählen.

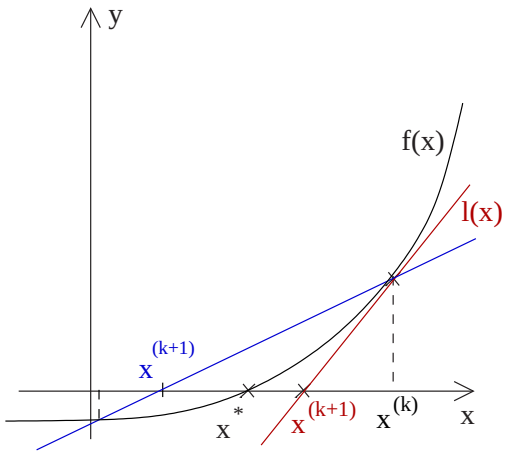


Abbildung 3.4: Sekantenverfahren, geometrische Interpretation

Bemerkung: Das Verfahren erfordert 2 Startwerte $x^{(0)}$, $x^{(1)}$ (siehe Abb. 3.4). Das Verfahren erfordert keine Auswertung von f pro Iterationschritt (speichern von $f(x^{(k-1)})$

Satz 3.4 (Konvergenz des Sekantenverfahrens)
Sei $f \in C^2(a,b)$ mit $x^* \in (a,b)$, $f(x^*) = 0$ und $m := \min\limits_{a \le x \le b} |f'(x)| > 0$, $M := \max\limits_{a \le x \le b} |f''(x)|$. Sei $q := \frac{M}{2m}\rho < 1$ für $\rho > 0$. Sei $x^{(0)}$, $x^{(1)} \in B_\rho(x^*)$, $x^{(0)} \neq x^{(1)}$

Sei $\left(x^{(k)}\right)_{k \in \mathbb{N}}$ definiert durch das Sekantenverfahren, dann gilt $x^{(k)} \in B_\rho(x^*)$ und $x^{(k)} \longrightarrow x^*$ für $k \longrightarrow \infty$

(a) A-priori Fehlerschranke $\left|x^{(k)} - x^*\right| \le \frac{2m}{M} q^{\gamma_k}$, wobei $(\gamma_k)_{k \in \mathbb{N}}$ die Folge der Fibonacci-Zahlen, d.h. $\gamma_0 = \gamma_1 = 1$; $\gamma_{k+1} = \gamma_k + \gamma_{k-1}$

(b) A-posteriori Fehlerschranke:

$$\left|x^{(k)} - x^*\right| \leq \frac{1}{m}\left|f(x^{(k)})\right| \leq \frac{M}{2m}\left|x^{(k)} - x^{(k-1)}\right| \cdot \left|x^{(k)} - x^{(k-1)}\right|$$

Folgerung 3.5 (Konvergenz des Sekantenverfahrens)
Für $f \in C^2(\mathbb{R})$ existiert eine Umgebung U um jede einfache Nullstelle, so dass $x^{(k)} \longrightarrow x^*$ für $x^{(0)},\ x^{(1)} \in U$ und es gilt $\left|x^{(k)} - x^*\right| \leq \frac{2m}{M}\widetilde{q}^{\alpha^k}$ mit $\widetilde{q} > 1$ und $\alpha = \frac{1}{2}(1+\sqrt{5}) \approx 1.618$

Beweis: (vgl. Folgerung 3.3). Mit der A-priori-Abschätzung von Satz 3.4 ist noch zu Zeigen $q^{\gamma_k} \leq \widetilde{q}^{\alpha^k},\ \widetilde{q} < 1$ geeignet.

Ansatz: $\gamma_k = \lambda^k$ mit $\lambda > 0$. Aus $\gamma_n - \gamma_{n-1} - \gamma_{n-2} = 0$ folgt

$$\lambda^n(\lambda^2 - \lambda - 1) = 0 \iff \lambda^2 - \lambda - 1 = 0 \implies \lambda_{1/2} = \frac{1}{2}(1 \pm \sqrt{5})$$

D.h. die allgemeine Lösung der Differenzengleichung $\gamma_n - \gamma_{n-1} - \gamma_{n-2} = 0$ ist

$$\gamma_n = c_1\lambda_1^k + c_2\lambda_2^k$$

für $c_1, c_2 \in \mathbb{R}$

Da $\gamma_0 = \gamma_1 = 1 \implies c_1 = \frac{\lambda_1}{\sqrt{5}},\ c_2 = -\frac{\lambda_2}{\sqrt{5}}$. Also $\gamma_k = \frac{1}{5}\left(\lambda_1^{k+1} - \lambda_2^{k+2}\right)$. Da $|\lambda_2| < |\lambda_1|$ und für k groß genug gilt

$$\begin{aligned} q^{\gamma_k} &\leq q^{\frac{1}{\sqrt{5}}\left(\lambda_1^k\right)},\ \text{da } q < 1 \\ &= \widetilde{q}^{\left(\gamma_1^k\right)} \qquad (\alpha = \lambda_1) \end{aligned}$$ □

Beweis: (Satz 3.4) (Ähnlich wie Satz 3.2)

(i) $|x - y| \leq \frac{1}{m}|f(x) - f(y)|\ \forall\ x, y \in B_\rho(x^*)$

(ii) $\left|\frac{f(x)-f(y)}{x-y} - \frac{f(x)-f(z)}{x-z}\right| \overset{\text{Taylor}}{\leq} |z - y|\,\frac{M}{2}\ \forall\ x, y, z \in B_\rho(x^*)$

$\Phi(x, y) = x - f(x)\frac{x-y}{x-z}\ x, y \in B_\rho(x^*)$

$\implies |\Phi(x, y) - x^*| = \frac{|x-y|\cdot|x-x^*|}{|f(x)-f(y)|}\left|\frac{f(x)-f(y)}{x-y} - \frac{f(x)-f(x^*)}{x-y^*}\right|$

$\implies |\Phi(x, y) - x^*| \leq \frac{1}{m}|x - x^*| \cdot |y - x^*|\,\frac{M}{2} < \rho$

$\implies x^{(k+1)} \in B_\rho(x^*)$ falls $x^{(0)}, x^{(1)} \in B_\rho(x^*)$

Mit $\rho^{(k)} = \frac{M}{2m}\left|x^{(k)} - x^*\right| \implies \rho^{(k+1)} \leq \rho^{(k)} \leq q^{\gamma_{n+1}}$

$\implies$ A-priori Abschätzung.

zu b) $\left|x^{(k)} - x^*\right| \overset{1.}{\leq} \frac{1}{m}\left|f(x^{(k)})\right| \overset{2.}{\leq} \frac{M}{2m}\left|x^{(k)} - x^{(k-1)}\right| \cdot \left|x^{(k)} - x^{(k-1)}\right|$.
Weitere Details siehe Übungsblätter □

Zusammenfassung

ISV: $\left|x^{(k)} - x^*\right| \leq \frac{(b-a)}{2}\left(\frac{1}{2}\right)^k$ bei einer Funktionsauswertung für $x^{(k)} \longrightarrow x^{(k+1)}$

Newton-Verfahren: $\left|x^{(k)} - x^*\right| \leq \frac{M}{2m}q^{2^k}$ bei einer Funktionsauswertung und einer Funktionsauswertung der Ableitung.

Sekanten-Verfahren: $\left|x^{(k)} - x^*\right| \leq \frac{M}{2m}\widetilde{q}^{1.628^k},\ (\widetilde{q} < 1)$ bei einer Funktionsauswertung.

Annahme: Die Auswertung von f' ist ungefähr so aufwändig wie die Auswertung von f, d.h. das Newton-Verfahren hat doppelten Aufwand.

Def. $z^{(k)} := x^{2k}$ bei ISV bzw. beim Sekanten-Verfahren $\Longrightarrow$ der Aufwand um aus $z^{(k)}$ den Wert $z^{(k+1)}$ zu berechnen erfordert zwei Funktionsauswertungen.

ISV: $\left|z^{(k)} - x^*\right| \leq \frac{(b-a)}{2}\left(\frac{1}{4}\right)^k$

$$\begin{array}{rcl} \text{Sekanten} - \text{Verfahren}: \left|z^{(k)} - x^*\right| & \leq & \frac{M}{2m}\widetilde{q}^{(\lambda_1^{2k})}, \\ & \leq & \frac{M}{2m}\widetilde{q}^{(\lambda_1+1)^k} \\ & \leq & \frac{M}{2m}\widetilde{q}^{(2.618)^k} \end{array} \left[\begin{array}{c} \text{da } \lambda_1^2 - \lambda_1 - 1 = 0 \\ \Longrightarrow\ \lambda_1^2 = \lambda_2 + 1 \end{array}\right.$$

D.h. bei gleichen Aufwand konvergiert das Sekanten-Verfahren schneller als das Newton- oder ISV-Verfahren. Auch wenn das Sekanten-Verfahren schneller konvergiert als die anderen 2, sollte man es ganz gezielt anwenden, weil das Sekanten-Verfahren nicht immer die beste Lösung ist.

Problem beim Sekanten-Verfahren

Fehleranalyse wurde bisher ohne Berücksichtigung von Rundungsfehlern gemacht. Seien $x^{(k)},\ x^{(k-1)}$ sehr nah an x^*, dann liegen auch $f(x^{(k)}),\ f(x^{(k-1)})$ nahe zusammen und haben bei dem selben Vorzeichen Schwierigkeiten eine richtige Differenz $f(x^{(k)}) - f(x^{(k-1)})$ zu bilden, weil es Auflöschungen auftretten.

Einschub: Konvergenz von Iterationsverfahren

Allgemein: $x^{(k+1)} = \Phi\left(x^{(k)}, \ldots, x^{(k-j)}\right),\ j \geq 0$ fest (1)
Startwerte: $x^{(0)}, \ldots, x^{(j)},\ \Phi : X^j \longrightarrow X,\ X$ ist die Iterationsvorschrift in einem Banach-Raum X.

Definition 3.6
Das Iterationsverfahren (1) **konvergiert total** *gegen $x^* \in X$, falls es eine Umgebung U von x^* gibt, so dass für alle $x^{(0)}, \ldots, x^{(j)} \in U$: $x^{(k)} \longrightarrow x^*$ $(k \longrightarrow \infty)$*

Definition 3.7 (Konvergenzordnung)
Sei X ein Banach-Raum und $\left(x^{(k)}\right)_{k\in\mathbb{N}}$ Folge in X, die gegen ein $x^* \in X$ konvergiert. Die Folge hat mindestens die **Konvergenzordnung** *$p \geq 1$ falls*

$$\lim_{k \longrightarrow \infty} \sup \frac{\left\|x^{(k+1)} - x^*\right\|}{\left\|x^{(k)} - x^*\right\|^p} = c$$

mit $c < 1$ für $p = 1$ und $c < \infty$ für $p > 1$.

Die Ordnung ist genau p, falls $c \neq 0$. Der Fall $p = 1$ wird als **lineare Konvergenz** *bezeichnet und falls für ein $p > 1$ gilt:*

$$\lim_{k \longrightarrow \infty} \sup \frac{\left\|x^{(k+1)} - x^*\right\|}{\left\|x^{(k)} - x^*\right\|^p} = 0$$

so spricht man von **super linearer Konvergenz**.

Beispiel: Das Newton-Verfahren konvergiert quadratisch ($p = 2$), da $\left|x^{(k+1)} - x^*\right| \leq c \cdot \left|x^{(k)} - x^*\right|^2$
Das ISV Verfahren konvergiert linear, da $\left|x^{(k+1)} - x^*\right| \leq \frac{1}{2}\left|x^{(k)} - x^*\right|$

Satz 3.8
Sei $I = [a, b],\ \Phi : I \longrightarrow I,\ \Phi \in C^p(I)$

$x^{(0)} \in I,\ x^{(k+1)} = \Phi(x^{(k)}),\ k \geq 0$ mit $x^{(k)} \longrightarrow x^*$ mit $x^* = \Phi(x^*)$

(i) $p = 1,\ x^{(k)} \longrightarrow x^*$ mind. linear $\iff |\Phi'(x^*)| < 1$

(ii) $p \geq 2,\ x^{(k)} \longrightarrow x^*$ mind. mit der Ordnung $p \iff \Phi^{(\nu)}(x^*) = 0,\ (1 \leq \nu \leq p-1)$

Beweis: (i) Siehe den Banachschen Fixpunktsatz.

(ii) "$\Longrightarrow$" Annahme: $\exists\ j < p$ mit $\Phi^{(j)}(x^*) \neq 0$ (j soll minimal sein).

$$\begin{aligned} x^{(k+1)} - x^* &= \Phi(x^{(k)}) - \Phi(x^*) \\ &= \sum_{\nu=1}^{p-1} \frac{\Phi^{(\nu)}(x^*)}{\nu!}\left(x^{(k)} - x^*\right)^\nu + \Phi^{(p)}(\xi_k)\frac{(x^{(k)}-x^*)^p}{p!} \end{aligned}$$

mit ξ_k zwischen $x^{(k)}$ und x^*

$$\Longrightarrow \quad x^{(k+1)} - x^* = \frac{\Phi^{(j)}(x^*)}{j!}\left(x^{(k)} - x^*\right)^j \cdot \underbrace{\left[1 + \left(x^{(k)} - x^*\right)\sum_{\nu \leq j+1}^{p} a_\nu \left(x^{(k)} - x^*\right)^{\nu-j-1}\right]}_{:=b}$$

Für k groß: $|b| \geq \frac{1}{2}$, da $x^{(k)} \longrightarrow x^*$ und a_ν unabhängig von k beschränkt.

$\Longrightarrow \quad \left|x^{(k+1)} - x^*\right| \geq \frac{1}{2}\left|x^{(k)} - x^*\right|^j \frac{\left|\Phi^{(j)}(x^*)\right|}{j!}$. Da $x^{(k)} \longrightarrow x^*$ mit Ordnung p, $\infty > c \geq \frac{\left|x^{(k+1)}-x^*\right|}{\left|x^{(k)}-x^*\right|} \geq \frac{1}{2j!}\left|\Phi^{(j)}(x^*)\right| \cdot \left|x^{(k)} - x^*\right|^{j-p} \longrightarrow \infty$ ist Widerspruch zur Annahme (mit $j - p < 0$)

"$\Longleftarrow$"

$$\begin{aligned} x^{(k+1)} - x^* &= \sum_{\nu=1}^{p-1} \frac{\Phi^{\nu}(x^*)}{\nu!} \left(x^{(k)} - x^*\right)^{\nu} + \frac{\Phi^{(p)}(\xi_k)}{p!} \left(x^{(k)} - x^*\right)^{p} \\ &= \frac{\Phi^{(p)}(\xi_k)}{p!} \left(x^{(k)} - x^*\right)^{p}, \text{ da } \Phi^{(\nu)}(x^*) = 0 \ (1 \le \nu \le p) \\ \Longrightarrow \quad \frac{\left|x^{(k+1)} - x^*\right|}{\left|x^{(k)} - x^*\right|^{p}} &= \frac{1}{p!}\Phi^{(p)}(\xi_k) \longrightarrow \frac{1}{p!}\Phi^{(p)}(x^*) \ (k \longrightarrow \infty) \end{aligned}$$

da ξ_k zwischen $x^{(k)}$ und x^* und $x^{(k)} \longrightarrow x^*$ $(k \longrightarrow \infty)$ □

e) **Verfahren höher Ordnung** $(p = 3)$[2]

1. Idee: Linear Kombination von 2 Verfahren zweiter Ordnung. Seien $\Phi_0,\ \Phi_1$ Iterationsverfahren, die quadratisch konvergieren.

$\Phi_s(x) = (1-s)\Phi_0(x) + s\Phi_1(x)$

$\Longrightarrow$ $\Phi_s'(x^*) = (1-s)\Phi_0(x^*) + s\Phi_1(x^*) = 0$, da nach Satz 3.7 $\Phi_0(x^*) = 0,\ \mathbb{I}_1'(x^*) = 0$

Wähle s so, dass $I_s''(x^*) = 0$ $\Longrightarrow$ Φ_s konvergiert mit 3. Ordnung nach Satz 3.7 (Beispiel: siehe Übungsblatt).

Idee mit dem Newton-Verfahren

$$\begin{aligned} \Phi(x) &= x - \tfrac{f(x)}{f'(x)};\ f(x^*) = 0 \implies \Phi(x^*) = x^*,\ f'(x^*) \neq 0 \\ \Phi'(x) &= 1 - \tfrac{f'(x)^2 - f(x)f''(x)}{f'(x)^2} = 1 - 1 - 0 = 0 \text{ da } f(x^*) = 0 \end{aligned}$$

$\Longrightarrow$ mit Satz 3.7 konvergiert das Newton-Verfahren quadratisch.i

2. Idee: $\Phi(x) = x - g(x)f(x) - h(x)f(x)^2$ mit $g(x) \neq 0,\ h(x) \neq 0$ in einer Umgebung von x^*. $\Phi(x^*) = x^* \iff f(x^*) = 0$

Bestimme $g,\ h$, so dass $\Phi'(x^*) = \Phi''(x^*) = 0$

Sei x^* einfache Nullstelle. $\Phi'(x) = 1 - f'(x)g(x) - f(x)g'(x) - 2f(x)f'(x)h(x) - h'(x)f^2(x)$ $\Longrightarrow$ $\Phi'(x^*) = 0 \overset{f(x^*)=0}{\Longleftrightarrow} 1 - f'(x^*)g(x^*) = 0.$

Wähle $g(x) = \frac{1}{f'(x)}$

$$\begin{aligned} \Phi''(x) &= -f'(x)\left(g'(x) + h(x)f(x) + 2h(x)f'(x)\right) - f(x)(\cdots) - \underbrace{\underbrace{\left(g(x)f'(x)\right)'}_{=1}}_{=0}, \quad g''(x) = \tfrac{f''(x)}{f'(x)^2} \\ &= \tfrac{f''(x)}{f'(x)} - 2h(x)f'(x) - f(x)(\cdots) \\ \Longrightarrow \quad \Phi''(x) &= \tfrac{f''(x)}{f'(x)} - 2h(x^*)f'(x^*)^2 = 0 \end{aligned}$$

Wähle: $h(x) = \frac{1}{2f'(x)^2}\frac{f''(x)}{f'(x)} = \frac{f''(x)}{2f'(x)^3}$

[2]Das ist die Fortsetzung der Verfahren ab Seite 52

Folgerung 3.9
Das Verfahren

$$\Phi(x)x - \frac{f(x)}{f'(x)} - \frac{1}{2}\frac{f(x)^2 f''(x)}{f'(x)^3}$$

konvergiert mit dritter (3) Ordnung in einer Umgebung einer einfachen Nullstelle x^* von f.

f) **Newton-Verfahren für mehrfache Nullstellen**[3]

Definition 3.10
Sei $f \in C^n(I)$ $(n \geq 1)$ *hat eine* **Nullstelle** x^* **der Ordnung** $(n-1)$ *bzw. der* **Vielfachheit** n *gdw.* $f^{(\nu)}(x^*) = 0,\ 0 \leq \nu \leq n-1$ *und* $f^{(n)}(x^*) \neq 0$.

Satz 3.11
Sei $f \in C^{n+1}(a,b),\ n \geq 1$ und $x^* \in (a,b)$ eine einfache Nullstelle von f und $f^{(k)}(x) \neq 0$ $(x \neq x^*)$ $0 \leq k \leq n$ und $f^{(n+1)}(x) \neq 0$ für $x \in (a,b)$. Dann existiert eine Umgebung von x^*, so dass der Newton-Verfahren mindestens linear konvergiert.

Beweis:

$$\Phi(x) = \begin{cases} x - \frac{f(x)}{f'(x)} & : \ x \neq x^* \\ x^* & : \ x = x^* \end{cases}$$

Behauptung: $\Phi \in C^1(a,b)$ und $|\Phi(x)| < 1$. Zusammen mit Satz 3.8 folgt die Behauptung des Satzes.

$$\lim_{\substack{x \longrightarrow x^* \\ x \neq x^*}} \Phi(x) = x^* - \lim_{x \longrightarrow x^*} \frac{f(x)}{f'(x)} \overset{\text{L'Hôspital}}{=} x^* - \lim_{x \longrightarrow x^*} \frac{f^{(n-1)}(x)}{f^{(n)}(x)} = x^* + \frac{0}{f^{(n)}(x^*)} = x^*$$

$$\Phi'(x) = \frac{1 - f'(x)^2 - f(x)f'(x)}{f'(x)^2} = \frac{f(x)f''(x)}{f'(x)^2}, \qquad x \neq x^*$$

$\lim_{x \longrightarrow x^*} \Phi'(x) = 1 - \frac{1}{n} \implies |\Phi'(x^*)| < \frac{1}{n}$. Der Beweis geht mit Taylor.

$f^{(p)}(x) = f^{(n)}(\xi_p)\frac{(x-x^*)(n-p)}{(n-p)!},\ p = 0,1,2$ mit ξ_p zwischen x und x^*, d.h. $\lim_{x \longrightarrow x^*} \xi_p = x^* \qquad (p = 0,1,2)$

1. Fall: Die Vielfachheit.

Dann ist $\Phi'(x) = 1 - \frac{1}{n}$

Wähle $\Phi_\alpha(x) = x - \alpha\frac{f(x)}{f'(x)},\ \alpha \in \mathbb{R}$ fest $\implies \Phi'_\alpha(x^*) = 1 - \frac{\alpha}{n} = 0 \iff \alpha = n$ □

Folgerung 3.12
Sei $f \in C^{n+1}(a,b),\ n \geq 1$ und x^* eine einfache Nullstelle, dann konvergiert das Verfahren

$$x^{(k+1)} = x^{(k)} - n\frac{f(x^{(k)})}{f'(x^{(k)})}$$

quadratisch gegen x^*.

[3]Das ist die Fortsetzung der Verfahren ab Seite 57

2. Fall: Die genaue Ordnung der Nullstelle ist nicht bekannt. Ansatz: $F(x) = \frac{f(x)}{f'(x)}$. Dann gilt $F(x^*) = 0$ (vgl. Satz 3.11) und $F'(x) \longrightarrow \frac{1}{n}$ $(x \longrightarrow x^*) \implies x^*$ ist eine einfache Nullstelle.

Folgerung 3.13
Das Verfahren

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})f'(x)^2}{f(x^{(k)})(f'(x^{(k)}))^2 - f''(x^{(k)})f(x^{(k)})} = x^{(k)} - \frac{F(x^{(k)})}{F'(x^{(k)})}$$

konvergiert lokal quadratisch gegen x^*

3.2 Nullstellen bei Polynomen

Gegeben: $p(x) = a_n x^n + \cdots + a_1 x + a_0 = \sum_{k=0}^{n} a_k x^k$

Gesucht: $x^* \in \mathbb{R}$ (oder $x^* \in \mathbb{C}$) mit $p(x^*) = 0$

Bemerkung: Häufig $p(x) = \sum_{k=0}^{n} \hat{a}_{n-k} x^k$ in der Literatur ohne Einschränkung $a_0 \neq 0$ (sonst ist $x^* = 0$ eine Nullstelle)

Definition 3.14
p heißt **Polynom n-ten Grades**, *falls $a_n \neq 0$. Die Menge aller Polynome höchstens von Grad n ist P_n, $\dim P_n = n + 1$ und $\{1, x, x^2, \ldots, x^n\}$ ist eine Basis von P_n (Basis der Monome)*

Satz 3.15 (Fundamentalsatz der Algebra)
Ein Polynom n-ten Grades hat genau n Nullstellen über $\mathbb{C}$, wenn x^* eine Vielfachheit q hat, dann wird $x^* q$ Mal gezählt. Daher hat p die Gestalt $p(x) = \prod_{i=1}^{n}(x - z_i)$, wobei $z_1, \ldots, z_n$ die Nullstellen von p sind.

Sind $z_i \in \mathbb{C}\backslash\mathbb{R}$, dann ist auch $\overline{z_i} \in \mathbb{C}\backslash\mathbb{R}$ eine Nullstelle von p. Die beiden Faktoren

$$(x - z_i)(x - \overline{z_i}) = x^2 + b_i x + c_i,\ b_i, c_i \in \mathbb{R}$$

$z_{n_1+1}, \ldots, z_n$ komplexe Nullstellen, dann

$$p(x) = \prod_{i=1}^{n}(x - z_i) = \prod_{k=1}^{n_2}(x^2 + a_i x + b_i),\ n_2 = n - n_1$$

Auswertung von Polynomen (Horner-Schema)

Sei $x_o \in \mathbb{R}$ fest. Zur Effizienzsteigerung und Verringerung von Rundungsfehlern wird $p(x_0)$ ausgewertet in der Form:

$$p(x_0) = a_0 + x_0\,(a_1 + x_0\,(a_2 + x_o\,(\cdots x_0 a_n)\cdots)\cdots)$$

Dies wird als **einfaches Horner-Schema** bezeichnet.

Algorithmus:

$a = a_n\ \ (=: a'_n)$
$i = n-1, \ldots, 0:\ \ a = a_i + a x_0\ \ (=: a'_i)$
`Dann ist` $a = a'_0 = p(x_0)$

Folgerung 3.16
Das einfache Horner-Schema liefert die Entwicklung $p(x) = a'_0 + (x - x_0)\left(a'_1 + a'_2 x + \cdots + a'_n x^{n-1}\right)$ mit $a'_n = a_n$ und $a'_k = a_k + a'_{k+1} x_0^{n-1}$ $(k = 0, \ldots, n-1)$

Es gilt: $p'(x_0) = a'_1 + a'_2 x_0 + \cdots + a'_n x_0^{n-1}$

Beweis: Taylor: $p(x) = p(x_0) + \sum_{k=1}^{n} \frac{p^{(k)}(x_0)(x-x_0)^k}{k!} = p(x_0) + (x - x_0)q(x)$

Ansatz: $q(x) = a'_1 + a'_2 x + \cdots + a'_n x^{n-1} = \sum_{k=1}^{n} a'_k x^{k-1}$

Damit: $p(x_0) + (x - x_0) \sum_{k=0}^{n} a'_k x^{k-1} \overset{\text{z.z}}{=} \sum_{k=0}^{n} a_k x^k$
gdw. $p(x_0) + \sum_{k=1}^{n} a'_k x^k - x_0 \sum_{k=0}^{n} a'_{k+1} x^k \overset{!}{=} \sum_{k=0}^{n} a_k x^k$

Koeffizienten: $a_k = a'_k - x_0 a'_{k+1}$ $(k = 1, \ldots, n-1)$

$k = n$**:** $a'_n = a_n$
$k = 0$**:** $p(x_0) - x_0 a_1 = a_0 \implies a'_0 - x_0 a'_1 = a_0$

$\implies k = n : a'_n = a_{n_1},\ 0 \le k \le n-1 :\ a'_k - x_0 a'_{k+1} = a_k$. Nach Def. ist a'_k gegeben.

Der Wert von $p'(x_0)$ läßt sich wieder mit dem einfachen Horner-Schema für $q(x) = a'_1 + a'_2 x + \cdots + a'_n x^{n-1}$. auswerten.

Liefert $a''_j = a'_j + a''_{j+1} x_0$ $(j = n-1, \ldots, 1),\ a''_n = a'_n$ mit
$p(x) = a''_0 + (x - x_0)a''_1 + (x - x_0)^2(a''_2 + a''_3 x) + \cdots + a''_n x^{n-2}$ mit $\frac{1}{2}p(x) = a''_2 + a''_3 x_0 + \cdots + a'_n x_0^{n-2}$ (analog zu Folgerung 3.16)

Durch wiederholtes Auswerten entsteht das **vollständige Honer-Schema**

$$p(x) = a'_0 + (x - x_0)a''_1 + (x - x_0)^2 a'''_2 + \cdots + a_n^{(n+1)}(x - x_0)^n$$

mit $a_k^{(k+1)} = \frac{p^{(k)}(x_0)}{k!}$.

Algorithmus:

`Gegeben: Vektor` $a = (a_0, \ldots, a_n),\ m \le n$
$k = 0, \ldots, m$
$l = n-1, \ldots, k\,[a_l = a_l + x_0 a_{l+1}]$
`Dann ist` $a = \left(\frac{p^{(0)}(x_0)}{0!}, \frac{p^{(1)}(x_0)}{1!}, \ldots, \frac{p^{(m)}(x_0)}{m!}\right)$

Newton-Verfahren mit Horner-Schema: $m = 1 \implies$

$$x^{(k+1)} = x^{(k)} - \frac{p(x^{(k)})}{p'(x^{(k)})}$$

Satz 3.17
Sei p en Polynom mit n reellen Nullstellen $z_1 \ge z_2 \ge \ldots \ge z_n$ und $x^{(i)} > z_i$, dann ist $\left(x^{(k)}\right)_{k \in \mathbb{N}}$ monoton fallend und $x^{(k)} \longrightarrow z$

Beweis: Siehe Übungsblätter

Satz 3.18 Schranke für Nullstelle
Sei $p(x^*) = 0$, ohne Einschränkung $a_0 \neq 0,\ a_n = 1$. Dann gilt: $0 < |x^*| \leq \min\{A, B, C\},\ A = \sum_{k=0}^{n-1} |a_k|,$
$B = 1 + \max_{0\leq k\leq n-1} |a_k|\,,\ c = 2 \max_{0\leq k\leq n-1} |a_k|^{\frac{1}{k}-k}$

Da $a_0 \neq 0 \implies |x^*| > 0$ und da $B \geq 1 \implies |x^*| \leq B$ für $|x^*| < 1$. Sei $|x^*| \geq 1$. Es gilt:
$|x^*|^n = |x^*|^n = \left|-\sum_{k=0}^{n-1} a_k (x^*)^k\right| \leq \sum_{k=0}^{n-1} |a_k|\,|x^*|^k \qquad (a_n = 1)$

z.z: $|x^*| \leq A,\ |x^*| \leq B,\ |x^*| \leq C$

Satz 3.19 (A-posteriori Abschätzung)
Sei $p \in P_n$, $grad(p) = n$ und $z_1, \ldots, z_n \in \mathbb{C}$ die Nullstellen von p und $a \neq 0,\ a_n = 1$ (d.h. p ist normiert). Sei z^* die Nährung einer Nullstelle, d.h. $|p(z^*)| < \varepsilon$. So gibt es mindestens eine Nullstelle z_k, so dass für den relativen Fehler gilt:

$$\frac{|z^* - z_k|}{|z_k|} \leq \sqrt[n]{\frac{\varepsilon}{|a_0|}}$$

Beweis: $p(x) = \prod_{k=1}^{n} (x - z_k),\ p(0) = a_0$

$\implies a_0 = p(0) = (-1)^n \prod_{k=1}^{n} z_k \implies |p(x)| = \prod_{k=1}^{n} |x - z_k|\,,\ |a_0| = \prod_{k=1}^{n} |z_k|$

Annahme: $\forall\ k \in \{1, \ldots, n\} :\ \frac{|z^*-z_k|}{|z_k|} > \sqrt[n]{\frac{\varepsilon}{|a_0|}}$

$\implies \frac{\varepsilon}{|a_0|} \overset{\text{Voraussetzung}}{\geq} \frac{|p(z^*)|}{|a_0|} = \prod_{k=1}^{n} \frac{|z^*-z_k|}{|z_k|} > \left(\sqrt[n]{\frac{\varepsilon}{|a_0|}}\right)^n = \frac{\varepsilon}{|a_0|}$
Widerspruch zur Annahme! □

g) **Deflation**[4]

Ist eine Nullstelle z_m von p berechnet, so ist $p(x) = (x - z_m)q(x)$ mit $grad(q) = n-1$ und $q(x) = \frac{p(x)}{x-z_m}$

Weitere Nullstellen durch das Newton-Verfahren angewandet auf q

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - \frac{q(x^{(k)})}{q'(x^{(k)})} = x^{(k)} - \frac{p(x^{(k)})}{(x^{(k)}-z_m)\ \frac{p'(x^{(k)})}{z^{(k)}-z_m} - \frac{p(x^{(k)})}{(x^{(k)}-z_m)^2}} \\ &= x^{(k)} - \frac{p(x^{(k)})}{p'(x^{(k)}) - \frac{p(x^{(k)})}{x^{(k)}-z_m}} \end{aligned}$$

D.h. das Verfahren kann benutzt werden, ohne q zu bestimmen.

Problem: Fehler, falls z_m nur approximativ bekannt.

Ist z_m^* eine Approximation, dann erhalten wir eine Division mit Rest: $p(x) = (x - z_m^*)q(x) + r(x)$. Daher $(*)$ verwenden, um Startwert für Newton-Iteration angewandet auf p zu erhalaten.

[4]Das ist die Fortsetzung der Verfahren ab Seite 58

h) **Verfahren von Barrstow**

Idee: quadratischen Faktor $h(x) = x^2 + bx + c$ direkt abspalten.

Division mit Rest: $p(x) = h(x)q(x) + r_1(b,c)x + r_2(b,c)$ mit $grad(q) = n-2$. Falls b, c so gewählt wurden, dass $r_1(b,c) = r_2(b,c) = 0 \implies$ die Nullstellen von h sind die Nullstellen von p, wobei die Restfunktionen ein nicht lineares Gleichungssystem für 2 Unbekannten (b,c) erzeugen.[5]

3.3 Nicht lineare Gleichungssysteme

Problem: $D \subset \mathbb{R}^n$ abgeschlossen und konvex $(n \geq 2)$,
$f : D \longrightarrow \mathbb{R}^n$, d.h. $f(x) = (f_1(x), \ldots, f_n(x))^\top,\ x \in D$

Gesucht: $x^* \in D$ mit $f(x^*) = 0$, d.h. $f_i(x^*) = 0\ (i = 1, \ldots, n)$
Annahme: ein solches x^* existiert in D.

i) **Newton-Verfahren** $(n \geq 2)$ [6]

Idee: $(n = 2),\ f(x) = (f_1(x_1, x_2), f_2(x_1, x_2))^\top$

Setze $f_{i,k} := \frac{\partial f_i}{\partial x_k},\ i, k = 1, \ldots, n$

$$Df(x) := \begin{pmatrix} f_{1,1}(x) & \cdots & f_{1,n}(x) \\ \vdots & \ddots & \vdots \\ f_{n,1}(x) & \cdots & f_{n,n}(x) \end{pmatrix} \in \mathbb{R}^{n \times n} \quad (\text{Jaccobi} - \text{Matrix})$$

Startwert $x^{(0)} \in \mathbb{R}^2$.
Taylor: $f_i(x_1, x_2) = f_i(x^{(0)}) + f_{i,1}(x^{(0)})(x_1 - x_1^{(0)}) + f_{i,2}(x^{(0)})(x_2 - x_2^{(0)}) + R_{i,1}(x_1, x_2) + R_{i,2}(x_1, x_2)$

Vernachlässige $R_{i,1}, R_{i,2}$ und setze $(x_1, x_2)^\top = x^*$

$$\implies 0 \approx f_i(x^{(0)}) + f_{i,1}(x^{(0)})(x_1 - x_1^{(0)}) + f_{i,2}(x^{(0)})(x_2 - x_2^{(0)})$$

Bestimme $x^{(1)}$, so dass die Null angenommen wird.

$$f_{1,1}(x^{(0)}) + f_{1,2}(x^{(0)})x_2^{(1)} = f_{1,1}(x^{(0)})x_1^{(0)} + f_{1,2}(x^{(0)})x_2^{(0)} - f_1(x^{(0)})$$

$$f_{2,1}(x^{(0)})x_1^{(1)} + f_{2,2}(x^{(0)})x_2^{(1)} = f_{2,1}(x^{(0)})x_1^{(0)} + f_{2,2}(x^{(0)})x_2^{(0)} - f_2(x^{(0)})$$

$$\text{bzw. } Df(x^{(0)}) \begin{pmatrix} x_1^{(0)} \\ x_2^{(1)} \end{pmatrix} = Df(x^{(0)} \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} - \begin{pmatrix} f_1(x^{(0)}) \\ f_2(x^{(0)}) \end{pmatrix}$$

$$\text{Ist } \det Df(x^{(0)}) \neq 0 \implies x^{(1)} = x^{(0)} - Df(x^{(0)})^{-1} f(x^{(0)})$$

[5] Mehr dazu im Numerik Buch vom Stör
[6] Das ist die Fortsetzung der Verfahren ab Seite 61

Algorithmus:

$x^{(0)}$ `gegeben. Für` $k \geq 0$
`1) Löse` $Df(x^{(0)})y^{(k)} = -f(x^{(k)})$
`2) Setze` $x^{(k+1)} = x^{(k)} + y^{(k)}$
`d.h.` $Df(x^{(k)})x^{(k+1)} = Df(x^{(k)})x^{(k)} - f(x^{(k)})$

Problem: In jedem Schritt muss ein $n \times n$ LGS gelöst werden. Häufig wird Df für einige Schritte festgehalten. Mit der L-R-Zerlegung können diese Schritte sehr einfach gelöst werden.

Satz 3.20
Sei $f_i \in C^2(\mathbb{R})$ und $Df(x^*)$ regulär (d.h. x^* einfache Nullstelle).

Dann existiert ein $r > 0$, so dass für alle $x^{(0)} \in B_r(x^*) := \{x \mid \|x - x^*\| < r\}$ gilt: $x^{(k)} \in B_r(x^*)$ und $x^{(k)} \overset{k}{\longrightarrow}{}^{\infty} x^*$ mindestens quadratisch.

Beweis: Analog zum Satz 3.2

Kapitel 4

Interpolation

Sei $\{\Phi(x_j, a_0, \ldots, a_n) \mid a_0, \ldots, a_n \in \mathbb{R}\}$ eine Familie von Funktionen mit $x_j \in \mathbb{R}$, deren Elemente durch $(n+1)$ Parameter $a_0, \ldots, a_n \in \mathbb{R}$ gegeben sind.

Aufgabe: Zu $(x_k, f_k) \in \mathbb{R}^2,\ k = 0, \ldots, n$ mit $x_i \neq x_k$ für $i \neq k$, finde Parameter $a_0, \ldots, a_n$, so dass

$$\Phi(x_k, a_0, \ldots, a_n) = f_k \text{ für } k = 0, \ldots, n.$$

Falls Φ linear von seinen Parametern abhängig, spricht man von einem **linearen Interpolationsproblem**. (x_k, f_k) sind zum Beispiel Messdaten oder diskrete Werte aus einem anderen numerischen Verfahren, oder $f_k = f(x_k)$ für eine (komplizierte) Funktion $f \in C^0$ und $x_0, \ldots, x_n$ sind die **Stützstellen**, an denen f interpoliert werden soll.

Beispiel: $V \subset C^0(\mathbb{R})$ Vektorraum und $\dim(V) = n+1,\ \varphi_0, \ldots, \varphi_n$ Basis von V.
$\Phi(x, a_0, \ldots, a_n) = \sum_{k=0}^{n} a_k \varphi_k(x)$, etwa $V = \mathbb{P}_n,\ \varphi_n(x) = x^k$, d.h. $\Phi(x, a_0, \ldots, a_n) = \sum_{k=0}^{n} a_k x^k$

Problem: Finde ein Polynom höchstens n-ten Grades, so dass $p(x_k) = f_k$ (**Polynominterpolation**)

Weitere wichtige Beispiele:

Trigonomerische Interpolation

$$\Phi(x, a_0, \ldots, a_n) = a_0 + a_1 e^{ix} + a_2 e^{2ix} + \cdots + a_n e^{nix} = a_0 + \sum_{k=1}^{k} a_k (\cos(kx) + i \cdot \sin(kx))$$

Exponentielle Interpolation (nicht linear)

$$\Phi(x, a_0, \ldots, a_n, \lambda_0, \ldots, \lambda_n) = a_0 e^{\lambda_0 x} + \cdots + a_n e^{\lambda_n x}$$

Rationale Interpolation (nicht linear)

$$\Phi(x, a_0, \ldots, a_n, b_0, \ldots, b_m) = \frac{a_0 + a_1 x + \cdots + a_n x^n}{b_0 + b_1 x + \cdots + b_m x^m}$$

Erweitertes Problem: Hermit-Interpolation

Es werden nicht nur Funktionswerte f_k an den Stützstellen x_k vorgeschrieben, sondern auch Werte für die Ableitung von Φ.

Beispiel: $p(x) = \sum_{k=0}^{N} a_k x^k$ mit $p(x_k) = f_k,\ p'(x_k) = d_k$ für gegebene $(x_k, f_k, d_k) \in \mathbb{R}^3$ $(k = 0, \ldots, n)$, etwa $N = 2(n+1) - 1$

Spline-Interpolation

Sei $q \in \mathbb{N}$ fest.

Gesucht: $\Phi \in C^q$ mit $\Phi(x_k) = f_k$, d.h. $\Phi_{|_{[x_k, x_{k+1}]}} \in P_{q+1}$, d.h. Φ ist stücksweise polynomial.

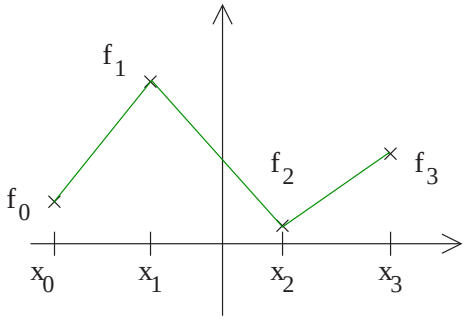


Abbildung 4.1: Spline-Interpolation

Abbildung 4.1 zeigt das Problem, denn $\Phi \notin C^1$.

4.1 Polynominterpolation

Gegeben: $(x_0, f_0), \ldots, (x_n, f_n) \in \mathbb{R}^2$ mit $x_k \neq x_i$ $(k \neq i)$

Gesucht: $p \in \mathbb{P}_N$ mit $p(x_i) = f_i$ $(i = 0, \ldots, n)$ mit N minimal.

Beispiel: $(x_0, f_0) = (0, 0),\ (x_1, f_1) = (1, 1)$

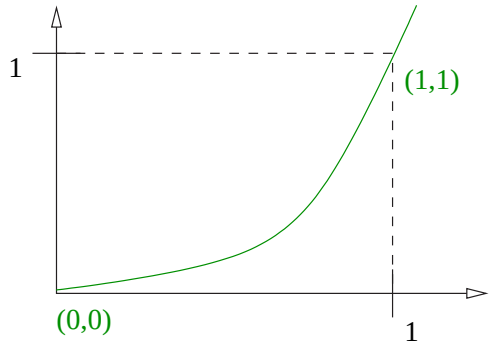


Abbildung 4.2: Polynominterpolation, Beispiel 1

Abbildung 4.2 verdeutlich das Beispiel, denn $p(x) = x^N$ erfüllt das Interpolationsproblem für alle $N \geq 1$. Gesuchtes Polynom ist dann: $p(x) = x$

Satz 4.1
Es existiert genau ein $p \in \mathbb{P}_n$ mit $p(x_i) = f_i \ (i = 0, \dots, n)$

Beweis: Sei $\varphi_0, \dots, \varphi_n$ eine Basis von $\mathbb{P}_n$, dann ist das Interpolationsproblem äquivalent dazu, einen Vektor $a = (a_0, \dots, a_n)^\top$ zu finden, welcher das LGS $Aa = f$ löst mit $A = (\alpha_{ik}) \in \mathbb{R}^{(n+1)\times(n+1)}$, $\alpha_{ik} = \varphi_k(x_i)$ und $f = (f_0, \dots, f_n)^\top \in \mathbb{R}^{n+1}$

$$\begin{aligned} \text{Es gilt : } Aa = f \quad &\iff \quad \sum_{k=0}^{n} \alpha_{ik} a_k = f \\ &\iff \quad \sum_{k=0}^{n} a_k \varphi_k(x_i) = f_i \\ &\iff \quad p(x_i) = f_i \text{ mit } p(x) = \sum_{k=0}^{n} a_k \varphi_k(x) \end{aligned}$$

Existenz und Eindeutigkeit: Es reicht zu zeigen, dass A regulär ist.

Sei also $a = (a_0, \dots, a_n)^\top$ Lösung von $\sum_{k=0}^{n} a_k \varphi_k(x_i) = 0 \ (i = 0, \dots, n)$

$$\implies p(x) = \sum_{k=0}^{n} a_k \varphi_k(x) \in P_n \text{ hat die } (n+1)\text{-Nullstellen } x_0, \dots, x_n \implies p \equiv 0 \implies a_0 = \dots = a_n = 0$$

Also $Aa = 0 \implies a = 0 \implies A$ injektiv $\implies A$ regulär.
$\implies$ Das Interpolationsproblem ist eindeutig lösbar $\square$

Bemerkung: Der Beweis von Satz 4.1 erlaubt es, Verfahren zur Lösung des Interpolationsproblems zu konstruieren, dazu muss man eine Basis $\varphi_0, \dots, \varphi_n$ von $\mathbb{P}_n$ wählen und das $(n+1) \times (n+1)$ LGS $Aa = f$ lösen.
Wählt man folgende Basis $\varphi_0(x) = 1, \varphi_1(x) = x, \varphi_2(x) = x^2, \dots, \varphi_n(x) = x^n$ (also $\varphi_i(x) = x^i$), dann gilt $p(x) = \sum_{i=0}^{n} \alpha_i \varphi_i(x)$ und es entsteht folgende Matrix:

$$A = \begin{pmatrix} \varphi_0(x_0) & \cdots & \varphi_n(x_0) \\ \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \cdots & \varphi_n(x_n) \end{pmatrix} = \begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}$$

A heißt die **Vandermondsche Matrix**; sie ist sehr schlecht konditioniert und voll besetzt. Daher ist das LGS $Aa = f$ sehr aufwändig zu lösen.

Das Interpolationspolynom $\sum_{k=0}^{n} a_k x^k$ heißt in **Normalform**. Diese Darstellung wird praktisch nicht verwendet. Andere Darstellungen sind üblicher, aber das Interpolationspolynom ist immer dasselbe!

a) **Lagrange-Form des Interpolationsproblems**

Am einfachsten ist $Aa = f$ zu lösen, falls $A = \mathbb{I}$, d.h. $\varphi_k(x_i) = \delta_{ki} \ (0 \le k, i \le n)$

$$\implies \varphi_k(x_i) = 0 \text{ für } k \neq i \implies \varphi_k(x) = c \prod_{\substack{i=0 \\ i\neq k}}^{n} (x - x_i). \text{ Aus } \varphi_k(x_k) = 1$$

$$\implies c = \left(\prod_{\substack{i=0 \\ i\neq k}}^{n} (x - x_i) \right)^{-1} \implies \varphi_k(x) = \prod_{\substack{i=0 \\ i\neq k}}^{n} \tfrac{(x-x_i)}{x_k - x_i}, \ (k = 0, \dots, n)$$

Definition 4.2 (und Satz)
Die Polynome

$$l_k^n(x) := \prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{(x-x_i)}{x_k - x_i}$$

heißen **Lagrange-Polynome** *und* $(l_0^n, \ldots, l_n^n)$ *bilden eine Basis von* $\mathbb{P}_n$ *und* $p(x) = \sum_{k=0}^{n} f_k l_k^n(x)$ *ist das Interpolationsproblem zu* $(x_0, f_0), \ldots, (x_n, f_n)$.

Es gilt: $l_i^n(x_j) = \delta_{ij}$

Beweis: Klar, wegen Konstruktion

Bemerkung: Diese Darstellung ist für die Theorie sehr brauchbar. Mit dieser Konstruktion zu arbeiten ist angenehm, weil für $p(x) = \sum_{i=1}^{n} f_i l_i^n(x)$ gilt: $p(x_j) = \sum_{i=1}^{n} f_i l_j^n(x_j) = f_j$
Nachteil: Die Basispolynome ändern sich bei Zunahme der Anzahl der Stützstellen.

b) **Newton-Form des Interpolationsproblems**

Wähle eine Basis von $\mathbb{P}_n$, so dass A eine untere $\triangle$-Matrix wird

$$\varphi_k(x) := \prod_{j=0}^{k-1}(x - x_j), \ (k = 0, \ldots, n \implies \varphi_k \in \mathbb{P}_k)$$

$$\begin{aligned}
\text{etwa} : \varphi_0(x) &= 1 \left(\text{verwende Konvention } \prod_{j=j_0}^{j_n} a_j = 1 \text{ falls } j_n < j_0 \right) \\
\varphi_1(x) &= (x - x_1) \\
\varphi_2(x) &= (x - x_1)(x - x_2) \\
&\vdots \quad \text{damit ist } A \text{ untere } \triangle\text{−Matrix, da} \\
\varphi_n(x_k) &= 0 \text{ für } i < k
\end{aligned}$$

Definition 4.3

$$N_k^n := \prod_{j=0}^{k-1}(x - x_j)$$

heißen **Newton-Polynome** *und das Interpolationspolynom* $p(x) = \sum_{k=0}^{n} a_k N_k^n(x)$ *heißt in* **Newton-Form**.

$$\begin{aligned}
\text{Es gilt} : a_0 &= \frac{f_0}{\varphi_0(x_0)} f_0 \\
a_1 &= \frac{(f_1 - \varphi_0(x_1)a_0)}{\varphi_1(x_1)} = \frac{f_1 - f_0}{x_1 - x_0} = f[x_0, x_1] \\
a_2 &= \frac{(f_2 - \varphi(x_2)a_0 - \varphi_1(x_2)a_1)}{\varphi_2(x_2)} \\
&= \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0} = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = f[x_0, x_1, x_2]
\end{aligned}$$

Diese Koeffizienten werden berechnet über die **dividierten Differenzen** $f[x_0, \ldots, x_n]$
(Siehe §4.3)

Bemerkung: Diese Darstellung ist wegen der Rekursion schwer zu handhaben, deshalb wird diese meistens programmiert, weil Computer viel besser mit Rekursion umgehen können als Menschen.

4.2 Interpolation von Funktionen durch Polynome

Gegeben: Stützstellen $x_0, \ldots, x_n$ und f stetig

Gesucht: Interpolationspolynom zu $(x_0, f_0), \ldots, (x_n, f_n)$

Satz 4.4 (Fehlerdarstellung)
Sei $f \in C^{n+1}(a,b)$ und $p \in \mathbb{P}_n$ das Interpolationspolynom zu den Stützstellen $x_0, \ldots, x_n$ paarweise verschieden. Dann existiert zu jedem $x \in (a,b)$ ein $\xi_x \in (a,b)$ mit

$$(*) \quad f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \cdot \prod_{k=0}^{n} (x - x_k)$$

Beweis: Für $x = x_i$ $(i = 0, \ldots, n)$ ist nichts zu zeigen, da $f(x_i) = p(x_i)$ und $\prod_{k=0}^{n} (x - x_k) = 0$

Sei also $x \neq x_i$: Setze

$$\omega(t) := \prod_{i=0}^{n} (t - x_i)$$

und betrachte

$$\Phi(t) := f(t) - p(t) - \lambda \omega(t)$$

mit $\lambda = \frac{f(x)-p(x)}{\omega(t)} \in \mathbb{R}$ (beachte x fest)

$\Longrightarrow$ $\Phi(x) = 0$ und $\Phi(x_i) = 0$ $(i = 0, \ldots, n)$

$\Longrightarrow$ Φ hat $n+2$ Nullstellen. Nach dem Satz von Rolle

$\Longrightarrow$ Φ' hat $n+1$ Nullstellen und daher $\Phi^{(n+1)}$ eine Nullstelle $\xi_x \in (a,b)$ mit

$$\Phi^{(n+1)} = f^{(n+1)}(\xi_x) = f^{(n+1)}(\xi_i) - (n+1)! \frac{f(x) - p(x)}{\omega(t)}$$

$\Longrightarrow$ $(*)$ wird erreicht (durch Umformen) $\square$

Folgerung 4.5
Vorraussetzungen wie in 4.4

(i) Dann gilt $\|f - p\|_\infty \leq \frac{1}{(n+1)!} \left\|f^{(n+1)}\right\|_\infty \|\omega\|_\infty$ mit $\omega(x) = \prod_{j=0}^{n} (x - x_j)$ (ω heißt **Knotenpolynom**).

(ii) Sind die Ableitungen von $f \in C^\infty$ gleichmäßig beschränkt, d.h. $\left\|f^{(n)}\right\|_\infty \leq C$, $C > 0$ unabhängig von n, dann gilt: $\|f - p_n\|_\infty \longrightarrow 0$, wobei $p_n \in \mathbb{P}_n$ ein Interpolationspolynon zu $(n+1)$ Stützstellen ist, d.h. f kann gleichmäßig durch Polynomen approximiert werden.

Beweis: Folgt direkt aus dem Satz 4.4

Beispiel 4.6
$\left(x_0^{(n)}, x_1^{(n)}, \ldots, x_n^{(n)}\right)$ $(n \in \mathbb{N})$ Folge von Stützstellen, $p_n \in \mathbb{P}_n$ das zugehörige Interpolationspolynom.

(i) Es existiert eine stetige Funktion f, so dass $p_n \not\longrightarrow f$ gleichmäßig.

(ii) Für jede stetige Funktion f existiert eine Folge von Stützstellen mit $f_n \longrightarrow f$ gleichmäßig.

(iii) Für $x \notin [\min\{x_0, \ldots, x_n\}, \max\{x_0, \ldots, x_n\}]$ dann wächst das Knotenpolynom $\omega(x) = \prod_{k=0}^{n} (x - x_k)$ sehr schnell für große n. D.h. Vorsicht bei Extraprodukt, da der Fehler sehr groß wird.

Zu (i) Runge: $f(x) = \frac{1}{1+x^2}$, $-5 \leq x \leq 5$ und $x_n^{(n)} := -5 + kh_n$, $0 \leq k \leq n$ mit $h_n := \frac{10}{n}$ (gleichmäßige Stützstellenverteilung), $p_n(x_k^{(n)}) = f(x_k^{(n)})$

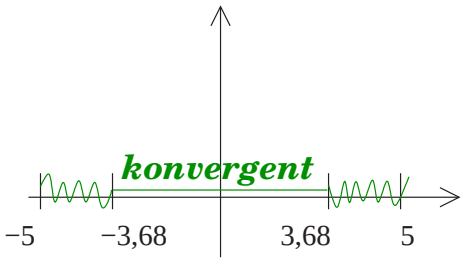


Abbildung 4.3: Interpolation von Funktionen, Beispiel 4.6

Grund (Siehe Abbildung 4.3): f als komplexe Funktion ist nicht so schön glatt. Bei anderer Wahl der Stützstellen konvergiert $p_n \longrightarrow f$

Idee: Wähle Stützstellen $x_0, \ldots, x_n$, so dass $\|w\|_\infty$ möglichst minimal ist.

Bemerkung: Es reicht sich das Interval $[-1, 1]$ anzuschauen, da ein $\psi \in P_1$ mit $\psi(a) = -1$, $\psi(b) = 1$ und diese Transformation ändert den Grad des Interpolationspolynoms nicht.

Definition 4.7 (Tschebyschev-Polynome)
$T_0(x) = 1,\ T_1(x) = x, \quad x \in [-1, 1]$

$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad x \in [-1, 1]$

$\hat{T}_n(x) = 2^{1-n} T_n(x)$ *D.h.*

$T_2(x) = 2x^2 - 1$

$T_3(x) = 2x(2x^2 - 1) - x = 4x^3 - 2x - x = 4x^3 - 3x$ *und*

$\hat{T}_2(x) = \frac{1}{2} T_2(x) = x^2 - \frac{1}{2}, \quad \hat{T}_3(x) = \frac{1}{4} T_3(x) = x^3 - \frac{3}{4}x$

Satz 4.8
Für $x \in [-1,1]$ gilt:

$$(*) \qquad T_n(x) = \cos\left(n \cos^{-1}(x)\right)$$

Weiterhin gilt:

(i) $|T_n(x)| \leq 1$

(ii) $T_n\left(\cos\left(\frac{j\pi}{n}\right)\right) = (-1)^j \quad (0 \leq j \leq n)$

(iii) $T_n\left(\cos\left(\pi \frac{2j-1}{2n}\right)\right) = 0 \quad (1 \leq j \leq n)$

(iv) $T_n \in P_n(-1,1)$

(v) $\hat{T}_n \in P_n(-1,1)$ und $\hat{T}_n$ ist ein normiertes Polynom.

Beweis: Additionstheorem:

$$\cos(A+B) = \cos(A)\cos(B) - \sin(A)\sin(B)$$

$\implies \cos((n+1)\Theta) = \cos(n\Theta)\cos(\Theta) - \sin(n\Theta)\sin(\Theta)$ und
$\cos((n-1)\Theta) = \cos(n\Theta)\cos(\Theta) + \sin(n\Theta)\sin(\Theta)$

$\implies \cos((n+1)\Theta) + \cos((n-1)\Theta) = 2\cos(n\Theta)\cos(\Theta)$

Setze $\Theta := \cos^{-1}(x)$ bzw $x := \cos(\Theta)$

$\implies \cos\left((n+1)\cos^{-1}(x)\right) = 2\cos(\cos^{-1}(x))x - \cos((n-1)\cos^{-1}(x))$

d.h. $F_n := \cos\left(n\cos^{-1}(x)\right)$ genügt der Rekursionsformel von Definiton 4.7

$F_0(x) = 1,\ F_1(x) = \cos(\cos^{-1}(x)) = x \implies F_n = T_n \implies (*)$

Die Eigenschaften (i), (ii), (iii) folgen direkt durch Einsetzen aus $(*)$; (iv) und (v) folgen ebenfalls durch Einsetzen aus der Rekursionsformel für T_n □

Lemma 4.9
Sei $p \in \mathbb{P}_n$ ein normiertes Polynom auf $[-1,1]$. Dann gilt: $\|p\|_\infty = \max\limits_{-1 \leq x \leq 1} |p(x)| \geq 2^{1-n}$ und $\left\|\hat{T}_n\right\|_\infty = 2^{1-n}$

Satz 4.10
Mit den Stützstellen $x_k = \cos\left(\pi \frac{2k-1}{2(n+1)}\right), \quad k = 1, \ldots, n+1$ als die Nullstellen von T_{n+1} gilt, dass das Knotenpolynom gerade $\hat{T}_{n+1}$ ist, d.h. die Maximumsnorm des Knotenpolynoms ist.

Bemerkung: Die Nullstellen von $\hat{T}_{n+1}$ liegen dichter zusammmen an den Intervallgrenzen $-1, 1$.

Beweis: (Von Lemma 4.9)

Annahme: $|p(x)| < 2 \quad \forall\, x \in [-1,1]$

Sei $x_i = \cos\left(\frac{i\pi}{n}\right)$. Nach Satz 4.8 (ii)

$\implies \hat{T}_n(x_i) = 2^{1-n} T_n(x_i) = (-1)^i 2^{1-n}$

$\Longrightarrow \quad (-1)^i p(x_i) \leq |p(x_i)| < 2^{1-n} = (-1)^i \hat{T}_n(x_i)$

$\Longrightarrow \quad (-1)^i \left(\hat{T}(x_i) - p(x_i)\right) > 0$ für $0 \leq i \leq n$

D.h. das Polynom $\hat{T}_n - p$ wechselt $(n+1)$-Mal das Vorzeichen. $\hat{T}_n$ und p sind normierte Polynome

$\Longrightarrow$ $\hat{T}_n - p$ ein Polynom $(n-1)$-tes Grades ist und das ist ein Widerspruch, da das $\hat{T}_n - p$ höchstens $(n-1)$ Nullstellen hat. □

Bemerkung: Das Knotenpolynom $\omega(t) = \prod_{k=0}^{n} (t - x_k)$ ist ein normiertes Polynom $(n+1)$-tes Grades und die Stützstellen $x_0, \ldots, x_n$ sind die Nullstellen von ω.

4.3 Dividierte Differenzen

Wiederholung: Newton-Verfahren des Interpolationspolynom: $p(x) = \sum_{k=0}^{n} a_k N_k(x)$ mit

$$N_k(x) = \begin{cases} 1 & : \quad k = 0 \\ \prod_{j=0}^{k-1} (x - x_j) & : \quad k \geq 1 \end{cases}$$

Gesucht: Algorithmus zur Berechnung von $a_0, \ldots, a_n$

Bemerkung: Setze $p_m(x) = \sum_{k=0}^{m} a_k N_k(x)$ für $m \leq n$

$\Longrightarrow$ $p_m(x_j) = f_j \quad (0 \leq j \leq m)$ mit $p_m \in \mathbb{P}_m$, da $N_k \in \mathbb{P}_m$ für $0 \leq k \leq m$. D.h. p_m ist **das** Interpolationspolynom in P_m zu den Daten $(x_0, f_0), \ldots, (x_m, f_m)$. Daher hängt a_k nur von $(x_0, f_0), \ldots, (x_k, f_k)$ für $0 \leq k \leq m$. Es wird daher die Schreibweise $f[x_0, \ldots, x_n]$ für a_k benutzt.
Beachte, a_m ist der Koeffizient vor dem x^m im Polynom p_m

Definition 4.11
Seien $i_o, \ldots, i_n \in \{0, \ldots, n\}$ paarweise verschieden und sei $p_{i_0,\ldots,i_k}$ das Interpolationspolynom zu den Daten $(x_{i_0}, f_{i_0}), \ldots, (x_{i_k}, f_{i_k})$. Mit $f[x_{i_0}, \ldots, x_{i_k}]$ bezeichnen wir den Koeffizienten vor x^k im Polynom $p_{i_0,\ldots,i_k}$

$f[i_0, \ldots, i_k]$ wird als **dividierte Differenz** *der Ordnung k bezeichnet.*

Satz 4.12
Die Polynome $p_{i_0,\ldots,i_k}$ genügen der Rekursionsformel

(i) $p_{i_0,\ldots,i_k}(x) = \frac{(x - x_{i_0}) p_{i_0,\ldots,i_k}(x) - (x - x_{i_k}) p_{i_0,\ldots,i_k}}{x_{i_k} - x_{i_0}}$

Die dividierten Differenzen genügen der Rekursionsformel

(ii) $f[x_{i_0}, \ldots, x_{i_k}] = \frac{f[x_{i_1},\ldots,x_{i_k}] - f[x_{i_0},\ldots,x_{i_{k-1}}]}{x_{i_k} - x_{i_0}}$

$f[x_{i_l}] = x_{i_l}$

Die dividierten Differenzen sind unabhängig von der Reihenfolge ihrer Koeffizienten, d.h. ist $x_{j_0}, \ldots, x_{j_n}$ eine Permutation von $x_{i_0}, \ldots, x_{i_n}$, so gilt $f[x_{j_0}, \ldots, x_{j_n}] = f[x_{i_0}, \ldots, x_{i_n}]$

Bemerkung: Die dividierten Differenzen können in der Form eines Tableaus geschrieben werden.

$$\begin{array}{llll|lll} x_0 & a=f_0 & a_1=f[x_0,x_1] & a_2=f[x_0,x_1,x_2] & a_3=f[x_0,x_1,x_2,x_3] \\ x_1 & f_1 & f[x_1,x_2] & f[x_1,x_2,x_3] & \\ x_2 & f_2 & f[x_2,x_3] & & \\ x_3 & f_3 & & & \end{array}$$

$f[x_1,x_2,x_3] = \frac{f[x_2,x_3]-f[x_1,x_3]}{x_3-x_1}$. Beachte, $f_k = f[x_k]$ und $p(x) = \sum_{k=0}^{n} f[x_0,\ldots,x_k]N_k(x)$ ist das gesuchte Interpolationspolynom.

Beispiel 4.13

x	3	1	5
f	1	-3	2

Die dividierten Differenzen:

$$\begin{array}{ll|ll} 3 & 1 & 2=\frac{-3-1}{1-3} & -\frac{3}{8}=\cdots \\ 1 & -3 & \frac{5}{4}=\cdots & \\ 5 & 2 & & \end{array}$$

d.h. das Interpolationspolynom ist $p(x) = 1 + 2(x-3) - \frac{3}{8}(x-3)(x-1)$

Füge eine Stützstelle hinzu:

x	3	1	5	6
f	1	-3	2	4

Die dividierten Differenzen:

$$\begin{array}{ll|lll} 3 & 1 & 2 & -\frac{3}{8} & \frac{7}{40} \\ 1 & -3 & \frac{5}{4} & \frac{3}{20} & \\ 5 & 2 & 2 & & \\ 6 & 4 & & & \end{array}$$

d.h. das Interpolationspolynom ist $p(x) = 1 + 2(x-3) - \frac{3}{8}(x-3)(x-1) + \frac{7}{40}(x-3)(x-1)(x-5)$

Beweis: Von Satz 4.12

(i) Setze $R(x)$ als rechte Seite von (i). Zu zeigen: $p_{i_0,\ldots,i_n} = R(x)$

Notation:
$P_k = p_{i_0,\ldots,i_k}$, $p_{k-1} = p_{i_0,\ldots,i_{k-1}}$ $q_k = p_{i_1,\ldots,i_k}$

Dann ist

$$R(x) = \frac{(x-x_{i_0})q_k(x) - (x-x_{i_k})p_{k-1}(x)}{x_{i_k}-x_{i_0}}$$

$$\begin{aligned} \Longrightarrow \quad R(x_{i_0}) &= \frac{0-(x_{i_0}-x_{i_k})f_{i_0}}{x_{i_k}-x_{i_0}} = f_{i_0} \\ R(x_{i_k}) &= \frac{(x_{i_k}-x_{i_0})f_{i_k}}{x_{i_k}-x_{i_0}} = f_{i_k} \\ R(x_{i_l}) &= \frac{(x_{i_l}-x_{i_k})f_{i_l}-(x_{i_l}-x_{i_k})f_{i_l}}{x_{i_k}-x_{i_0}} = f_{i_l} \quad \forall\ 0<l<k \end{aligned}$$

$\Longrightarrow$ R ist das Interpolationspolynom zu $(x_{i_0},f_{i_0}),\ldots(x_{i_n},f_{i_n})$

$\Longrightarrow$ Eindeutigkeit $\Longrightarrow$ $p_{i_0,\ldots,i_n} = R$

(ii) Der Koeffizient von x^k in $R(x)$ ist $\frac{f[x_{i_1},\ldots,x_{i_k}]-f[x_{i_0},\ldots,x_{i_{k-1}}]}{x_{i_k}-x_{i_0}}$

Nach Definition ist dieser Koeffizient gleich $f[x_{i_0},\ldots,x_{i_k}]$ $\Longrightarrow$ (ii) □

Satz 4.14 (Weitere Eigenschaften der dividierten Differenz)
Sei $f \in C^0(a,b)$, $x_0, \ldots, x_n \in (a,b)$ paarweise verschieden und t fest mit $t \neq x_n \ \forall\ k = 0$

(i) Wenn p das Interpolationspolynom von f an der Stützstellen $x_0, \ldots, x_n$ ist, so gilt:

$$f(t) - p(t) = f[x_0, \ldots, x_n] \prod_{j=0}^{n} (t - x_j)$$

(ii) Ist $f \in C^n(a,b)$ so existiert ein $\xi \in (a,b)$ mit $f[x_0, \ldots, x_n] = \frac{1}{n!} f^{(n)}(\xi)$

Beweis: Siehe Übungsaufgaben

Algorithmus 4.15 (Dividierte Differenzen)
Ziel: Das ganze Tableau soll berechnet und in eine Matrix gespeichert werden. Wenn eine weitere Stützstelle hinzugefügt wird, dann reicht es die Diagonale der dividierten Differenzen auszurechnen.

$c_{i0} := f_i$

Für $f = 1, \ldots, n$
  Für $i = 0, n - j$
    $c_{ij} := \frac{c_{i+1,j-1} - c_{i,j-1}}{x_{i+j} - x_i}$

$\Longrightarrow \quad c_{ij} = f[x_i, \ldots, x_{i+j}]$

Nach Zunahme einer Stützstelle (x_{n+1}, f_{n+1})

$c_{n+1,0} := f_{n+1}$
Für $j = 0, \ldots, n+1$
  $c_{n+1-j,j} := \frac{c_{n+1-j,j-1} - c_{n-j,j-1}}{x_{n+1} - x_{n+1-j}}$

Satz 4.16 (Auswertung des Interpolationspolynoms)
1. Fall: Die Koeffizienten $a_0, \ldots, a_n$ seien bekannt. Dann kann man ein Schema (ähnlich dem Horner-Schema §3.2) benutzt werden, so dass gilt

$$\begin{aligned} p(x) &= \sum_{k=0}^{n} a_k \prod_{j=0}^{k-1} \underbrace{(x - x_j)}_{:=\chi_j} \\ &= \left(\left(\cdots\left(\left(a_n \gamma_{n-1} + a_{n-1}\right)\gamma_{n-2} + a_{n-2}\right)\cdots\right) x_1 + a_1\right) x_0 a_0) \end{aligned}$$

Algorithmus:
$p := a_n$
Für $k = n-1, \ldots, 0$
  $p := p(x - x_k) + a_n$

2. Fall: Das Interpolationspolynom p soll nur an einer Stelle ausgerechnet werden ohne vorher die Koeffizienten zu berechnen.

Neville-Schema

Sei $p_{i_0,\dots,i_k} \in P_n$ das Interpolationspolynom zu $(x_{i_0}, f_{i_0}) =, \dots, (x_{i_0}, f_{i_0})$. Das Neville-Schema verwendet die Rekursion aus 4.14 (i)

$$\begin{array}{cc|ccc} x_0 & f_0 = p_0(x) & p_{01}(x) & \cdots & p_{0,1,\dots,n}(x) \\ x_1 & f_1 = p_1(x) & p_{12}(x) & \cdots & \\ \vdots & \vdots & \vdots & & \\ x_n & f_n = p_n(x) & p_{n,n+1}(x) & & \end{array}$$

$p_{0,1,\dots,n}(x)$ ist gesucht, also der letzte Eintrag in der Tabelle.

Beispiel 4.17
Gegeben:

x_i	3	1	5	6
f_i	1	-3	2	4

Gesucht: $p(0)$

(i) **Mit dividierten Differenzen** erhält man

$$p(x) = 1 + 2(x-3) - \frac{3}{8}(x-3)(x-1) + \frac{7}{40}(x-3)(x-1)(x-5)$$

Mit dem Horner-Schema:

$$\begin{array}{rcl} p(0) & = & \left(\left(\frac{7}{40}(0-5) - \frac{3}{8}\right)(-1) + 2\right)(-3) + 1\big) \\ & = & \left(\left(\frac{5}{4} + 2\right)(-3) + 1\right) \\ & = & \left(-\frac{39}{4} + 1\right) = -\frac{35}{4} \end{array}$$

(ii) **Mit dem Neville-Schema** erhält man

$$\begin{array}{cc|ccc} x_0 & f_0 & & & \\ 3 & 1 & \frac{(0-3)(-3)-(0-1)1}{1-3} = -5 & -\frac{79}{8} & \frac{35}{4} \\ 1 & -3 & \frac{(0-1)2-(0-5)(-3)}{5-1} = -\frac{7}{2} & & \\ 5 & 2 & \frac{(0-5)4-(0-6)2}{6-5} = -8 & & \\ 6 & 4 & & & \end{array}$$

4.4 Hermite Interpolation

Gegeben: $x_0, \dots, x_n$ paarweise verschieden und für jede Stützstelle x_i Werte $c_{ij} \in \mathbb{R}$ für $0 \le j \le k_{i-1}$. Die Anzahl der Bedingungen ist $m+1 := R_0 + R - 1 + \cdots + R_n$, d.h. es macht Sinn $p \in \mathbb{P}_m$ zu suchen.

Satz 4.18
Es existiert genau ein $p \in \mathbb{P}_m$, welches die Bedingungen des Hermite Interpolationspolynoms erfüllt.

Beweis: Analog zu Satz 4.1

Satz 4.18b (Fehlerabschätzung für Hermite Interpolation)
Es sei $f \in C^{N+1}(a,b)$ und $a \le x_0 < \dots < x_m \le b$. Mit $m_0, \dots, m_m \in \{1, \dots, N+1\}$ und $n+1 = \sum_{j=0}^{n} m_j$

Sei $p_n \in \mathbb{P}_n$ das Hermite Interpolationspolynom zu den Daten

$$\begin{array}{llll} (x_0, f(x_0)), & \dots & , (x_0, f^{(m_0)}(x_0)) \\ (x_0, f(x_1)), & \dots & , (x_1, f^{(m_1)}(x_1)) \\ \vdots & & \vdots \\ (x_0, f(x_m)), & \dots & , (x_0, f^{(m_m)}(x_m)) \end{array}$$

Dann existiert für alle $x \in [a, b]$ ein $\xi_x \in [a, b]$ mit

$$f(x) - p_n(x) = \frac{f^{(N+1)}(\xi_x)}{(N+1)!}\Omega(x)$$

wobei $\Omega(x) = \prod_{k=0}^{n} (x - x_k)^{m_k}$
Dabei ist es zu beachten, dass für Fehlerabschätzung $N = n$ gilt.

Beweis: Analog zu Satz 4.4 (vgl. Schebach-Werner)

Beispiel 4.19 (Newtonform und dividierte Differenzen)
Gesucht: $p \in \mathbb{P}_2$ mit $p(x_0) = c_{00},\ p'(x_0) = c_{01},\ p(x_1) = c_{10}$

Durch dividierte Differenzen:

$$\begin{array}{cc|ll} x_i & f_i & & \\ x_0 & f_0 & f[x_0, x_0] & f_[x_0, x_0, x_1] \\ x_0 & f_0 & f[x_0, x_1] & \\ x_1 & f_1 & & \end{array}$$

Nach Satz 4.14 gilt für $t \in (a, b) :\ \exists\, \xi \in (x_0, t)$ mit $f[x_0, t] = f'(\xi)$

Ist $f' \in C^0(a, b)$ so gilt:

$$\lim_{t \longrightarrow x_0} f[x_0, t] = f'(x_0)$$

Daher macht es Sinn $f[x_0, x_0] = f'(x_0)$ zu setzen. Im Beispiel

$$\begin{array}{cc|cc} x_0 & c_{00} & c_{01} & \frac{f[x_0,x_0]-f[x_0,x_1]}{x_1-x_0} \\ x_0 & c_{00} & \frac{c_{10}-c_{00}}{(x_1-x_0)^2} - \frac{c_{01}}{x_1-x_0} & \end{array}$$

Wir setzen das Interpolationspolynom in der Newtonform ein.

$$p(x) = f[x_0] + f[x_0, x_0](x - x_0) + f[x_0, x_0, x_1](x - x_0)^2$$

Dieser Ansatz läßt sich verallgemeinern zu:

$$p(x) = \sum_{k=0}^{n} f[z_0, \dots, z_n] \prod_{j=0}^{k-1} (x - z_j)$$

$$\begin{array}{rcl} \text{wobei } z_0 & = & \cdots = z_{k_0-1} = x_0 \\ z_{k_0} & = & \cdots = z_{k_0-k_1-1} = x_1 \\ & \vdots & \\ & \text{usw.} & \end{array}$$

Satz 4.20 (Rekursionsformel für die dividierten Differenzen)
Sei $i_0, \dots, i_n \in \{0, \dots, n\}$ und o.B.d.A $z_{i_0} \le z_{i_1} \le \cdots \le z_{i_k}$. Dann gilt

$$f[z_{i_0}, \dots, z_{i_k}] = \begin{cases} \frac{f[z_{i_1}, \dots, z_{i_k}] - f[z_{i_0}, \dots, z_{i_{k-1}}]}{z_{i_k} - z_{i_0}} & : \quad z_{i_k} \neq z_{i_0} \\ \frac{1}{k!} f^{(k)}(z_{i_0}) & : \quad z_{i_k} = z_{i_0} \end{cases}$$

Bemerkung 4.21

(i) Bei der Hermite Interpolation werden gerade die Werte vorgeschrieben, die bei den Dividierten Differenzen Tableau nicht durch die Rekursion gegeben sind.

(ii) Interpolationsprobleme, bei denen nicht für alle $j = 0, \dots, k-1$ die Werte $p^{(j)}(x)$ vorgeschrieben werden, sind nicht so einfach zu lösen (vergleiche Birkoff-Interpolation in den Übungsaufgaben).

Wiederholung

Gegeben: Daten $(x_0, y_0), \dots, (x_n, y_n),\ x_0, \dots, x_n$ paarweise verschieden.

Gesucht: $p \in \mathbb{P}_n$ mit $p(x_n) = y_n,\ k = 0, \dots, n$

Lagrange Darstellung

$p(x) = \sum_{k=0}^{n} y_k L_k^n(x)$ und $L_k^n(x) = \prod_{\substack{l=0 \\ l \neq k}}^{n} \frac{x - x_l}{x_k - x_l} \in P_n$ mit $L_k^n(x_l) = \delta_{kl}$

Neville Schema

Auswertung von p an einer Stelle x

Sei $p_i^{(k)} = p_{k,k+1,\dots,k+i}(x)$ das Interpolationspolynom zu $(x_k, y_k), \dots, (x_{k+i}, y_{k+i})$. Dann gilt die Rekursionsformel

$$p_i^{(k)} = \frac{(x - x_k) p_{i-1}^{(k+1)} - (x - x_{k+i}) p_{i-1}^{(k)}}{(x_{k+1} - x_k)}$$

und $p_0^{(k)} = y_k \implies p(x) = p_n^{(0)}$

Fehlerabschätzung

Ist $f \in C^{n+1}(a, b)$ und $y_k = f(x_k) \implies f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{k=0}^{n} (x - x_k)$

4.5 Richardson Extrapolation

Gegeben: $a : (0, \infty) \longrightarrow \mathbb{R}$

Gesucht: $a(0) = \lim_{h \searrow 0} a(h)$

Idee: Wähle $h_0, \dots, h_n$, setze $a_k = a(h_k)$ und bestimme das Interpolationspolynom zu $(h_0, a_0), \dots, (h_n, a_n)$ und approximiere $a(0)$ durch $p(0)$.

Beispiel 4.22

(i) **L'Hôpital Regel**

Berechne $\lim\limits_{x \longrightarrow 0} \frac{\cos(x)-1}{\sin(x)}$, d.h. $a(h) = \frac{\cos(h)-1}{\sin(h)}$

$$\begin{aligned} \text{Setze}: h_0 &= \tfrac{1}{8} \quad , \quad a_0 = -6.258151 \cdot 10^{-2} \\ h_1 &= \tfrac{1}{16} \quad , \quad a_1 = -3.126018 \cdot 10^{-2} \\ h_2 &= \tfrac{1}{32} \quad , \quad a_2 = -1.562627 \cdot 10^{-2} \\ \Longrightarrow \quad & p(0) = -1.02\ldots \cdot 10^{-2} \end{aligned}$$

Es ist $a(0) = \lim\limits_{h \searrow 0} \frac{\cos(h)-1}{\sin(h)} = \lim\limits_{h \searrow 0} \frac{-\sin(h)}{\cos(h)} = 0$

(ii) **Numerische Verfahren** (etwa Differenziation von $f \in C^1$)

Wähle $a(h) = \frac{f(h)-f(-h)}{2h}$

Ist f analytisch, so gilt die **asymptotische Entwicklung**, $a(h) = a(0) + \sum\limits_{i=1}^{\infty} \alpha_{2i} h^{2i}$ mit $a(0) = f'(0)$ und

$$\begin{aligned} f(h) &= f(0) + \sum_{i=1}^{\infty} f^{(i)}(0) h^i \\ f(-h) &= f(0) + \sum_{i=1}^{\infty} f^{(i)}(0)(-h)^i = \sum_{i=1}^{\infty} f^{(i)}(0)(-1)^i h^i \end{aligned}$$

Das heißt, $a(h)$ ist eine gerade Funktion, d.h. $a(h) = a(-h)$ und das Interpolationspolynom solle nur h^{2k}-Terme enthalten.

Sei $f(x) = \sin(x)$, $a(h) = \frac{\sin(h)-\sin(-h)}{2h} = \frac{\sin(h)}{h}$

$$\begin{aligned} h_0 &= \tfrac{1}{8} \quad , \quad a_0 = 0.9973 \\ h_0 &= \tfrac{1}{16} \quad , \quad a_0 = 0.99934 \\ h_0 &= \tfrac{1}{32} \quad , \quad a_0 = 0.99983 \end{aligned}$$

und $p(0) = 0.999999926$. p ist Polynom in h^2, d.h. sei g das Interpolationspolynom zu $(h_0^2, a_0), \ldots, (h_n^2, a_n)$ und $p(x) = q(x^2)$

Satz 4.23 (Extrapolationfehler)

Gelte für $a : (0, \infty) \longrightarrow \mathbb{R}$ die asymptotische Entwicklung $a(h) = a(0) + \sum\limits_{j=1}^{n} \alpha_j h^j + a_{n+1}(h) h^{q(n+1)}$ mit $q > 0$ und $a_{n+1}(h) = \alpha_{n+1} + o(1)$, $\left[\gamma = o(h) \iff \frac{\gamma(h)}{h} \longrightarrow 0 \text{ für } h \longrightarrow 0\right]$

Dabei $\alpha_1, \ldots, \alpha_{n+1} \in \mathbb{R}$ unabhängig von h. Sei $(h_k)_{k \in \mathbb{N}}$ eine monoton fallende Folge, $h_k > 0$ und $\frac{h_{k+1}}{h_k} \leq \rho < 1$ für $\rho > 0$ unabhängig von k.

Mit $p_h^{(k)} \in \mathbb{P}_n$ bezeichnen wir das Interpolationspolynom in h zu den Daten $(h_k^q, a(h_k)), \ldots, (h_{k+n}^q, a(h_{k+n}))$. Dann gilt: $\left|a(0) - p_h^{(k)}(0)\right| = O(h_k^{q(n+1)})$ für $k \longrightarrow 0$

Beweis: Setze $z = h^q$, $z_k = h_k^q$. In der Lagrange Darstellung ist: $p_n^{(k)}(z) = \sum\limits_{i=0}^{n} a(h_{k+i}) L_{k,i}^n(z)$ mit

$L_{k,i}^n(z) = \prod_{\substack{l=0 \\ l\neq i}}^{n} \frac{z-z_{k+l}}{z_{k+i}-z_{k+l}}$. Mit der asymptotischen Entwicklung gilt:

$$\begin{aligned} p_n^{(k)}(z) &= \sum_{i=0}^{n} \left(a(0) + \sum_{j=1}^{n} \alpha_j z_{k+i}^j + \alpha_{n+1} z_{k+i}^{n+1} + o(1) z_{k+i}^{n+1} \right) L_{k,i}^n(z) \\ &= a(0) \sum_{i=0}^{n} L_{k,i}^n(z) + \sum_{j=1}^{n+1} \alpha_j \sum_{i=0}^{n} L_{k,i}^n(z) z_{n+j}^j + o(1) \sum_{i=0}^{n} z_{n+i}^{n+1} L_{k,i}^n(z) \end{aligned}$$

Für die Summen $\sum_{i=0}^{n} z_{k+i}^j L_{k,i}^n(0)$ für $j = 0, \ldots, n+1$ benutzt man die Fehlerabschätzung aus Satz 4.4 mit $f(z) = z^r,\ r = 0, \ldots, n+1$ und dem Interpolationspolynom $q_n^{(k)}$ zu $(z_k, f(z_k)), \ldots, (z_{k+n}, f(z_{k+n}))$, d.h.
$q_n^{(k)}(z) = \sum_{i=0}^{n} f(z_{k+i}) L_{k,i}^n(z)$ bzw
$q_n^{(k)}(z) = \sum_{i=0}^{n} z_{k+i}^r L_{k,i}^n(0)$

Es gilt: $f(0) - q_n^{(k)}(0) = \frac{1}{(k+1)!} f^{(n+1)}(\xi_0) \prod_{i=0}^{n} (0 - z_{k+i})$

$$\implies -\sum_{i=0}^{n} z_{k+i}^r L_{k,i}^n(0) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_0)(-1)^{n+1} \prod_{i=0}^{n} z_{k+i} - f(0)$$

$$= \begin{cases} -1 & : \ r = 0 \\ 0 & : \ r = 1, \ldots, n \\ (-1)^{n+1} \prod_{i=0}^{n} z_{k+i} & : \ r = n+1 \end{cases}$$

$$\implies p_n^{(k)}(0) = a(0) + \alpha_{n+1}(-1)^n \prod_{i=0}^{n} z_{k+i} + \sum_{i=0}^{n} o(1) z_{n+i}^{n+1} L_{k,i}^n(0)$$

Es gilt:

$$\begin{aligned} \left| \alpha_{n+1}(-1)^n \prod_{i=0}^{n} z_{k+i} \right| &\leq |\alpha_{n+1}| \prod_{i=0}^{n} z_k = |\alpha_{n+1}| \, z_k^{(n+1)} \\ &= |\alpha_{n+1}| - h_k^{q(n+1)} = O(h_k^{q(n+1)}) \end{aligned}$$

Auch gilt: $\left| L_{k,i}^n(0) \right| = \prod_{\substack{l=1 \\ l\neq i}}^{n} \frac{1}{\frac{z_{k+1}}{z_{k+l}} - 1} \leq C(\rho, n, q)$ aber unabhängig von k

$$\begin{aligned} \implies \left| \sum_{i=0}^{n} o(1) L_{i,k}^n(0) z_{k+i}^{n+1} \right| &\leq C(\rho, n, q) \sum_{i=0}^{k} o(1) z_{k+i}^{n+1} \\ &\leq C(\rho, n, q) \sum_{i=0}^{k} o(1) z_n^{n+1} \\ &\leq C(\rho, n, q) o(z_n^{n+1}) = o(h_k^{q(n+1)}) \end{aligned}$$

$$\implies \left| p_n^{(k)}(0) - a(0) \right| = O(h_k^{q(n+1)}) \quad \square$$

Algorithmus: Zur Berechnung von $p_n^{(k)}(0)$ eignet sich das Neville Schema:

$$p_n^{(k)}(0) = p_{n-1}^{(k)}(0) + \frac{p_{n-1}^{(k)}(0) - p_{n-1}}{\frac{z_k}{z_{k+1}} - 1}$$

Mit $a_{k,n} := p_n^{(k-n)}(0) \implies a_{k,n} = a_{k,n-1} + \frac{a_{k,n-1} - a_{k-1,n-1}}{\frac{h_{k-n}}{h_k}^q - 1}$

Als Tableau mit Startwert $a_{k,0} = a(h_k)$

$$\begin{array}{lllll} h_0 & a_{0,0} & & & \\ h_1 & a_{1,0} & a_{1,1} & & \\ h_2 & a_{2,0} & a_{2,1} & a_{2,2} & \\ \vdots & \vdots & \vdots & \vdots & \ddots \\ h_k & a_{k,0} & a_{k,1} & a_{k,2} & \cdots & a_{k,k} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \end{array}$$

Beispiel 4.24
Berechnung von $e = \lim\limits_{n \longrightarrow \infty}(1+\frac{1}{n})^n = \lim\limits_{n \longrightarrow \infty}(1+n)^{\frac{1}{n}}$, d.h. $a(h) = (1+h)^{\frac{1}{h}}$

Wähle $h_k = 2^{-k} \implies a_{k,0} = a(h_k) = (1+2^{-k})^{2k}$

$\implies a_{0,0} = 2,\ a_{1,0} = \frac{9}{4},\ a_{2,0} = \frac{625}{256} \approx 2.44$

Als Tableau:

$$\begin{array}{llllll} k=0 & h_0=1 & 2 & & & \\ k=1 & h_1=\frac{1}{2} & \frac{9}{4} & \frac{5}{2} & & \\ k=2 & h_2=\frac{1}{4} & \frac{625}{256} & \frac{337}{128} & \frac{257}{96} \approx 2.67708 & \\ & & n=0 & n=1 & n=2 & \end{array}$$

$a_{1,1} = a_{1,0} + \frac{a_{1,0}-a_{0,0}}{\frac{h_0}{h_1}-1} = \frac{9}{4} + \frac{\frac{9}{4}-2}{2-1} = \frac{5}{2}$

$a_{2,1} = a_{2,0} + \frac{a_{2,0}-a_{1,0}}{2-1} = \frac{337}{128}$

Allgemein: $\frac{h_{k-n}}{h_k} = 2^{-k+n+k} = 2^n$

$a_{2,2} = a_{2,1} + \frac{a_{2,1}-a_{1,1}}{2^2-1} = \frac{257}{96}$

$e = 2.718281828,\ a_{5,5} = 2.71827$ mit $a_{5,0} = 2.6769$

Es gilt: $|a_{80} - a(0)| \approx 0.0053$

Annahme: Aufwand zur Berechnung von $a(h_k) \sim \frac{1}{h_k} = 2^k$

$\implies$ Aufwand für a_{80} ist 256 und für $a_{33} \sim a(h_0) + a(h_1) + a(h_3) = 15$

4.6 Trigonometrische Interpolation

Gegeben: $(x_0, y_0), \ldots, (x_n, y_n),\ x_k$ paarweise verschieden, $x_k \in [0, \omega),\ \omega > 0$

Gesucht: Periodische Funktion $t_n : \mathbb{R} \longrightarrow \mathbb{R}$ mit Periode ω, welche die Daten interpoliert. $\forall\ x \in \mathbb{R}$: $t_n(x+\omega) = t_n(x)$ und $t_n(x_k) = y_k,\ k = 0, \ldots, n$

O.B.d.A $\omega = 2\pi$

Die gesuchte Funktion t_n aus den Funktionen

$$1, \cos(x), \cos(2x), \ldots$$
$$\sin(x), \sin(2x), \ldots$$

zusammensetzen.

Suche: $(a_k, b_k) : t_n(x) = \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos(kx) + b_k \sin(kx)) + \frac{\Theta}{2} a_{m+1} \cos((m+1)x)$, wobei

$$\Theta := \begin{cases} 0 & : \ n \text{ gerade} \\ 1 & : \ n \text{ ungerade} \end{cases}, \quad m := \begin{cases} \frac{n}{2} & : \ n \text{ gerade} \\ \frac{n-1}{2} & : \ n \text{ ungerade} \end{cases}$$

Viele Aussagen lassen sich kompakter über $\mathbb{C}$ formulieren, wobei die **Eulersche Formel**

$$e^{iz} = \cos(z) + i \cdot \sin(z)$$

benutzt wird.

Definition 4.25 (Trigonometrische Polynome)

$$T_n := \left\{ t^* : \ \mathbb{C} \longrightarrow \mathbb{C} \ \middle| \ t^*(z) = \sum_{k=0}^{n} c_k e^{ikz} \right\}$$

Mit $w := e^{iz} \implies t^*(z) = \sum_{k=0}^{n} c_k w^k$

Lemma 4.26

(i) Seien $(a_k)_{k=0}^{\infty}$, $(b_k)_{k=0}^{\infty}$ reelle Folgen. Setze $b_0 = 0$, $a_{-k} = a_k$, $b_{-k} = b_k$ und $c_k = \frac{1}{2}(a_k - i \cdot b_k)$ für $k \in \mathbb{Z}$. Dann gilt:

$$\frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos(kx) + b_k \sin(kx)) = \sum_{k=-m}^{m} c_k e^{ikx}$$

(ii) Sei $(c_k)_{k=-m}^{m}$, $c_k \in \mathbb{C}$. Setze $a_k = c_k + c_{-k}$, $b_k = i \cdot (c_k - c_{-k})$, $k = 0, \ldots, m$. Dann gilt:

$$\frac{1}{2} a_0 + \sum_{k=1}^{m} (a_k \cos(kx) + b_k \sin(kx)) = \sum_{k=-m}^{m} c_k e^{ikx}$$

Beweis: Siehe Übungsaufgaben

Voraussetzungen in diesem Abschnitt

- Äquidistante Stützstellen, d.h. $x_k = \frac{2\pi}{n+1} k, \ k = 0, \ldots, n$
- $w(x) = e^{ix}, \ E_k(x) = e^{ikz} \ (k \in \mathbb{Z})$
- $\hat{w} := e^{i \cdot \frac{2\pi}{n+1}} \in \mathbb{C}, \ w_k := e^{ikx} = e^{ik\frac{2\pi}{n+1}} = \hat{w}^k$

Lemma 4.27

(i) $(E_k)_{k \in \mathbb{Z}}$ sind ein **Orthonormalsystem**, d.h. $\langle E_k, E_l \rangle = \delta_{kl}$

(ii) $w_k^{n+1} = 1$, d.h. $w_0, \ldots, w_n$ sind die $(n+1)$-ten Einheitswurzel und $w_0, \ldots, w_n$ paarweise verschieden

(iii) $w_k^l = w_l^k, \ w_{n+1-k}^l = w_{-k}^l, \ w_k^{-e} = \overline{w_k^l}$

(iv) $\frac{1}{n+1}\sum_{j=0}^{n} w_j^{k-l} = \delta_{kl},\ 0 \le k,l \le n$

(v) Für festes $j \in \mathbb{N}$ fest: $\sum_{k=0}^{n} \sin(jx_k) = 0,\ \sum_{k=0}^{n} \cos(jx_k) = \begin{cases} n+1 & : \ (n+1) \mid j \\ 0 & : \ \text{sonst} \end{cases}$

Beweis: Siehe Übungsaufgaben

Satz 4.28 (Trigonometrische Interpolation in $\mathbb{C}$)
Zu gegebenen Daten $y_0, \dots, y_n \in \mathbb{C}$ existiert genau ein $t_n^* \in T_n$ mit $t_n^*(x_k) = y_k$ für $k = 0, \dots, n$.

Die Koeffizienten c_k sind gegeben durch:

$$c_k = \frac{1}{n+1}\sum_{j=0}^{n} y_j e^{-ijx_k} \left(= \frac{1}{n+1}\sum_{j=0}^{n} y_i w_k^{-j} \right)$$

Beweis: Um die Existenz und Eindeutigkeit zu zeigen, verwenden wir Satz 4.1, der auch im Komplexen gezeigt werden kann. Es existiert daher ein $p \in \mathbb{P}_n$ $p(x) = \sum_{k=0}^{n} c_k x^k$ mit $c_k \in \mathbb{C}$ und $p(w_k) = y_k$ $(k = 0, \dots, n)$ [Interpolationspolynom zu $(w_0, y_0), \dots, (w_n, y_n)$]

Mit $t_n^*(x) = \sum_{k=0}^{n} c_k e^{ikx}$ gilt: $t_n^*(x_l) = \sum_{k=0}^{n} c_k e^{ikx_l} = \sum_{k=0}^{n} c_k w_l^k = p(w_l) = y_l$

Um die explizite Darstellung der Koeffizienten zu zeigen, verwendet man Lemma 4.27.

$$\begin{aligned} \sum_{j=0}^{n} y_j w_k^{-j} &= \sum_{j=0}^{n} p(w_j) w_k^{-j} = \sum_{j=0}^{n} \left(\sum_{l=0}^{n} c_l w_j^l \right) w_k^{-l} \\ &= \sum_{l=0}^{n} c_l \left(\sum_{k=0}^{n} w_j^{l-k} \right) = \sum_{l=0}^{n} c_l (n+1)\delta_{lk} = (n+1)c_k \end{aligned}$$

$$\Longrightarrow \quad c_k = \tfrac{1}{n+1}\sum_{j=0}^{n} y_j w_k^{-j} \qquad \square$$

Satz 4.29 (Trigonometrische Interpolation in $\mathbb{R}$)
Für $n \in \mathbb{N}$ gegeben, setze $m = \begin{cases} \frac{n}{2} & : \ n \text{ gerade} \\ \frac{n-1}{2} & : \ n \text{ ungerade} \end{cases}$, und $\Theta = \begin{cases} 0 & : \ n \text{ gerade} \\ 1 & : \ n \text{ ungerade} \end{cases}$

Zu gegebenen Daten $y_0, \dots, y_k \in \mathbb{R}$ existiert genau eine Funktion
$t_n(x) = \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos(kx) + b_k \sin(kx)) + \frac{\Theta}{2} a_{m+1} \cos((m+1)x)$ mit $t_n(x_k) = y_k,\ k = 0, \dots, n$

Für die Koeffizienten a_k, b_k gilt:

$$a_k = \frac{2}{n+1}\sum_{j=0}^{n} y_j \cos(jx_k),\ b_k = \frac{2}{n+1}\sum_{j=0}^{n} y_j \sin(jx_k)$$

Beweis: 1. Sei t_n^* das komplexe Interpolationspolynom zu $(x_0, y_0), \ldots, (x_n, y_n)$. Nach Satz 2.28 gilt:

$$t_n^*(x) = \sum_{j=0}^{n} c_j e^{ijx} \text{ mit } c_k = \frac{1}{n+1} \sum_{j=0}^{n} y_j w_k^{-j}$$

Setze $c_{-k} = c_{n+1-k},\ k = 1, \ldots, m$, d.h. $c_{-1} = c_n,\ c_{-2} = c_{n-1}, \ldots, c_{-m} = \begin{cases} c_{m+1} & : \ n \text{ gerade} \\ c_{m+2} & : \ n \text{ ungerade} \end{cases}$

Setze $a_k = c_k + c_{-k},\ b_k = i \cdot (c_k - c_{-k}),\ k = 0, \ldots, m$ und $a_{m+1} = \begin{cases} 0 & : \ n \text{ gerade} \\ 2c_{m+1} & : \ n \text{ ungerade} \end{cases}$

Nach dem Lemma 4.26 gilt:

$$(*)\ \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos(kx) + b_k \sin(kx)) = \sum_{k=-m}^{m} c_k e^{ikx}$$

$$\begin{aligned} y_l = \sum_{k=0}^{n} c_k w_l^k &= \sum_{k=0}^{m} c_k w_k^l + \sum_{k=1}^{m} c_{-k} \frac{w_{n+1-k}^l}{w_{-k}^l} + \Theta c_{m+1} w_{m+1}^l \\ &= \sum_{k=-m}^{m} c_k w^l + \Theta c_{m+1} w_{m+1}^l \end{aligned}$$

Für n ungerade gilt: $m + 1 = \frac{n+1}{2}$ und daher

$$\begin{aligned} w_{m+1}^l &= \cos((m+1)x_l) + i \cdot \sin((m+1)x_l) \\ &= \cos((m+1)x_l) + i \cdot 0, \text{ da } (m+1)x_l = \tfrac{n+1}{2} l \tfrac{2\pi}{n+1} = l\pi \end{aligned}$$

$$\begin{aligned} \Longrightarrow\ y_l &= \sum_{k=-m}^{m} c_k w_k^l + \Theta C_{m+1} w_{m+1}^l \\ &\overset{(*)}{=} \tfrac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos(kx_l) + b_k \sin(kx_l)) + \tfrac{\Theta}{2} a_{m+1} \cos((m+1)x_l) \\ &= t_n(x_l) \end{aligned}$$

2. Eindeutigkeit folgt, da das LGS, welches die Koeffiziente a_k, b_k bestimmt, für jede rechte Seite $y_0, \ldots, y_n$ lösbar ist. Daher ist die Matrix regulär.

3. Die explizite Darstellung folgt aus:

$$c_{-k} = c_{n+1-k} = \frac{1}{n+1} \sum_{j=0}^{n} y_j w_{n+1-k}^l = \frac{1}{n+1} \sum_{j=0}^{n} \sum_{y_j} w_k^l$$

Daraus folgt:

$$a_k = c_k + c_{-k} = \frac{1}{n+1} \left(\sum_{j=0}^{n} y_j \left(e^{-ijx_k} + e^{ijx_k} \right) \right) = \frac{2}{n+1} \sum_{j=0}^{n} y_j \cos(jx_k)$$

$$b_k = i \cdot (c_k + c_{-k}) = \frac{1}{n+1} \left(\sum_{j=0}^{n} y_j \left(e^{-ijx_k} - e^{ijx_k} \right) \right) = \frac{2}{n+1} \sum_{j=0}^{n} y_j \sin(jx_k) \quad \square$$

Bemerkung: $t_n(x_l) = t_n^*(x_l)$, aber im Allgemeinen $t_n(x) \neq t_n^*(x)$ für $x \neq x_l$ $(l = 0, \ldots, n)$, sogar $t_n(x) \neq Re(t_n^*(x))$.

Gegeben: $y_0, \ldots, y_n \in \mathbb{R}$

Gesucht: t_n mit $t_n(x_k) = y_k,\ 2\pi$ periodisch, wobei $x_k = k\frac{2\pi}{n+1}$

Im Komplexen: Es existiert genau ein $t_n^* \in T_n,\ t_n^*(x) = \sum_{k=0}^{n} c_k e^{ikx},\ c_k = \frac{1}{n+1}\sum_{k=0}^{n} y_j e^{-ijx_k}$

Im Reellen: Es existiert genau ein $t_n(x) = \frac{a_0}{2} + \sum_{k=1}^{m}(a_k\cos(kx) + b_k\sin(kx)) + \Theta\frac{a_{m+1}}{2}\cos((m+1)x)$ mit

$$\Theta = \begin{cases} 0 & : \ n \text{ gerade} \\ 1 & : \ n \text{ ungerade} \end{cases}, \quad m = \begin{cases} \frac{n}{2} & : \ n \text{ gerade} \\ \frac{n-1}{2} & : \ n \text{ ungerade} \end{cases}$$

$a_k = \frac{2}{n+1}\sum_{j=0}^{n} x_j\cos(jx_k) = c_k + c_{n+1-k}$
$b_k = \frac{2}{n+1}\sum_{j=0}^{n} y_j\sin(jx_k) = i\cdot(c_k - c_{n+1-k})$
$a_{m+1} = \begin{cases} 0 & : \ n \text{ gerade} \\ 2c_{m+1} & : \ n \text{ ungerade} \end{cases}$

Beispiel 4.30
$n = 2,\ x_0 = 0,\ x_1 = \frac{2}{3}\pi,\ x_2 = \frac{4}{3}\pi$

Es gilt: $\cos(x_1) = \cos(x_2) = -\frac{1}{2},\ \sin(x_1) = -\sin(x_2) =: s,\ 2x_1 = x_2$ und $2x_2 = x_1 \bmod 2\pi$

$c_0 = \frac{1}{3}\left(y_0e^{-i0} + y_1e^{-i0} + y_2e^{-i0}\right) = \frac{1}{3}(y_0 + y_1 + y_2)$

$c_1 = \frac{1}{3}\left(y_0e^{-i0} + y_1e^{-i\frac{2}{3}\pi} + y_2e^{-i2\frac{2}{3}\pi}\right) = \frac{1}{3}y_0 - \frac{1}{6}(y_1 + y_2) + i\cdot\frac{5}{3}(y_2 - y_1)$

$c_2 = \frac{1}{3}\left(y_0e^{-i0} + y_1e^{-i\frac{4}{3}\pi} + y_2e^{-i2\frac{4}{3}\pi}\right) = \frac{1}{3}y_0 - \frac{1}{6}(y_1 + y_2) + i\cdot\frac{5}{3}(y_1 - y_2)$

Im Reellen: $m = 1, \Theta = 0$

$a_0 = \frac{2}{3}(y_0 + y_1 + y_2),\ a_1 = \frac{2}{3}\left(y_0 - \frac{1}{2}y_1 - \frac{1}{2}y_2\right)$

$b_1 = \frac{5}{3}(y_1 - y_2)$

$y_0 = 0,\ y_1 = -y_2 = \frac{5}{2}$

$Re(t_2^*(x)) = 1 - \frac{1}{2}(\cos(x) + \cos(2x)) = Re\left(\sum_{k=0}^{3} c_k e^{ikx}\right) \overset{c_k\in\mathbb{R}}{=} \sum_{k=0}^{3} c_k\cos(kx)$

$t_2(x) = 1 - \cos(x)$

Dieses Beispiel (Abb. 4.4) bezieht sich auf die letzte Bemerkung.

Schnelle Fourier Transformation

Die Schnelle Fourier Transformation wird auch **FFT** (Fast Fourier Transformation) genannt.

Ziel: Effiziente Berechnung von $c_0, \ldots, c_n$. (a_k, b_k) können dann im zweiten Schritt schnell bestimmt werden.

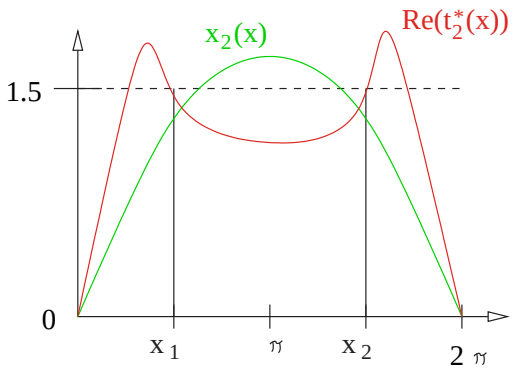


Abbildung 4.4: Beispiel 4.30

Idee: *Divide and conquer*-Verfahren: Das Problem der Größe n wird in 2 äquivalente Probleme der Größe $\frac{n}{2}$ aufgeteilt und separat gelöst, dann werden die beiden Lösungen wieder zu einer gesammten Lösung zusammengefügt. Am einfachsten ist die FFT darstellbar, falls $n = 2^Q - 1$, d.h. 2^Q Daten $y_0, \ldots, y_n$.

Sei n ungerade, dann sei $m = \frac{n-1}{2}$ und sei $l \in \{0, \ldots, n\}$ fest.

$$c_l = \frac{1}{n+1} \sum_{j=0}^{n} y_j w_j^{-l} = \frac{1}{n+1} \left(\sum_{j=0}^{m} y_{2j} w_{2j}^{-l} + \sum_{j=0}^{m} y_{2j+1} w_{2j+1}^{-l} \right) = \frac{1}{n+1} \left(\sum_{j=0}^{m} y_{2j} w_{2j}^{-l} + \hat{w}^{-l} \left(\sum_{j=0}^{m} y_{2j+1} w_{2j}^{-l} \right) \right) \text{ mit } \hat{w} = e^{i\frac{2\pi}{n+1}}$$

Da $n + 1 = 2(m + 1)$ folgt: $c_l = \frac{1}{2} \left(\frac{1}{n+1} \sum_{j=0}^{m} y_{2j} w_{2j}^{-l} + \hat{w}^{-l} \frac{1}{m+1} \sum_{j=0}^{m} y_{2j+1} w_{2j}^{-l} \right)$

Sei $l_1 \equiv l$ modulo $(m + 1)$, d.h. $l_1 \in \{0, \ldots, m\}$ und $l = \lambda(m + 1) + l_1$ $(\lambda \in \mathbb{N})$

$\Longrightarrow$ $l = \frac{1}{2}\lambda(n+1)+l$, $w_{2j}^{-l} = e^{-il2j\frac{2\pi}{n+1}} = e^{-i\lambda_j 2\pi - i\cdot l_1 2j\frac{2\pi}{n+1}} = (1+i\cdot 0)w_{2j}^{-l_1}$ $\Longrightarrow$ $c_l = \frac{1}{2}\left(c_{l_1}^{even} + c_{l_1}^{odd}\hat{w}^{-l}\right)$ (2 Operationen) wobei $c_{l_1}^{even}$, $c_{l_1}^{odd}$ geraade die Koeffizienten des komplexen trigonometrischen Polynoms zu den Daten $(x_0, y_0), \ldots, (x_{n-1}, y_{n-1})$ und $(x_0, y_1), (x_2, y_3), \ldots, (x_{n-1}, y_n)$.

Algorithmus: Siehe nächste Seite.

Allgemein: Pro Level $2(n+1)$ Operationen bei $\log_2(n)$ Levels $\Longrightarrow$ Anzahl der Operationen zur Berechnung von $c_0, \ldots, c_n$ beträgt $2(n + 1)\log_2(n) = O(n \log_2 n)$

Satz 4.31
Sei $n = 2m + 1$ und $y_0, \ldots, y_n$ gegeben. $t_n^*(x) = \sum_{j=0}^{n} c_j e^{ijx}$ sei das komplexe trigonometrische Interpolationspolynom zu $(x_0, y_0), \ldots, (x_n, y_n)$.

Sei $t_n^{even}(x) = \sum_{j=0} c_j^{even} e^{ijx}$ das Interpolationspolynom zu $(x_0, y_0), \ldots, (x_{2m}, y_{2m})$ und $t_n^{odd}(x) = \sum_{j=0}^{m} c_j^{odd} e^{ijx}$ zu $(x_0, y_1), \ldots, (x_{2m}, y_{2m+1})$.

Dann gilt:

$$(*)\ t_n^*(x) = \frac{1}{2}\left(1 + e^{i\cdot(m+1)x}\right) t_n^{even}(x) + \frac{1}{2}\left(1 - e^{i\cdot(m+1)x}\right) t_n^{odd}\left(x - \frac{1}{m+1}\right)$$

und es gilt $c_l = \frac{1}{2}\left(c_l^{even} + \hat{w}^{-l}c_l^{odd}\right)$, $c_{l+m+1} = \frac{1}{2}\left(c_l^{even} - \hat{w}^{-l}c_l^{odd}\right)$ mit $l = 0, \ldots, n$ und $\hat{w} = e^{i\frac{\pi}{m+1}}$

Algorithmus:

Stützstellen ($Q = 3$)

$y_0,\ y_1,\ y_2,\ y_3,\ y_4,\ y_5,\ y_6,\ y_7$

$8 \cdot 2$

$y_0,\ y_2,\ y_4,\ y_6$

$4 \cdot 4$

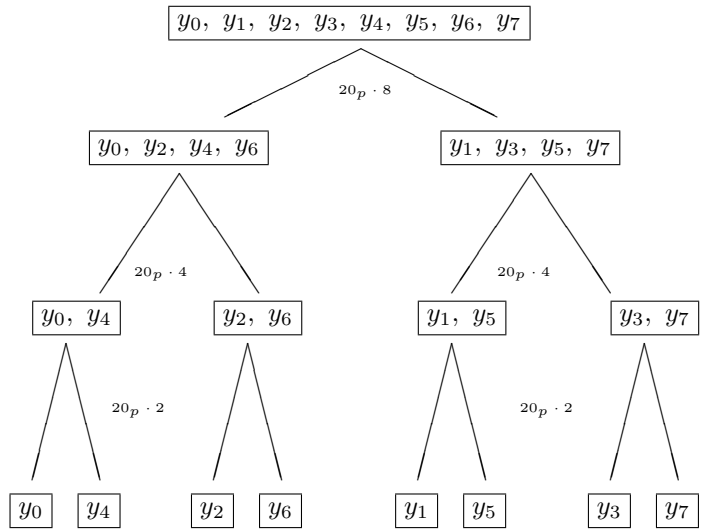
$y_0,\ y_4$

$2 \cdot 8$

Rechenaufwand: 0

$48 = (3 \cdot 2 \cdot (n+1))$

Daten



$$\Longrightarrow \quad t_0^6 = y_0,\ t_1^6 = y_4,\ t_2^0 = y_2,\ \ldots,\ t_7^0 = y_7$$

Beweis: Sei r_n die rechte Seite von $(*)$, d.h.

$$\begin{aligned}
r_n(x) &= \tfrac{1}{2}\sum_{j=0}^{m}\left[\left(1+e^{i(m+1)x}\right)c_j^{odd}e^{ij\left(x-\frac{\pi}{m+1}\right)}\right] \\
&= \tfrac{1}{2}\sum_{j=0}^{m}\left[c_j^{even}\left(e^{ijx}+e^{i(j+m+1)x}\right)+c_j^{odd}\left(e^{ijx}-e^{i(j+m+1)x}\right)e^{-ij\frac{\pi}{m+1}}\right] \\
&= \tfrac{1}{2}\sum_{j=0}^{m}\left(c_j^{even}+e^{-ij\frac{\pi}{m+1}}c_j^{odd}\right)e^{ijx}+\tfrac{1}{2}\sum_{j=,+1}^{1m+1}\left(c_{j-(m+1)-l}^{even}-e^{-ij\frac{\pi}{m+1}}c_{-j(m+1)}^{odd}e^{ijx}\right) \\
&= \sum_{j=0}^{n}\hat{c}_j e^{ijx} \in T_n
\end{aligned}$$

Wegen der Eindeutigkeit des Interpolationspolynom folgt $t_n = r_n$, falls r_n die Interpolationsbediengung erfüllt. Für $x_l = \frac{2\pi}{m+1}$ gilt:

$$e^{-ij(m+1)x_l} = e^{i\frac{2\pi}{2m+1}(m+1)l} = e^{il\pi} = \begin{cases} 1 & : \ l \text{ gerade} \\ -1 & : \ l \text{ ungerade} \end{cases}$$

$$\Longrightarrow \quad r_n(x_l) \overset{(*)}{=} \begin{cases} t_n^{even}(x_l) & : \ l \text{ gerade} \\ t_n^{odd}\left(x_l - \frac{\pi}{m+1}\right) & : \ l \text{ ungerade} \end{cases}$$

$$= \begin{cases} t_n^{even}(x_l) & : \ l \text{ gerade} \\ t_n^{odd}(x_l - 1) & : \ l \text{ ungerade} \end{cases}$$

Also $r_n(x_l) = y_l$ und damit $t_n \equiv r_n \implies \hat{c}_j = c_j$, da $\left\{e^{ijx}\right\}_{j=0}^{m}$ sind linear unabhängig (siehe Übungsaufgaben)

Da $e^{-ij\frac{\pi}{m+1}} = \hat{w}^{-j}$ folgt die Formel für c_l aus der Definition von $\hat{c}_l$ □

Satz 4.32 (Aufwand der FFT)
Sei $n = 2^Q - 1$ für $Q \geq 1$ und seien $y_0, \ldots, y_n$ gegeben. Mittels der FFT können die $(n+1)$-Koeffizienten $c_0, \ldots, c_n$ mit $O(n \log_2 n)$ komplexen Operationen berechnet werden.

Beweis: Wir bezeichnen mit $R(q)$ die Anzahl der Operationen zur Berechnung von $2^q - 1$ Koeffizienten.

Behauptung: $R(q) \leq 2q2^q$ Beweis folgt durch Induktion.

$q = 0 : c_0 = y_0$, d.h. $R(0) = 0$

$$q-1 \rightsquigarrow q: \quad R(q) = 2R\underbrace{(q-1)}_{c_l^{even}, c_l^{odd}} + 2\underbrace{(2^q-1)}_{\hat{w}^{-l}c_l^{odd/even} + c_l^{odd/even}}$$

$$\begin{aligned} \implies R(q) &\leq 2 \cdot 2(q-1)2^{q-1} + 2(2^q - 1) \\ &\leq 2(q-1)^q + 22^q = 2q2^q \end{aligned}$$

Da $n+1 = 2^q$, d.h. $Q = \log_2(n+1)$, folgt $R(Q) = 2 \cdot (n+1)\log_2(n+1) = O(n \log_2 n)$ □

Algorithmus:

`Für` $q = 0, \ldots, Q$ `sei` $t_k^q(x) = \sum_{j=0}^{2^{q-1}} c_{k,j}^q e^{ijx}, \quad k = 0, \ldots, 2^{Q-q} - 1$
`das Interpolationspolynom zu` $\left(x_{j2^{Q-q}}, y_{2j^{Q-q}+k}\right)_{j=0}^{2^{q-1}}$
`Nach Satz 4.31 mit` $m = 2^q - 1$ `bzw` $n = 2^{q+1} - 1$ `gilt:`

$$\begin{aligned} c_{k,l}^{q+1} &= \tfrac{1}{2}\left(c_{k,l}^q + e^{-i\frac{2\pi}{2^{q+1}}l} c_{k+2^{Q-q-1},l}^q\right) \quad l = 0, \ldots, 2^{q-1} \\ c_{k,l+2}^{q+1} &= \tfrac{1}{2}\left(c_{k,l}^q - e^{-i\frac{2\pi}{2^{q+1}}l} c_{k+2^{Q-q-1},l}^q\right) \end{aligned}$$

`Start der Iteration:` $\boxed{c_{k,0}^0 = y_k}$ `(sofort lösbar)`

Speicherbedarf: Für den Schritt $q \longrightarrow q+1$ müssen Matrizen berechnet werden:

$$C^q = \begin{pmatrix} q \\ c_{k,l} \end{pmatrix}, \; C^{q+1} = \begin{pmatrix} q+1 \\ c_{k,l} \end{pmatrix}$$

$C^q \in \mathbb{C}^{2^{Q-q} \times 2^q}$ bzw. $C^{q+1} \in \mathbb{C}^{2^{Q-q-1} \times 2^{q+1}}$

Beide Matrizen sind von der selben Dimension: $2^{Q-q}2^q = 2^Q = n+1$ und $2^{Q-q-q}2^{q+1} = 2^Q = n+1$

Daher sollen die Koeffizienten $c_{k,l}^q$, $c_{k,l}^{q+1}$ in Vektoren der Dimension $n+1$ gespeichert werden.

$$c_{k,l}^q =: C[2^q k + l]$$

$$c_{k,l}^{q+1} =: D[2^{q+1} k + l]$$

Es gilt: $e^{-\frac{2\pi}{2^{q+1}}l} = e^{-l\frac{2\pi}{2^Q}2^{Q-q-q}l} =: W[2^{Q-q-q}l]$, wobei der Vektor $W[l] := e^{-i\frac{2\pi}{2^Q}l},\ l = 0, \ldots, 2^q - 1$

Sei $\hat{w} := e^{-i\frac{2\pi}{2^Q}}$

Für $l = 0, \ldots, 2^{Q-1}$:

$q = 0$ [$C[l] = y_l$; $W[l] = \hat{w}^l$]

Für $q = 0, \ldots, Q-1$

$q \longrightarrow q+1$ [für $k = 0, \ldots, 2^{Q-(q+1)} - 1$ [für $l = 0, \ldots, 2^q - 1$ [$(*)$ [$u = C[2^q k + l]$; $v = W[2^{Q-q-1}l]C[2^q(k + 2^{Q-q-1}) + l]$; $D[2^{q+1}k + l] = \frac{1}{2}(u+v)$; $D[2^{q+1}k + l + 2^q] = \frac{1}{2}(u-v)$]]]

Aufwand: $(*)$ benötigt 3 Operationen. Anzahl der Durchläufe von $(*)$

$$Q2^{Q-q-1}2^q = Q2^{q-1} = \log_2(n+1)\frac{n+1}{2}$$

Daher ist der gesamte Aufwand gleich

$$3\log_2(n+1)\frac{n+1}{2} = O(n\log_2 n)$$

Bemerkung: Der Algorithmus kann so umgeschrieben werden, dass der Vektor D nicht gebraucht wird, und es existiert auch Varianten für den Fall $n \neq 2^Q - 1$

4.7 Spline-Interpolation

Motivation: Bei großen Werten von n führt die Polynominterpolation zu stark oszillierenden Interpolationspolynomen, da $p_n \in C^\infty(I)$. Das Problem tritt besonders dann auf, wenn die Stützstellen vorgegeben sind. Daher verwendet man häufig polynomielle Funktionen, d.h.

$$P|[x_{i-1}, x_i] \in P_r$$

mit $r \ll n$. Die Interpolationsbedingung $p(x_i) = y_i$ führt zu $p \in C^0(I)$, aber p ist i.a. nicht in $C^\infty(I)$, sondern $p \in C^4(I)$. Die Parameter (r, q) sind geeignet zu wählen.

$$P|_{[x_{i-1},x_i]} \in P_r,\ p(x) = \begin{cases} p_1(x) & : \quad x \in (x_0, x_1] \\ p_2(x) & : \quad x \in (x_2, x_3] \\ \vdots & \\ p_2(x) & : \quad x \in (x_{n-1}, x_n] \end{cases}$$

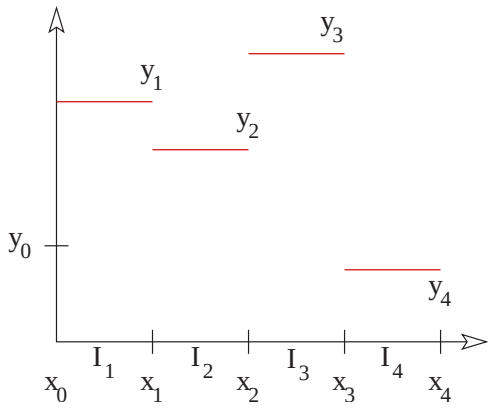


Abbildung 4.5: Beispiel 4.33: Treppenfunktionen

$p_i \in \mathbb{P}_r$ hat folgende Interpolationsbedingungen: $p_i(x_{i-1}) = y_{i-1},\ p_i(x_i) = y_i \iff p(x_k) = y_k,\ k = 0, \ldots, n$

Notation: $\Delta = (x_0, \ldots, x_n)$ ist eine Zerlegung von $I = [a, b]$ mit $x_0 = a,\ x_n = b,\ x_{i-1} < x_i\ (1 \leq i \leq n)$

Mit $h_i := x_i - x_{i-1} > 0$ bezeichnen wir die Länge des Teilintervals $I_i := (x_{i-1}, x_i),\ I_0 := \{a\},\ i = 1, \ldots, n$. Die Feinheit der Zerlegung ist gegeben durch:

$$h = \max_{1 \leq i \leq n} h_i$$

Für $r, q \in \mathbb{N}$ definieren wir den Raum

$$S_\Delta^{r,q} := \left\{ p \in C^q(I) \;\middle|\; P|_{I_i} =: p_i \in P_r \text{ für } 1 \leq i \leq n \right\}$$

Gegeben: Zerlegung Δ der Daten $y_0, \ldots, y_n$ und $r, q \in \mathbb{N}$

Gesucht: $P_\Delta \in S_\Delta^{r,q}$ mit $P_\Delta(x_k) = y_k,\ k = 0, \ldots, n$

Beispiel: Die Abbildungen 4.4 zeigen 4 verschiedene Interpolationen mit ihren Fehlern (polynomiel mit gleichmäßig verteilte Stützstellen, Tschebyschev-Interpolation, trigonometrische Interpolation und Spline-Interpolation). Dabei gilt es: $f(x) = \frac{1}{1+x^2},\ y_k = f(x_k), I = [0, 5],\ n = 11$ (Siehe Seite 96, am Ende dieses Kapitels)

Beispiel 4.33
$\underline{r = 0}$: Die einzige Interpolation durch stückweise konstanten Funktionen ist gegeben durch $P_\Delta(x) = y_i$ für $x \in I_i$ bzw. $p_i(x) = y_i$.
Abbildung 4.5 zeigt die entstandene Treppenfunktion, in diesem Fall ist P_Δ nicht einmal stetig!

$\underline{r = 1}$: Es soll $p_i \in \mathbb{P}_1$ und $p_i(x_{i-1}) = y_{i-1},\ p(x_i) = y_i$.
Abbildung 4.6 zeigt die eindeutig bestimmen p_i Funktionen durch $p_i(x) = y_i + \frac{y_i - y_{i-1}}{h_i}(x - x_i)$. Damit gilt: $P_\Delta \in S_\Delta^{1,0}$

$\underline{r = 3}$: Annahme: $y_k = f(x_k)$ mit $f \in C^k(I)$

(i) **Fall**: Wähle für $i = 1, \ldots, n$ Werte $x_{ij} \in I_i$ für $j = 1, 2$ und definiere p_i als Interpolationspolynom zu $(x_{i-1}, y_{i-1}), (x_{i1}, f(x_{i1})), (x_{i1}, f(x_{i1})), (x_{i2}, f(x_{i2})), (x_{i+1}, f(x_{i+1})) \implies p_i \in P_3$ und $P_\Delta \in S_\Delta^{3,0}$. Nach Satz 4.4 gilt:

$$\begin{aligned} |f(x) - P_\Delta(x)| &= |f(x) - p_i(x)| = f^{(4)}(\xi x)\tfrac{1}{4!}h_i^4 \text{ für } x \in I_i \\ &\leq \left\|f^{(4)}\right\|_\infty \tfrac{1}{4!}h^4 \end{aligned}$$

(ii) **Fall**: Wähle $p_i \in \mathbb{P}_3$ durch Hermite Interpolation zu $(x_{i-1}, y_{i-1}), (x_{i-1}, f'(x_{i-1})), (x_i, y_i), (x_i, f'(x_i)) \implies P_\Delta \in S_\Delta^{3,1}$ und $\|f - P_\Delta\|_\infty \leq h^4 \frac{1}{4!} \left\|f^{(4)}\right\|_\infty$

Frage: $P_\Delta \in S_\Delta^{3,4}$?

Bemerkung: Sei $n > r$, dann ist das Interpolationsproblem in $S_\Delta^{r,q}$ für $q \geq r$ i.a. schlecht gestellt (in dieser Situation nicht lösbar).

Freiheitsgrade: $p_i \in \mathbb{P}_r$ führt auf $(r+1)$ Koeffizienten, also: $n(r+1)$ Freiheitsgrade.

Anzahl der Bedingungen:

Auf I_1 : 2 Interpolationsbedingungen
I_2 : 2 Interpolationsbedingungen + q Stetigkeitsbedingunge in x_1
$\vdots$
I_n : 2 Interpolationsbedingungen + q Stetigkeitsbedingunge in x_n

$\implies 2n + q(n-1) = n(q+1) - q$

Ist $q \geq r$ so folt: $2n + q(n-1) \geq 2n + r(n-1) = n(r+1) + n - r > n(r+1)$, Falls $n - r > 0$, d.h. mehr Bedingungen als Freiheitsgrade und das Problem ist nicht lösbar.

Spezialfall: $q = r - 1$ (Spline-Interpolation)

Bedingungen: $n(q+1) - q = n(r+1) - q$, d.h. q mehr Freiheitsgrade als Bedingungen.

Kubische Spline-Interpolation

Gegeben: $\Delta = (x_0, \ldots, x_n)$ Zerlegung des Intervals $[a, b] = I$ und Daten $y_0, \ldots, y_n \in \mathbb{R}$

Gesucht ist $P_\Delta \in S_\Delta^{3,2}$ mit $P_\Delta(x_i) = y_i$ $(0 \leq i \leq n)$ und

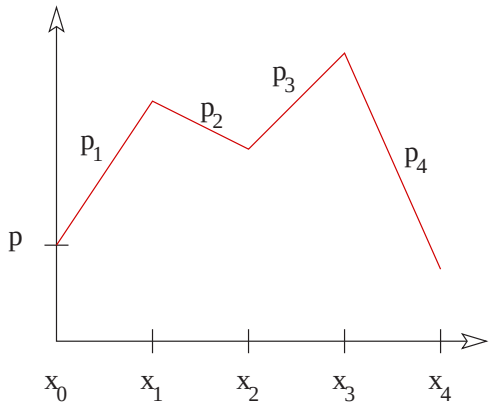


Abbildung 4.6: Beispiel 4.33: Gerade

a) $P''_\Delta(a) = M_a,\ P''_\Delta(b) = M_b$ für $M_a, M_b \in \mathbb{R}$ gegeben. Im Fall $M_a = M_b = 0$ spricht man vom **natürlichem kubischen Spline**

b) $p'(a) = g(a),\ p'(b) = g_b$ für $g_a, g_b \in \mathbb{R}$ gegeben

c) P_Δ ist **periodisch fortsetzbar** in $\mathbb{C}^2(\mathbb{R})$ (nur falls $y_0 = y_n$), dann $p'(a) = p'(b),\ p''(a) = p''(b)$

d) **not-a-knot**-Bedingung: $P_{\Delta|_{[I_1 \cup I_2]}} \in \mathbb{P}_3,\ P_{\Delta|_{[I_{n-1} \cup I_n]}} \in \mathbb{P}_3$, d.h. die Zusatzbedingungen werden verwendet, um die Sprünge in P'''_Δ für $x = x_1,\ x = x_{n-1}$ zu eliminieren.

Satz 4.34 (Existenz und Eindeutigkeit)
Zu gegebener Zerlegung Δ und Daten $y_0, \dots, y_n$ existiert genau ein $P_\Delta \in S^{3,2}_\Delta$ mit $p(x_k) = y_k$ und P_Δ erfüllt die Zusatzbedingungen a), b), c), oder d). Im Fall c) muss gelten: $y_0 = y_n$

Beweis: **Idee:** Stelle LGS für jeden sogenannten Moment $M_j = P''_\Delta(x_j)$ auf.

Da p''_j linear auf $I_j = (x_{j-1}, x_j]$ ist, muss gelten: $p''_j(x) = \frac{1}{h_j}\left(M_j(x - x_{j-1}) + M_{j-1}(x_j - x)\right)$.

Durch zweimalige Integration folgt für geeignete Integrationskonstanten $a_j, b_i \in \mathbb{R}$:

$$p_j(x) = \frac{1}{6h_j}\left(M_j(x - x_{j-1})^3 + M_{j-1}(x_j - x)^3\right) + b_j\left(x - \frac{x_j + x_{j-1}}{2}\right) + a_j \ (*)$$

Aus den Interpolationsbedingungen $p_j(x_{j-1}) = y_{j-1},\ p_j(x_j) = y_j$ folgt:

$$\begin{aligned} y_{j-1} &= \tfrac{1}{6h_j} M_{j-1} h_j^3 - b_j \tfrac{1}{2} h_j + c_j \\ y_j &= \tfrac{1}{6h_j} M_j h_j^3 + b_j \tfrac{1}{2} h_j + a_j \end{aligned}$$

Dieses ist ein 2×2 LGS für a_j, b_j mit der Lösung

$$\begin{aligned} (**)\ a_j &= \tfrac{1}{2}(y_j + y_{j-1}) - \tfrac{1}{12} h_j^2 (M_j + M_{j-1}) \\ b_j &= \tfrac{1}{h_j}(y_j - y_{j-1}) - \tfrac{1}{6} h_j (M_j - M_{j-1}) \end{aligned}$$

Damit hängen die p_j nur von den Momenten $M_0, \dots, M_n$ ab.

Es bleiben noch die $n-1$ Bedingungen $p'(x_j) = p'_{j+1}(x_j)$ für $j = 1, \dots, n-1$.
Aus $(*)$ und $(**)$ folgt: $p'_j(x) = \frac{1}{2j}\left(M_j (x - x_{j-1})^2 - M_{j-1}(x_j - x)^2\right) + \frac{1}{h_j}(y_j - y_{j-1}) - \frac{1}{6} h_j (M_j - M_{j-1})$

Daher ist $p'_j(x_j) = p'_{j+1}(x_j)$ äquivalent zu
$\frac{1}{2} M_j (h_{j+1} + h_j) + \frac{1}{6} h_{j+1}(M_{j+1} - M_j) - \frac{1}{6} h_j (M_j - M_{j-1}) = \frac{1}{h_{j+1}}(y_{j+1} - y_j) - \frac{1}{h_j}(y_j - y_{j-1})$
für $j = 1, \dots, n-1$ bzw. $\frac{1}{6} h_j M_{j-1} + \frac{1}{3}(h_j + h_{j+1}) M_j + \frac{1}{6} h_j M_{j+1} = y[x_j, x_{j+1}] - y[x_{j-1}, x_j]$

Mit der zweiten dividierten Differenz $y[x_{j-1}, x_j, x_{j+1}] = \frac{y[x_j, x_{j+1}] - y[x_{i-1} - x_j]}{x_{j+1} - x_{j-1}}$ und da $x_{j+1} - x_{j-1} = h_j + h_{j+1}$ folgt:

$$\mu_j M_{j-1} + M_j + \lambda_j M_{j+1} = 3y[x_{j-1}, x_j, x_{x+1}]$$

mit $\mu_j = \frac{h_j}{2(h_j + h_{j+1})},\ \lambda = \frac{j_{j+1}}{2(h_j + h_{j+1})}$

Wir erhalten ein $(n-1) \times (n+1)$ LGS für die Momente $M_0, \dots, M_n$

Fall a) $M_0 = M_a,\ M_n = M_b$

Dies führt auf das $(n-1)\times(n-1)$ LGS für $M_1,\dots,M_{n-1}$ der Form

$$A\begin{pmatrix} M_1 \\ \vdots \\ M_{n-1}\end{pmatrix} = \begin{pmatrix} 3y[x_0,x_1,x_2]-\mu_1 M_a \\ 3y[x_1,x_2,x_3] \\ \vdots \\ 3y[x_{n-2},x_{n-1},x_n]-\lambda_{n-1}M_b\end{pmatrix}$$

mit

$$A = \begin{pmatrix} 1 & \lambda_1 & & 0 \\ \mu_2 & \ddots & \ddots & \\ & \ddots & \ddots & \lambda_{n-2} \\ 0 & & \mu_{n-1} & 1\end{pmatrix}$$

A ist regulär nach dem folgenden Lemma, da $\mu_j+\lambda_j<1$ und $\lambda_1<1,\ \mu_{n-1}<1$

Die Fälle b,c,d führen analog auf einfach strukturierte LGS mit regulären Matrizen, d.h. $p_1,\dots,p_n$ eindeutig duch $(*),(**)$ festgelegt □

Lemma 4.35
Sei $A\in\mathbb{R}^{n\times n}$ eine **tridiagonale Matrix**, d.h.

$$A = tridiag(b_i,a_i,c_i) = \begin{pmatrix} a_1 & c_1 & & 0 \\ b_2 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ 0 & & b_n & a_n\end{pmatrix}$$

Es gelte: $|a_1|>|c_1|>0$ und $|a_n|>|b_n|>0$ und $|a_i|\geq|b_i|+|c_i|,\ b_i\neq 0,\ c_2\neq 0,\ 2\leq i\leq n-1$

Dann:

(i) A ist regulär.

(ii) $A=LR$ mit $L=tridiag(b_i,\alpha_i,0)$ und $R=tridiag(0,1,\gamma_i)$ mit $\alpha_1=a_1,\ \gamma_1=c_1\alpha_1^{-1}$ und für $2\leq i\leq n:\ \alpha_i=a_i-b_i\gamma_{i-1},\ \gamma_i=c_i\alpha_i^{-1}$

Daher kann $Ax=b$ in $O(n)$ Operationen gelöst werden.

Beweis: Siehe Übungsaufgaben

Lemma 4.36
Die Spline-Interpolation mit kubischen Splines und Zusatzbedingungen a), b), c) oder d) kann mit $O(n)$ Operationen gelöst werden.

Beweis: a) folgt aus 4.24, 4.35
b), c), d): Beispiel im Schabach-Werner

Historisch: Interpolation durch biegsamen Stab (engl: spline) und Brett mit Nägeln bei (x_k,y_k). Der Stab hat minimale Krümmung, d.h. die Funktion minimiert $\int_I \frac{(y''(t))^2}{1+(y'(t))^2}\mathrm{d}t$ über alle glatten Funktionen y mit $y(x_k)=y_k$. Für den Fall kleiner erster Ableitungen entspricht das näherungsweise. $\int_I y''(t)^2\mathrm{d}t$

Satz 4.37 (Minimierungseigenschaft kubischer Splines)
Sei $\Delta = (x_0, \ldots, x_n)$ eine Zerlegung von $I = [a, b]$ und $y_0, \ldots, y_n \in \mathbb{R}$ gegeben. Sei $P_\Delta \in S_\Delta^{3,2}$ ein kubischer Spline mit $P_\Delta(x_k) = y_k$ und

(i) $P_\Delta''(a) = 0,\ P_\Delta''(b) = 0$

(ii) P_Δ periodisch fortsetzbar in $C^2(\mathbb{R})$

(iii) $P_\Delta'(a) = g_a,\ P_\Delta'(b) = g_b$

Dann gilt für alle $f \in C^2(a, b)$ mit $f(x_k) = y_k$ und $\int_a^b |f''|^2 \leq \infty$,
d.h. $f''' \in L^2(a, b)$: $\int_a^b |f''(x)|^2 \,\mathrm{d}x \geq \int_a^b |P_\Delta''(x)|^2 \,\mathrm{d}x$

Daher muss im Fall (ii) $f \in C^2(\mathbb{R})$ periodisch mit Periode $b - 1$ sein, bzw. im Fall (i) muss $f'(a) = g_a$, $f'(b) = g_b$ gelten.

Beweis: Wir benötigen das folgende Lemma

Lemma 4.38 (Holladay Identität)
Sei $f \in C^2(a, b)$ mit $\int_a^b |f'''|^2 < \infty$ und $P_\Delta \in S_\Delta^{3,2}$, dann gilt:

$$\int_a^b |f'' - P_\Delta|^2 = \int_a^b |f''|^2 \int_a^b |P_\Delta''|^2 - 2 \cdot \left\{ [(f'(x) - P_\Delta'(x))P_\Delta''(x)]_{x=a}^b - \sum_{i=1}^n [(f(x) - P_\Delta(x))P_\Delta'''(x)]_{x=x_{i-1}^+}^{x_i^-} \right\}$$

Dabei wurden die folgenden Abkürzungen benutzt:

$$[g(x)]_{x=a}^b = g(b) - g(a)$$

$$[g(x)]_{x=x_{i-1}^+}^{x_i^-} = \lim_{x \nearrow x_i} g(x) - \lim_{x \searrow x_{i-1}} g(x) \text{ beachte : } P_\Delta''' \text{ ist unstetig!}$$

Beweis: (Fortsetzung des Beweises von Lemma 4.38)

In den 3 Fällen a), b), c) verschwindet der Term $2 \cdot \{\cdots\}$ in der Holladay Identität
$\Longrightarrow\ 0 \leq \int_a^b |f'' - P_\Delta''|^2 = \int_a^b |f''|^2 - \int_a^b |P_\Delta''|^2$ □

Satz 4.39 (Fehlerabschätzung)
Sei Δ eine Zerlegung von I mit $h \leq Kh_i$ $(1 \leq i \leq n)$ für ein $K > 0$. Sei $f \in C^4(a, b)$ mit $\left|f^{(4)}\right| < L$ für $x \in (a, b)$.
Sei $P_\Delta \in S_\Delta^{3,2}$ mit $P_\Delta(x_k) = f(x_k)$ und $P_\Delta'(a) = f'(a),\ P_\Delta'(b) = f'(b)$.

Dann gilt für $l = 0, 1, 2, 3$: $\left|f^{(l)}(x) - P_\Delta^{(l)}(x)\right| \leq 2LKh^{4-l}$. Also: $|f(x) - P_\Delta(x)| \leq 2LKh^4$

Beweis: Beweis von Lemma 4.37

$$\begin{aligned}
\int_a^b |f'' - P''_\Delta|^2 &= \int_a^b -2\int_a^b f'' P''_\Delta + \int_a^b |P''_\Delta|^2 \\
&= \int_a^b |f''|^2 - \int_a^b |P''_\Delta|^2 - 2\int_a^b (f'' - P''_\Delta)P''_\Delta \\
&= \int_a^b |f''|^2 - \int_a^b |P''_\Delta|^2 - 2\sum_{i=1}^n \underbrace{\int_{I_i} (f'' - p_i'')p_i}_{:=p_i}
\end{aligned}$$

$$\begin{aligned}
A_i = \int_{x_{i-1}}^{x_i} (f'' - p_i'')p_i'' &= [(f' - p_i')p_i'']_{x=x_{i-1}}^{x_i} - \int_{x_{i-1}}^{x_i} (f' - p_i')p_i''' \\
&= [(f' - p_i')p_i'']_{x=x_{i-1}}^{x_i} - [(f - p_i)p_i''']_{x_{i-1}}^{x_i} + \sum_{x_{i-1}}^{x_i} (f - p_i)p_i^{(4)}
\end{aligned}$$

Da $p_i^{(4)} \equiv 0$, da $p_i \in P_3$ und

$$\begin{aligned}
\sum_{i=1}^n [(f' - p_i')p_i'']_{x=x_{i-1}}^{x_i} &\overset{(f,p_i',p_i''\in C^0)}{=} \sum_{i=1}^n [(f' - P'_\Delta]P''_\Delta]_{x=x_{i-1}}^{x_i} \\
&= \sum_{i=1}^n [(f'(x_i) - p'_\Delta(x_i))P''_\Delta(x_i) - (f'(x_{i-1} - P'_\Delta(x_{i-1}))P''_\Delta(x_{i-1})] \\
&\overset{\text{Teleskopsumme}}{=} (f'(x_n) - P'_\Delta(x_n))P''_\Delta(x_n) - f'(x_0) - P'_\Delta(x_0))P''_\Delta(x_0) \\
&= [(f'(x) - P'_\Delta(x))P''_\Delta(x)]_{x=a}^b \quad \text{da } x_0 = a,\ x_n = b \\
\Longrightarrow \sum_{i=1}^n A_i &= [(f'(x) - P'_\Delta(x))P''_\Delta(x)]_{x=a}^b - \sum_{i=1}^n [(f(x) - p_i(x))p_i'''(x)]_{x=x_{i-1}}^{x_i}
\end{aligned}$$

$\Longrightarrow$ Hollady Identität $\square$

B-Splines(Skizze)

Ziel: Konstruktion einer einfachen Basis von $S_\Delta^{k,k-1}$ mit

1. positiven Basisfunktionen für numerische Stabilität
2. möglichst kleinem Träger

Definition 4.40 (B-Splines)
Sei $(t_i)_{i\in\mathbb{Z}}$ *eine monoton nicht-fallende Folge mit* $\lim_{i \longrightarrow \pm\infty} t_i = \pm\infty$. *Dann sind die* **B-Splines**

$B_{i,k} : \mathbb{R} \longrightarrow \mathbb{R}$ vom Grad $k \in \mathbb{N}$ rekursiv definiert durch $B_{i,0}(x) = \begin{cases} 1 & : \ t_i \le x \le t_{i+1} \\ 0 & : \ \text{sonst} \end{cases}$

und $B_{i,k} = \omega_{i,k}(x)B_{i,k-1}(x) + (1 - \omega_{i+1,k}(x))B_{i+1,k-1}(x)$ mit

$\omega_{i,k}(x) = \begin{cases} \frac{x-t_i}{t_{i+k}-t_i} & : \ t_{i+k} \neq t_i \\ 0 & : \ \text{sonst} \end{cases}$

Beispiel:
Die Abb. 4.7 zeigt 6 verschiedene Beispiele, die bei den B-Splines auftreten können.

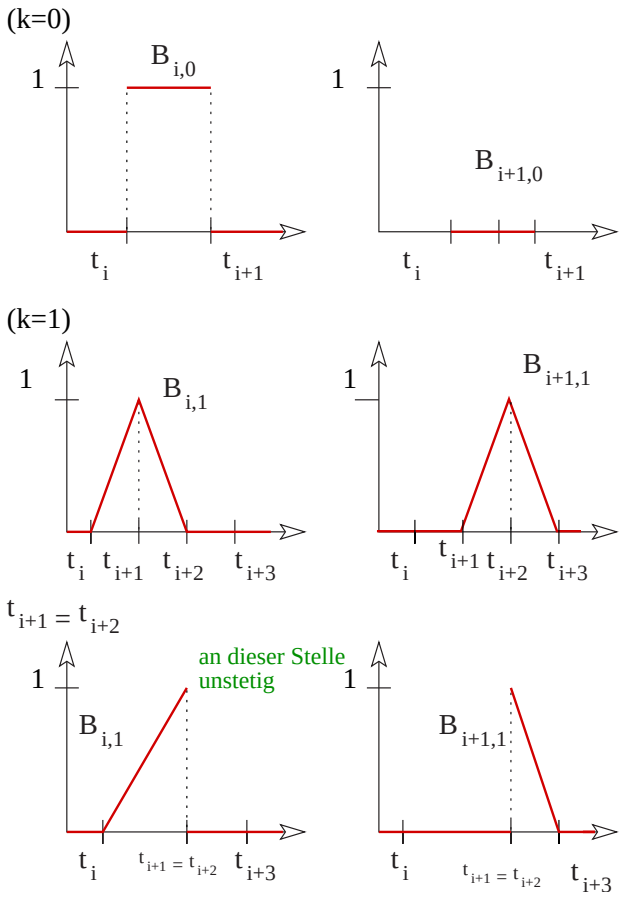


Abbildung 4.7: B-Splines

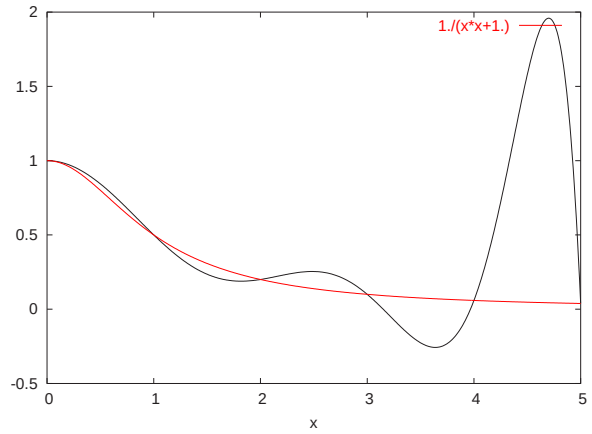
Satz 4.41 (Eigenschaften der B-Splines)

(i) $B_{ik}|_{[t_j,t_{j+1}]} \in \mathbb{P}_n \ \forall\ i,j \in \mathbb{Z}$

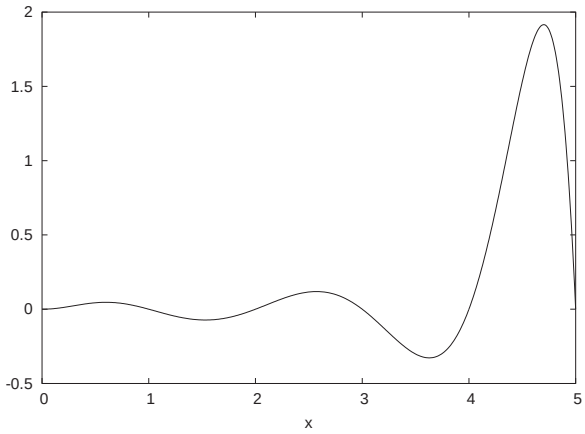
(ii) $supp(B_{i,k}) = [t_i, t_{i+k+1}]$, falls $t_i \leq t_{i+k+1}$, sonst ist $B_{i,k} \equiv 0$

(iii) $B_{ik} \geq 0,\ \sum\limits_{i\in\mathbb{Z}} B_{ik}(x) = 1,\ \forall\ x \in \mathbb{R}$ (Zerlegung der 1)

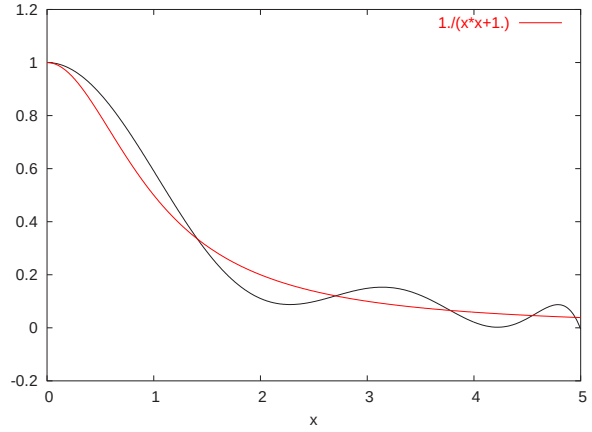
(iv) Falls $\forall\ i \in \mathbb{Z}:\ t_i < t_{i+1}$, dann ist $B_{i,k} \in C^{k-1}$ und $(B_{i,k})_{i\in\mathbb{Z}}$ bilden Basis in $S_\Delta^{k,k-1}$, sie sind also linear unabhängig.



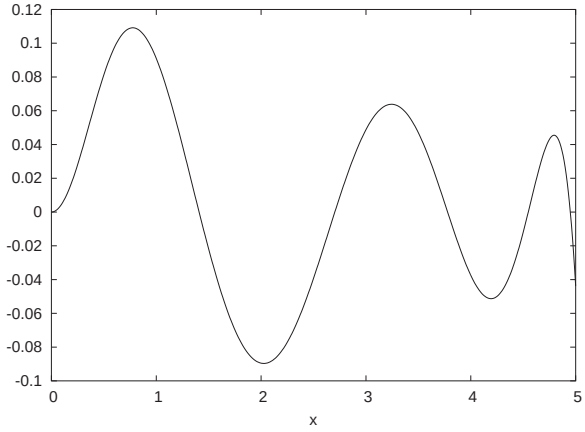
Polynomielle mit gleichmäßig verteilten Stützstellen



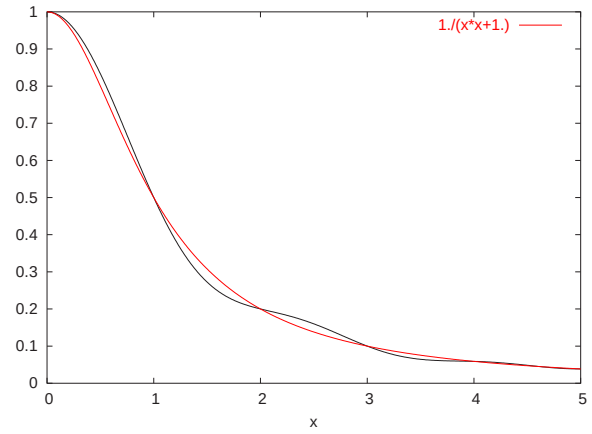
Fehler der Interpolation



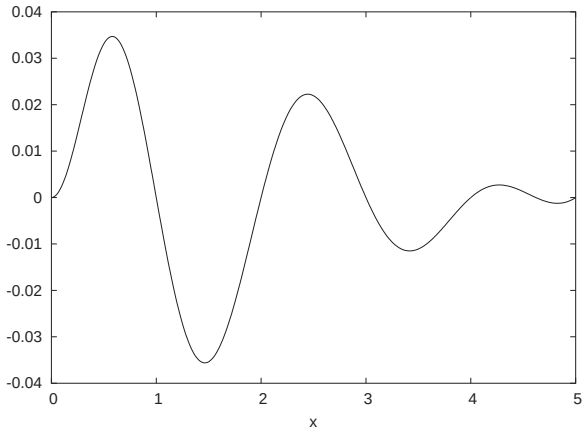
Tschebyschev-Interpolation



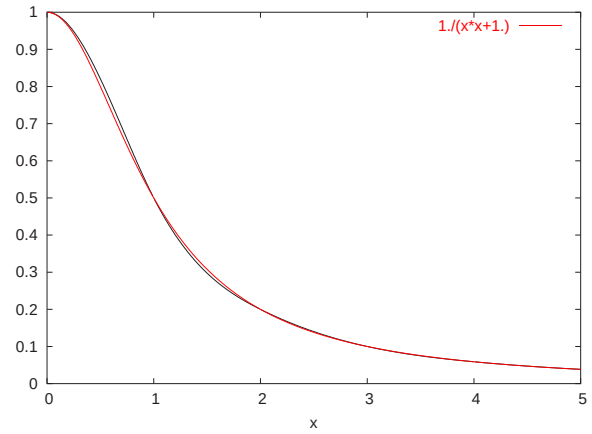
Fehler der Interpolation



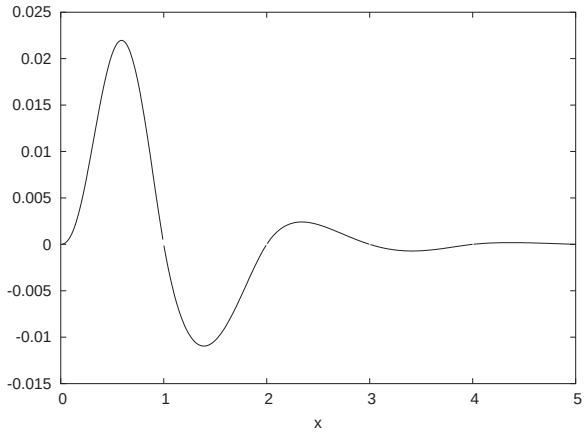
Trigonometrische Interpolation



Fehler der Interpolation



Spline-Interpolation



Fehler der Interpolation

Abbildung 4.8: Unterschiede einiger Interpolationen

Kapitel 5

Numerische Intergration

Ziel: Approximation von $I(f) := \int_a^b \omega(x) f(x) \mathrm{d}x$ für $f \in C^k(a,b)$ und einer festen Gewichtsfunktion $\omega \in L^1(a,b)$ (einfach $\omega \in C^0(a,b)$)

Ansatz: Approximiere $I(f)$ durch eine Summe

$$I_n(f) := \sum_{j=0}^{m} \sum_{l=0}^{m_j} f^{(k)}(x_j) \omega_j^l$$

Definition 5.1
Eine Funktion $I_n : C^k(a,b) \longrightarrow \mathbb{R}$ *der Form*

$$I_n(f) := \sum_{j=0}^{m} \sum_{l=0}^{m_j} f^{(k)}(x_j) \omega_j^l$$

heißt **Quadraturformel** *mit den Stützstellen* $x_j \in [a,b]$ *und den* **Gewichten** $\omega_j^l \in \mathbb{R}$. *Dabei ist* $m \in \mathbb{N}$ *und* $m \in \{0, \dots, k\}$ *und* $n+1 = \sum_{j=0}^{m} m_j$.

Die Quadraturen heißen **exakt** *für* $\mathbb{P}_n$ *(bezüglich* ω*)* $\iff$ $I_n(p) = I(p) \ \forall \ p \in \mathbb{P}_n$.

$$R(f) = I(f) - I_n(f)$$

ist das zu I_n *gehörende* **Fehlerfunktional**.

Bemerkung: Für die allgemeine Definition der Quadratur siehe die Definition der Hermite Interpolation. Im folgenden betrachten wir meistens Quadraturen der Form

$$I_n(f) = \sum_{l=0}^{n} \omega_l f(x_l), \text{ d.h. } m_j = 0$$

Beispiel 5.2 ($n \equiv 1$)
Die Abbildung 5.1 verdeutlicht diese Beispiele

(i) Mittelpunktregel: $I_1(f) = (b-a) f\left(\frac{a+b}{2}\right)$

(ii) Trapezregel: $I_2(f) = \frac{b-a}{2}\left(f(a) + f(b)\right)$

(iii) Simpsonregel: $I_3(f) = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$

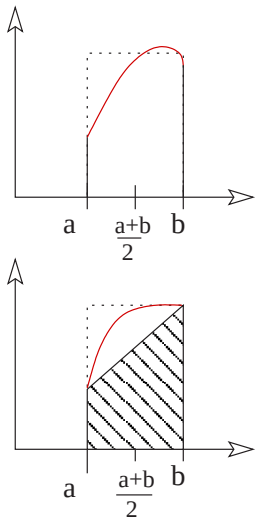


Abbildung 5.1: Beispiel 5.1

Satz 5.3
Gegeben seien $\omega \in L^1(a,b)$ und Stützstellen $x_0, \ldots, x_n$. Dann existiert genau eine Quadraturformel der Form

$$I_n(f) = \sum_{l=0}^{n} \omega_l f(x_l)$$

welche exakt auf dem Polynomraum $\mathbb{P}_n$ ist und deren Gewichte sind gegeben durch

$$\omega_j := \int_a^b \omega(x) L_j^n(x) \mathrm{d}x,$$

wobei $L_j^n(x) = \prod_{\substack{l=0 \\ l \neq j}} \frac{(x-x_l)}{(x_j-x_l)}$ (Lagrange Basispolynome)

Beweis: I_n exakt auf $\mathbb{P}_n$
$\iff$ $I_n(p) = I(p)\ \forall p \in \mathbb{P}_n$
$\iff$ $I_n(L_l^n) = I(L_l^n)$ für $l = 0, \ldots, n$ $(*)$
$\int_a^b \omega(x) L_l^n(x) = \sum_{j=0}^{n} \omega_j L_l^n(x_j) = \omega_l$

Da $L_l^n(x_i) = \delta_{ij}$

$(*)$ $p = \sum_{l=0}^{n} a_l L_l^n$; $I_n(p) = \sum_{l=0}^{n} a_l I_n(L_l^n) = \sum_{l=0}^{n} a_l I(L_l^n) = I(p)$

Bemerkung: Es ist $I_n(f) = I(p_n)$, wobei $p_n \in \mathbb{P}_n$ das eindeutig bestimmte Interpolationspolynom zu $(x_0, f(x_0)), \ldots, (x_n, f(x_n))$ ist.

$$I_n(f) = \sum_{l=0}^{n} \omega_l f(x_l) = \sum_{l=0}^{n} \int_a^b \omega(x) L_l^n(x) f(x_l) \mathrm{d}x = \int_a^b \omega(x) \underbrace{\sum_{l=0}^{n} L_L^n(x) f(x)}_{p_n(x)} \mathrm{d}x = \int_a^b \omega(x) p_n(x) \mathrm{d}x = I(p_n)$$

Definition 5.4
Eine Quadraturformel $I_n(f) = \sum_{l=0}^{n} \omega_l f(x_l)$ *zu gegebenen Stützstellen* $a \leq x_0 < x_1 < \ldots < x_n \leq b$ *und Gewichtsfunktionen* $\omega \in L^1(a,b)$ *heißt* **Interpolationsquadratur**, *wenn sie auf* $\mathbb{P}_n$ *exakt ist. Nach Satz 5.3 ist sie eindeutig.*

Satz 5.5
Seien $x_0, \ldots, x_n \in [a,b]$ und $\omega \in L^1(a,b)$ gegeben mit den Symmetrieeigenschaften

(i) $x_j - a = b - x_{n-j}$ $(0 \leq j \leq n)$ (Symmetrie bzgl. $\frac{a+b}{2}$)

(ii) $\omega(x) = \omega(a+b-x)$ $(x \in [a,b])$ (gerade Funktion bzgl. $\frac{a+b}{2}$)

Dann gilt $\omega_{n-j} = \omega_j$ $(0 \leq j \leq n)$, d.h. die Interpolationsquadratur ist symmetrisch.

Falls n gerade, so ist I_n exakt auf $\mathbb{P}_{n+1}$

Beweis:

(a) Sei $\tilde{I}_n(f) := \sum_{j=0}^{n} \omega_{n-j} f(x_j)$. Dann gilt $\tilde{I}_n(p) = I_n(p)\ \forall\ p \in P_n$. Damit ist aber $\tilde{I}_n$ exakt auf P_n und nach Satz 5.3 gilt $\tilde{I}_n = I_n \implies \omega_{n-j} = \omega_j$.

(b) Sei nun $n = 2m$ und damit $x_m = \frac{a+b}{2}$ wegen (a). Sei $p_n \in \mathbb{P}_n$ das Interpolationspolynom zu $(x_0, f(x_0)), \ldots, (x_n, f(x_n))$ und sei $q_{n+1} \in \mathbb{P}_{n+1}$ das Hermite Interpolationspolynom zu $(x_0, f(x_0)), \ldots, (x_{m-1}, f(x_{m-1})), (x_m, f(x_m)), (x_m, f'(x_m)), (x_{m+1}, f(x_{m+1})), \ldots, (x_n, f(x_n))$. Mit

$$a := \frac{f'(x_m) - p_n'(x_m)}{\prod\limits_{\substack{l=0 \\ l \neq m}}^{n} (x_m - x_l)}$$

und $N(x) = \prod_{l=0}^{n} (x - x_l) \in \mathbb{P}_{n+1}$. Setze $\tilde{q}_{n+1}(x) - p_n(x) + aN(x)$. Es ist $\tilde{q}_{n+1} \in \mathbb{P}_{n+1}$ und $\tilde{q}_{n+1}(x_l) = p_n(x_l) + aN(x_l) = f(x_l) + 0$ und $\tilde{q}_{n+1}'(x_m) = p_n'(x_m) + a \prod\limits_{\substack{l=0 \\ l \neq m}}^{n} (x_m - x_l) = f'(x_m) \implies \tilde{q}_{n+1}$ erfüllt dieselben Interpolationsbedingungen q_{n+1} und da die Hermite Interpolation eindeutig ist, gilt $q_{n+1} = \tilde{q}_{n+1}$.

Es gilt wegen (a): $N(x) = \prod_{l=0}^{n} (x - x_l) \left(x - \frac{a+b}{2}\right) \prod_{l=0}^{n} (x - (a+b-x_l))$ und $N(a+b-x) = (-1)^{n+1} N(x) = -N(x)$

Wegen (b) gilt:

$$\begin{aligned} \int_a^b \omega(x)N(x)\mathrm{d}x &= \int_a^{x_m} \omega(x)N(x)\mathrm{d}x + \int_{x_m}^b \omega(x)N(x)\mathrm{d}x \\ &\overset{t=a+b-x}{=} \int_a^{x_m} \omega(x)N(x)\mathrm{d}x - \int_{x_m}^b \omega(a+b-t)N(a+b-t)\mathrm{d}t \\ &= \int_a^{x_m} \omega(x)N(x)\mathrm{d}x + \int_a^{x_m} \omega(t)(-N(t))\mathrm{d}t = 0 \end{aligned}$$

Daher gilt: $\int_a^b q_{n+1}(x)\omega(x)\mathrm{d}x = \int_a^b p_n(x)\omega(x)\mathrm{d}x = I(p_n) = I(f)$, da p_n Polynominterpolation zu f.

Sei nun $f \in \mathbb{P}_{n+1} \implies f = q_{n+1}$ und daher $I(f) = I(q_{n+1}) = I(p_n) = I_n(f)\ \forall f \in \mathbb{P}_{n+1}$ □

Satz 5.6 (Fehlerabschätzung)
Sei I_n eine Interpolationsquadratur (I.Q.) auf $\mathbb{P}_n(a,b)$ mit Gewichtsfunktion $\omega \equiv 1$. $R_n(f) := I_n(f) - I(f)$ sei das zugehörige Fehlerfunktional. Dann gilt:

(i) $|R_n(f)| \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}(b-a)^{n+2}$ für alle $f \in C^{n+1}(a,b)$, falls n ungerade ist.

(ii) $|R_n(f)| \leq \frac{\|f^{(n+2)}\|_\infty}{(n+2)!}(b-a)^{n+3}$ für alle $f \in C^{n+2}(a,b)$. falls n gerade ist.

Beweis:

(i) Es ist $I_n(f) = I(p_n)$, wobei $p_n \in \mathbb{P}_n$ das Interpolationspolynom zu den Daten $(x_i, f(x_i))\ i = 0, \ldots, n$ ist.

$$\begin{aligned} \implies |R_n(f)| &= \left|\int_a^b (f - p_n)\right| \leq \int_a^b |f(x) - p_n(x)|\,\mathrm{d}x \\ &\overset{\text{Satz 4.4}}{=} \int_a^b \left|\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\prod_{k=0}^{n}(x - x_k)\right|\mathrm{d}x \\ &\leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}(b-a)^{n+2} \end{aligned}$$

(ii) Aus dem Beweis vom Satz 5.5 folgt: $I_n(f) = I(q_{n+1})$. Dann folgt die Behauptung mit Satz 4.18b (Siehe Seite 75) □

Bemerkung:

1. Die Abschätzungen lassen sich leicht verallgemeinern auf den Fall $\omega \in L^1(a,b)$.
2. Die Abschätzung $\left|\prod_{k=0}^{n}(x - x_n)\right| \leq (b-a)^{n+1}$ kann für gegebene $x_0, \ldots, x_n$ deutlich verbessert werden zu $\left|\prod_{K=0}^{n}(x - x_n)\right| \leq K(b-a)^{n+1}$ mit $K \ll 1$

Satz 5.7 (Koordinatentransformation)
Sei $\hat{I}_n(\hat{f}) = \sum_{k=0}^{n} \hat{\omega}_k \hat{f}(t_k)$ mit $t_k \in [-1,1]$ eine I.Q. auf dem „Einheitsintervall“ $[-1,1]$

Dann wird durch $I_n(f) := \sum_{k=0}^{n} \omega_n f(x_k)$ mit $\omega_k = \frac{b-a}{2}\hat{\omega}_k,\ x_k = \frac{b-a}{2}t_k + \frac{b+a}{2}$ eine I.Q. auf dem Intervall $[a,b]$ definiert.

Gilt für das Fehlerfunktional $\hat{R}_n$ zu $\hat{I}_n$ die Abschätzung $|R_n(f)| \leq K\left\|\hat{f}^{(m)}\right\|_\infty 2^{m+1}$, so gilt für R_n zu I.Q. I_n: $|R_n(f)| = K\left\|f^{(m)}\right\|_\infty (b-a)^{m+1}$

Beweis: Sei $p_n \in \mathbb{P}_n$ und $\hat{p}(t) = p(x(t))$ mit $x(t) := \frac{b-a}{2}t + \frac{b+a}{2}$.

Da $x(t)$ linear ist, gilt: $\hat{p} \in \mathbb{P}_n$ und

$$\begin{aligned} I(p) = \int_a^b p(x)\mathrm{d}x &= \int_{-1}^{1} p(x(t))x'(t)\mathrm{d}t \\ &= \tfrac{(b-a)^2}{2}\int \hat{p}^{-1}(t)\mathrm{d}t = \tfrac{b-a}{2} I_m(\hat{p}) \\ &= \sum_{k=0}^{n} \tfrac{b-a}{2}\hat{\omega}_k - \hat{p}(t_k) \\ &= \sum_{k=0}^{n} \omega_n p(x_n) = I_n(p) \end{aligned}$$

Daher ist I_n exakt auf $\mathbb{P}_n$ und $I_n(f) = \frac{b-a}{2}\hat{I}_n(\hat{f})$ mit $\hat{f}(t) = f(x(t))$

Es ist dann $\hat{f}'(t) = x'(t)f'(x(t))$

$$\Longrightarrow \quad \hat{f}^{(m)}(t) = \left(\tfrac{b-a}{2}\right)^m f^{(m)}(x(t))$$

$$\Longrightarrow \quad \left\|\hat{f}^{(m)}\right\|_\infty = 2^{-m}(b-a)^m \left\|f^{(m)}(x(t))\right\|_\infty$$

$$\begin{aligned} \Longrightarrow \quad |R_n(f)| &= \left|\int_a^b f(x)\mathrm{d}x - I_n(f)\right| = \left|\tfrac{b-a}{2}\left[\int_{-1}^{1} f(t)\mathrm{d}t - \hat{I}_n(\hat{f})\right]\right| \\ &= \tfrac{b-a}{2}\left|\hat{R}_n(\hat{f})\right| \le \tfrac{b-a}{2}\left\|\hat{f}^{(n)}\right\|_\infty K 2^{m+1} \\ &= (b-a)^{m+1}\left\|f^{(m)}\right\|_\infty K \qquad \square \end{aligned}$$

Bemerkung:

1. Es reicht also aus I.Q.en auf $[-1,1]$ zu konstruieren. Zu $-1 \le t_0 < \ldots < t_n \le 1$ wird durch

$$\hat{\omega}_j := \prod_{\substack{k=0 \\ k\neq j}} \frac{t - t_k}{t_j - t_k}\mathrm{d}t$$

   die I.Q. auf $[-1,1]$ zu $\mathbb{P}_n$ definiert. Mit $(\omega_j, x_j)_{j=0}^n$ wie im Satz 5.7 wird dann die I.Q. auf $[a,b]$ definiert.

2. Da für jede I.Q. $I_n(1) = (b-a)$ gilt, muss $\sum_{k=0}^{n} \omega_k = (b-a)$ gelten.

3. Wie bei der Polynom Interpolation für große Werte von n auf, wie z.B. negative Gewichte (vgl. Übungsaufgaben). Daher geht man dazu über, Quadraturen auf Teilintervallen aufzusummieren:

$$\int_a^b f(x)\mathrm{d}x = \sum_{i=1}^{n}\int_{a_{i-1}}^{a_i} f(x)\mathrm{d}x \quad a = a_0 < \ldots < a_n = b$$

Satz 5.8 (Zusammengesetze Quadraturen)
Sei $\hat{I}_n(\hat{f}) = \sum_{k=0}^{n} \hat{\omega}_k \hat{f}(t_n)$ eine I.Q. auf $[-1,1]$ mit $\left|\hat{R}_n(\hat{f})\right| \le K\left\|\hat{f}^{(m)}\right\|_\infty 2^{m+1}$ zu $a < b$, $N \in \mathbb{N}$. Setze $a_l := a + lH$, $l = 0, \ldots, N$ mit $H := \frac{b-a}{2}$.

Dann ist $I_n(f) := \frac{H}{2}\sum_{l=1}^{N}\sum_{k=0}^{N} \hat{\omega}_k f\left(\frac{H}{2}(t_k - 1) + a + lh\right)$ eine Quadraturformel mit der Abschätzung
$|R_n(f)| := |I_n(f) - I(f)| \le K\left\|f^{(m)}\right\|_\infty (b-a)H^m$

Beweis: Wir wenden Satz 5.7 auf $[a_{l-1}, a_l]$ an:

$$\begin{aligned} \Longrightarrow \quad I_n^l(f) &:= \sum_{k=0}^{n} \tfrac{a_l - a_{l-1}}{2} \hat{\omega}_k f\left(\tfrac{a_l - a_{l-1}}{2} t_k + \tfrac{a_l + a_{l-1}}{2}\right) \\ &= \tfrac{H}{2} \sum_{k=0}^{n} \hat{w}_k f\left(\tfrac{1}{2} H(t_k - 1) + a + lH\right) \end{aligned}$$

Also gilt: $I_n(f) = \sum_{l=1}^{N} I_n^l(f)$ und es folgt:

$$\begin{aligned} |R_n(f)| &\leq \sum_{l=1}^{N} \left|R_n^l(f)\right| \overset{\text{Satz 5.7}}{\leq} K \left\|f^{(m)}\right\|_\infty \sum_{l=1}^{N} \underbrace{(a_l - a_{l-1})}_{:=}^{m+1} \\ &= K \left\|f^{(m)}\right\|_\infty \underbrace{NH}_{=b-a} H^m \quad \square \end{aligned}$$

5.1 Newton-Cotes Formel

- Die Newton-Cotes Formel sind eine mit **äquidistanten**[1] Stützstellen I.Q.; $x_k = a + kh,\ h = \frac{b-a}{n}$
- Als offene Newton-Cotes Formel bezeichnet man I.Q. zu den äquidistanten Stützstellen $x_k = a + (k+1)h,\ h = \frac{b-a}{2}$ (Nur innere Stützstellen), d.h. a, b sind keine Stützstellen.

1. $n = 1$ (Trapezregel)

$$x_0 = a,\ x_1 = b,\ \omega_0 = \int_a^b \tfrac{x-b}{a-b} \mathrm{d}x = \tfrac{b-a}{2},\ \omega_1 = \tfrac{b-a}{2}$$
$$T(f) = I_1(f) = \tfrac{b-a}{2}\left(f(a) + f(b)\right) \text{ (Siehe Abb. 5.1)}$$
$$\begin{aligned} |R_n(f)| &\leq \tfrac{\|f''\|_\infty}{2} \int_a^b |x-a|\,|x-b| \\ &= \tfrac{\|f''\|_\infty}{2} \tfrac{(b-a)^3}{6} = \tfrac{\|f''\|_\infty}{12}(b-a)^3 \end{aligned}$$

2. $n = 2$ (Simpson-Regel)

$$x_0 = a,\ x_1 = \tfrac{a+b}{2},\ x_2 = b,\ \omega_0 = \omega_2 = \tfrac{b-a}{2},\ \omega_1 = \tfrac{2(b-a)}{3}$$
$$S(f) = I_2(f) = \tfrac{b-a}{6}\left(f(a) + 4f\left(\tfrac{a+b}{2}\right) + f(b)\right),\ |R_5(f)| \leq \tfrac{\|f^{(4)}\|_\infty}{2880}(b-a)^5$$

Zusammengesetze Newton-Cotes Formel

1. **Zusammengesetze Trapezregel** (Satz 5.8, $N = n,\ H = h$)

$$T_h(f) = \tfrac{h}{2} \sum_{l=1}^{n} \left[f(a + lh - h) + f(a + lh)\right] = \tfrac{1}{2}\left(f(a) + 2\sum_{l=1}^{n-1} f(a + lh) + f(b)\right)$$
$$|R_n(f)| \leq \tfrac{\|f''\|_\infty}{12}(b-a)h^2$$

[1] Der Abstand zwischen den Stützstellen ist gleich.

2. **Zusammengesetze Simpson-Regel** (Ssatz 5.8, $N = \frac{n}{2},\ H = 2h$)

$$S_h(f) = \frac{h}{3}\left(f(a) + 2\sum_{l=1}^{N-1} f(x_{2l}) + 4\sum_{l=1}^{N} f(x_{2l-1}) + f(b)\right)$$
$$|R_n(f)| \leq \frac{\|f^{(4)}\|_\infty}{180}(b-a)h^4$$

Bemerkung: Bei den Newton-Cotes Formeln bleiben die Gewichte bis $n = 6$ positiv. Dei den offenen Newton-Cotes Formeln nur bis $n = 2$

5.2 Gauß-Quadraturen

Idee: Wir suchen eine Quadratur Q_n, welche für $\mathbb{P}_n$ mit möglichst großem m exakt ist. Dies ist **nicht** möglich für $m = 2n + 2$ (vgl ÜA). Aber für $m = 2n + 1$ wird dies mit der Gauß-Quadrature (G.Q.) erreicht.

Definition 5.9 (Gauß-Quadraturen)
Sei $\omega \in L^1(a,b)$ gegeben. Eine Quadraturformel $Q_n : C([a,b]) \longrightarrow \mathbb{R},\ Q_n(f) := \sum_{k=0}^{n} \omega_k f(x_k)$ heißt **Gauß-Quadratur**, *falls Q_n auf $\mathbb{P}_{2n+1}$ exakt ist.*

Satz 5.10
Sei $\omega \in L^1(a,b)$ und eine Quadratur $Q_n(f) := \sum_{k=0}^{n} \omega_k f(x_k)$ gegeben. Setze $p_{n+1}(x) = \prod_{k=0}^{n}(x - x_k)$. Dann sind äquivalent:

(i) Q_n ist Gauß-Quadratur.

(ii) Q_n ist Interpolationsquadratur und $\int_a^b \omega(x)p_{n+1}(x)q(x)\mathrm{d}x = 0\ \forall\ q \in \mathbb{P}_n$.

Beweis: „(i) $\Longrightarrow$ (ii)“

Sei $q \in \mathbb{P}_n$. Dann ist $\int_a^b \omega(x)p_{n+1}(x)q(x)\mathrm{d}x = Q_n(p_{n+1}q) = \sum_{k=0}^{n} \omega_k \underbrace{p_{n+1}(x_k)}_{=0 \text{ nach Def.}} q(x) = 0$

„(ii) $\Longrightarrow$ (i)“

Sei $p \in \mathbb{P}_{2n+1}$. Mit Polynomdivision gilt: $p = qp_{n+1} + r$ mit $q, r \in P_n$

$$\begin{aligned}
\Longrightarrow \int_a^b \omega(x)p(x)\mathrm{d}x &= \int_a^b \omega(x)\Big(\underbrace{q(x)p_{n+1}(x)}_{=0} + r(x)\Big)\mathrm{d}x \\
&\overset{\text{Vor. (ii)}}{=} 0 + \int_a^b \omega(x)r(x)\mathrm{d}x \\
&= 0 + Q_n(r) \\
&= Q_n(p_{n+1}q) + Q_n(r) \\
&= Q_(p_{n+1}q + r) = Q_n(p) \qquad \square
\end{aligned}$$

Definition 5.11

(i) Eine Funktion $\omega \in L^1(a,b)$ heißt **zulässige Gewichtsfunktion** *, falls gilt $\omega \geq 0$, also $\int_a^b w(x)\mathrm{d}x > 0$ (fast überall > 0).*

(ii) Ist ω eine zulässige Gewichtsfunktion, so wird durch

$$\langle p, q \rangle_\omega := \int_a^b \omega(x)p(x)q(x)\mathrm{d}x$$

ein Skalarprodukt auf $\mathbb{P}_n$ definiert. Die Wohldefiniertheit des Skalarprodukts kann durch Nachrechnen geprüft werden.

Satz 5.12
Sei ω eine zulässige Gewichtsfunktion. Dann liefert die durch das Gram-Schmidtsche Orthogonalisierungsverfahren[2] definierte Folge $(p_n)_{n\in\mathbb{N}}$

$$p_{n+1}(x) = x^{n+1} - \sum_{i=0}^{n} \frac{\langle x^{n+1}, p_i \rangle_\omega}{\langle p_i, p_i \rangle_\omega} p_i(x),$$

wobei $p_0 \equiv 1$ ist, das eindeutig bestimmte Polynom $p \in \mathbb{P}_{n+1}$ der Form

$$(*)\ p(x) = \prod_{k=0}^{n}(x - x_k) \quad x_k \in \mathbb{C},\ 0 \leq k \leq n$$

mit

$$(**)\ \langle p, q \rangle_\omega = 0 \quad \forall\ q \in \mathbb{P}_n$$

Außerdem ist $\{p_0, p_1, \ldots, p_{n+1}\}$ eine **Orthogonalbasis**[3] von P_{n+1} bezüglich $\langle \cdot, \cdot \rangle_\omega$

Beweis: (Induktion über n)

$n = 0$: klar.

$n-1 \longrightarrow n$: Sei $\{p_0, \ldots, p_n\}$ eine Orthogonalbasis von P_n. Setze

$$P_n^\perp := \left\{ p \in \mathbb{P}_{n+1} \;\middle|\; \langle p, q \rangle_\omega = 0 \quad \forall\ q \in \mathbb{P}_n \right\}$$

$\Longrightarrow \quad \dim(P_n^\perp) = 1$

Da $(*)$ verlangt, dass der Koeffizient vor x^{n+1} gleich 1 ist, gibt es genau ein $p \in \mathbb{P}_{n+1}$, welches $(*)$ und $(**)$ erfüllt. Nach Konstruktion ist $p = p_{n+1}$, da $\langle p_{n+1}, p_n \rangle_\omega = 0 \quad \forall\ 0 \leq k \leq n$.
Also folgt $\langle p_{n+1}, q \rangle_\omega = 0 \quad \forall\ q \in \mathbb{P}_n$ $\square$

[2] Das Verfahen ist leicht geändert, weil die Vektoren nicht normiert werden.
[3] Hierbei handelt es sich um keine normierte Basis.

Satz 5.13
Sei ω eine zulässige Gewichtsfunktion, dann gilt: die Nullstellen $x_0, \ldots, x_n$ von p_{n+1} aus Satz 5.12 sind reell, einfach und liegen im Intervall (a, b)

Beweis: Wir setzen $q(x) = 1,\ k = -1$, falls es keine reelle Nullstelle ungerader Vielfachheit von p_{n+1} in (a, b) gibt.

$q(x) = \prod_{j=0}^{n} (x - x_j)$ andernfalls, wobei x_j die Nullstellen sind für $0 \leq j \leq k$.

Zu zeigen: $k = n$

Annahme: $k < n$: Nach Definition hat $p := p_{n+1}q$ kein Vorzeichenwechsel in (a, b). Da $k < n$, folgt für $q \in \mathbb{P}_n : \langle p_{n+1}, q \rangle_\omega = 0 \implies \omega p_{n+1} q = 0$ (fast überall) und $\int_a^b \omega(x)\mathrm{d}x = 0$ und das ist ein Widerspruch! □

Satz 5.14
Sei ω eine zulässige Gewichtsfunktion. Dann gibt es genau eine G.Q. Q_n für ω, nämlich die, deren Stützstellen $x_0, \ldots, x_n$ die Nullstellen von p_{n+1} aus Satz 5.12 sind und deren Gewichte definiert sind durch

$$\omega_j := \int_a^b \omega(x) L_j(x) \mathrm{d}x$$

mit

$$L_j(x) := \prod_{\substack{k=0 \\ k \neq j}} \frac{(x - x_k)}{(x_j - x_k)}$$

Es gilt $\omega_j > 0 \quad \forall\ j$.

Beweis: Folgt aus den Sätzen 5.10, 5.12, 5.13 und aus dem Satz 5.3

Noch zu zeigen: $\omega_j > 0 \quad \forall\ j$: Da $L_j^2 \in \mathbb{P}_{2n}$ ist, folgt:

$$0 < \int_a^b \omega(x) L_j^2(x) \mathrm{d}x = Q_n(L_j^2) = \sum_{k=0}^{n} \omega_k L_j^2(x_k) = \omega_j \quad \square$$

Satz 5.15 (Deutung der Gauß-Quadraturen als Interpolationsquadraturen)
Sei p das eindeutige bestimmte Polynom in $\mathbb{P}_{2n+1}$ mit den Eigenschaften $p(x_i) = f(x_i),\ p'(x_i) = f'(x_i)$ für $i = 0, \ldots, N$ und x_i Nullstellen von p_{n+1} sind. Dann gilt: $Q_n(f) = Q_n(p) = I(p)$

Beweis: Siehe Übungsaufgaben

Folgerung 5.16
Für $f \in C^{2n+2}(a, b)$ gibt es ein $\xi \in (a, b)$ mit $I(f) - Q_n(f) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \langle p_{n+1}, p_{n+1} \rangle_\omega$

Beweis: Siehe Übungsaufgaben

Bemerkung: Die G.Q.en sind für stetige Funktionen auf kompakten Intervallen konvergent bei Graderhöhung, d.h. $|I(f) - Q_n(f)| \overset{k}{\longrightarrow}^{\infty} 0$

Beispiel 5.17

1. $\omega(x) = 1$, $[-1, 1]$
   Es gilt $p_n(x) = \frac{(2n)!}{2^n (n!)^2} P_n(x)$, wobei $P_n(x)$ die **Legendre-Polynome** sind mit $P_0(x) = 1,\ P_1(x) = x$, $P_{n+1}(x) = \frac{2n+1}{n+1} P_n(x) - \frac{n}{n+1} P_{n-1}(x)$

   Es gilt: $I(f) - q_n(f) = 2^{2n} \frac{n+1}{2n+2} \frac{(n!)^4}{((2n+1)!)^3} f^{(2n+2)}(\xi)$

   Für $n = 1$: $Q(f) = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)$ („2-Punkt-Gauß-Quadratur“)

   $n = 2:\ Q_2(f) = \frac{1}{9}\left(5f\left(-\sqrt{\frac{3}{5}}\right) + 8f(0) + 5f\left(\sqrt{\frac{3}{5}}\right)\right)$

   Die G.Q. auf $[a, b]$ erhält man durch Koordinatentransformation (vgl. 5.7)

2. **Gauß-Tschebyscheff-Quadraturen**

   $\omega(x) = \sqrt{1 - x^2}^{-1},\ [-1, 1]$
   $p_n(x) = \frac{1}{2^{n-1}} T_k(x)$ und T_k die Tschebyscheff-Polyome 1. Art sind.

   $T_0(x) = 1,\ T_1(x) = x,\ T_{n+1}(x) = 2xT_n(x) - T_{n+1}(x),\ n > 1$
   $\Longrightarrow\ T_n(x) = \cos(n \cdots \arccos(x))$

   Nullstellen von $p_{n+1}:\ x_j^{(n)} = \cos\left(\frac{2j+1}{2n+1}\pi\right) \quad j = 0, \ldots, n$

   Gewichte: $\omega_j^{(n)} = \frac{\pi}{n+1}$

   Fehler: $I(f) - Q_n(f) = \frac{\pi}{2^{2n+1}(2n+2)!} f^{(2n+2)}(\xi)$

3. **Laguerre-Quadratur**

   $\omega(x) = e^{-x},\ [0, \infty)$
   $p_n(x) = (-1)^n L_n(x)$ und L_n Lagrange-Polynome mit

   $L_0(x) = 1,\ L_1(x) = 1 - x,\ L_{n+1}(x) = (1 + 2n - x)L_n(x) - n^2 L_{n-1}(x)$

   Fehler: $I(f) - Q_n(f) = \frac{n+1}{2} \frac{(n!)^2}{(2n+1)!} f^{(2n+2)}(\xi)$

4. **Hermite-Quadraturen**

   $\omega(x) = e^{-x^2},\ (-\infty, \infty)$
   $p_n(x) = 2^n H_n(x)$ und H_n die Hermite Polynome mit

   $H_0(x) = 1,\ H_1(x) = 2x,\ H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x)$

   Fehler: $I(f) - Q_n(f) = \frac{\sqrt{\pi} n!}{2^{n+1}(2n+1)!} f^{(2n+2)}(\xi)$

5. $[-1,1],\ \alpha,\beta > -1,\ \omega(x) = (1-x)^\alpha(1+x)^\beta$ („Jacobi-Polynome“) definiert

$$J_n(x,\alpha,\beta) := \frac{1}{2^n n!\omega(x)} \frac{d^k}{dx^n}\left((x^2-1)^n\omega(x)\right)$$

Definition 5.18 (Zusammengesetze Gauß-Quadraturen)
Idee: *Zerlege $[a,b]$ in Teilintervalle und wende dort die G.Q. an.*

Zusammengesetze 2-Punkt G.Q. $(n=1)$. Setze $h = \frac{b-a}{N},\ a_j = a + jh$ für $j = 0,\dots,N$

Dann ist die zusammengesetze 2-Punkt G.Q. gegeben durch:

$$Q_n(f) = \frac{h}{2}\sum_{j=0}^{N-1}\left(f(a_j+h') + f(a_{j+1}-h')\right)$$

mit $h' = \frac{h}{2}\left(1 - \frac{1}{\sqrt{3}}\right)$

5.3 Romberg Verfahren

Idee: Die Richardson Extrapolation auf eine zusammengesetze Quadraturformel anwenden. Besonders geeignet ist die zusammengesetze Trappezregel $T_h(f)$, da sie eine asympotische Entwicklung in h^2 erlaubt, d.h. $q = 2$ in Satz 4.23

Definition 5.19 (Bernoulli Polynome/Zahlen)
Die durch $B_0(t) = 1$ *und* $\frac{\partial}{\partial x}B_k(t) = B_{k-1}(t),\ \int_0^1 B_k(t)\mathrm{d}t = 0$ *definierten Polynome heißen* **Bernoulli Polynome**

$$B_0(t) = 1, \quad B_1(t) = t - \frac{1}{2}, \quad B_2(t) = \frac{1}{2}t^2 - \frac{1}{2}t + \frac{1}{12}, \quad \dots$$

Die **Bernoulli Zahlen** *sind gegeben durch*

$$B_k := k! \cdot B_k(0)$$

Lemma 5.20

(i) $B_k(0) = B_k(1)$ für $k \geq 2$

(ii) $B_k(t) = (-1)^k B_k(1-t)$ für $k \geq 0$

(iii) $B_{2k+1}(0) = B_{2k+1}\left(\frac{1}{2}\right) = B_{2k+1}(1) = 0$ für $k \geq 1$

Satz 5.21 (Euler-MacLaurin'sche Summenformel)
Sei $f \in C^{2m}(a,b),\ m \in \mathbb{N}$ und $h := \frac{b-a}{n},\ n \in \mathbb{N}$. Dann gilt:

$$T_h(f) = \int_a^b f(x)\mathrm{d}x - \sum_{k=1}^{m-1} h^{2k} \frac{B_{2k}}{(2k)!} \left(f^{(2k-1)}(b) - f^{(2k-1)}(a) \right) + O(h^{2m})$$

Bemerkung: Diese zeigt die asymptotische Entwicklung. Die Formel zeigt, dass die Trapezregel auch ohne Extrapolation gut geeignet für periodische Funktionen ist.

Beweis: Sei $\varphi \in C^{2m}(0,1)$ beliebig. Dann gilt mit $B_1' = B_0,\ B_1(0) = \frac{1}{2},\ B_1(0) = -\frac{1}{2}$:

$$\begin{aligned}
\int_0^1 \varphi(t)\mathrm{d}t &= \int_0^1 B_0(t)\varphi(t)\mathrm{d}t \\
&= [B_1(t)\varphi(t)]_{t=0}^1 - \int_0^1 B_1(t)\varphi'(t)\mathrm{d}t \\
&= \tfrac{1}{2}\left(\varphi(1) + \varphi(0)\right) - [B_2(t)\varphi'(t)]_{t=0}^1 + \int_0^1 B_2(t)\varphi''(t)\mathrm{d}t \\
&\overset{5.19.i}{=} \tfrac{1}{2}\left(\varphi(1) + \varphi(0)\right) - B_2(0)\left(\varphi'(1) - \varphi'(0)\right) + \underbrace{[B_3(t)\varphi''(t)]_{t=0}^1}_{=0 \ 5.19.iii} - \int_0^1 B_3(t)\varphi'''(t)\mathrm{d}t \\
&= \cdots \\
&= \tfrac{1}{2}\left(\varphi(1) - \varphi(0)\right) - \sum_{k=1}^{m-1} B_{2k}(0)\left(\varphi^{(2k-1)}(1) - \varphi^{(2k-1)}(0)\right) + \int_0^1 B_{2m}(t)\varphi^{(2m)}(t)\mathrm{d}t
\end{aligned}$$

Setze $\varphi_j(t) := hf(x_{j-1}, th), \quad 1 \le j \le n$, dann gilt:

- $\int_0^1 \varphi_j(t)\mathrm{d}t = \int_{x_{j-1}}^{x_j} f(x)\mathrm{d}x$
- $\varphi_j^{(2k-1)}(t) = h^{2k} f^{(2k-1)}(x_{j-1}, th)$
- $\varphi_j(1) = f(x_j) = \varphi_{j+1}(0)$
- $\varphi_j^{(2k-1)}(1) = \varphi_{j+1}^{(2k-1)}(0)$

Daher gilt:

$$\begin{aligned}
\int_a^b f(x)\mathrm{d}x &= \sum_{j=1}^n \int_{x_{j-1}}^{x_j} f(x)\mathrm{d}x = \sum_{j=1}^n \int_0^1 \varphi_j(t)\mathrm{d}t \\
&= \sum_{j=1}^n \tfrac{1}{2}\left(\varphi_j(0) + \varphi_j(1)\right) - \sum_{j=1}^n \sum_{k=1}^m B_{2k}(0)\left(\varphi_j^{(2k-1)}(1) - \varphi_j^{(2k-1)}(0)\right) + \sum_{j=1}^n \int_0^1 B_{2m}(t)\varphi_j^{(2m)}(t)\mathrm{d}t \\
&= \sum_{j=1}^n \tfrac{h}{2}\left(f(x_j) + f(x_{j-1})\right) - \sum_{k=1}^{m-1} B_{2k}(0)\left(\varphi_k^{(2k-1)}(1) - \varphi_k^{(2k-1)}(0)\right) \\
&\quad + \sum_{j=1}^n \int_0^1 B_{2m}(t)h^{2m+1} f^{(2m)}(x_{j-1} + th)\mathrm{d}t \\
&= T_h(f) - \sum_{k=0}^{m-1} B_{2k}(0)\left(f^{(2k-1)}(b) - f^{(2k-1)}(a)\right)h^{2k} + h^{2m}\left[h\sum_{j=1}^m \int_0^1 B_{2m}(t) f^{(2m)}(x_{j-1} + (h))\mathrm{d}t\right]
\end{aligned}$$

Letztes Term ist $O(h^{2m})$, falls $[\cdot]$ konstant ist unabhängig von h

$$\begin{aligned}\left|h\sum_{j=1}^{n}\int_0^1 B_{2m}(t)f^{(2m)}(x_{j-1}(h))\mathrm{d}t\right| &\leq h\sum_{j=1}^{n}\|B_{2m}\|_\infty\|f(2m)\|_\infty\\ &= n\cdot h\cdot\|B_{2m}\|_\infty\cdot\left\|f^{(2m)}\right\|_\infty\\ &= (b-a)\|B_{2m}\|_\infty\cdot\left\|f^{(2m)}\right\|_\infty = \text{konstant}\end{aligned}$$ □

5.4 Fehlerdarstellung nach Peano

Definition 5.22
Ein lineares Funktional $R : C^{k+1}(a,b) \longrightarrow \mathbb{R}$ heißt **zulässig**, *gdw es entweder aus einer Auswertung $f^{(\nu)}(x_0)$, $x_0 \in [a,b]$, $0 \leq \nu \leq k$, aus einem gewichteten Integral $\int_{a_0}^{b_0}\omega(x)f^{(\nu)}(x)\mathrm{d}x$, $a_0, b_0 \in [a,b]$, $0 \leq \nu \leq k$ oder aus einer endlichen Linearkombination solcher Funktionalen besteht.*

Beispiel 5.23
(i) Fehlerfunktionale von Quadraturformeln sind zulässig, $R(f) = I(f) - I_n(f)$

(ii) Fehlerfunktional von finite-differnzen Approximationen: $R(f) = \frac{f(b)-f(a)}{b-a} - f'(x_0)$ für $x_o \in [a,b]$

Bemerkung: Im folgenden tauchen Funktionen $K \in C^{k+1}\left((a,b)^2\right)$ auf; für $t \in [a,b]$ ist $K(t,\cdot) \in C^{k+1}(a,b)$ und $\nu(t) := R(K(,\cdot))$ ist eine Abbildung von $[a,b] \longrightarrow \mathbb{R}$. Analog wird durch
$w(x) := \int_a^b K(t,x)u(t)\mathrm{d}t \in C^{k+1}(a,b)$ und $R\left(\int_a^b K(t,\cdot)u(t)\mathrm{d}t\right) := R(w) \in \mathbb{R}$

Lemma 5.24
Für ein nach Definition 5.21 zulässiges Funktional gilt die Vertauschungsregel
$R\left(\int_a^b K(b,\cdot)u(t)\mathrm{d}t\right) = \int_a^b R(K(t,\cdot))u(t)\mathrm{d}t$ für alle $K \in C^{k+1}\left((a,b)^2\right)$, $u \in C^0(a,b)$

Beweis: $w(x) = \int_a^b K(t,x)u(t)\mathrm{d}t$, $v(t) = R(K(,\cdot))$. Zu zeigen: $R(w) = \int_a^b \nu(t)u(t)\mathrm{d}t$

Wegen der Linearität des Integrals reicht es die 2 Fälle zu untersuchen:

(i) $R(f) = f^{(\nu)}(x_0)$

(ii) $R(f) = \int_{a_0}^{b_0} w(x)f^{(\nu)}\mathrm{d}x$

Zu (i): $R(w) = \int_a^b \partial_x^{(\nu)}K(t,x_0)u(t)\mathrm{d}t = \int_a^b \nu(t)u(t)\mathrm{d}t$

Zu (ii): analog □

Lemma 5.25

Sei $(x-t)_+^l := \begin{cases} (x-t)^l & : \quad x \geq t \\ 0 & : \quad \text{sonst} \end{cases}$

Dann gilt für $f \in C^{k+1}(a,b)$:

$$(x) = (P_k f)(x) + \left(K_k(\cdot, x), f^{(k+1)}\right)$$

mit $(P_k f)(x) = \sum_{j=0}^{k} f^{(j)}(a)\frac{(x-a)^j}{j!} \in \mathbb{P}_k$ und $K_k(t,k) = \frac{1}{k!}(x-t)_+^k$

Beweis: Taylorentwicklung mit Integralrestterm:

$$\begin{aligned} f(x) &= \sum_{j=0}^{k} f^{(j)}(a)\tfrac{(x-a)^j}{j!} + \int_a^x \tfrac{(x-t)^k}{k!} f^{(k+1)}(t)\mathrm{d}t \\ &= (P_k f)(x) + \int_a^x \tfrac{(x-t)^k}{k!} f^{(k+1)}(t)\mathrm{d}t \\ &= (P_k f)(x) + \left(K_k(\cdot, x), f^{(k+1)}\right) \end{aligned}$$

Satz 5.26

Sei R ein nach Definition 5.22 zulässiges Funktional auf $C^{k+1}(a,b)$, welches auf dem Raum der Polynome $\mathbb{P}_k$ identisch verschwindet, d.h. $R(p) = 0 \quad \forall\, p \in \mathbb{P}_k$

Dann gibt es für alle $f \in C^{k+1}(a,b)$

$$R(f) = \int_a^b K(t) f^{(k+1)}(t)\mathrm{d}t$$

mit $k(t) := \frac{1}{k!} R\left(\cdot, -t_+^k\right)$. $k(t)$ heißt **Peano Kern** von R und ist unabhängig von f

Beweis: $R(f) \overset{5.25}{=} R((P_k f) + (K_k(\cdot, x), f^{(k+1)}) \overset{R \text{ linear}}{=} R(P_k f) + R\left(\int_a^b K_k(t, \cdot) f^{(k+1)}(t)\mathrm{d}t\right)$

$\overset{\text{Vor. } 5.24}{=} \int_a^b R(K_k(t,\cdot)) f^{(k+1)}(t)\mathrm{d}t = \int_a^b K(t) f^{(k+1)}(t)\mathrm{d}t \quad \square$

Folgerung 5.27

Die Voraussetzungen für die Folgerung stimmen mit denen, von Satz 5.26 überein.

Hat der Peano Kern $K(t)$ für $t \in [a,b]$ kein Vorzeichenwechsel, so gilt: $\forall\, t \in C^{n+1}(a,b), \quad \exists\, \xi \in [a,b]$ mit

$$R(f) = f^{(n+1)}(\xi)\frac{1}{(n+1)!} R(x^{n+1})$$

Beweis:

$$\begin{aligned} R(f) &= \int_a^b K(t) f^{(n+1)}(t)\mathrm{d}t \\ &\overset{\text{MWS}}{=} f^{(n+1)}(\xi) \int_a^b K(t)\mathrm{d}t \quad \text{da } K \text{ kein VZW hat} \end{aligned}$$

Anwenden auf x^{n+1} ergibt: $R(x^{n+1}) = (n+1)! \int_a^b K(t)\mathrm{d}t$

$$\Longrightarrow \int_a^b K(t)\mathrm{d}t = \tfrac{R(x^{n+1})}{(n+1)!}$$

$$\Longrightarrow R(f) = f^{(n+1)}(\xi) \int_a^b K(t)\mathrm{d}t = f^{(n+1)}(\xi)\tfrac{R(x^{n+1})}{(n+1)!} \quad \square$$

Beispiel 5.28 (Simpsonregel)

$R(f) = \frac{1}{3}f(-1) + \frac{4}{3}f(0) + \frac{1}{3}f(1) - \int_{-1}^{1} f(x)\mathrm{d}x$

$\Longrightarrow R(p) = 0 \quad \forall\, p \in \mathbb{P}_4$, d.h. $k = 3, a = -1, b = 1$ in Satz 5.26

$\Longrightarrow R(f) = \int_{-1}^{1} u(t) f^{(4)}(t)\mathrm{d}t$ mit

$$K(t) = \frac{1}{6}R\left((\cdot - t)_+^3\right) = \frac{1}{18}(-1-t)^3 + \frac{2}{9}(-t)^3 + \frac{1}{18}(1-t)^3 - \frac{1}{6}\int_{-1}^{1}(x-t)^3\mathrm{d}x$$

Für $t \in [-1,1]$ gilt: $(-1-t)_+^3 = 0,\ (1-t)_+^3 = (1-t)^3$

$$(-t)_+^3 = \begin{cases} -t^3 & : \ -1 \le t \le 0 \\ \\ 0 & : \ 0 \le t \le 1 \end{cases},$$

$\int_{-1}^{1}(x-t)_+^3\mathrm{d}x = \int_t^1 (x-t)^3\mathrm{d}x = \frac{1}{4}(1-t)^4$

$$\Longrightarrow K(t) = \begin{cases} \frac{1}{72}(1-t)^3(1+3t) & : \ 0 \le t \le 1 \\ \\ K(-t) & : \ -1 \le t < 0 \end{cases}$$

$\Longrightarrow K(t) \ge 0$ für $t \in [-1,1]$ nach Folgerung 5.27

$$R(f) = f^{(4)}(\xi)\frac{1}{24}R(x^4) = f^{(4)}(\xi)\frac{1}{24}\left(\frac{1}{3}\cdot 1 + \frac{4}{3}\cdot 0 + \frac{1}{3}\cdot 1 - \int_{-1}^{1} x^4\mathrm{d}x\right) = f^{(4)}(\xi)\frac{1}{90}$$

Definition 5.29 (Experimentelle Konvergenzordnung *EOC*)
Sei $f \in C^k(a,b)$ und $I : C^k(a,b) \longrightarrow \mathbb{R}$ ein Funktional, I_h eine Quadraturformel, die I approximiert.

Die experimentelle Konvergenz $EOC\left((e_{H \longrightarrow h}\right)$ *(engl.* ***e****xperimental* ***o****rder of* ***c****onvergence) für den Fehler $e_h := |I(f) - I_h(f)|$ ist definiert durch*

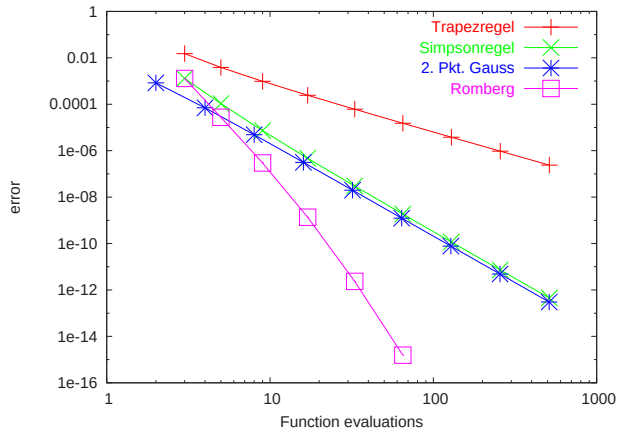
$$EOC\left(e_{H \longrightarrow h}\right) := \frac{\log\left(\frac{e_H}{e_h}\right)}{\log\left(\frac{H}{h}\right)}$$

für ein $H > 0$ vorgegeben.

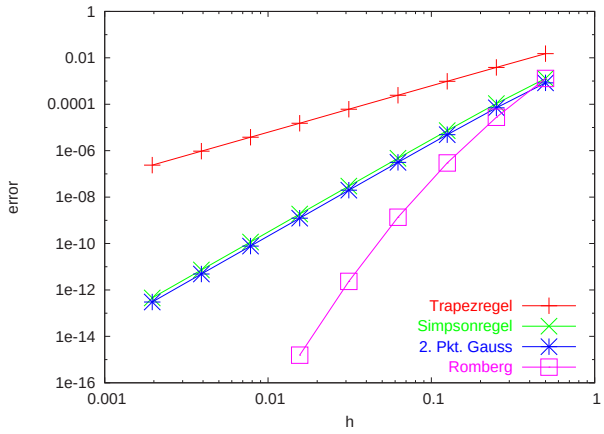
Bemerkung: Für $h \longrightarrow 0$ verhält sich der Fehler wie h^p, wobei p vom angewandten Verfahren abhängt. Mit der EOC hat man die Möglichkeit, p numerisch zu bestimmen.

Beispiel 5.30 (Fehler der Approximierung der Integration)

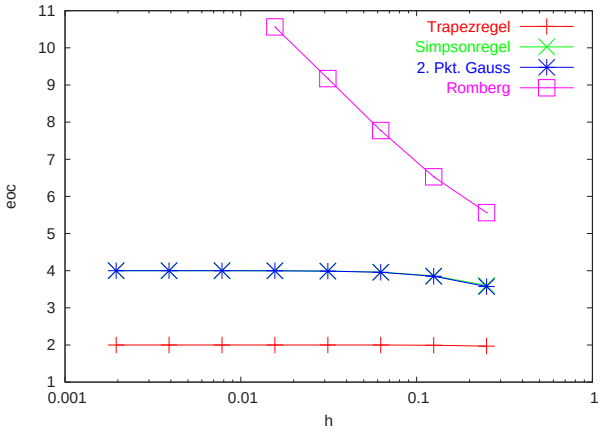
Gegeben seien $I = [0, 1]$ und $f(x) := \frac{1}{x+1}$, $g(x) := \frac{3}{2}\sqrt{x}$. Es gilt $\int_0^1 \frac{1}{x+1} \mathrm{d}x = \ln(2)$, $\int_0^1 \frac{3}{2}\sqrt{x} \mathrm{d}x = 1$. Die Abbildung 5.2 ziegt verschiedene Fehlerverhältnisse von 4 Verfahren: Trapezregel (rot), Simposon-Regel (grün), zwei-Punkt Quadratur (blau) und Romberg Verfahren (lila). ***Typ 1*** ist der Fehler im Vergleich zu h, d.h. zu der Unterteilung bei den zusammengesetzen Quadraturen. ***Typ 2*** ist der Fehler im Vergleich zu der Anzahl der Funktionauswertungen, im Prinzip ein Maß für den Berechnungsaufwand. ***Typ 3*** ist die EOC im Verhältnis zu h.



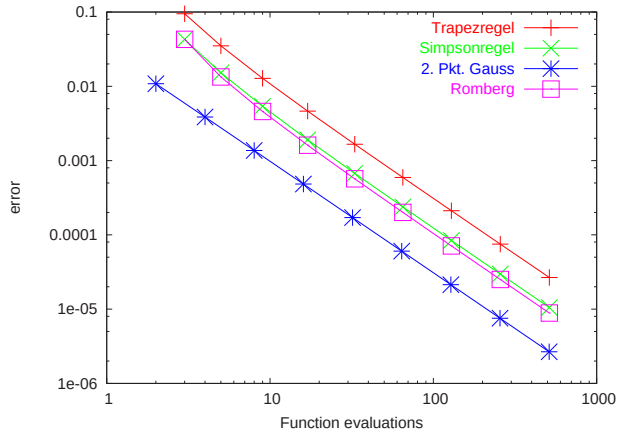
f Typ 1



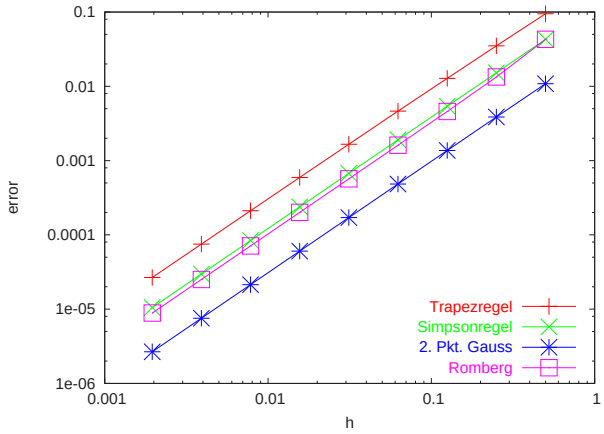
f Typ 2



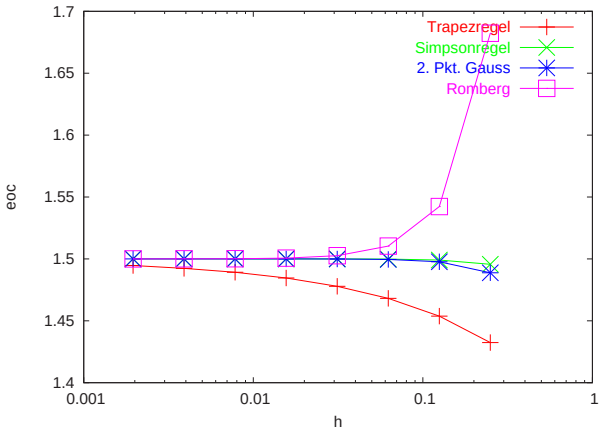
f Typ 3



g Typ 1



g Typ 2



g Typ 3

Abbildung 5.2: Fehler der Quadraturen

Kapitel 6

Lösung gewöhnlicher Differentialgleichungen

Achtung: Dieses Kapitel ist nur ein (kurzer) Einblick in die Lösungen der gewöhnlichen Differentialgleichungen. Viele Sätze und Lemmas werden deshalb nicht gezeigt. Der Aufbau der Theorie plus alle Sätze und Beweise werden erst in der Vorlesung Numerik II gemacht.

6.1 Numerische Verfahren für Anfangswertprobleme

Problem A: Anfangswertproblem (AWP)
Gegeben: $I = [a, b],\ J \subset \mathbb{R}$ zusammenhängende, offende Teilmenge, $S := I \times J,\ f : S \longrightarrow \mathbb{R}$ stetig, $y_0 \in J$

Gesucht: $y \in C^1(I),\ y(I) \subseteq J$ mit $y'(t) = f(t, y(t)),\ t \in I,\ y(a) = y_0$

Beispiel 6.1
Die Existenz und Eindeutigkeit einer Lösung von **A** hängt sehr von den Eigenschaften von f ab!

(i) $f(y) = \alpha y_2,\ \alpha \in \mathbb{R}$, d.h. Problem **A** ist: $y'(t) = \alpha y(t)^2$ mit Anfangswert $y(0) = 1$

Mit $z(t) = \frac{1}{y(t)} \implies z'(t) = -\frac{y'(t)}{y(t)^2} = -\frac{\alpha y(t)^2}{y(t)^2} = -\alpha$

$\implies z(t) = -\alpha t + 1 \implies y(t) = \frac{1}{1-\alpha t}$

Für $\alpha \leq 0 \implies y \in C^1((0, \infty))$, für $\alpha > 0 \implies \lim\limits_{t \longrightarrow \frac{1}{\alpha}} y(t) = \infty$

D.h. für $I = \left[0, \frac{1}{\alpha} + \varepsilon\right],\ \varepsilon > 0$ gilt nicht $y(I) \subseteq J$, d.h ex existiert eine Lösung von **A** lokal in der Zeit.

(ii) $f(y) = \sqrt{y},\ (\implies J = [0, \infty))$. Anfangswert: $y(0) = 0$

Problem **A**: $y'(t) = \sqrt{y(t)},\ y(0) = 0$

Lsg: $y_1(t) = 0$, aber auch $y_2(t) = \frac{1}{4}t^2$ ist auch eine Lösung, d.h. keine Eindeutigkeit

Bedingungen für die Existenz und Eindeutigkeit

(M) $|f(t, y)| \leq M \quad \forall\ (t, y) \in S$

(L) $|f(t, y) - f(t, z)| \leq L\,|y - z| \quad \forall\ (t, s) \in S$

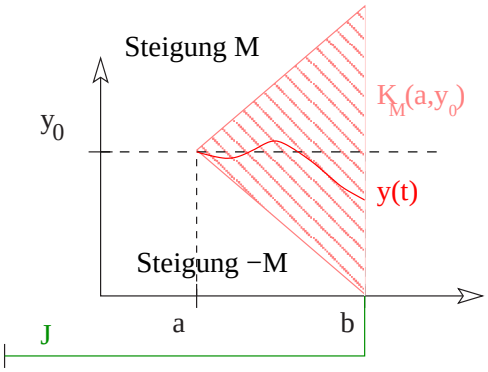


Abbildung 6.1: Definition 6.4

Lemma 6.2
Das Problem **A** kann als Fixpunktgleichung formuliert werden. D.h. y ist Lösung von **A** gdw. $T(y) = y$ mit dem Operator $T : C^0(I) \longrightarrow C^0(I);\ (Ty)(t) = y_0 + \int_a^t f(\tau, y(\tau))\mathrm{d}\tau$

Beweis:

$$T(y) = y \iff y(t) = y_0 + \int_a^b f(\tau, y(\tau))\mathrm{d}\tau \iff y'(t) = f(t, y(t)) \quad \square$$

Satz 6.3 (Picard-Lindelöf, lokale Version)
Erfülle f auf S die Bedingung (L), dann hat **A** lokal eine eindeutige Lösung $\tilde{y}$, d.h. $\exists\ \varepsilon > 0$, so dass auf $I = [a, a + \varepsilon]$ eine Lösung **A** existiert.

Beweis:

$$\begin{aligned} |(Ty)(t) - (Tz)(t)| &\overset{(\mathrm{L})}{\leq} L \int_a^t |y(\tau) - z(\tau)|\,\mathrm{d}\tau \\ &\leq L \max_{\tau \in I} |y(\tau) - z(\tau)|\,(t - a) \\ &\leq L\varepsilon \left\|y - z\right\|_\infty, \text{ für } t \in I_\xi \end{aligned}$$

Wähle $\varepsilon > 0$, sind $L\varepsilon < 1 \implies T$ ist eine Kontraktion $\square$

Definition 6.4

$$K_M(a, y_0) := \left\{ (t, y) \in I \times \mathbb{R} \;\middle|\; |y - y_0| \leq M(t - a) \right\}$$

wobei M die Konstante aus Bedingung (M) ist.

$S := I \times J$ heißt **zulässig** *für das AWP **A** gdw. $K_M(a, y_0) \subset S$*

Abbildung 6.1 zeigt die Menge $K_M(a, y_0)$, y_0 darf sich nur innerhalb dieses „Kegels“ befinden.

Lemma 6.5
Ist $\tilde{y}$ Lösung von **A**, so liegt der Graph von $\tilde{y}$ in $K_M(a, y_0)$ (siehe Abbildung 6.1), d.h.
$(t, \tilde{y}(t)) \in K_M(a, y_0) \quad \forall\, t \in I$

Beweis: $\tilde{y}(t) = y_0 + \int_a^t f(\tau, \tilde{y}(\tau))\mathrm{d}\tau \implies |\tilde{y}(t) - y_0| \leq \int_a^t |f(\tau, \tilde{y}(\tau)|\,\mathrm{d}\tau$

$\implies |\tilde{y}(t) - y_0| \leq M(t-a) \implies (t, \tilde{y}(t)) \in K_M(a, y_0)$ □

Satz 6.6 (Peano)
Das AWP **A** erfülle auf einem zulässigen Rechtseckgebiet $S = [a, b] \times [y_0 - \sigma_1, y_0 + \sigma_2]$ die Bedingung (M). Dann existiert eine Lösung $\tilde{y}$ von **A** auf I

Bemerkung: Die Lösung ist nicht unbedingt eindeutig.

Satz 6.7 (Picard-Lindelöf)
Zustätzlich zu den Voraussetzungen vom Satz 6.3 erfülle f die Bedingung (K). Dann gilt:

(i) **A** hat auf I eine eindeutige bestimmte Lösung $\tilde{y}$

(ii) Die Folge $(y_k)_{k\in\mathbb{N}}$ mit $y_0(t) = y_0,\ y_k := Ty_{n-1}$ konvergiert gleichmäßig gegen $\tilde{y}$ auf I

Satz 6.8 (Statigkeitssatz für das Anfangswertproblem)
f genüge auf S der Bedingung (L), $\tilde{y}$ sei die Lösung von **A**, $\tilde{z}$ sei die Lösung des **gestörten AWP**

$$Z'(t) = f(t, z(t) + \varepsilon(t),\ Z(a) = z_0 \in J$$

Ferner sei $|z_0, y_0| \leq \varepsilon_0$ und $\varepsilon(t)$ sei stetig für $t \in I$ mit $|\varepsilon(t)| \leq \varepsilon_1$. Dann gilt:

$$|\tilde{z}(t) - \tilde{y}(t)| \leq (\varepsilon_0 + \varepsilon_1(t-a))\, e^{L(t-a)}$$

Beweis: Wir benötigen das folgende Lemma:

Lemma 6.9 (Grounwalls Lemma)
Seien $p, q \in C^0(a, b)$ mit $p, q \geq 0$. Erfülle $e : [a, b] \longrightarrow \mathbb{R}$ die Integralungleichungen:

$$0 \leq e(t) \leq p(t) + \int_a^t q(\tau)e(\tau)\mathrm{d}\tau$$

für $t \in [a, b]$, so gilt:

$$e(t) \leq p(t) + \int_a^b q(\tau)p(\tau)\exp\left(\int_a^t q(s)\mathrm{d}s\right)\mathrm{d}\tau$$

Beweis: (Stetigkeitssatzes)

Setze $e(t) := |\tilde{z}(t) - \tilde{y}(t)|$

$$\begin{aligned}
\Longrightarrow \quad e(t) &= \left| z_0 + \int_a^t f(\tau, \tilde{z}(\tau)) - \varepsilon(\tau)\mathrm{d}\tau - y_0 - \int_a^t f(\tau, \tilde{y}(\tau))\mathrm{d}\tau \right| \\
&\leq |z_0 - y_0| + \int_a^t \varepsilon(\tau)\mathrm{d}\tau + \int_a^t |f(\tau, \tilde{\tau}\mathrm{d}\tau| \\
&\leq \varepsilon_0 + \varepsilon_1(t-a) + L\int_a^t (-\tilde{z}(\tau) - \tilde{y}(\tau))\mathrm{d}\tau \\
&= p(t) + \int_a^t q(\tau)e(\tau)\mathrm{d}\tau \text{ mit} \\
&\quad p(\tau) = \varepsilon_0 + \varepsilon_1(t-a),\ q(\tau) = L
\end{aligned}$$

Aus dem Lemma von Gronwall folgt:

$$e(t) \leq \varepsilon_0 + \varepsilon_1(t-a) + L\int_a^t (\varepsilon_0 + \varepsilon_1(\tau - a))e^{L(\tau-a)}\mathrm{d}\tau \quad \square$$

Lemma 6.10 (Grounwalls Lemma, diskrete Version)
Seien $(p_n)_{n\in\mathbb{N}}$, $(q_n)_{n\in\mathbb{N}}$, $(e_n)_{n\in\mathbb{N}}$ positive Folgen mit $0 \leq e_{n+1} \leq (1+q_n)e_n + p_n$ für $n < N$. Dann gilt:

$$e_n \leq \left(e_0 + \sum_{j=0}^{n-1} p_j\right) \exp\left(\sum_{j=0}^{n-1} q_j\right),\ n < N$$

Beweis: Durch Induktion

Im folgenden gelte (M) und (L) und $\tilde{y}$ sei die Lösung von $y'(t) = f(t, y(t))$, $y(a) = a_0$, $\quad (t \in [a,b] =: I)$

Gitter: $I_h := \{t_0, \ldots, t_n\} \subseteq I,\ t_{k+1} = t_k + h_k$ mit $h_k > 0,\ k = 0, \ldots, n-1,\ t_0 = a,\ t_n = b$ (Vergleiche die Zerlegung bei der Spline-Interpolation).
Schrittweitenvektor: $\overline{h} = (h_0, \ldots, h_{n-1})$

Feinheit: $h := \max\limits_{k=0,\ldots,n-1} h_k$

Numerishces Verfahren: Φ besteht aus

1. Konstruktion von I_k (eventuell Beschränkungen an $\overline{h}$)
2. Zuordnung einer Gitterfunktion $u_h : I_h \longrightarrow \mathbb{R},\ u_h(t_k) = u_k \approx \tilde{y}(t_k) =: y_k$

Globaler Fehler: $e_h : I_h \longrightarrow \mathbb{R},\ e_h(t_h) := y_h - u_h,\ \|e_h\|_h := \max\limits_{t\in I} |e_h(t)| = \max\limits_{o\leq k\leq n} |y_k - u_k|$ ist der **Diskretisierungsfehler** von h_i auf I_h

Φ heißt **konvergent** gdw. $\|e_h\|_h \longrightarrow 0,\ |h \longrightarrow 0|$ und ist $\|e_h\|_h = O(h^p)$, so dass Φ die **Konvergenzordnung** p hat.

Bemerkung: Bei AWP wird I_h von links nach rechts abgearbeitet, d.h. entlang der Zeitachse mit $u_k(t_0) = u_k(a) = y = \tilde{y}(a)$. Zur Berechnung von u_{k+1} werden bereits bekannte Werte $(t_l, u_l)_{k-r<l<k}$ verwendet.

Beim **Einschrittverfahren** ist $r = 1$. Ist $r > 1$, so spricht man vom **Mehrschrittverfahren**

Ansatz für Taylorentwicklung für $\tilde{y}$

$$\tilde{y}(t) = \tilde{y}(s) + \tilde{y}'(s)(t-s) + \tilde{y}''(s)\tfrac{1}{2}(t-s)^2 + \tilde{y}'''(s)\tfrac{1}{6}(t-s)^3 + \tau$$
$$y'(s) = f(s, \tilde{y}(s))$$
$$y''(s) = \tfrac{\partial}{\partial s} f(s, \tilde{y}(s)) = \partial_t f(s, \tilde{y}(s)) + \partial_y f(s, \tilde{y}(s)) \underbrace{\tilde{y}'(s)}_{=f(s,\tilde{y}(s))}$$
$$y'''(s) = \partial_t f(s, \tilde{y}(s)) + 2\partial_{ty} f(s, \tilde{y}(s))\tilde{y}'(s) + \partial_x^2 f(s, \tilde{y}(s))\tilde{y}'(s)^2 + \partial_y f(s, \tilde{y}(s))\tilde{y}''(s)$$

Damit kann $\tilde{y}(s)$ allein durch Ableitungen von f an der stelle $(s, \tilde{y}(s))$ ausgedrückt werden.

Problem: f muss ausreichend viele partielle Ableitungen haben.

Frage: Wie sind t, s zu wählen?

a) Explizites Verfahen

$s = t_k,\ t = t_{k+1},\ t - s = h_k$
$\Longrightarrow\ \tilde{y}(t_{k+1}) = y_{k+1} \overset{\text{Taylor}}{=} \tilde{y}(t_k) + y_k^{(1)} h_k + y_k^{(2)} \tfrac{1}{2} h_k^2 + y_k^{(3)} \tfrac{1}{6} h_k^3 + \tau_k h_k$, $\tau_k h_k$ heit **Abschneidefehler**

$$y_k^{(1)} = f(t_k, y_k)$$
$$y_k^{(2)} = \partial_t f(t_k, y_k) + \partial_y f(t_k, y_k) y_k^{(1)}$$
$$y_k^{(3)} = \cdots$$

Verfahren

$$u_{k+1} = u_k + u_k^{(1)} h_k + u_k^{(2)} \tfrac{1}{2} h_k^2 + u_k^{(3)} \tfrac{1}{6} h_k^3$$

$$u_k^{(1)} = f(t_k, u_k)$$
$$u_k^{(2)} = \partial_t f(t_k, u_k) + \partial_y f(t_k, u_k) u_k^{(1)}$$
$$u_k^{(3)} = \cdots$$

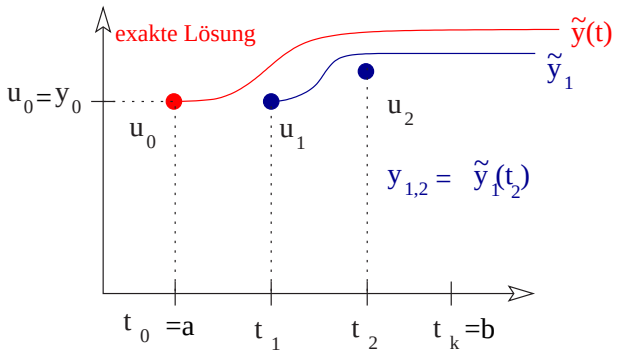


Abbildung 6.2: Explizites Verfahren: Fehler der Schritten

In einem Schritt wird der lokale Fehler $h_k \tau_k$ gemacht. Die lokale Fehler aus früheren Schritten werden weitertransportiert. (siehe Abbildung 6.2)

$\tilde{y}_1$ ist Lösung von $y'(t) f(t, y(t)),\ y(t_1) = u_1$

Problem: lokale Fehler dürfen sich nicht zu sehr akkumulieren.

Beispiel 6.11 (Explizites Euler Verfahren)
$\tilde{y}(t) = \tilde{y})(s) + f(s, \tilde{y}(s))(t-s) + \tau \implies u_{k+1} = u_k + f(t_k, u_k)h_k$

$\tilde{y}$ erfüllt: $y_{k+1} = y_k + f(t_k, y_k)h_k + h_k\tau_k$, d.h. $\tau_k = \underbrace{\frac{y_{k+1} - y_k}{h_k}}_{\approx \tilde{y}'s} - f(t_k, y_k) = O(h_k)$

$$\begin{aligned}
\implies |e_{k+1}| &= |y_{k+1} - u_{k+1}| = |y_k + f(t_k, y_k)h_k + h_k\tau_k - (u_k + f(t_k, u_k)h_k)| \\
&\leq |y_k - u_k| + h_k \left|f(t_k, y_k) - f(t_k, u_k)h_k\right| \\
&\overset{(L)}{\leq} |e_k| + h_k \cdot \underbrace{L}_{\text{Lipschitz Konst.}} \cdot |y_k - u_k| + h_k |\tau_k| \\
&= (1 + h_k L) |e_k| + h_k |t_k| \\
&= (1 + q_k) |e_k| + p_k \quad a_k = h_k L,\ p_k = h_k |\tau_k|
\end{aligned}$$

$$\overset{\text{Groundwall}}{\implies} |e_k| \leq \left(|e_0| + \sum_{j=0}^{n-1} h_j |\tau_j|\right) \exp\left(\sum_{j=0}^{n-1} L h_j\right)$$

$$\implies |e_k| \leq \left(0 + (b-a) \max_{0 \leq h \leq k-1} |\tau_h|\right) \exp\left(L(b-a)\right), \text{ da}$$
$$e_0 = u_0 - y_0 = 0,\ \sum_{j=0}^{k-1} h_j \leq (b-a)$$

$\implies \|e_k\|_k \leq (b-a) \max_{0 \leq k \leq n-1} |\tau_k| \exp\left(L(b-a)\right)$ gdw. $\|e_k\|_k \longrightarrow 0$ gdw. $\max_{o \leq k \leq n} |\tau_k| \longrightarrow 0$

Es gilt: $|\tau_k| = O(h) \implies \|e_k\|_k = O(h)$, d.h. das Verfahren konvergiert mit der Ordnung 1.

Tylorentwicklung:

$\tilde{y}(t)\tilde{y}(s) + f(s, \tilde{y}(s))(t-s) + \tau$

Explizites Euler Verfahren $s = t_k,\ t = t_{k+1},\ (t-s) = h_k$

Implizites Euler Verfahren $s = t_{k+1},\ t = t_k,\ (t-s) = -h_k$

$\implies$ Verfahren: $u_k = u_{k+1} + f(t_{k+1}, u_{k+1})(-h_k)$ bzw. $u_k - u_{k+1} + f(t_{k+1}, u_{k+1})h_k = 0$

d.h. u_{k+1} ist Nullstelle der Funktion $F_k(u) = u_k - f(t_{k+1}, u)h_k$

Berechnung etwa mit dem Newton-Verfahren (Kap. 3)

Frage: Warum überhaupt Implizites Verfahren?

Beispiel: $f(t, y) = -\lambda_j,\ \lambda \in \mathbb{R}$

Exp. Euler Verfahren: $u_{k+1} = u_k + (-\lambda u_k)h_k = (1 - \lambda h_k)u_k$

$$\begin{aligned}
\implies u_{k+1} &= (1 - \lambda h)u_k = (1 - \lambda h)^2 u_{k-1} \\
&= \cdots = (1 - \lambda h)^{k+1} u_0 = (1 - \lambda h)^{k+1} y_0 \\
&\approx e^{(k+1)h\lambda} y_0 = e^{t_{k+1}\lambda} y_0 = \tilde{y}(t_{k+1}),\ \text{da} \\
\tilde{y}(t) &= e^{-\lambda t} \text{ Lösung des AWP } y' = \lambda y, y(0) = y_0
\end{aligned}$$

Falls $1 - \lambda h \leq 0 \iff \lambda > 0$ und $h > \frac{1}{\lambda}$, dann oszilliert u_k um die Null (Abb. 6.3)

Ist $1 - \lambda h < -1 \iff \lambda > 0$ und $h > \frac{2}{\lambda}$, dann gilt sogar: $|u_k| \longrightarrow \infty$ ($k \longrightarrow \infty$), aber $\tilde{y}(t) \longrightarrow 0$ für $t \longrightarrow \infty$, falls $\lambda > 0$

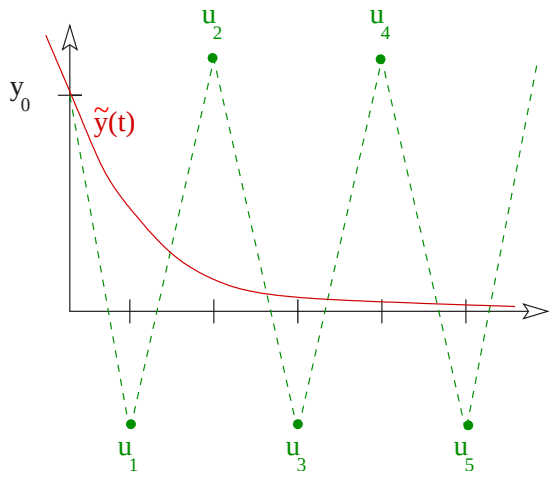


Abbildung 6.3: Explizites Euler Verfahren

Das heißt, falls $\lambda > 0,\ h > \frac{2}{\lambda}$, dann gilt $u_k(t_k) \longrightarrow \infty\ (k \longrightarrow \infty)$ und $\tilde{y}(t) \longrightarrow 0,\ t \longrightarrow \infty$.
Ist $h < \frac{1}{\lambda}$, dann $|u_k - y_k| \longrightarrow 0$. D.h. für λ groß, muss h sehr klein sein, damit das Verfahren gute Lösungen produziert, dann müssen aber sehr viele Schritte durchgeführt werden, um $t_n = b$ zu erreichen, da $n = \frac{b-a}{h}$

Implizites Euler Verfahren

$u_k - u_{k+1} + (-\lambda u_{k+1})h = 0$

$\implies (1 + \lambda h)u_{k+1} = u_k$

$\implies u_{k+1} = \frac{u_k}{1+\lambda h} = \cdots = \frac{u_0}{(1+\lambda h)^{k+1}} = \frac{y_0}{(1+\lambda h)^{k+1}}$

$\implies u_{k+1} = \frac{y_0}{(1+\lambda h)^{k+1}} \approx \frac{y_0}{e^{\lambda h(k+1)}} = e^{-\lambda t_{k+1}} y_0 = \tilde{y}^{(t_{k+1})}$

Falls $\lambda > 0 \implies 0 \leq \frac{1}{(1+\lambda h)^{k+1}} \longrightarrow 0\ (k \longrightarrow \infty)$

Also ist für $\lambda > 0$ das implizite Verfahren geiegneter, da keine Beschränkung an die Schrittweite h gestellt werden müssen.
Alternative zur Taylorentwicklung

Alternative zur Taylorentwicklung nur im $\tilde{y}$, ist eine zusätzliche Taylorentwicklung in f

$$f(t+h, y+hf(t,y)) = f(t,y) + f\partial_t f(t,y) + hf(t,y)\partial_x f(t,y) + O(h^2)$$

$$\begin{aligned}\tilde{y}(t+h) &= \tilde{y}(t) + \tilde{y}'(t)h + \tilde{y}''(t)\tfrac{1}{2}h^2 + O(h^3)\\ &= \tilde{y}(t) + f(t,\tilde{y}(t))h + \tfrac{1}{2}h^2(f(t,\tilde{y}(t)) + h\partial_t f(t,\tilde{y} == +hf(t,\tilde{y}(t))\partial_x f(t,\tilde{y}(t))) + O(h^3)\\ &= \tilde{y}(t) + \tfrac{1}{2}f(t,\tilde{y}(t))h + \tfrac{1}{2}hf(t+h, hf(t,\tilde{y}(t))) + O(h^3)\end{aligned}$$

Daraus erhält man das **Heun-Verfahren**

$$u_{k+1} = u_k + \frac{1}{2}h_k f(t_k, u_k) + \frac{1}{2}h_k f(t_{k+1}, u_k + h_k f(t_k, u_k))$$

Abschneidfehler ist $O(h_k^3)$

Bemerkung: Das Verfahren konvergiert mit der Ordnung h^2. Es wird nur die Funktion f benötigt, aber keine Ableitungen von f. Die systematische Untersuchung solcher Verfahren führt zu dem **Runge-Kutta-Verfahren**. (vgl. die Vorlesung Numerik II)

Herleitung von Mehrschrittverfahren

Bei der Herleitung von Mehrschrittverfahren werden häufig Interpolationsquadraturen verwendet:

$$\tilde{y}(t_{k+1}) = \tilde{y}(t_{k-r+1}) + \int_{t_{k-r+1}}^{t_{k+1}} f(s, \tilde{y}(s))\mathrm{d}s$$

Ersetze $\int_{t_{k-r+1}}^{t_{k+1}} f(s, \tilde{y}(s))ßuds$ durch eine Interpolationsquadratur mit Stützstellen $t_{k-r+1}, \ldots, t_k, t_{k+1}$

$$\Longrightarrow \quad \tilde{y}(t_{k+1}) \approx y(t_{k-r+1}) + \sum_{l=0}^{r-1} \omega_l f(t_{k-l+1}, \tilde{y}(t_{k-l+1}))$$

Verfahren:

$$u_{k+1} = u_{k-r+1} + \sum_{l=0}^{r-1} \omega_l f(t_{k-l}, u_{k-l})$$

Beispiel 6.12

$r = 1$ **Einschrittverfahren**

$$\int_{t_k}^{t_{k+1}} f(s, \tilde{y}(s))\mathrm{d}s \approx \begin{cases} (t_{k+1} - t_k)\, f(t_k, \tilde{y}(t_k)) \\ (t_{k+1} - t_k)\, f(t_{k+1}, \tilde{y}(t_{k+1})) \\ \frac{1}{2}\,(t_{k+1} - t_k)\,(f(t_k, \tilde{y}(t_k)) + f(t_{k+1}, \tilde{y}(t_{k+1})) & \text{Trapezregel} \end{cases}$$

Wahl 1: Explizites Euler Verfahren
Wahl 2: Implizites Euler Verfahren
Wahl 3: **Crank-Nicholson Verfahren**

$$u_{k+1} = u_k + \frac{1}{2} h_k f(t_k, u_k) + \frac{1}{2} h_k f(t_{k+1}, u_{k+1})$$

Das Crank-Nicholson Verfahren ist ein implizites Verfahren und konvergiert mit 2. Ordnung.

$r = 2$ Zu approximieren ist die **Mittelpunktregel**

$$\int_{t_{k-1}}^{t_{k+1}} f(s, \tilde{y}(s))\mathrm{d}s \approx (t_{k+1} - t_{k-1}) f(t_k, \tilde{y}(t_k))$$

Dies führt zum expliziten Verfahren

$$u_{k+1} = u_k + (h_k + h_{k-1}) f(t_k, u_k). \quad k \geq 1$$

Problem: Wie berechnet man u_1? u_0 ist bekannt duch das AWP aber $u_1 \approx \tilde{y}(t_1)$ muss mit einem passenden Einschrittverfahren berechnet werden, das mit der mindestens selben Ordung konvergiert, wie das verwendete Mehrschrittverfahren.

Verallgemeinerungen:

1. Systeme von gewöhnlichen Differentialgleichungen 1. Ordnung.

   Gegeben: $I = [a, b],\ J \subseteq \mathbb{R}^n$ zusammenhängend, offen. $S := I \times J,\ y_0 \in J,\ y_0 = (y_{0,1}, \ldots, y_{0,n})$

   Gesucht: $\tilde{y} \in C^1(I, \mathbb{R}^n)$ mit $\tilde{y}(I) \subseteq J,\ \tilde{y}'(t) = f(t, \tilde{y}(t)),\ \tilde{y}(a) = y_0$, d.h.
   $\tilde{y}_i'(t) = f_i(t, \tilde{y}(t)),\ \tilde{y}_i(a) = y_{0,i},\ f : S \longrightarrow \mathbb{R}^n$

   Die meisten Sätze und Verfahren lassen sich auf Systeme verallgemeinern.

2. AWP höherer Ordnung

   Gegeben: $I = [a, b],\ J \subseteq\in \mathbb{R}^n$ zusammenhängend, offen, $S := I \times J,\ f : S \longrightarrow \mathbb{R},\ y_0 \in J$

   Gesucht: $\tilde{y}_k := \tilde{y}^{(k)}$, d.h. $\tilde{y}(t)(f(t), \tilde{y}'(t), \ldots, \tilde{y}^{(m-1}(t))^\top$

   $$F(t, y_0, \ldots, y_{m-1}) = \begin{pmatrix} y_1 \\ \vdots, y_{m-1} \\ f(t, y_0, \ldots, y_{m-1}) \end{pmatrix}$$

   Es gilt: $\tilde{y}'(t) = \tilde{y}^{(k+1)}(t) = \tilde{y}_{k+1} = F_k(t, y_0, \ldots, y_{m-1})$

   Für $k = 0, \ldots, m-2$ und $y_{m-2}'(t) = \tilde{y}^{(m)}(t) = f(t, y(t), \ldots, y^{(m-1)}(t)) = F_{m-1}(t, y_0(t), \ldots, y_{m-1}(t))$

   D.h. $\tilde{y} = (\tilde{y}_0, \ldots, \tilde{y}_{m-1})$ ist Lösung des AWP $\tilde{y}' = F(t, \tilde{y}),\ \tilde{y}(a) = y_0$

   Daher können AWP höherer Ordnung wie Systeme von AWP 1. Ordnung behandelt werden.

6.2 Numerische Verfahren für Randwertprobleme

Problem B: Gesucht: $u \in C^2(I),\ I = [a, b]$ mit $u''(x) = f(x, u(x), u'(x)),\ x \in [a, b]$ mit Randwerten $u(a) = u_a \in \mathbb{R}$ und $u(b) = u_b \in \mathbb{R}$

Beispiel: Temperaturverteilung in einem Stab der Länge $L = b - a$ bei vorgegebener Temperaturen u_a, u_b an den Enden des Stabes.

Lösungsansatz:

1. Differenzenquotienten: $u''(x_i) \approx \frac{u(x_{i-1}) - 2u(x_i) + u(x_{i+1}}{h^2},\ u'(x_i) \approx \frac{u(x_{i+1}) - u(x_{i-1})}{2h}$

   Damit erhält man ein Verfahren zur Berechnung einer Approximation $u_i \approx u(x_i)$ durch $\frac{u_{i+1} - 2u_i - u_{i-1}}{h^2} = f(x_i, u_i, \frac{u_{i+1} - u_{i-1}}{2h}$

   Falls f unabhängig von u, u' ist, ergibt sich ein LGS für $u_1, \ldots, u_{n-1}$ mit $u_0 = u_a,\ u_n = u_b$

   (siehe Abbildung 6.4). Ansonsten erhält man ein System von nicht-linearen Gleichungen, welches durch ein geeignetes Nullstellenverfahren gelöst werden muss.

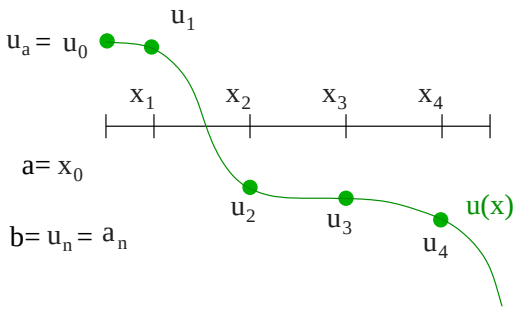


Abbildung 6.4: Differenzenquotienten

2. **Shooting-Verfahren**

Setze $y_1 = u,\ y_2 = u' \implies y_1' = y_2$ und $y_2' = f(x, y_1, y_2),\ y_1(a) = u_a,\ y_2(a) =?$
Sei $\tilde{y}_\alpha = (\tilde{y}_{\alpha,1}, \tilde{y}_{\alpha,2})^\top$ die Lösung des AWP $(*)$ mit Anfangswerten $\tilde{y}_{\alpha,1}(a) = u_a,\ \tilde{y}_{\alpha,2}(a) = \alpha$ für $\alpha \in \mathbb{R}$ beliebig.

Ist $y_{\alpha,1}(b) = u_b$, so ist $y_{\alpha,1}$ Lösung des Randwertproblems.
Setze $F(\alpha) := y_{\alpha,1}(b) - u_b$

Gesucht: Nullstellen von F. (Siehe Abbildung 6.5)

Verfahren: etwa Sekantenverfahren (vgl. Kap3)

$\alpha_0, \alpha_1 \in \mathbb{R}$ vorgebegen.

$$\begin{aligned} \alpha_{k+1} &= \alpha_k - \frac{\alpha_k - \alpha_{k-1}}{F(\alpha_k) - F(\alpha_{k-1}} F(\alpha_k) \\ &= \alpha_k - \frac{\alpha_k - \alpha_{k-1}}{y_{\alpha_k,1}(b) - y_{\alpha_{k-1},1}(b)} (y_{\alpha_k,1}(b) - u_b) \end{aligned}$$

Verfahren: Wähle $\alpha_0, \alpha_1 \in \mathbb{R},\ \alpha_0 \neq \alpha_1$, approximiere Lösung $\tilde{y}_{\alpha_0},\ \tilde{y}_{\alpha_1}$ von $(*)$. Für $k \geq 1$ setze

$$\alpha_{k+1} = \alpha_k - \frac{\alpha_k - \alpha_{k-1}}{u_{\alpha_k,1}(b) - u_{\alpha_{k-1},1}(b)} (u_{\alpha_k,1}(b) - u_b)$$

Approximiere $\tilde{y}_{\alpha_{k+1}}(b)$ durch $u_{\alpha_{k+1}}(b)$

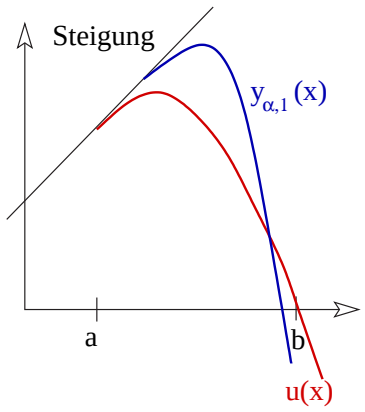


Abbildung 6.5: Shooting-Verfahren

Index