Einführung in die Numerische Mathematik

Vorlesung vom WS 2007/2008

Mario Ohlberger

Institut für Numerische und Angewandte Mathematik
Fachbereich Mathematik und Informatik
Westfälische Wilhelms-Universität Münster

Dieses Skript beruht auf meiner Vorlesung *Einführung in die Numerische Mathematik* vom Wintersemester 2007/2008 an der Westfälische Wilhelms-Universität Münster.
Es ist eine üeberarbeitete und ergänzte Version des Skripts zur Vorlesung "Numerik I" gelesen von Andreas Dedner unter meiner Mitwirkung an der Albert-Ludwigs-Universität Freiburg im Wintersemester 2004/2005. Das Skript wurde in seiner ersten Fassung von Pablo Yanez Trujillo getippt und kann in der ursprünglichen Version von seiner Webpage heruntergeladen werden: `http://klingsor.informatik.uni-freiburg.de/numi.php`.
Es besteht keine Garantie auf Richtigkeit und/oder Vollständigkeit des Manuskripts.

Mario Ohlberger

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 0

Einleitung

Die Numerische Mathematik, oder auch Numerik genannt, beschäftigt sich mit der numerischen Lösung endlichdimensionaler Probleme, sowie mit der Approximation unendlichdimensionaler Probleme durch endlichdimensionale. Die Numerik ist somit eine mathematische Schlüsseldisziplin zur Behandlung von Anwendungsproblemen mit Hilfe des Computers.
Der Numerik geht stehts die Modellierung voraus, deren Ziel es ist ein Anwendungsproblem in der mathematischen Sprache zu formulieren. Dieses Prozedere wird in der Abbildung 1 verdeutlicht.

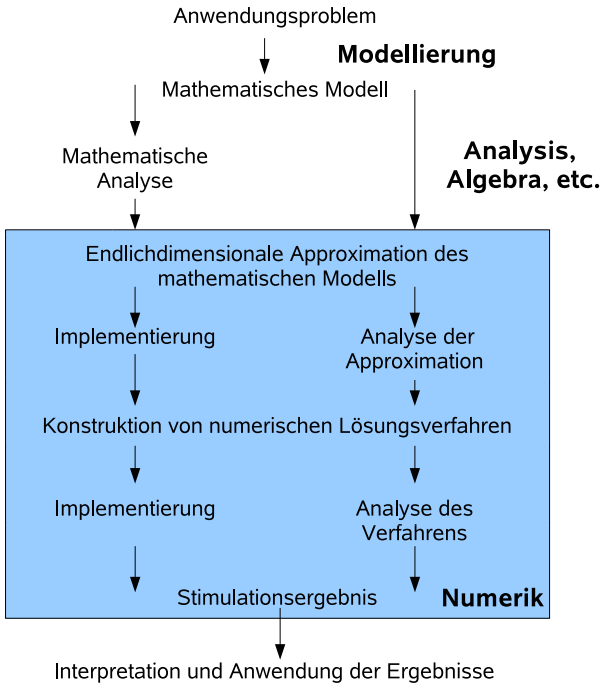


Abbildung 1: Illustration der Vorgehensweise zur Lösung eines Anwendungsproblems

Im folgenden wollen wir an einem einfachen Anwendungsbeispiel dieses Vorgehen skizzieren.

Beispiel 0.1 (Berechnung des Wärmetransports in einem Draht)

Schritt 1: Modellierung

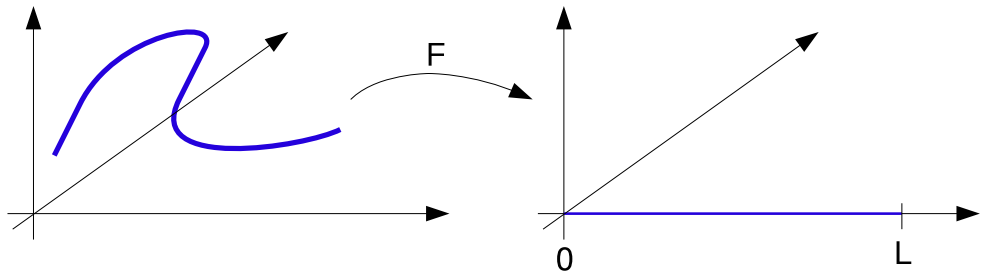


Abbildung 2: Koordinatentranformation zur mathematischen Betrachtung des Wärmetransports in einem Draht.

Wir betrachten den Wärmetransport in einem Draht. Nach einer Koordinatentransformationen, wie sie in Abbildung 2 skizziert ist, können wir den Draht eindimensional durch ein Intervall $I = [0, L]$ repräsentieren. Nach dem Fick'schen Gesetz gilt für die Wärmeleitung, dass der Wärmefluß q proportional zum Gradienten der Temperatur T ist, d.h.

$$q(x,t) = -\sigma \partial_x T(x,t), \forall x \in I, \forall t \in [0, T_{\max}].$$

Dabei bezeichnet $\sigma > 0$ die Wärmeleitfähigkeit und ist eine Materialkonstante. Außerdem gilt für jedes Teilintervall $[a, b] \in I$ und jeden Zeitabschnitt $[t_1, t_2] \in [0, T_{\max}]$:

$$\int_{[a,b]} T(x,t_2)dx - \int_{[a,b]} T(x,t_1)dx = \int_{[t_1,t_2]} q(a,t)dt - \int_{[t_1,t_2]} q(b,t)dt.$$

Sind die Temperatur und der Wärmefluß genügend oft differenzierbar, so folgt mit dem Hauptsatz der Differential- und Integralrechnung

$$\int_{[a,b]} \int_{[t_1,t_2]} \partial_t T(x,t)dxdt = -\int_{[a,b]} \int_{[t_1,t_2]} \partial_x q(x,t)dxdt.$$

Da dies für beliebige a, b, t_1, t_2 gilt, folgt mit dem Hauptsatz der Variationsrechnung

$$\partial_t T(x,t) = -\partial_x q(x,t) = \sigma\ \partial_{xx} T(x,t), \qquad \forall (x,t) \in I \times [0, T_{\max}].$$

Wir erhalten so als mathematisches Modell für die Wärmeleitung in einem Draht eine partielle Differentialgleichung, d.h. eine Gleichung, die die partiellen Ableitungen einer Funktion miteinander in Beziehung setzt. Eine mathematische Analyse zeigt, dass diese sogenannte Wärmeleitungsgleichung eine eindeutige Lösung T besitzt, falls man beispielsweise die Temperatur zum Zeitpunkt $t = 0$ und an den Endpunkten $x = 0, L$ vorgibt. Da die gesuchte Temperaturverteilung T eine differenzierbare Funktion in zwei Veränderlichen darstellt, handelt es sich hierbei um ein unendlichdimensionales Problem. Im nächsten Schritt wollen wir mit Hilfe von sogenannten Finite Differenzenverfahren eine endlichdimensionale Approximation angeben.

Schritt 2: Endlichdimensionale Approximation

Zur Approximation der Wärmeleitungsgleichungen führen wir Zerlegungen $T_h := \{x_i | x_i = hi, i = 0, ..., N+1\}$ und $J_k := \{t_n | t_n = kn, n = 0, ..., M\}$ des Ortsintervalls $[0, L]$ und des Zeitintervalls $[0, T_{\max}]$ ein. Dabei ist $h := L/(N+1)$ die Ortsschrittweite und $k := T_{\max}/M$ die Zeitschrittweite der jeweiligen Zerlegung.
Die Idee der Finite Differenzen besteht darin, alle Ableitungen in der Wärmeleitungsgleichungen durch Differezenquotienten zu ersetzen. Verwenden wir z.B.

$$\partial_t T(x,t_n) \approx (T(x,t_n) - T(x,t_{n-1}))/k$$

und

$$\partial_{xx}T(x_i,t) \approx (T(x_{i+1},t) - 2T(x_i,t) + T(x_{i-1},t))/h^2,$$

so erhalten wir für die Approximation T_i^n von $T(x_i,t_n)$ die Gleichung

$$(T_i^n - T_i^{n-1})/k = \sigma(T_{i+1}^n - 2T_i^n + T_{i-1}^n)/h^2, \quad \forall i = 1,...,N, n = 1,...,M.$$

Dies ist eine lineare Gleichung für T_i^n, die jedoch mit den Linearen Gleichungen des selben Typs für die ebenfalls unbekannten Werte T_i^{n-1} und T_{i+1}^n, T_{i-1}^n gekoppelt ist. Für die Anfangs- und Randwerte verwenden wir die vorgegebenen Werte der Temperatur, d.h.

$$T_i^0 := T(x_i,0), \qquad T_0^n := T(0,t_n), \qquad T_{N+1}^n := T(L,t_n).$$

Berechnet man sukzessive zunächst die Lösung zu den Zeitpunkten $t_1, t_2, t_3, ...$, so erhält man für jeden dieser Zeitschritte t_n ein lineares Gleichungssystem mit N Gleichungen für die Unbekannten T_i^n, $i = 1,...,N$. Wir haben das unendlichdimensionale Problem also durch das sukzessive Lösen von linearen Gleichungssystemen approximiert. Verfahren zur Approximation von Differentialgleichungen werden in Kapitel 6 vorgestellt und in der Vorlesung *Höhere Numerische Mathematik* im Sommersemester detailliert analysiert.

Schritt 3: Numerische Lösung des endlichdimensionalen Problems

Im letzten Lösungsschritt geht es darum die linearen Gleichungssysteme numerisch zu lösen. Hierzu kann z.B. die Gaußelimination verwendet werden. Falls jedoch die Dimension N des Systems sehr groß wird, sind andere Verfahren besser geeignet. Im zweiten Kapitel der Vorlesung wenden wir uns daher der numerischen Lösung von linearen Gleichungssystemen zu.

Wäre in unserem Beispiel der Wärmeleitfähigkeitskoeffizient temperaturabhängig, d.h. $\sigma = \hat{\sigma}(T(x,t))$, mit einer nichtlinearen Funktion $\hat{\sigma}$, so wäre das resultierende Gleichungssystem nichtlinear. Solchen Problemen ist das Kapitel 3 gewidmet.
Wählt man zur Approximation der Differentialgleichungen andere Verfahren, wie z.B. Finite Elemente Verfahren, so ist auch die numerische Approximation von Funktionen durch Polynome und die numerische Berechnung von Integralen Bestandteil der Verfahren. Diese Themen werden in den Kapiteln 4 und 5 behandelt.

Das Beispiel der Wärmeleitung in einem Draht verdeutlicht die Notwendigkeit von numerischen Methoden, wie z.B. die Lösung linearer, oder nichtlinearer Gleichungssysteme, oder die Approximation von Differentialgleichungen. Bevor wir uns diesen Methoden zuwenden, werden im nächsten Kapitel jedoch einige Grundlagen dargestellt. Hierbei handelt es sich zum einen um eine Zusammenstellung wichtiger Begriffe auf der Analysis und Linearen Algebra und zum anderen wird auf die numerische Approximation reeller Zahlen eingegangen und daraus resultierende Fehlerquellen diskutiert.

Kapitel 1

Grundlagen

Im folgenden Abschnitt werden wir Definitionen angeben, die wir in den weiteren Kapiteln benötigen.

1.1 Normierte Räume

Seien $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$ ein Körper und V ein Vektorraum über $\mathbb{K}$.

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

(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 $V = \mathbb{R}^n, v = (v_1, \ldots, v_n) \in \mathbb{R}^n$. Dann ist
$\|v\|_\infty := \max_{1 \leq i \leq n} |v_i|, \quad \|v\|_1 := \sum_{i=1}^{n} |v_i|,$
$\|v\|_2 = \left(\sum_{i=1}^{n} |v_i|^2 \right)^{\frac{1}{2}}, \quad \|v\|_p := \left(\sum_{i=0}^{n} |v_i|^p \right)^{\frac{1}{p}} \quad (1 \leq p < \infty).$

Beispiel 1.3
Sei $V = C^0(I), I = [a, b] \subset \mathbb{R}$. Dann ist
$\|v\|_\infty := \sup \{|v(x)| \mid x \in I\},$
$\|v\|_p := \left(\int_a^b |v(x)|^p \, \mathrm{d}x \right)^{\frac{1}{p}}.$

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

Definition 1.5 (Banachraum)
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** $:\Longleftrightarrow$
$\forall\ \varepsilon > 0\ \exists\ N\ \forall\ m, l > N : \|u_l - u_m\| < \varepsilon.$

Ein normierter Raum $(V, \|\cdot\|)$ *heißt vollständig, falls alle Cauchy-Folgen in* V *bzgl.* $\|\cdot\|$ *in* V *konvergieren. Ein vollständiger normierte Raum heißt* **Banachraum**.

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

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

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

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

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

(ii) T *heißt stetig in* $D \iff$
T *ist stetig für alle* $u \in D$.

(iii) T *heißt* **Lipschitz-stetig** $\iff$
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 7) 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) $\|Tu\|_V \leq \|T\|_{U,V} \|u\|_U$ 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

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

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 $A^* A U = UD$, da $U^{-1} = U^* \implies A^* A U_i = \lambda_i u_i$, 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 $\|A\|_{2,2} = \sup\limits_{\|x\|_2 = 1} \|Ax\|_2$ 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 (iii) (Seite 10) gibt eine a-priori Schranke, die man nutzen kann, um einen 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 $TOL < 1$

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

Meistens ist dies eine Überschätzung des Aufwands.

Satz 1.29 (iv) (Seite 10) kann als Abbruchkriterium während der Iteration benutzt werden, d.h. man bricht ab, falls $\frac{L}{1-L} \|u_k - u_{k-1}\| < TOL$ ist.

1.4 Taylorreihe

Definition 1.31
Sei $C^0(I), I = (a.b)$ *der Raum der* **stetigen Funktionen auf I** . *Mit* $C^m(I) := \{f : I \longrightarrow \mathbb{R} \mid f, f', f'', \ldots, f^{(m)} \text{ ex. und sind stetig}\}$ *bezeichnen wir den Raum der* ***m*-mal stetig differenzierbaren Funktionen**.

Kurzschreibweise: $C^m(a,b)$ *statt* $C^m((a,b))$. *Mit der Definition* $C^\infty(I) := \bigcap_{m\in\mathbb{N}} C^m(I)$ *folgt dann*

$$C^\infty(I) \subset \ldots \subset C^m(I) \subset \ldots \subset C^0(I).$$

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

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

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

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

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

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

Beweis: Für beide Sätze siehe Analysis I.

Folgerung 1.34 (Häufig verwendete Form)
Seien $f \in C^{m+1}(x_0 - h_0, x_0 + h_0),\ x_0 \in \mathbb{R}, h_0 > 0$. Sei $|h| \leq h_0$, dann existiert eine Abbildung $\omega_m : (-h_0, h_0) \longrightarrow \mathbb{R}$ mit $\lim_{h \longrightarrow 0} \omega_m(h) = 0$, so dass gilt

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

Beweis: Wende Satz 1.32 (Seite 11) an mit $x = x_0 + h$, d.h. es existiert ein ξ mit $|\xi| < |h|$ und

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

$$= \frac{f^{(m)}(x_0)}{m!}h^m - \frac{f^{(m)}(x_0)}{m!}h^m + \frac{f^{(m)}(\xi)}{m!}h^m$$

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

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

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 $f(x) - (f(x_0) + f'(x_0)(x - x_0)) = o\left(|x - x_0|^2\right)$ wegen Folgerung 1.34 (Seite 11) mit $h = x - x_0$ und $m = 1$.

Ist $f \in C^2(\mathbb{R})$, 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 11), da f'' beschränkt in einer Umgebung von x_0, d.h. $|f''(\xi)| < C$.

1.5 Approximationsfehler und Fehleranalyse

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.

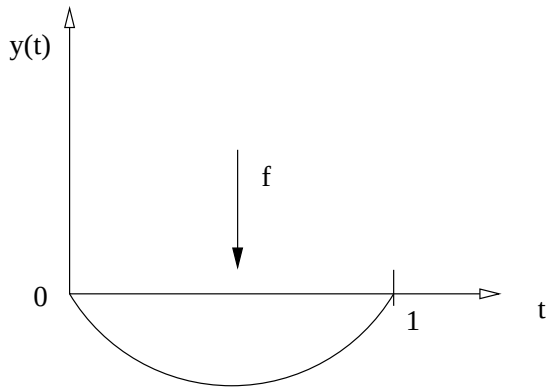


Abbildung 1.1: Modellfehler

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. Dies führt zu Modellfehlern, die jedoch in dieser Vorlesung nicht weiter betrachtet werden. Wir nehmen an, dass die Minimierung von $\overline{E}$ das zu lössende Problem sei: Als notwendige und hinreichende Bedingung für die Minimierung von $\overline{E}$ erhält man durch Variation

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

die Differentialgleichung

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

mit Randwerten $y(0) = y(1) = 0$.

3. **Datenfehler:** c ist eine Materialkonstante, die vom Material des Seils abhängt (aber auch von Temperatur und Luftfeuchtigkeit). 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.

Beispiel: Betrachten wir die Differentialgleichung $u'(t) = (c-u(t))^2,\ u(0) = 1\ c > 0$, so folgt durch Substitution

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

Studieren wir das **Verhalten von u in Abhängigkeit von c**, so sehen wir:

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

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

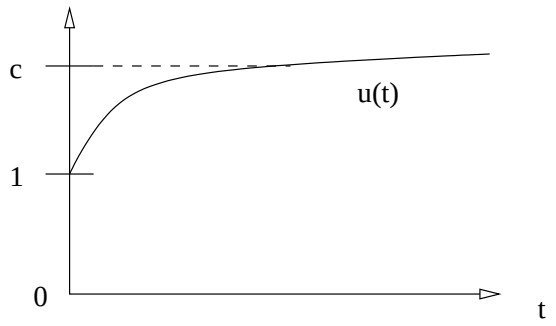


Abbildung 1.2: Auswirkung des Datenfehlers.

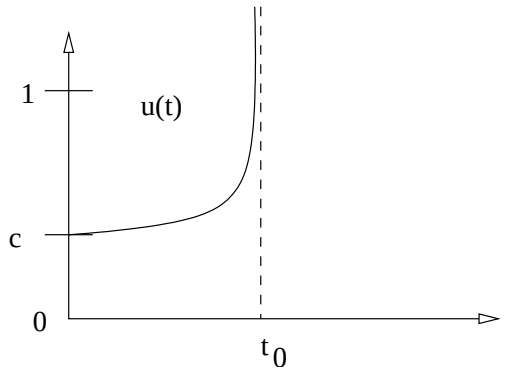


Abbildung 1.3: Auswirkung des Datenfehlers.

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

Durch Messfehler oder auch Approximationsfehler kann leicht $c > 1$ oder $c < 1$ eintreten, und man erhält qualitativ unterschiedliche Ergebnisse.

4. **Diskretisierungsfehler**: Zurück zu unserem Seiltänzerproblem. In der Numerik müssen wir Ableitungen durch etwas Berechenbares ersetzen, auch können wir $y(t)$ nicht für alle t bestimmen. Sei $N \in \mathbb{N}, x_i := ih, i = 0, \ldots, N+1$ mit $h = \frac{1}{N+1}$, so approximieren wir auch hier

$$y''(x_i) \approx \frac{1}{h^2} \left(y(x_{i+1}) - 2y(x_i) + y(x_{i-1}) \right).$$

Setze: $f_i \equiv f(x_i)$ und sei $y_i \approx y(x_i)$, dann ist eine Finite-Diffenrenzen Approximation von $-cy''(t) = f(t), \ t \in (0,1), \ y(0) = y(1) = 0$ gegeben durch

$$-\frac{c}{h^2} \left(y_{i+1} - 2y_i + y_{i-1} \right) = f_i, \ i = 1, \ldots, N, \quad y_0 = y_{N+1} = 0.$$

Die Differenz $|y_i - y(x_i)|$ ist der Diskretisierungsfehler im Punkt x_i. Es muss untersucht werden, wie sich dieser Fehler verhält wenn die Gitterweite h gegen 0 geht.

5. **Lösungsfehler/Abbruchfehler:**

Setze $A = \begin{pmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{N \times N}$ und $F = (f_i)_{i=1}^N \in \mathbb{R}^N$, so ergibt sich aus der

Finite Differenzen Diskretisierung das diskretes Problem: Finde $y_h \in \mathbb{R}^N$ mit

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

Zur Lösung verwenden wir die Identität

$$D y_h = D y_h - A y_h + \frac{h^2}{c} F, \text{ mit } D = diag(2) = \begin{pmatrix} 2 & & 0 \\ & \ddots & \\ 0 & & 2 \end{pmatrix}.$$

Sei y_h^0 ein beliebiger Startwert (z.B: $y_h^0 = 0$). Zur Berechnung von y_h betrachten wir folgende Iterationsvorschrift:

$$D {y_h}^{n+1} := D y_h^n - A y_h^n + \frac{h^2}{c} F$$

bzw.

$$y_h^{n+1} = y_h^n - D^{-1} \left(A y_h^n + \frac{h^2}{c} F \right).$$

Es muss gezeigt werden, dass ${y_h}^n \to y_h$ für $n \to \infty$.

In der Praxis können wir nur bis zu einem endlichen Wert $n_0 \in \mathbb{N}$ rechnen. Das heißt die Lösung eines Problems wird $y_h^{n_0}$ sein. Der Abbruchfehler in der Norm $\|\cdot\|$ ist $\left\| y_h^{n_0} - y_h \right\|$.

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

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

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 $c \approx 0.29998 \cdot 10^9 \frac{m}{s}$ oder Elektronenruhemasse $m_0 \approx 0.911 \cdot 10^{-30}$ kg.

2. Als Normierung nimmt man für $a \neq 0$ an, dass $m_1 \neq 0$ ist.

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 $rd(a) = \min\limits_{\overline{a} \in A} |\overline{a} - a|$ 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 $a \in (-a_{\min}, a_{\min})$ wird in der Regel $rd(a) = 0$ gesetzt ("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). 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 Maschinengenauigkeit.

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 $m_1 \neq 0$ folgt $|a| \geq b^{-1} b^E$ und weiter

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

Setzt man $\varepsilon := \frac{rd(a)-a}{a}$, so folgt $|\varepsilon| \leq eps$ und $rd(a) = \varepsilon a + a = a(1+\varepsilon)$.

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 $|\varepsilon| \leq eps$.

[1] Jedes Kästchen entspricht einem Bit

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

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

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

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

$$\begin{aligned} \bar{I}_0 &= 0.182 \cdot 10^0 \\ \bar{I}_1 &= 0.900 \cdot 10^{-1} \\ \bar{I}_2 &= 0.500 \cdot 10^{-1} \\ \bar{I}_3 &= 0.833 \cdot 10^{-1} \\ \bar{I}_4 &= -0.166 \cdot 10^0 \end{aligned}$$

Dabei bezeichnet $\bar{I}_k$ den berechneten Wert unter Berücksichtigung der Rundungsfehler. 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: $\bar{I}_{21} = -0.158 \cdot 10^{-1}$ und $\bar{I}_{39} = 8.960 \cdot 10^{10}$.

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

(B) Berechnet man die Werte I_k exakt, so ergibt sich bei einer Rundung auf drei Dezimalstellen $I_9 = I_{10}$ und eine Rückwärtsiteration $I_{k-1} = \frac{1}{5}\left(\frac{1}{k} - I_k\right)$ ergibt:

$$\begin{aligned} \bar{I}_4 &= 0.343 \cdot 10^{-1} \\ \bar{I}_3 &= 0.431 \cdot 10^{-1} \\ \bar{I}_2 &= 0.500 \cdot 10^{-1} \\ \bar{I}_1 &= 0.884 \cdot 10^{-1} \\ \bar{I}_0 &= 0.182 \cdot 10^0 \end{aligned}$$

Hier tritt **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 $\begin{pmatrix} \widetilde{x}_1 \\ \widetilde{x}_2 \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \end{pmatrix}$, d.h. wir erhalten ca. 100% Abweichung.

Dies bedeutet, dass kleine Änderungen der Eingabedaten zu großen Änderungen der Lösungen führen können. In diesem Kontext führen wir den Begriff der Kondition eines Problems ein.

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?
Die Matrix

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

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 Aufgabe (f, x_0) die effektive Berechnung von f an der Stelle x_0. Dabei sind x_0 die Eingabedaten.

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

Satz 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

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

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 11) 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. □

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*

$$\|f(x) - f(x_0)\| \leq L_{abs} \|x - x_0\| \qquad (*)$$

für alle $x \in B_\delta(x_0)$*. Gibt es keine solche Konstante, so heißt das Problem* **schlecht gestellt**.

Sei im folgenden $L_{abs}(\delta)$ *die kleinste Zahl mit der Eigenschaft* $(*)$.
Analog sei $L_{rel}(\delta)$ *die 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)
Wir definieren $K_{abs} := \lim\limits_{\delta \searrow 0} L_{abs}(\delta)$ *die* **absolute Konditionszahl** *und* $K_{rel} := \lim\limits_{\delta \searrow 0} L_{rel}(\delta)$ *die* **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 $f'(x) = A^{-1}$.
Damit folgt $K_{abs} = \|A^{-1}\|$; und hieraus mit $Ax = b$ und der Submultiplikativität der zugeordneten Norm:

$$K_{rel} = \|A^{-1}\| \frac{\|b\|}{\|A^{-1}b\|} = \|A^{-1}\| \frac{\|Ax\|}{\|x\|} \leq \frac{\|A^{-1}\| \cdot \|A\| \cdot \|x\|}{\|x\|} = \|A^{-1}\| \cdot \|A\|.$$

Wir definieren entsprechend die Kondition der Matrix A durch

$$cond(A) := \|A^{-1}\| \cdot \|A\|.$$

Beachte, dass ein $x \in \mathbb{R}^m$ existiert 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: $cond(A) = \|A^{-1}\| \|A\| \approx 10^9$. Das Problem ist also schlechte konditioniert.

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. Direkte Verfahren
2. Iterative Verfahren

Aus der Schule (und den Lineare Algebra-Vorlesungen) ist uns ein direkte Verfahren bekannt, das Gaußsche Eliminationsverfahren. Für kleine Gleichungssysteme eignet sich dieses Verfahren, jedoch kann das Verfahren für $n >> 1000$ sehr ineffizient werden, da das Verfahren einen Rechenaufwand der Ordnung n^3 hat. Aus diesem Grund werden wir andere Verfahren kennenlernen, mit denen man schneller ans Ziel kommen kann.

Wir werden Probleme folgender 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 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 Lösungen existieren).

(C) Geg: $A \in \mathbb{R}^{n \times n}$
Ges: A^{-1} (falls die Inverse 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 $Ax = b$.

Alle diese 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 direktes Verfahren mit dieser Komplexität.

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

   Ein Nachteil ist, dass während der laufenden Berechnung keine Näherung vorliegt, 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 darin, 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 ist $A + \Delta A$ regulär und es gilt

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

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

Bemerkung: $cond(A) := \|A\| \left\|A^{-1}\right\|$ ist der entscheidende Verstärkungsfaktor für den relativen Fehler.

2.1 Direkte Verfahren

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

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

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

Dann 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, so ist $Ax = b$ lösbar. Aus

$$b_i = \sum_{u=1}^{n} a_{iu} x_u = \sum_{u=i}^{n} a_{iu} x_u$$

erhalten wir den Algorithmus:

$$\begin{aligned} i = n: \quad & x_n = \tfrac{b_n}{a_{nn}}, \\ i < n: \quad & x_i = \tfrac{1}{a_{ii}} \left(b_i - \sum_{u=i+1}^{n} a_{iu} x_n \right). \end{aligned}$$

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 $\widetilde{A}x = \widetilde{b}$ dieselbe Lösung hat wie $Ax = b$.

Eine Lösung dieses Problems liefert der Gaußalgorithmus:

2.1.1 Gaußalgorithmus/LR-Zerlegung

Der Algorithmus startet mit

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

und führt durch sukzessive Manipulation auf $\left(A^{(p)}, b^{(p)}\right), p = 1, \ldots, n-1$, so dass aus $A^{(p-1)}x = b^{(p-1)}$ folgt $A^{(p)}x = b^{(p)}$. Um $\left(A^{(1)}, b^{(1)}\right)$ zu berechnen, wird zur i-ten Zeile für $i = 2, \ldots, n$ das $a_{i1}^{(0)}/a_{11}^{(0)}$-fache der ersten Zeile hinzuaddiert. Wir erhalten somit

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

$$\downarrow$$

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

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

Unsere Rechnung setzt voraus, dass stets gilt $a_{pp}^{(p-1)} \neq 0$ gilt, ansonsten müssen zuerst Zeilen vertauscht werden. 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 bedeuted dies, dass A singulär ist. Das Zeilenvertauschen wird als **Pivotisierung** bezeichnet. Häufig 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** oder **Spaltenpivotisierung** bezeichnet.

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

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

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

Mit Pivotisierung folgt hingegen nach Zeilentausch:

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

und schließlich nach der Elimination

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

Hier folgt also $x_2 = \frac{1-2\varepsilon}{1-\varepsilon} \approx 1$ und $x_1 = 2 - x_2 \approx 1$, da auf einem Computer $rd(1 - 2\varepsilon) = 1$ für sehr kleines ε gilt.

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. Hier 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.

Wir fassen das Gaußverfahren, wie folgt zusammen.

Algorithmus 2.4 (Gaußverfahren)
Setze $q_i = i \ \ (i = 1, \ldots, n)$.

Für $p = 1, \ldots, n-1$:

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

Vertausche die Zeilen $q_j \longleftrightarrow q_p$ [Spaltenpivotisierung]

Für $k = p+1, \ldots, n$:

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

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

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

Für $j = p+1, \ldots, n$:

Setze $a_{q_k j} = a_{q_k j} - l \cdot a_{q_p j}$ [Matrix $A^{(p)}$]

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

Die Lösung von $Ax = b$ wird anschließend durch Rückwärtseinsetzen wie folgt gelöst:

$$\begin{aligned} &\text{Setze } x_n = b_{q_n} / a_{q_n n}. \\ &\text{Für } k = n-1, \ldots, 1: \\ &x_k = \left(b_{q_k} - \sum_{i=k+1}^{n} a_{q_k i} x_i \right) \Big/ a_{q_k k}. \end{aligned}$$

Bemerkungen:

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

(ii) Anstelle der entstehenden Nullen wird der Multiplikationsfaktor l gespeichert.

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

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

Formaler Zugang:
1) Uminterpretation der Pivotisierung:
Im i-ten Schritt des Gaußalgorithmus werden Zeilen vertauscht $i \longleftrightarrow k$, $k > i$. Zur Umformulierung

betrachten wir folgende Matrix.

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

Lemma 2.5
Für die Matrizen P_{ik} gilt:

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

(ii) $B = AP_{ik}$ entspricht der Matrix nach der Vertauschung der i-ten und k-ten 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. Ist $P = P_{l_m,k_m}, \dots P_{l_1,k_1}$, eine Permutationsmatrix, so gilt $P^{-1} = P_{l_1,k_1}, \dots P_{l_m,k_m}$.

2) Berechnung im Gaußverfahren:
Wir betrachten die Matrix:

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

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

Lemma 2.7

(i) Sei $B = L_iA = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$, wobei $b_i \in \mathbb{R}^n$ Zeilenvktoren sind. Dann gilt

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

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

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

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

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

Beweis: Durch nachrechnen.

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

$$R = L_{n-1}^{-1} P_{n-1} \cdots L_1^{-1} P_1 A.$$

Dabei it $P_i = P_{ij}$ für ein $j \geq i$ und R eine obere $\triangle$-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: Ist $A = LR = MS$, wobei L, M untere $\triangle$-Matrizen mit Diagonalelementen 1 sind, und R, S obere $\triangle$-Matrizen sind, so folgt $L = M,\ R = S$.

Bemerkung: Ist $PA = LR$ gegeben, so kann man $Ax = b$ lösen, indem man in zwei Schritten durch Vorwärts-, bzw. Rückwärtseinsetzen löst:

(a) Löse $Lz = Pb$,

(b) löse $Rx = z$.

Dies gilt, da

$$\begin{array}{rcl} Ax = b & \iff & PAx = Pb, \\ & \iff & LRx = Pb, \\ & \iff & LRx = Lz, \\ & \iff & Rx = z. \end{array}$$

I.A. wird L in den frei werdenden Stellen von A gespeichert (die 1 auf der Diagonalen muss man nicht speichern). Also

$$A^{(n-1)} := \begin{pmatrix} & \tilde{R} \\ \tilde{L} & \end{pmatrix}.$$

Dabei ist $\tilde{R}$ die obere Dreiecksmatrix von R und $\tilde{L}$ die untere Dreiecksmatrix von L ohne die Diagonale. L und R benötigen also zusammen n^2 Speicherstellen.

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 $\implies$ $P_1 L_1 \dots P_{n-1}L_{n-1}R = A$.

Definiere Permutationsmatrix $P := P_{n-1} \dots P_1$ und $L := PP_1L_1 \dots P_{n-1}L_{n-1}$

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

Noch zu zeigen: L ist untere $\triangle$-Matrix mit Diagonaelementen 1.

Es ist:

$$\begin{aligned} L &= PP_1L_1 \dots P_{n-1}L_{n-1} = P_{n-1} \dots \underbrace{P_1P_1}_{=E_n} L_1P_2 \dots P_{n-1}L_{n-1} \\ &= P_{n-1} \dots P_2L_1P_2 \dots P_{n-1}L_{n-1}. \end{aligned}$$

Wir setzen:

$$\tilde{L}_1 := P_2L_1P_2 \text{ und für} p = 2, ..., n-1 : \ \tilde{L}_p := P_{p+1}\tilde{L}_{p-1}L_pP_{p+1}$$

. Dann hat $\tilde{L}_p$ die Gestalt:

$$\tilde{L}_p = \begin{pmatrix} 1 & & & & & & 0 \\ * & \ddots & & & & & \\ \vdots & \ddots & \ddots & & & & \\ \vdots & & * & \ddots & & & \\ \vdots & & \vdots & 0 & \ddots & & \\ \vdots & & \vdots & \vdots & \ddots & \ddots & \\ * & \cdots & * & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Außerdem gilt, dass $P_q\bar{L}_pP_q$ (mit $q > p$) ebenfalls diese Gestalt, da $P_q = P_{qk},\ q < k$.

Also folgt

$$\begin{aligned} L &= P_{n-1} \dots P_3P_2L_1P_2L_2P_3 \dots P_{n-1}L_{n-1} \\ &= P_{n-1} \dots P_3\widetilde{L}_1L_2P_3 \dots P_{n-1}L_{n-1} \\ &= P_{n-1} \dots P_4\widetilde{L}_2L_3P_4 \dots P_{n-1}L_{n-1} \\ &\vdots \\ &= \widetilde{L}_{n-1} \end{aligned}$$

und somit ist L untere $\triangle$-Matrix.

Zur Eindeutigkeit:
Es gilt: L^{-1} hat ebenfalls untere $\triangle$-Gestalt mit Diagonalelementen 1 (betrachte $L^{-1}L = E_n$). Analog folgt, dass S^{-1} eine obere $\triangle$-Matrix ist.

Also ist $L^{-1}M$ untere $\triangle$-Matrix mit Diagonalelementen 1 und RS^{-1} hat obere $\triangle$-Gestalt. Aus $LR = MS$ folgt $RS^{-1} = L^{-1}M = E_n$ und hieraus mit der Eindeutigkeit der Inversen: $R = S \wedge L = M$. $\square$

Weiter Anwendungen der LR-Zerlegung

(a) Determinantenberechnung einer Matrix A.
Hat R obere/untere $\triangle$-Gestalt, so gilt

$$\det(R) = \prod_{i=1}^{n} r_{ii}.$$

Aus der LR Zerlegung folgt

$$R = L_{n-1}^{-1} P_{n-1} \dots L_1^{-1} P_1 A$$

und somit

$$\det R = \det(L_{n-1}^{-1}) \det(P_{n-1}) \dots \det(L_1^{-1}) \det(P_1) \det(A).$$

Weiter gilt:

$$\det\left(L_i^{-1}\right) = 1 \quad \text{und} \quad \det(P_i) = \det(P_{ik}) = \begin{cases} 1 & : \quad i = k \\ -1 & : \quad i \neq k \end{cases}.$$

Also folgt:

$$\det(A) = \begin{cases} \det(R) & : \text{ gerade Anzahl von Zeilenvertauschungen} \\ -\det(R) & : \text{ ungerade Anzahl von Zeilenvertauschungen} \end{cases}$$

$$= \begin{cases} \prod_{i=1}^{n} r_{ii} & : \text{ gerade Anzahl von Zeilenvertauschungen} \\ -\prod_{i=1}^{n} r_{ii} & : \text{ ungerade Anzahl von Zeilenvertauschungen} \end{cases}$$

(b) 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 eventuell auch Spalten vertauscht werden, was den Rang der Matrix nicht verändert. Ist dies nicht möglich, so hat $A^{(p)}$ die Gestalt

$$A^{(p)} = \left(\begin{array}{c|c} * & * \\ \hline 0 & 0 \end{array}\right)$$

Dabei sind die ersten p Zeilenvektoren linear unabhängig, aber alle weiteren Zeilenvektoren sind linear abhängig. Es folgt $\text{Rang}\,(A) = \text{Rang}\left(A^{(p)}\right) = p$.

Achtung: Aufgrund von Rundungsfehlern kann dieses Verfahren numerisch zu falschen Ergebnissen führen:

(c) Berechnung der Umkehrmatrix A^{-1} einer Matrix A.

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

2. Berechnung durch simultane Elimination

$$\begin{array}{c|ccc} & 1 & & 0 \\ A & & \ddots & \\ & 0 & & 1 \end{array} \longrightarrow \quad \text{Vorwärtselimination} \quad \begin{array}{ccc|ccc} r_{11} & \cdots & * & * & & 0 \\ & \ddots & \vdots & \vdots & \ddots & \\ 0 & & r_{nn} & * & \cdots & * \end{array} \longrightarrow$$

$$\text{Rückwärtselimination} \quad \begin{array}{ccc|c} r_{11} & & 0 & \\ & \ddots & & * \\ 0 & & r_{nn} & \end{array} \longrightarrow \begin{array}{ccc|c} 1 & & 0 & \\ & \ddots & & A^{-1} \\ 0 & & 1 & \end{array}$$

2.1.2 Gauß-Jordan Verfahren

Diese Methode beruht darauf, durch Matrixumformungen von $Ax = b$ zu $Bb = x$ mit $B = A^{-1}$ überzugehen. Die Idee des Verfahren ist folgende: Ist $a_{pq} \neq 0$, so kann die p-te Gleichung nach x_q aufgelöst werden:

$$x_q = -\frac{a_{p1}}{a_{pq}}x_1 - \ldots - \frac{a_{pq-1}}{a_{pq}}x_{q-1} + \frac{1}{a_{pq}}b_q - \frac{a_{pq+1}}{a_{pq}}x_{q+1} - \ldots - \frac{a_{pn}}{a_{pq}}x_n.$$

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

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

Man erhält also eine Matrix $\widetilde{A}$ mit

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

Kann dieser Schritt z.B. mit $p = q$ n-mal durchgeführt werden, so ergibt sich

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

Dies entspricht einem Algorithmus ohne Pivotisierung, d.h. $a_{ii} \neq 0$, Varianten mit Pivotisierung sind ebenfalls möglich (siehe z.B. Stoer, Numerische Mathematik I, Springer, 1989, Abschnitt 4.2).

2.1.3 Cholesky Verfahren für SPD-Matrizen

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

$$A = QQ^T.$$

Dabei ist $Q = LD^{1/2}$ mit $D^{1/2} = \text{diag}(\sqrt{r_{11}}, \ldots, \sqrt{r_{nn}})$.
Die Existenz einer solchen Zerlegung sieht man wie folgt ein:

$$A = LR \implies A = LD\tilde{R} = LDL^T = LD^{1/2}(LD^{1/2})^T = QQ^T,$$

wobei $\tilde{R} = L^T$ aus der Symmetrie von A folgt. Die positive Definitheit der Matrix A ist notwendig, damit $D^{1/2}$ wohldefiniert ist.
Unter Ausnutzung der Symmetrie erhält man folgenden Algorithmus, der die untere Dreiecksmatrix von Q anstelle der unteren Dreiecksmatrix von A abspeichert und $D^{-1/2}$ in einem Vektor d:

Algorithmus 2.10 (Cholesky Verfahren)

```
Für i = 1,...,n:
    Für j = i,...,n:
        Setze u := a_ij
        Für k = i-1,...,1:
            Setze u := u - a_jk a_ik
        Falls i = j,
            Setze d_i := 1/√u   (Abbruch, falls u ≤ 0)
        Sonst
            Setze a_ji := d_i u
```

Dieser Algorithmus hat aufgrund der Symmetrie den halben Aufwand im Vergeleich zum Gauß-Algorithmus.

2.1.4 LR-Zerlegung für Tridiagonalmatrizen

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 mittels LR-Zerlegung zerlegt werden. Diese Zerlegung kann explizit in Abhängigkeit von α, β und γ hingeschrieben werden (siehe Übungsaufgabe).

2.2 Überbestimmte Gleichungssysteme/Ausgleichsrechnung

Problem: Gegeben sind m Messdaten (zum Beispiel Zeit und Konzentration) $(x_1, y_1), \ldots, (x_m, y_m)$ und Funktionen $u_1, \ldots, u_n,\ (n, m \in \mathbb{N},\ n \leq m)$.

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

$$\triangle_2 := \left(\sum_{j=1}^{m} \left(u(x_j) - y_j \right)^2 \right)^{\frac{1}{2}} = \inf_{c_1,\ldots,c_n \in \mathbb{R}} \left(\sum_{j=1}^{m} \left(\sum_{i=1}^{n} \left(c_i u_i(x_j) \right) - y_j \right)^2 \right)^{\frac{1}{2}}$$

Dieses Problem wird als das **Gaußsche Ausgleichsproblem** oder als die Methode der kleinsten Quadrate (least squares) bezeichnet.

Bemerkung: Das **Tschebyscheffsche Ausgleichsproblem**, bei dem bezüglich der Maximumsnorm minimiert wird, d.h.

$$\triangle_\infty := \inf_{c_1,\ldots,c_n \in \mathbb{R}} \max_{i=1,\ldots,m} |c_i u_i(x_j) - y_j|,$$

ist deutlich schwieriger.

Sei:
$c = (c_1, \ldots, c_n)^\top \in \mathbb{R}^n$ (der gesuchte Lösungsvektor),
$x = (x_1, \ldots, x_m)^\top \in \mathbb{R}^m,\ y = (y_1, \ldots, y_m)^\top \in \mathbb{R}^m$,
$A = (a_{ij}) \in \mathbb{R}^{m \times n}$ mit $a_{ij} = u_i(x_j)$.

Dann ist das Ausgleichsproblem äquivalent zur Minimierung des Funktionals

$$F(c) := \|Ac - y\|_2 . \qquad (AGP)$$

Bemerkung: Sind $m = n,\ u_1, \ldots, u_n$ linear unabhängig und $x_1, \ldots, x_m$ paarweise verschieden, so ist A regulär und $c = A^{-1}y$ ist das gesuchte Minimum. Im Allgemeinen ist jedoch $n << m$, so dass $\text{Rang}(A) \leq n$ folgt. In solchen Fällen erwarten wir, dass (AGP) einen Minimierer hat, jedoch $Ac = y$ entweder keine oder sehr viele Lösungen hat.

Satz 2.11 (Normalengleichung)
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 $(Ax = b)$ mit kleinstem Fehlerquadrat, d.h. $\overline{x}$ minimiert $F(x) = \|Ax - b\|_2$. Dies ist äquvalent dazu, dass $\overline{x}$ die **Normalengleichung**

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

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

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

mit $y \in \text{Kern}(A)$. In diesem Fall wird meistens die Lösung x_A mit minimaler 2-Norm gesucht, d.h.

$$\|x_A\| = \inf\left\{\|x\|_2 \;\middle|\; x \text{ Lösung des Ausgleichsproblems}\right\}.$$

Diese Lösung ist eindeutig.

Lemma 2.12
Sei $A \in \mathbb{R}^{m\times n}$, dann gelten für $A^\top A \in \mathbb{R}^{n\times n}$ folgende Eigenschaften:

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

(ii) $A^\top A$ ist positiv semidefinit. Falls $\text{Rang}(A) = n$ ist, so ist $A^\top A$ 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 und es gilt dimKern$(A\top A - \lambda I)$ = dimKern$(AA\top - \lambda I)$ für alle Eigenwerte $\lambda > 0$.

(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 2.12(iv))

Es gilt $\mathbb{R}^m = \text{Bld}(A) \oplus \text{Bld}(A)^\perp$, d.h. es ist zu zeigen: $\text{Bld}(A)^\perp = \text{Kern}(A^\top)$

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

Beweis: (Satz 2.11)

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

$$\begin{aligned}
\|b - Ax\|_2^2 &= \|b - A\overline{x} + A(\overline{x} - x)\|_2^2 \\
&= \langle b - A\overline{x} + A(\overline{x} - x), b - A\overline{x} + A(\overline{x} - x)\rangle \\
&= \langle b - A\overline{x}, b - A\overline{x}\rangle + 2\langle b - A\overline{x}, A(\overline{x} - x)\rangle + \langle A(\overline{x} - x), A(\overline{x} - x)\rangle \\
&= \|b - A\overline{x}\|_2^2 + \|A(\overline{x} - x)\|_2^2 + 2\left\langle A^\top(b - A\overline{x}), \overline{x} - x\right\rangle \\
&\geq \|b - A\overline{x}\|_2^2
\end{aligned}$$

Also gilt für alle $x \in \mathbb{R}^n :\ \|b - A\overline{x}\|_2 \leq \|b - Ax\|_2$.

Sei nun umgekehrt $\overline{x}$ eine Lösung des Minimierungsproblems, so folgt

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

Also folgt $A^\top A\overline{x} = A^\top b$ und somit ist $\overline{x}$ Lösung der Normalengleichung.
Insbesondere kann also die Existenz einer Lösung des AGPs durch das Lösen der Normalengleichung gezeigt werden.
Zur Lösung der Normalengleichung: Es ist $b \in \mathbb{R}^m = \mathrm{Bld}(A) \oplus \mathrm{Kern}(A^\top)$. Also können wir $b = s + r$ zerlegen mit $s \in \mathrm{Bld}(A),\ r \in \mathrm{Kern}(A^\top)$. Zu $s \in \mathrm{Bld}(A)$ existiert ein $\overline{x} \in \mathbb{R}^n$ mit $A\overline{x} = s$. Es folgt:

$$A^\top A\overline{x} = A^\top s + A^\top r = A^\top(r + s) = A^\top b,$$

d.h. $\overline{x}$ ist Lösung der Normalengleichung.

Ist $\mathrm{Rang}(A) = n$, so folgt, dass $A^\top A$ positiv definit (Lemma 2.12(ii)) und somit auch regulär ist. Insbesondere folgt daraus, dass $\overline{x} = (A^\top A)^{-1} A^\top b$ die eindeutige Lösung der Normalengleichung ist.

Ist $\mathrm{Rang}(A) < n$ und seien x_1, x_2 Lösungen der Normalengleichung. Dann gilt $b = Ax_i + (b - Ax_i) \in \mathrm{Bld}(A) \oplus \mathrm{Kern}(A^\top)$.

Da die Zerlegung $\mathbb{R}^m = \mathrm{Bld}(A) \oplus \mathrm{Kern}(A^\top)$ eindeutig ist, folgt $Ax_i = s = A\overline{x}$, also $A(x_i - \overline{x}) = 0$, d.h. $x_i - \overline{x} \in \mathrm{Kern}(A)$.

Wir definieren die Lösungsmenge K durch

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

Dann ist K kompakt und da die Norm $\|\cdot\|_2$ stetig ist, nimmt sie ihr Minimun auf K an. Sei also x_A die Lösung des AGPs mit

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

Sind nun $x_1, x_2 \in K$ Lösungen mit $\|x_1\|_2 = \|x_2\|_2 = \rho$, so folgt $\frac{x_1+x_2}{2} \in K$ und daher

$$\rho \le \left\| \frac{x_1 + x_2}{2} \right\|_2 \le \frac{1}{2}\|x_1\|_2 + \frac{1}{2}\|x_2\|_2 = \rho.$$

Wir erhalten somit $\left\|\frac{x_1+x_2}{2}\right\|_2 = \rho$ und es folgt

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

Beispiel 2.13 (Ausgleichsgerade)
Gegeben: Messdaten:

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

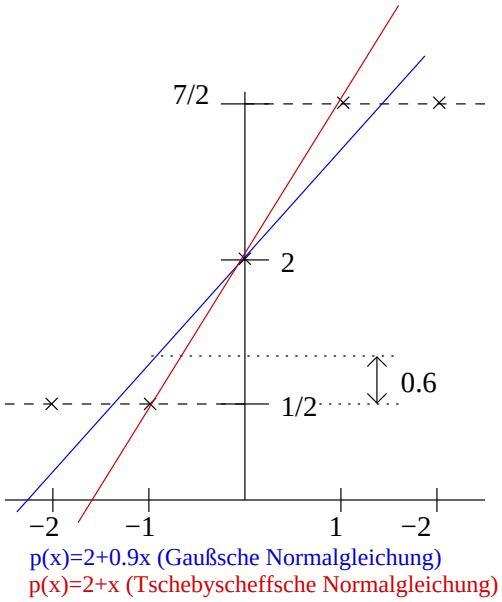


Abbildung 2.1: Ausgleichsgerade

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

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

Nach Satz 2.12 ist dann $(a, b)^\top$ die Lösung der Normalengleichung mit

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

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

Da $\text{Rang}(A) = 2$ ist, ist die Normalengleichung eindeutig lösbar. Aus $A^\top A \begin{pmatrix} a \\ b \end{pmatrix} = A^\top c$ folgt $\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 2 \\ 0.9 \end{pmatrix}$, d.h. $u(x) = 2 + 0.9x$ ist die Ausgleichsgerade.

Berechnet man die Abweichung, so folgt $\triangle_2 = \sqrt{0.9} < 1$, $\triangle_\infty = 0.6$.

Die Lösung des Tschebyscheffs-Problems ist gegeben durch $u(x) = 2 + x$. Hier erhält man $\triangle_2 = 1$; $\triangle_\infty = \frac{1}{2}$.

Bemerkung: Physikalisch könnte z.B. ein nichtlinearer Zusammenhang der Form $u(x) = \frac{a}{1+bx}$ sinnvoller sein.

[1] englischer Ausdruck der Ausgleichungsgerade

Mithilfe der Transformation $\hat{u}(x) = \frac{1}{u(x)} = \frac{1}{a} + \frac{b}{a}x = \hat{a} + \hat{b}x$ lassen sich jedoch solche Probleme oft auf die Berechnung der linearen Ausgleichungsgeraden zurückführen.

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 $\text{Rang}(A) = n,\ A \in \mathbb{R}^{n\times n}$, dann ist $cond(A^\top A) \sim cond(A)^2$.

2.2.1 QR-Zerlegung nach Householder

Idee: Sei $\text{Rang}(A) = n,\ A \in \mathbb{R}^{m\times n}$. Anstelle einer LR-Zerlegung sei

$$A = QR$$

mit einer oberen $\triangle$-Matrix $R \in \mathbb{R}^{m\times n}$ und einer orthogonalen Matrix $Q \in \mathbb{R}^{m\times m}$ gegeben, d.h. $Q^{-1} = Q^\top$. Dann folgt

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

wobei $\widetilde{R}$ eine reguläre obere $\triangle$-Matrix ist. Durch Einsetzen erhalten wir

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

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

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

so folgt aus $R^\top = \Big(\underbrace{\widetilde{R}^\top}_{n} \;\Big|\; \underbrace{0}_{m-n}\Big)$: $R^\top c = \begin{pmatrix} \widetilde{R}^\top \widetilde{c} \\ 0 \end{pmatrix}$.

$\widetilde{R}^\top$ ist regulär. Sei also $\overline{x} \in \mathbb{R}^n$ die Lösung von $\widetilde{R}\overline{x} = \widetilde{c}$, so ist $\overline{x}$ leicht zu berechnen, da $\widetilde{R}$ obere $\triangle$-Matrix ist.

Da $R\overline{x} = \begin{pmatrix} \widetilde{c} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ folgt $R^\top R\overline{x} = R^\top c = R^\top Q^\top b$ und somit ist $\overline{x}$ die Lösung der Normalengleichung.
Konzentrieren wir uns also auf die Berechnung einer QR Zerlegung.

QR-Zerlegung nach Householder

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

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 2.14 (Dyadisches Produkt)
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.*
Es gilt $A \in \mathbb{R}^{m\times m}$ und $a_{ij} = u_i v_j$ $(1 \leq i, j \leq m)$.

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

Folgerung 2.15
Seien $u, v \in \mathbb{R}^m$, $A = uv^\top$ und $w \in \mathbb{R}^m$. Dann gilt

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

Definition 2.16 (Householder Matrix)
Sei $v \in \mathbb{R}^m$, $v \neq 0$. Die Matrix $H(v) = \mathbb{I} - 2\frac{vv^\top}{\|v\|_2^2}$ heißt **Householder Matrix**.
Wir setzen $H(0) = \mathbb{I}$.

Folgerung 2.17
Sei $v \in \mathbb{R}^m$, dann gilt:

(i) $H(v)$ ist symmetrisch.

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

Beweis:

(i) Es ist $H(v)_{ij} = \delta_{ij} - 2\frac{v_i v_j}{\|v\|_2^2} = \delta_{ji} - 2\frac{v_j v_i}{\|v\|_2^2} = H(v)_{ij}^\top$. Dabei bezeichnet δ_{ij} das Kronecker Symbol.

(ii) Wegen (i) bleibt zu zeigen, dass $H(v)^2 = \mathbb{I}$ gilt. Es ist

$$\begin{aligned} H(v)^2 &= \left(\mathbb{I} - 2\frac{vv^\top}{\|v\|_2^2}\right)\left(\mathbb{I} - 2\frac{vv^\top}{\|v\|_2^2}\right) = \mathbb{I} - 4\frac{vv^\top}{\|v\|_2^2} + 4\frac{(vv^\top)^2}{\|v\|_2^4} \\ &= \mathbb{I} - \frac{4}{\|v\|_2^2}\left(vv^\top - \frac{(vv^\top)^2}{\|v\|_2^2}\right) \\ &= \mathbb{I} - \frac{4}{\|v\|_2^2}\left(vv^\top - \frac{\langle v,v\rangle vv^\top}{\|v\|_2^2}\right) \text{ (wegen 2.15(ii))} \\ &= \mathbb{I} \quad \square \end{aligned}$$

Satz 2.18
Sei $a \in \mathbb{R}^m$ und $u := a \pm \|a\|_2 e_k \in \mathbb{R}^m$ $(1 \le k \le m)$. Dann gilt

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

Beweis: Im Fall $u = 0$ gilt aufgrund der Definition von u $a = \mp\|a\|_2 e_k$. Also folgt $H(u)a = \mathbb{I}a = \mp\|a\|_2 e_k$.

Sei also $u \neq 0$, etwa $u = a - \|a\|_2 e_k$. Dann folgt

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

Und weiter

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

$$\begin{aligned} H(u)a &= \left(\mathbb{I} - \tfrac{1}{h}uu^\top\right)a = a - \tfrac{1}{h}(uu^\top)a \\ &= a - \tfrac{1}{h}\langle u, a\rangle u \qquad \text{(Folgerung 2.15(i))} \\ &= a - \tfrac{1}{h}\langle a - \|a\|_2 e_k, a\rangle u \\ &= a - \tfrac{1}{h}\left(\|a\|_2^2 - \|a\|_2\langle e_k, a\rangle\right)u \\ &= a - u = \|a\|_2 e_k \qquad \text{(Definition von } u). \quad \square \end{aligned}$$

Verfahren: QR Zerlegung

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

Schritt 1:

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

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

Schritt 2:

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

Dann ist $Q_2 \in \mathbb{R}^{m \times m}$, $H(u^{(1)}) \in \mathbb{R}^{m-1 \times m-1}$ und es folgt

$$Q_2 Q_1 A = Q_2 R^{(1)} = \begin{pmatrix} * & \cdots & \cdots & 0 \\ 0 & * & \cdots & * \\ \vdots & 0 & & \\ \vdots & \vdots & A^{(2)} & \\ 0 & 0 & & \end{pmatrix}.$$

Iterativ erhalten wir so nach n Schritten:

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

und $Q := Q_1 \cdots Q_n$ ist orthogonal, da alle Q_i orthogonal sind. Ausserdem gilt $A = QR$, da $Q_i^2 = \mathbb{I}$ und somit $QQ_n \cdots Q_1 = \mathbb{I}$.

Wir fassen die bisherigen Ergebnisse zusammen:

$$\begin{array}{lll} \text{Ausgleichsproblem} & \Longleftrightarrow & \text{Normalengleichung} \\ & \Longleftrightarrow & \tilde{R}\overline{x} = \widetilde{c} \text{ mit } A = QR = \left(\dfrac{\widetilde{R}}{0} \right), \ R \\ & & \text{und } R \text{ ist reguläre obere } \triangle\text{-Matrix,} \\ & & \text{falls } \mathrm{Rang}(A) = n \text{ gilt.} \end{array}$$

Betrachten wir nun die Kondition der Matrix R. Falls $A \in \mathbb{R}^{n \times n}$ und $\mathrm{Rang}(A) = n$ ist, gilt

$$\begin{array}{rcl} cond_2(A) & = & \|A\|_2 \left\| A^{-1} \right\|_2 \\ & = & \|QR\|_2 \left\| R^{-1} Q^{-1} \right\|_2 \\ & = & \|QR\|_2 \left\| R^{-1} Q^{\top} \right\|_2 \\ & = & \|R\|_2 \cdot \left\| R^{-1} \right\|_2 \end{array}.$$

Also folgt $\Longrightarrow$ $cond_2(A) = cond_2(R)$.

2.2.2 Singulärwertzerlegung einer Matrix

Die QR-Zerlegung liefert eine Möglichkeit, (AGP) numerisch zu lösen, falls $\mathrm{Rang}(A) = n$ ist. Für Probleme mit $\mathrm{Rang}(A) \leq n$ betrachten wir nun die Singulärwertzerlegung einer Matrix.

Satz 2.19 (Singulärwertzerlegung)

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

$$\Sigma = \text{diag}(\sigma_1,\dots,\sigma_p),$$

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

D. h. Σ hat die Form

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

mit $\text{diag}(\sigma_1,\dots,\sigma_r) \in \mathbb{R}^{r\times r}$.

Die Werte $\sigma_1,\dots,\sigma_r$ heißen **singuläre Werte** von A. Sie entsprechen gerade den Wurzeln aus den Eigenwerten von $A^\top A$ bzw. $AA^\top$ (Bem: nach 2.12(v) haben $A^\top A$ und $AA^\top$ dieselben positven und reellen Eigenwerte).

Beweis: Eindeutigkeit: Seien $U^\top AV = \Sigma$ und U, V orthogonal, dann gelten $Av_i = \sigma_i u_i$, da $AV = U\Sigma$ und $A^\top u_i = \sigma_i v_i$, da $A^\top U = V\Sigma$. Daraus ergibt sich

$$A^\top A v_i = \sigma_i A^\top u_i = \sigma_i^2 v_i \implies \sigma_i^2 \text{ ist Eigenwert von } A^\top A.$$

Analog folgt

$$AA^\top u_i = \sigma_i^2 u_i \implies \sigma_i^2 \text{ ist Eigenwert von } AA^\top.$$

Da nach 2.12(v) die Eigenwerte von $A^\top A$ und $AA^\top$ übereinstimmen, folgt die Eindeutigkeit.

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

Dann existieren $x_1 \in \mathbb{R}^n, y_1 \in \mathbb{R}^m$ mit $\|x_1\|_2 = 1$, $\|y_1\|_2 = 1$ und $Ax_1 = \sigma_1 y_1$. Sei $V = (x_1,\dots,x_n) \in \mathbb{R}^{n\times n}$ eine orthonormale Basis (ONB) des $\mathbb{R}^n$ und $U_1 = (y_1,\dots,y_m) \in \mathbb{R}^{m\times m}$ eine ONB des $\mathbb{R}^m$.

Dann folgt:

$$A_1 := U_1^\top A V_1 = \left(\begin{array}{c|c} \sigma_1 & w^\top \\ \hline 0 & B \end{array}\right),\ B \in \mathbb{R}^{m-1\times n-1},\ w \in \mathbb{R}^{n-1}.$$

Da U_1, V_1 orthogonal, gilt:

$$\|A_1\|_2 = \|A\|_2 = \sigma_1.$$

Desweiteren gilt

$$A_1 \begin{pmatrix} \sigma_1 \\ w \end{pmatrix} = \begin{pmatrix} \sigma_1^2 + w^\top w \\ Bw \end{pmatrix} = \begin{pmatrix} \sigma_1^2 + \|w\|_2^2 \\ Bw \end{pmatrix}$$

und daher

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

Insgesamt folgt also

$$\sigma_1^2 \geq \sigma_1^2 + \|w\|_2^2 \implies \|w\|_2^2 = 0 \implies w = 0$$

und wir erhalten

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

Die Aussage des Satzes folgt nun durch Induktion. □

Lösung des Ausgleichsproblems mit Singulärwertzerlegung:

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

$$\begin{array}{rcl} \|Ax - b\|_2^2 & = & \langle Ax - b, Ax - b\rangle \overset{\text{da } U^\top = \text{orth.}}{=} \langle U^\top(Ax - b), U^\top(Ax - b)\rangle \\ & \overset{VV^\top = \mathbb{I}}{=} & \langle U^\top A V(V^\top x) - U^\top b, U^\top A V(V^\top x) - U^\top b\rangle \\ & = & \langle \Sigma V^\top x - U^\top b, \Sigma V^\top x - U^\top b\rangle \\ & = & \|\Sigma V^\top x - U^\top b\|_2^2 \\ & = & \sum_{i=1}^{r} \left(\sigma_i (V^\top x)_i - u_i^\top b\right)^2 + \sum_{i=r+1}^{m} \left(u_i^\top b\right)^2 \\ & \geq & \sum_{i=r+1}^{m} \left(u_i^\top b\right) \end{array}.$$

Folgerung 2.20
$x \in \mathbb{R}^n$ ist genau dann Lösung des Ausgleichsproblems, wenn

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

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

Also ist $\|x\|_2$ minimal, g.d.w. $\alpha_{r+1} = \ldots = \alpha_n = 0$ ist. D.h. die eindeutige Lösung des AGPs mit minimaler 2-Norm ist gegeben durch

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

2.2.3 Pseudoinverse einer Matrix

Ziel: Verallgemeinerung der Inversen einer regulären Matrix auf beliebige Matrizen $A \in \mathbb{R}^{m \times n}$ mit

Hilfe der Singulärwertzerlegung.

Definition 2.21 (Pseudoinverse)
Zu $A \in \mathbb{R}^{m\times n}$ mit $Rang(A) = r$ sei

$$\Sigma := U^\top AV = \mathrm{diag}(\sigma_1, \ldots, \sigma_{\min\{m,n\}})$$

eine Singulärwertzerlegung von A. Wir definieren die $(n \times m)$-Matrix Σ^+ durch

$$\Sigma^+ := \left(\begin{array}{ccc|c} 1/\sigma_1 & & & \\ & \ddots & & 0 \\ & & 1/\sigma_r & \\ \hline & 0 & & 0 \end{array}\right),$$

*dann heißt $A^+ := V\Sigma^+U^\top \in \mathbb{R}^{n\times m}$ **Pseudoinverse** oder Penrose Inverse von A.*

Bemerkung: Die singulären Werte einer Matrix sind eindeutig bestimmt, die orthogonalen Matrizen U und V jedoch nicht. Die Eindeutigkeit der Pseudoinversen muß also noch gezeigt werden.

Satz 2.22
Sei $A \in \mathbb{R}^{m\times n}$ mit $Rang(A) = r$ gegeben. Dann gilt

(i) Ist $A^+ \in \mathbb{R}^{n\times m}$ eine Pseudoinverse zu A, so ist
$AA^+ = (AA^+)^\top, \quad A^+A = (A^+A)^\top, \quad AA^+A = A, \quad A^+AA^+ = A^+.$

(ii) Durch die "Penrose Bedingung"
$AB = (AB)^\top, \quad BA = (BA)^\top, \quad ABA = A, \quad BAB = B \qquad (*)$
ist eine Matrix $B \in \mathbb{R}^{n\times m}$ eindeutig bestimmt.
Insbesondere ist die Pseudoinverse wohldefiniert.

(iii) Sind $m \geq n$, $b \in \mathbb{R}^m$ und x_A die eindeutige lösung des Ausgleichsproblems, so ist $x_A = A^+b$.

(iv) Ist $m = n$ und $Rang(A) = n$, so ist $A^+ = A^{-1}$.

(v) Ist $m \geq n$ und $Rang(A) = n$, so ist $A^+ = (A^\top A)^{-1}A^\top$.

(vi) Sind $\sigma_1 \geq \ldots \sigma_r$ die singulären Werte von A, so ist $||A||_2 = \sigma_1$ und $||A^+||_2 = \frac{1}{\sigma_r}$.

Beweis: (Übungsaufgabe)

Bemerkung: Für die Pseudoinverse gilt auch $(A^+)^+ = A$, sowie $(A^\top)^+ = (A^+)^\top$, jedoch gilt im Allgemeinen nicht $(AB)^+ = B^+A^+$.

Definition 2.23 (Kondition einer singulären Matrix)
Für eine beliebige Matrix $A \in \mathbb{R}^{m\times n}$ definieren wir die Kondition von A bzgl. der Spektralnorm durch

$$cond_2(A) = ||A||_2||A^+||_2 = \frac{\sigma_1}{\sigma_r}.$$

2.3 Iterative Verfahren

Wir wollen uns nun iterativen Verfahren zur Lösung linearer Gleichungssysteme zuwenden.

Idee: Formuliere $Ax = b$ äquivalent als Fixpunktgleichung, so dass die Kontraktionsbedingung (vgl. Satz 1.29 auf Seite 10) erfüllt ist.

Ansatz: Wir zerlegen $A = M - N$ mit einer regulären Matrix M (i.A. ist M vorgegeben und $N := M - A$). Wir erhalten:

$$\begin{aligned} Ax = b \quad &\iff \quad (M-N)x = b \quad \iff \quad Mx - Nx = b \quad \iff \quad Mx = Nx + b \\ &\iff \quad x = M^{-1}Nx + M^{-1}b. \end{aligned}$$

Wir definieren $T := M^{-1}N = \mathbb{I} - M^{-1}A$, $c := M^{-1}b$ sowie eine Abbildung $F : \mathbb{R}^n \to \mathbb{R}^n$ durch $F(x) = Tx + c$.

Dann gilt: x ist die Lösung von $Ax = b$, g.d.w. x ein Fixpunkt von F ist, d.h. wenn gilt $x = F(x) = Tx + c$.

Iterationsverfahren: Sei ein Startwert $x^0 \in \mathbb{R}^n$ gegeben, dann definieren wir die Iteration für $k \in \mathbb{N} \setminus \{0\}$ durch

$$x^{k+1} = F(x^k),$$

d.h. x^{k+1} wird berechnet durch folgende Schritte:

$$\begin{aligned} &1) \quad My^k &&= r^k, \quad \text{mit dem Residuum } r^k := b - Ax^k, \\ &2) \quad x^{k+1} &&= x^k + y^k. \end{aligned}$$

Bemerkung: Aus den Lösungsschritten erkennt man, dass solche Iterationsverfahren nur dann von Interesse sind, wenn die Defektgleichung $My^k = r^k$ leicht zu lösen ist. Dies ist z.B. der Fall, falls M eine Diagonalmatrix oder eine obere $\triangle$-Matrix ist.

Satz 2.24
Die Folge $(x^k)_{k\in\mathbb{N}}$ sei durch eine Fixpunktiteration zu $x = F(x) = Tx + c$ gegeben. Sei $\|\cdot\|$ eine Norm auf dem $\mathbb{R}^n$, so dass für die induzierte Matrixnorm gilt

$$q := \|T\| < 1.$$

Dann konvergiert x^k gegen ein x mit $Ax = b$ und es gelten die Fehlerabschätzungen

$$\left\|x - x^k\right\| \leq \frac{q^k}{1-q} \left\|x^1 - x^0\right\|,$$

bzw.

$$\left\|x - x^k\right\| \leq \frac{q}{1-q} \left\|x^k - x^{k-1}\right\|.$$

Beweis: Der Beweis folgt mit dem Banachschen Fixpunktsatz 1.29 da gilt

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

Aus $q < 1$ folgt also, dass F eine Kontraktion ist. □

Problem: Die Kontraktionsbedingung ist abhängig von der Norm, die Konvergenz nicht, da alle Normen auf $\mathbb{R}^n$ äquivalent sind.
Gesucht: Notwendige und hinreichende Bedingung für die Konvergenz.

Definition 2.25 (Spektralradius)
Für eine Matrix $B \in \mathbb{R}^{n\times n}$ definieren wir den **Spektralradius** $\rho(B)$ *durch*

$$\rho(B) := \max\{|\lambda| \mid \lambda \in \mathbb{C} \text{ ist Eigenwert von } B\}.$$

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

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

also eine Ähnlichkeitstransformation (Jordansche Normalform).

Lemma 2.26
Für $B \in \mathbb{R}^{n\times n}$ gilt

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

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

Beweis:

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

$$\begin{aligned} Bu = \lambda u \quad &\Longrightarrow \quad |\lambda|\,\|u\| = \|\lambda u\| = \|Bu\| \le \|B\|\,\|u\| \\ &\Longrightarrow \quad |\lambda| \le \|B\| \text{ für alle EWe} \quad \Longrightarrow \quad \rho(B) \le \|B\|. \end{aligned}$$

(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 $D_\delta = diag(\delta^0, \ldots, \delta^{n-1})$. Für $G \in \mathbb{C}^{n\times n}$ gilt dann $\left(D_\delta^{-1} G D_\delta\right) = g_{ij}\delta^{j-i}$. Also folgt

$$(UD_\delta)^{-1} B (UD_\delta) = D_\delta^{-1}\left(U^{-1}BU\right) D_\delta = \begin{pmatrix} \lambda_1 & \delta r_{12} & \delta^2 r_{13} & \cdots & \delta^{n-1} r_{1n} \\ 0 & \lambda_2 & \delta r_{23} & \cdots & \delta^{n-2} r_{2n} \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \delta r_{n-1,n} \\ 0 & \cdots & \cdots & \cdots & \lambda_n \end{pmatrix}.$$

Sei $\varepsilon > 0$ vorgegeben. Dann existiert ein $\delta > 0$ mit $\sum_{k=i+1}^{n} \delta^{k-i} |r_{ik}| \leq \varepsilon$ für alle $i \in \{1, \ldots, n-1\}$.

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:

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

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

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

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

Aus Lemma 2.26 folgt folgender Satz über die Konvergenz geometrischer Matrixfolgen.

Satz 2.27
Sei $T \in \mathbb{C}^{n \times n}$ eine reguläre Matrix. Dann sind äquivalent

(i) $T^\nu \longrightarrow 0$ für $\nu \longrightarrow \infty$.

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

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

(iv) Es existiert eine Norm auf $\mathbb{C}^n$, so dass für die induzierte Matrixnorm $\|T\| < 1$ gilt.

Beweis:
(i) $\Longrightarrow$ *(ii)*: Es gilt $\|T^\nu u\| \leq \|T^\nu\| \, \|u\| \longrightarrow 0$, da $T^\nu \longrightarrow 0$. Also folgt $T^\nu u \longrightarrow 0$ (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 $u \in \mathbb{C}^n \setminus \{0\}$ der zugehörige EV, dann gilt $T^\nu u = T^{\nu-1}(Tu) = T^{\nu-1}(\lambda u) = \lambda T^{\nu-1} u = \cdots = \lambda^\nu u$. Weiter folgt $\|T^\nu u\| = \|\lambda^\nu u\| = |\lambda|^\nu \, \|u\| \not\to 0$, da $|\lambda| \geq 1$. Dies ist ein Widerspruch zu *(ii)*.
(iii) $\Longrightarrow$ *(iv)*: Lemma 2.26.

(iv) $\Longrightarrow$ *(i)*: Es gilt $\|T^\nu\| \overset{\text{Submultipl.}}{\leq} \|T\|^\nu \longrightarrow 0$, da $\|T\| < 1$ $\square$

Folgerung 2.28
Das Iterationsverfahren konvergiert genau dann wenn $\rho(T) < 1$.

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

(a) $\rho(T) = \rho(\mathbb{I} - M^{-1}A)$ möglichst klein.

(b) Das Gleichungssystem $My^k = r^k$ muss leicht zu lösen sein.

Dies sind widersprüchliche Forderungen: Optimal für Bedingung (a) wäre $M = A \implies \rho(T) = 0$, aber dann ist die Bediengung (b) nicht erfüllt.
Auf der anderen Seite ist (b) erfüllt, falls M eine Diagonalmatrix ist. Dies führt uns zu folgendem Verfahren.

2.3.1 Gesamtschritt Verfahren (GSV)/ Jacobi Verfahren

Sei A regulär und $a_{ii} \neq 0$ für alle $i = 1, \ldots, n$. Setze

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

Dies führt zu folgender Iterationsvorschrift:

Sei ein Startvektor $x^0 \in \mathbb{R}^n$ gegeben.
Iteration:

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

Satz 2.29 (Hinreichende Bediengung für die Konvergenz des Jacobi-Verfahrens)
Falls entweder

(a) $\max_i \left(\sum_{k \neq i} \frac{|a_{ik}|}{|a_{ii}|}\right) < 1$ (**starkes Zeilensummenkriterium**)

oder

(b) $\max_i \left(\sum_{k \neq i} \frac{|a_{ki}|}{|a_{ii}|}\right) < 1$ (**starkes Spaltensummenkriterium**)

gilt, so konvergiert das Jacobi-Verfahren.

Falls (a) oder (b) erfüllt ist, heißt die Matrix A **stark diagonal dominant**, da in diesem Fall die Beträge der Diagonaleinträge größer sind als die Summe der Zeilen bzw. Spalten der Matrix.

Beweis: Gelte (a), so folgt

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

Gelte (b), so folgt

$$\rho(T) \le \|T\|_1 = \max_i \left(\sum_{k \neq i} \tfrac{|a_{ki}|}{|a_{ii}|} \right) < 1 \text{ (wegen (b)).} \quad \square$$

Die starke Diagonaldominanz ist nur eine hinreichende Bedingung. Betrachten wir folgendes Beispiel.

Beispiel 2.30
Bei der Diskretisierung $-\partial_{xx} u = f$ in §1.5 musste ein LGS $Ax = b$ gelöst werden mit

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

Für A gilt $\sum\limits_{k \neq i} \frac{|a_{ik}|}{|a_{ii}|} = 1$ für $i = 2, \ldots, n-1$ und für $i = 1, n$ gilt $\sum\limits_{k \neq i} \frac{|a_{ik}|}{|a_{ii}|} < 1$. Für dieses Beispiel ist also die starke Diagonaldominanz nicht erfüllt. Wir wollen nun eine schwächere Bedingung herleiten, die auch Matrizen eines solchen Typs mit beinhaltet.

Definition 2.31 (Zerlegbare Matrizen)
Eine Matrix $A = (a_{ik})$ heißt **zerlegbar**, *falls es eine Zerlegung von $N := \{1, \ldots, n\}$ in zwei Teilmengen $N_1, N_2 \subset N$ gibt mit $a_{ik} = 0 \ \forall (i,k) \in N_1 \times N_2$.*
(Zerlegung heißt: $N_1 \neq \emptyset$, $N_2 \neq \emptyset$, $N = N_1 \cup N_2$, $N_1 \cap N_2 = \emptyset$).

Lemma 2.32
Für $A \in \mathbb{R}^{n \times n}$ sind äquivalent

(i) A ist zerlegbar.

(ii) Der zugehörige gerichtete Graph
$$G(A) := \big(\text{Knoten } P_1, \ldots, P_n, \text{ gerichtete Kanten } \overline{P_j P_k} \iff a_{jk} \neq 0\big)$$
ist nicht zusammenhängend, d.h. es existieren Knoten P_j und P_k, so dass kein Pfad zwischen ihnen existiert. Es gibt also keine Folge $l_0, \ldots, l_L \in \{1, \ldots, N\}$ mit $l_0 = j, l_L = k$ und $a_{l_i l_{i+1}} \neq 0$ für alle $i = 0, \ldots, L$.

(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 $p + q = n$.

Beweis: Wir zeigen $(i) \iff (ii)$. Die Äquivalenz mit (iii) wird in den Übungen behandelt.

$(i) \implies (ii)$: Sei A zerlegbar, d.h. es existieren $N_1 \neq \emptyset$, $N_2 \neq \emptyset$, $N = N_1 \cup N_2$, $N_1 \cap N_2 = \emptyset$ und es gilt $a_{ik} = 0 \; \forall (i,k) \in N_1 \times N_2$. Sei $(i,k) \in N_1 \times N_2$. Annahme: $G(A)$ ist zusammenhängend. Dann existiert eine Folge $l_0, \ldots, l_L \in \{1, \ldots, N\}$ mit $l_0 = i$ und $l_L = k$ und $a_{l_i l_{i+1}} \neq 0$ für alle $i = 0, \ldots, L$. Nach Voraussetzung ist $l_0 = i \in N_1$. Da $a_{l_0 l_1} \neq 0$, ist $l_1 \notin N_2 \implies l_1 \in N_1$ und induktiv folgt somit $l_L = k \in N_1$. Dies ist ein Widerspruch zur Annahme $k \in N_2$.

$(ii) \implies (i)$: Sei nun $G(A)$ nicht zusammenhängend, existiere etwa kein Pfad zwischen P_j und P_k. Setze $N_1 := \{l |$ es existiert ein Pfad zwischen P_j und $P_l\} \cup \{j\}$, $N_2 := N \setminus N_2$. Dann gilt $k \in N_2$. Also folgt $N_1 \neq \emptyset$, $N_2 \neq \emptyset$, $N = N_1 \cup N_2$, $N_1 \cap N_2 = \emptyset$. Sei $(i,l) \in N_1 \times N_2$. Wäre $a_{il} \neq 0$, so würde ein Pfad von P_j zu P_l existieren und somit wäre $l \in N_1$. Dies ist ein Widerspruch und folglich gilt $a_{il} = 0$ für alle $(i,l) \in N_1 \times N_2$. □

Beispiel 2.33
Wir betrachten die Matrix

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

Dann ist der Graph $G(A)$ wie in Abb. 2.2 gegeben. Es gelten:

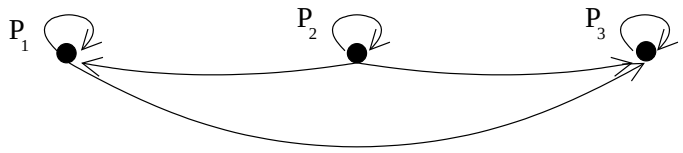


Abbildung 2.2: Graph, Beispiel 2.33

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

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

(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 2.34
Sei A eine Tridiagonalmatrix ohne Nullen auf der Diagonalen und den Nebendiagonalen, d.h.

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

Dann ist A unzerlegbar, da der Graph zusammenhängend ist. (Man kommt von jedem inneren Knoten zu den Nachbarn rechts und links. Siehe Übungsaufgabe)

Satz 2.35 (Schwaches Zeilensummenkriterium)
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

$$\sum_{k\neq r} \frac{|a_{rk}|}{|a_{rr}|} < 1.$$

Dann kann das Jacobi-Verfahren angewendet werden und es konvergiert für alle Startvektoren $x^0 \in \mathbb{R}^n$.

Beweis: Wir zeigen zunächst, dass gilt $|a_{ii}| > 0$. Dazu nehmen wir an, dass gilt $\sum_{k\neq i} |a_{ik}| \leq |a_{ii}|$. Da A unzerlegbar ist, folgt $\sum_{k\neq i} |a_{ik}| > 0$, und somit $|a_{ii}| > 0$. Insbesondere kann also das Jacobi-Verfahren angewendet werden.
Analog zum Beweis von Satz 2.29 können wir zeigen: $\rho(T) \leq 1$ mit $T = M^{-1}N$. Wir müssen also noch zeigen, dass $\rho(T) \neq 1$ ist.
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 $i = 1, \ldots, n$:

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

Also folgt

$$|v_i| = |\lambda|\,|v_i| \leq \frac{1}{|a_{ii}|} \sum_{k\neq i} |a_{ik}|\,|v_k|\,. \qquad (*)$$

Da $G(A)$ zusammenhängend ist, existiert ein Pfad zwischen P_s und P_r, d.h. $l_0 = s, \ldots, l_L = r$ und $a_{l_i, l_{i+1}} \neq 0$ $(i = 0, \ldots, L-1)$. Mit $(*)$ folgt also:

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

$$\begin{aligned}
|v_{l_{L-1}}| &\overset{*}{\leq} \frac{1}{|a_{l_{L-1}, l_{L-1}}|} \left(\sum_{k\neq l_{L-1}; k\neq l_L} |a_{l_{L-1},k}|\,|v_k| + |a_{l_{L-1},l_L}|\,|v_{l_L}| \right) \\
&< \frac{\|v\|_\infty}{|a_{l_{L-1}, l_{L-1}}|} \left(\sum_{k\neq l_{L-1}} |a_{l_{L-1},k}| \right) \leq \|v\|_\infty\,, \\
&\vdots \\
\|v\|_\infty = |v_s| &= |v_{l_0}| < \|v\|_\infty\,.
\end{aligned}$$

Dies ist ein Widerspruch und somit muss $\rho(T) < 1$ sein. Mit Folgerung 2.28 folgt dann die Behauptung. □

Als nächstes Iterationsverfahren wählen wir M als die untere Dreiecksmatrix von A. Wir erhalten das Einzelschritt Verfahren.

2.3.2 Einzelschritt Verfahren (ESV) / Gauß-Seidel-Verfahren

Sei A regulär mit $a_{ii} \neq 0$. Wir zerlegen A additiv in $A = L + D + R$ und setzen

$$M = L + D = \begin{pmatrix} a_{11} & & 0 \\ \vdots & \ddots & \\ a_{1n} & \cdots & a_{nn} \end{pmatrix}.$$

Da $a_{ii} \neq 0$ ist, ist M regulär und $N = M - A = -R$.
Dies führt zu folgender Iterationsvorschrift:

Sei ein Startvektor $x^0 \in \mathbb{R}^n$ gegeben.
Iteration (ESV):

$$\begin{aligned} x^{k+1} &= (L+D)^{-1}\left(b - Rx^{(k)}\right) \\ \Longrightarrow \quad x_i^{k+1} &= \tfrac{1}{a_{ii}}\left(b_i - \sum_{l=1}^{i-1} a_{il}x_l^{(k+1)} - \sum_{l=i+1}^{n} a_{il}x_l^{(k)}\right),\ i = 1,\dots,n. \end{aligned}$$

Vergleiche mit (GSV):

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

Satz 2.36
Die Matrix A erfülle das starkes Zeilensummenkriterium. Dann konvergiert das Einzelschrittverfahren.

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

Wir zeigen: $\|T\|_\infty \leq q < 1$ und folglich ist T Kontraktion.

Es ist $\|T\|_\infty = \sup_{\|x\|_\infty = 1} \|Tx\|_\infty$. Sei also $x \in \mathbb{R}^n$ mit $\|x\|_\infty = 1$. Setze $y := Tx$. Zu zeigen $\|y\|_\infty \leq q$, d.h. $|y_i| \leq q$ $(1 \leq i \leq n)$.

Induktion:
I.A. $y_1 = -\frac{1}{a_{11}}\left(\sum_{k=2}^{n} a_{1k}x_k\right)$

$$\Longrightarrow \quad |y_1| \leq \tfrac{1}{|a_{11}|}\sum_{k=2}^{n} |a_{1k}| \underbrace{|x_k|}_{\leq 1} \leq \tfrac{1}{|a_{11}|}\sum_{k\neq 1} |a_{1k}| \leq q.$$

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

$$\Longrightarrow \quad |y_i| \leq \tfrac{1}{|a_{ii}|}\left(\sum_{k=i+1}^{n} |a_{ik}| \underbrace{|x_k|}_{\leq 1} + \sum_{k=1}^{i-1} |a_{ik}| \underbrace{|y_k|}_{\leq q}\right) \leq q. \qquad \square$$

Wie für das Gesamtschrittverfahren erhalten wir auch hier folgenden Konvergenzsatz.

Satz 2.37 (Ohne Beweis)
Sei $A \in \mathbb{R}^{n\times n}$ unzerlegbar und erfülle das schwache Zeilensummenkriterium, dann konvergiert das Gauß-Seidel-Verfahren.

Darüberhinaus können wir für das Gauß-Seidel-Verfahren auch folgenden Satz zeigen, der für das Jacobi-Verfahren nicht gilt.

Satz 2.38
A sei symmetrisch und positiv definit, dann konvergiert das Gauß-Seidel-Verfahren.

Beweis: Sei $A = L + D + R$. Da A symmetrisch ist 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 $\mathbb{C}$. Sei $x \in \mathbb{C}^n$ der zugehörige Eigenvektor mit $\|x\|_2 = 1$. Dann gilt: $-(L + D)^{-1}Rx = \lambda x$, bzw. $-Rx = \lambda Dx + \lambda Lx = \lambda Dx + \lambda R^\top x$. Es folgt

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

Setze $\alpha := \langle Ax, x\rangle > 0$, da A positiv definit, $\sigma := \langle Rx, x\rangle = \sigma_1 + i\sigma_2 \in \mathbb{C}$, $\delta := \langle Dx, x\rangle$. Dann folgt

$$-\sigma = \lambda\alpha - \lambda\sigma = \lambda(\alpha - \sigma)$$

Hieraus folgt $\alpha - \sigma \neq 0$, da sonst $-\sigma = 0$ und somit $\alpha = \sigma = 0$ wäre. Dies ist ein Widerspruch zur positiven Definitheit. Also folgt

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

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

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

wobei $\mu > 0$ der kleinste Eigenwert von A ist.

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

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

da $\mu\underline{\delta} > 0$. Also folgt insgesamt $|\lambda| < 1$ und somit $\rho(T) < 1$. □

2.4 Zusammenfassung

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

$$Ax = b$$

kennengelernt und näher betrachtet,
(a) für $A \in \mathbb{R}^{n\times n}$ regulär

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

Verfahren:

(a) Direkte Verfahren oder Direktlöser (LR-, Cholesky, QR-Zerlegung)

(b) 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: 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:

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

  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 näherungsweise bestimmt und die Geschwindigkeit der Konvergenz hängt von $\rho(T)$ (Spektralradius) ab.

  Allerdings ist der zusätzliche Speicheraufwand sehr klein (das Gauß-Seidel-Verfahren hat gar kein zusätzlicher Speicheraufwand). Iterative Verfahren benötigen ein gutes Abbruchkriterium, z.B. $\left\|Ax^{(k)} - b\right\| < TOL$.

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.

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 einen Relaxationsparameter $w > 0$ und setzt

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

Dies führt auf das **SOR-Verfahren** [2].

Weitere Verfahren

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

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

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

[2] **S**uccesive **O**ver **R**elaxation.

Kapitel 3

Nichtlineare Gleichungen/ Nullstellensuche

In diesem Kapitel wollen wir uns mit der Berechnung von Nullstellen einer gegebenen Funktion

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

beschäftigen. Da man die Lösung linearer und nichtlinearer Gleichungen auch als Nullstellensuche einer geeigneten Funktion f auffassen kann, ist diese Fragestellung eine direkte Verallgemeinerung der Lösung linearer Gleichungsprobleme, die in Kapitel 2 behandelt wurde. Bei der Bestimmung von Nullstellen unterscheiden wir zwei Problemstellungen.

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

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

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.

3.1 Verfahren in einer Raumdimension

3.1.1 Intervallschachtelungsverfahren (ISV)

Voraussetzungen: Sei $f \in C^0(a, b)$ mit $a < b$ und $f(a)f(b) < 0$.

Verfahren: Setze $a_0 := a, b_0 := b$ und $x^{(0)} := \frac{1}{2}(a_0 + b_0)$.

Für $n = 0, \ldots, N$:

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

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

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

Setze $x^{(n+1)} := \frac{1}{2}(a_{n+1} + b_{n+1})$.

Satz 3.1 (Konvergenz von ISV)
Seien $(a_n)_{n\in\mathbb{N}}$, $(b_n)_{n\in\mathbb{N}}$, $\left(x^{(n)}\right)_{n\in\mathbb{N}}$ durch das Intervallschachtelungsverfahren definiert. Dann gilt

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

Es gilt die Fehlerabschätzung:

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

Beweis: Es gilt $(a_n)_{n\in\mathbb{N}}$ monoton wachsend und $(b_n)_{n\in\mathbb{N}}$ monoton fallend und $0 < b_n - a_n = 2^{-n}(b-a)$, $a_n < b, a < b_n$. Daraus folgt $(a_n)_{n\in\mathbb{N}}$, $(b_n)_{n\in\mathbb{N}}$ konvergieren und $\lim_{n\longrightarrow\infty} a_n = \lim_{n\longrightarrow\infty} b_n =: x^*$ Da $f \in C^0$ folgt weiter $f(x^*) = \lim_{n\longrightarrow\infty} f(a_n) = \lim_{n\longrightarrow\infty} f(b_n)$.

Nach Konstruktion gilt stets $f(a_n)f(b_n) < 0$, also folgt

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

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

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

Bemerkung: Das Verfahren ist sehr robust aber auch sehr langsam: Man benötigt etwa 3 Schritte, um eine Dezimalzahlstelle Genauigkeit zu gewinnen.

Aufwand: Eine Auswertung von f pro Schritt.

Das Verfahren wird in der Regel eingesetzt, um grob ein Intervall $[a, b]$ zu bestimmen, in dem eine Nullstelle liegt. Zur genaueren Berechnung der Nullstellen werden dann effizientere Verfahren eingesetzt.

3.1.2 Newton Verfahren

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

$$0 = f(x^*) = f(x^{(k)} + h) = f(x^{(k)}) + f'(x^{(k)})h + \mathcal{O}(h^2).$$

Unter Vernachlässigung der Terme höherer Ordnung folgt

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

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

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

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

Aufwand: Eine Auswertung von f und f' pro Schritt, d.h. es muß gelten $f \in C^1$ 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 $l(x) = ax + b$. Dann folgt aus der Taylorentwicklung $l(x) = f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)})$.

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

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

Dies ist gerade das Newton Verfahren.

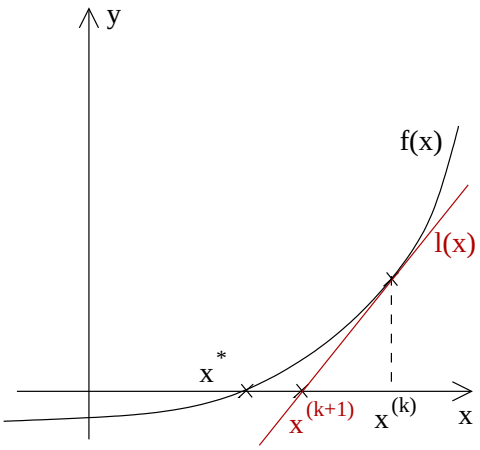


Abbildung 3.1: Newton Verfahren, Beispiel 1

Abbildung 3.1 veranschaulischt das Newton Verfahren. Mit dieser Anschauung sehen wir direkt ein, dass das Newton Verfahren nicht für alle Startwerte $x^{(0)}$ konvergieren wird. Abbildung 3.2 zeigt eine solche Situation. Hier gilt $\left|x^{(k)}\right| \longrightarrow \infty$. In Abbildung 3.3 gibt es zwei Nullstellen, aber es ist nicht klar, welche der beiden Nullstellen gefunden wird.

Satz 3.2 (Konvergenz des Newton-Verfahrens)
Sei $f \in C^2(a,b)$ und es es existiere ein $x^* \in (a,b)$ mit $f(x^*) = 0$. Sei $m := \min\limits_{a \le x \le b} |f'(x)| > 0$ und $M := \max\limits_{a \le x \le b} |f''(x)|$. Sei $\rho > 0$ so gewählt, dass $B_\rho(x^*) := \{x \mid |x - x^*| < \rho\} \subset [a,b]$ und $q := \frac{M}{2m}\rho < 1$. Dann konvergiert das Newton-Verfahren für jeden Startwert $x^{(0)} \in B_\rho(x^*)$. Es gilt die a-priori Fehlerschranke

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

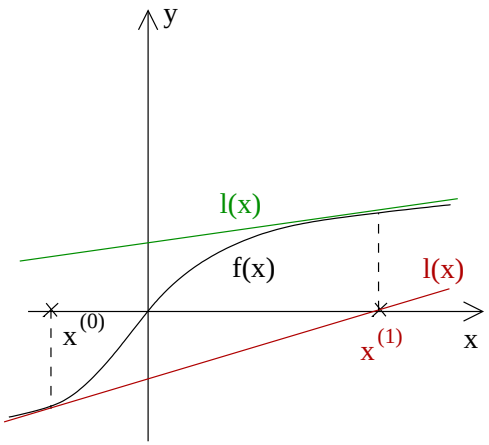


Abbildung 3.2: Newton Verfahren, Beispiel 2

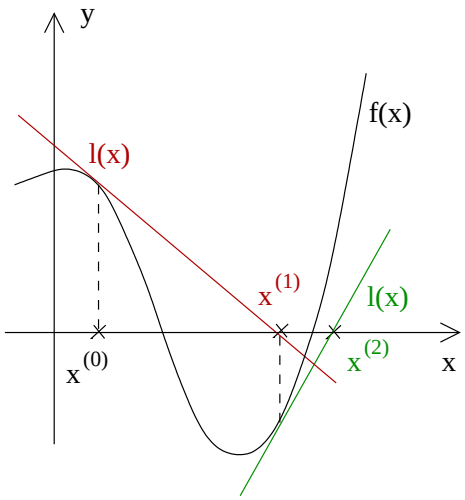


Abbildung 3.3: Newton Verfahren, Beispiel 3

und die a-posteriori Fehlerschranke

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

Bemerkung: Aus dem Mittelwertsatz folgt $\left|\frac{f(x)-f(y)}{x-y}\right| = |f'(\xi)| \geq m \;\forall\; x, y \in B_\rho(x^*);\; x \neq y \implies |x-y| \leq \frac{1}{m}|f(x)-f(y)|$. Folglich ist x^* ist die einzige Nullstelle in $B_\rho(x^*)$ und x^* ist **einfache** Nullstelle, d.h. $f(x^*) = 0$ und $f'(x^*) \neq 0$.

Beweis: Nach Taylorentwicklung (Satz 1.33) gilt

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

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

Also folgt für alle $x, y \in B_\rho(x^*)$

$$(2)\quad |R(y,x)| \le |y-x|^2 M \int_0^1 (1-s)\mathrm{d}s = \tfrac{M}{2}|y-x|^2$$

Für $x \in B_\rho(x^*)$ setze $\Phi(x) := x - \frac{f(x)}{f'(x)}$. Dann folgt

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

Also folgt für $x \in B_\rho(x^*)$

$$(3)\quad \begin{aligned} |\Phi(x)-x^*| &\le \tfrac{M}{2m}|x-x^*|^2, \\ &\le \tfrac{M}{2m}\rho^2 =: q\rho < \rho, \text{ da } q = \tfrac{M}{2m}\rho < 1. \end{aligned}$$

Insbesondere folgt $x^{(k)} \in B_\rho(x^*)$, falls $x^{(0)} \in B_\rho(x^*)$.

Sei $\rho^{(k)} := \frac{M}{2m}\left|x^{(k)}-x^*\right|$, so gilt wegen (3)

$$\begin{aligned} \rho^{(k)} &= \tfrac{M}{2m}\left|\Phi\left(x^{(k-1)}\right)-x^*\right| \le \tfrac{M}{2m}\left(\tfrac{M}{2m}\left|x^{(k-1)}-x^*\right|^2\right) \\ &= \left(\rho^{(k-1)}\right)^2 \le \ldots, \le \left(\rho^{(0)}\right)^{2^k} \\ \Longrightarrow \left|x^{(k)}-x^*\right| &\le \tfrac{2m}{M}\rho^{(k)} \le \tfrac{2m}{M}\left(\rho^{(0)}\right)^{2^k} = \tfrac{2m}{M}\left(\tfrac{M}{2m}\left|x^{(0)}-x^*\right|\right)^{2^k} \\ &\le \tfrac{2m}{M}\Big(\underbrace{\frac{M}{2m}\rho}_{=q}\Big)^{2^k} = \tfrac{2m}{M}q^{2^k}. \end{aligned}$$

Dies ist die a-priori Abschätzung. Da $q < 1$ ist, folgt $q^{(2^k)} \longrightarrow 0 \implies x^{(k)} \longrightarrow x^*$.
Für die a-posteriori Abschätzung benutzen wir nochmal (1) mit $y = x_{(k)}$ und $x = x^{(k-1)}$. Es folgt

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

und somit

$$\begin{aligned} \left|x^{(k)}-x^*\right| &\overset{MWS}{\le} \tfrac{1}{m}\left|f(x^{(k)})-f(x^*)\right| = \tfrac{1}{m}\left|f(x^{(k)})\right|, \\ &\overset{(2)}{=} \tfrac{1}{m}R(x^{(k)},x^{(k-1)}) \le \tfrac{M}{2m}\left|x^{(k)}-x^{(k-1)}\right|^2. \end{aligned}$$ □

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 $\left|x^{(10)}-x^*\right| \le \frac{2m}{M}q^{1024} \sim \frac{2m}{M}10^{-303}$.

Vergleichen wir dies mit dem Intervallschachtelungsverfahren, so folgt mit dem selben Startintervall: $|b-a| = \rho = q\frac{2m}{M} = \frac{m}{M}$, falls $q = \frac{1}{2}$ gilt. Nach 10 Schritten gilt also
$\left|x^{(10)}-x^*\right| = 2^{-11}|b-a| = 2^{-11}\frac{m}{M} \sim 10^{-4}\frac{2m}{M}$.

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: Übungsaufgabe.

Offene Fragen:

(a) Wie kann ρ effektiv bestimmt werden?

(b) Kann die Berechnung von f' umgangen werden?

(c) Was passiert, falls $f'(x^*) = 0$ ist, d.h falls x^* mehrfache Nullstelle ist?

(d) Gibt es noch schnellere Verfahren?

Kombination von Newton Verfahren und Intervallschachtelung

Idee: ISV einsetzen, um ausreichend nahe an eine Nullstelle x^* zu kommen, so dass das Newton Verfahren schnell konvergiert.

Verfahren: Seien $a < b$ 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` $|f_0| > TOL$:
$\quad$ `Falls` $f_a f_0 < 0$, `dann` $\widetilde{b} := x$, `sonst` $(\widetilde{a} := x;\ f_a := f_0)$
$\quad x := x - \frac{f_0}{f'(x)}$
$\quad f_1 := f(x)$
$\quad$ `Falls` $(|f_1| > |f_0|$ `oder` $x \notin (a,b))$, `dann`
$\qquad\qquad \left[a := \widetilde{a},\ b := \widetilde{b},\ x := \frac{1}{2}(a+b),\ f_1 := f(x)\right]$
$\quad f_0 := f_1$

3.1.3 Sekantenverfahren

Ein Nachteil des Newton-Verfahrens ist die Auswertung von $f'(x^{(k)})$.

Idee: Ersetze die Ableitung durch einen Differenzenquotient

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

Sekantenverfahren:

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

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 $y = y(x)$ die Gerade durch die Punkte $\left(x^{(k-1)}, f(x^{(k-1)})\right)$, $\left(x^{(k)}, f(x^{(k)})\right)$ ist, d.h.

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

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

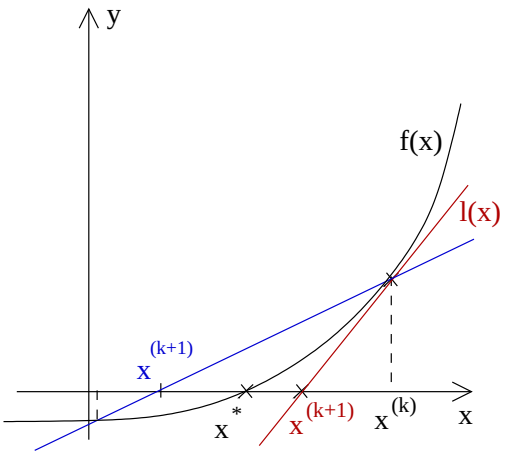


Abbildung 3.4: Sekantenverfahren, geometrische Interpretation

Bemerkung: Das Verfahren erfordert 2 Startwerte $x^{(0)}$, $x^{(1)}$ (siehe Abb. 3.4). Das Verfahren erfordert eine 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 $\rho > 0$. Seien $x^{(0)}$, $x^{(1)} \in B_\rho(x^*)$, $x^{(0)} \neq x^{(1)}$
und $\left(x^{(k)}\right)_{k\in\mathbb{N}}$ die Folge definiert durch das Sekantenverfahren.
Dann gilt $x^{(k)} \in B_\rho(x^*)$ und $x^{(k)} \longrightarrow x^*$ für $k \longrightarrow \infty$. Es gelten folgende Fehlerabschätzungen.

(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 ist, d.h. $\gamma_0 = \gamma_1 = 1$; $\gamma_{k+1} = \gamma_k + \gamma_{k-1}$.

(b) A-posteriori Fehlerschranke:

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

Beweis: Übungsaufgabe.

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 $\left|x^{(k)} - x^*\right| \le \frac{2m}{M}\widetilde{q}^{\alpha^k}$ mit $\widetilde{q} < 1$ und $\alpha = \frac{1}{2}(1 + \sqrt{5}) \approx 1.618$.

Beweis: Zunächst zeigt man analog zu Satz 3.3, dass es ein $\rho > 0$ gibt, so dass die Voraussetzungen von Satz 3.4 auf $U = B_\rho(x^*)$ erfüllt sind. Mit der A-priori-Abschätzung von Satz 3.4 ist noch zu zeigen, dass gilt $q^{\gamma_k} \leq \widetilde{q}^{\alpha^k}$ mit $\widetilde{q} < 1$ geeignet gewählt.

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

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

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

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

für $c_1, c_2 \in \mathbb{R}$.

Mit $\gamma_0 = \gamma_1 = 1$ folgt $c_1 = \frac{\lambda_1}{\sqrt{5}}$, $c_2 = -\frac{\lambda_2}{\sqrt{5}}$. Also $\gamma_k = \frac{1}{\sqrt{5}}\left(\lambda_1^{k+1} - \lambda_2^{k+1}\right)$. Aus $|\lambda_2| < |\lambda_1|$ folgt $\gamma_k \geq \frac{\lambda_1}{2\sqrt{5}}\lambda_1^k$ und somit

$$\begin{aligned} q^{\gamma_k} &\leq q^{\frac{\lambda_1}{2\sqrt{5}}\left(\lambda_1^k\right)}, \text{ da } q < 1 \\ &= \widetilde{q}^{\left(\alpha^k\right)} \end{aligned}$$

mit $\widetilde{q} := q^{\frac{\lambda_1}{2\sqrt{5}}} < 1$ und $\alpha = \lambda_1$. □

3.1.4 Zusammenfassung

Für die betrachteten Verfahren gilt:

ISV: $\left|x^{(k)} - x^*\right| \leq \frac{(b-a)}{2}\left(\frac{1}{2}\right)^k$, eine Auswertung von f pro Schritt.

Newton: $\left|x^{(k)} - x^*\right| \leq \frac{2m}{M}q^{2^k}$, je eine Auswertung von f und f' pro Schritt.

Sekanten: $\left|x^{(k)} - x^*\right| \leq \frac{2m}{M}q^{1.618^k}$, eine Auswertung von f pro Schritt.

Annahme: Die Auswertungen von f und f' seien gleich aufwändig. Dann hat das Newton-Verfahren pro Schritt den doppelten Aufwand.

Vergleich der Verfahren:
Definiere $z^{(k)} := x^{2k}$ beim ISV und Sekanten-Verfahren. Dann ist auch hier der Aufwand in einem Schritt durch zwei Auswertung von f gegeben. Es gilt:

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

Sekanten: $\left|z^{(k)} - x^*\right| \leq \frac{M}{2m}\widetilde{q}^{\left(\lambda_1^{2k}\right)} = \frac{M}{2m}\widetilde{q}^{\left(2.618^k\right)}$.

Bei gleichen Aufwand konvergiert das Sekanten-Verfahren also schneller als das Newton- oder IS-Verfahren.

Problem beim Sekanten-Verfahren

Die Fehleranalyse wurde 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 gleiches Vorzeichen. Da die Differenz in diesem Fall schlecht konditioniert ist, kann es hier zu Auslöschungen kommen.

3.2 Konvergenzordnung von Iterationsverfahren

Wir betrachten allgemeine Iterationsverfahren auf einem Banach-Raum X der Form

$$x^{(k+1)} = \Phi\left(x^{(k)}, \ldots, x^{(k-j)}\right), \; k \geq j \geq 0 \qquad (1)$$

mit Startwerten $x^{(0)}, \ldots, x^{(j)}$ und Iterationsfunktion $\Phi : X^j \longrightarrow X$.

Definition 3.6 Lokale Konvergenz
Das Iterationsverfahren (1) **konvergiert lokal** *gegen ein $x^* \in X$, falls es eine Umgebung U von x^* gibt, so dass für alle $x^{(0)}, \ldots, x^{(j)} \in U$ die Folge $(x^{(k)})_{k\in\mathbb{N}}$ gegen x^* konvergiert.*

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*

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

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

Die ***Ordnung ist genau*** **p**, *falls $c \neq 0$.*
Der Fall $p = 1$ heißt **lineare Konvergenz** *und falls für ein $p \geq 1$ gilt*

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

so spricht man von **superlinearer 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 (Konvergenzordnung von Iterationsverfahren)
Sei $I = [a, b]$, $\Phi : I \longrightarrow I$, $\Phi \in C^p(I)$. Sei $x^{(0)} \in I$, $x^{(k+1)} = \Phi(x^{(k)})$, $k \geq 0$ und es gelte $x^{(k)} \longrightarrow x^*$ mit $x^* = \Phi(x^*)$. Dann gilt:

(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, g.d.w.

$$\Phi^{(\nu)}(x^*) = 0, \; (1 \leq \nu \leq p - 1).$$

Beweis: (i) Es existiert eine Zwischenstelle ξ_k zwischen x^k und x^*, so dass gilt

$$\lim_{k\to\infty} \frac{\left\|\Phi(x^{(k)}) - x^*\right\|}{\left\|x^{(k)} - x^*\right\|} = \lim_{k\to\infty} \left|\Phi'(\xi_k)\right| = \left|\Phi'(x^*)\right|.$$

Also folgt die Behauptung.

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

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

mit ξ_k zwischen $x^{(k)}$ und x^*. Mit der Annahme folgt

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

mit $a_\nu := \frac{j!\Phi^{(\nu)}(x^*)}{\nu!\Phi^{(j)}(x^*)}$. Für große k gilt $|b| \geq \frac{1}{2}$, da $x^{(k)} \longrightarrow x^*$ und a_ν unabhängig von k beschränkt. Wir erhalten also

$$\left|x^{(k+1)} - x^*\right| \geq \frac{1}{2} \left|x^{(k)} - x^*\right|^j \frac{\left|\Phi^{(j)}(x^*)\right|}{j!}.$$

Da $x^{(k)} \longrightarrow x^*$ mit Ordnung p, gilt

$$\infty > c \geq \frac{\left|x^{(k+1)} - x^*\right|}{\left|x^{(k)} - x^*\right|^p} \geq \frac{1}{2j!} \left|\Phi^{(j)}(x^*)\right| \cdot \left|x^{(k)} - x^*\right|^{j-p} \longrightarrow \infty.$$

Dies ist ein Widerspruch, da nach Annahme $j - p < 0$.

"$\Longleftarrow$": Es gilt

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

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

3.2.1 Verfahren höher Ordnung ($p = 3$)

1. Idee: Konstruiere Verfahren 3. Ordnung aus einer Linearkombination von 2 Verfahren zweiter Ordnung. Seien $\Phi_0,\ \Phi_1$ Iterationsverfahren, die quadratisch konvergieren. Betrachte

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

Dann gilt $\Phi_s'(x^*) = (1 - s)\Phi_0'(x^*) + s\Phi_1'(x^*) = 0$, da nach Satz 3.7 $\Phi_0'(x^*) = \Phi_1'(x^*) = 0$.

Bestimmt man s so, dass $\Phi_s''(x^*) = 0$ gilt, so konvergiert nach Satz 3.7 Φ_s mindestens mit Ordnung 3. (Beispiel: siehe Übungsblatt).

2. Idee: (Verbessertes Newton-Verfahren)

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

Idee: Bestimme $g,\ h$, so dass $\Phi'(x^*) = \Phi''(x^*) = 0$ für eine einfache Nullstelle x^*.

Sei x^* einfache Nullstelle, so gilt

$$\Phi'(x) = 1 - f'(x)g(x) - f(x)g'(x) - 2f(x)f'(x)h(x) - h'(x)f^2(x).$$

Also folgt $\Phi'(x^*) = 0 \overset{f(x^*)=0}{\iff} 1 - f'(x^*)g(x^*) = 0 \implies g(x) = \frac{1}{f'(x)}$.

Analog folgt für Φ'' mit dieser Wahl von g:

$$\begin{aligned}\Phi''(x) &= -f'(x)\left(g'(x) + h(x)f(x) + 2h(x)f'(x)\right) - f(x)(\cdots) - \underbrace{\Big(\underbrace{g(x)f'(x)}_{=1}\Big)'}_{=0} \\ &= \frac{f''(x)}{f'(x)} - 2h(x)f'(x) - f(x)(\cdots).\end{aligned}$$

Aus

$$0 = \Phi''(x^*) = \frac{f''(x^*)}{f'(x^*)} - 2h(x^*)f'(x^*)^2$$

folgt also für die Wahl von h:

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

Folgerung 3.9 (Verbessertes Newton-Verfahren)
Das Verfahren, definiert durch

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

konvergiert in der Umgebung einer einfachen Nullstelle x^* von f mit mindestens dritter Ordnung.

3.2.2 Newton-Verfahren für mehrfache Nullstellen

Definition 3.10 (Ordnung und Vielfachheit einer Nullstelle)
Sei $f \in C^n(I)$ $(n \geq 1)$. f hat eine **Nullstelle x^* der Ordnung $(\mathbf{n-1})$** *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 n-fache Nullstelle von f mit $f^{(k)}(x) \neq 0$ $(x \neq x^*)$ $0 \leq k \leq n-1$ und $f^{(n)}(x) \neq 0$ für $x \in (a,b)$. Dann existiert eine Umgebung von x^*, so dass das Newton-Verfahren mindestens linear konvergiert.

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

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

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

Weiter folgt mit L'Hopital und $\Phi'(x) = 1 - \frac{f'(x)^2 - f(x)f''(x)}{f'(x)^2} = \frac{f(x)f''(x)}{f'(x)^2}$

$$\lim_{x\to x^*} \Phi'(x) = 1 - \frac{1}{n} \implies |\Phi'(x^*)| < 1. \qquad \square$$

Idee: Modifiziere das Verfahren je nach Vielfachheit:

Wähle $\Phi_\alpha(x) = x - \alpha\frac{f(x)}{f'(x)},\ \alpha \in \mathbb{R}$ fest

$$\implies \Phi'_\alpha(x^*) = 1 - \frac{\alpha}{n} = 0 \iff \alpha = n \qquad \square$$

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

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

quadratisch gegen x^*.

3.3 Nichtlineare 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: Es existiere ein $x^* \in D$ mit $f(x^*) = 0$.

3.3.1 Newton-Verfahren für nichtlineare Systeme

Idee: Iteriere analog zum skalaren Fall mit

$$x^{(k)} = x^{(k)} - Df(x^{(k)})^{-1} f(x^{(k)}).$$

Dabei ist $Df(x)$ die Jaccobi-Matrix von f und $Df(x)^{-1}$ die Inverse.

Algorithmus: (Newton-Verfahren für Systeme)

`Sei` $x^{(0)} \in \mathbb{R}^n$ `gegeben. Für` $k \geq 0$ `iteriere`

`1) Löse Defektgleichung:` $Df(x^{(k)})\ y^{(k)} = -f(x^{(k)})$

`2) Setze neu Iterierte:` $x^{(k+1)} = x^{(k)} + y^{(k)}$

Problem: In jedem Schritt muss ein $n \times n$ LGS gelöst werden. Häufig wird Df auch für einige Schritte festgehalten. Mit der LR-Zerlegung können diese Schritte dann sehr effizient gelöst werden.

Satz 3.13
Sei $f_i \in C^2(\mathbb{R})$ und $Df(x^*)$ regulär (d.h. x^* einfache Nullstelle).
Dann existiert ein $\rho > 0$, so dass für alle $x^{(0)} \in B_\rho(x^*) := \{x \mid \|x - x^*\| < \rho\}$ gilt: $x^{(k)} \in B_\rho(x^*)$ und $x^{(k)} \overset{k\to\infty}{\longrightarrow} x^*$ mindestens quadratisch.

Beweis: Analog zum Satz 3.2

Kapitel 4

Interpolation

Sei $\{\Phi(x, a_0, \ldots, a_n) \mid a_0, \ldots, a_n \in \mathbb{R}\}$ eine Familie von Funktionen mit $x \in \mathbb{R}$, 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:(Polynominterpolation)
$V \subset C^0(\mathbb{R})$ Vektorraum und $\dim(V) = n+1,\ \varphi_0, \ldots, \varphi_n$ Basis von V.
Setze $\Phi(x, a_0, \ldots, a_n) := \sum_{k=0}^{n} a_k \varphi_k(x)$.
Sei 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 $p(x_k) = f_k$.

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: Hermite-Interpolation

Es werden nicht nur Funktionswerte f_k an den Stützstellen x_k vorgeschrieben, sondern auch Werte für die Ableitung von Φ.

Beispiel: $p(x) = \sum_{k=0}^{N} a_k x^k$, $N = 2(n+1)-1$ mit $p(x_k) = f_k,\ p'(x_k) = d_k$ für gegebene $(x_k, f_k, d_k) \in \mathbb{R}^3$ $(k = 0, \ldots, n)$.

Spline-Interpolation

Sei $q \in \mathbb{N}$ fest.

Gesucht: $\Phi \in C^q$ mit $\Phi(x_k) = f_k$ und $\Phi_{|_{[x_k, x_{k+1}]}} \in \mathbb{P}_r$, d.h. Φ ist stücksweise polynomial.

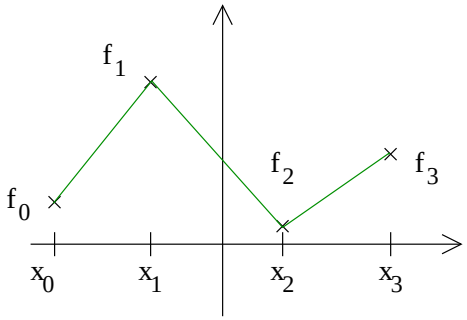


Abbildung 4.1: Spline-Interpolation

Abbildung 4.1 zeigt das Problem für $q = 0$ und $r = 1$, d.h. $\Phi \notin C^1$.

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: $(x_0, f_0) = (0, 0)$, $(x_1, f_1) = (1, 1)$

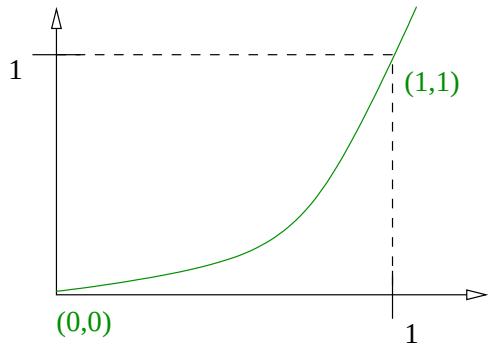


Abbildung 4.2: Polynominterpolation, Beispiel 1

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 $p(x_i) = f_i$ $(i = 0, \ldots, n)$.

Beweis: Sei $\varphi_0, \ldots, \varphi_n$ eine Basis von $\mathbb{P}_n$. Dann ist das Interpolationsproblem äquivalent dazu, einen Vektor $a = (a_0, \ldots, a_n)^\top$ 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 $f = (f_0, \ldots, f_n)^\top \in \mathbb{R}^{n+1}$

$$\begin{aligned} \text{Es gilt : } Aa = f \quad &\Longleftrightarrow \quad \sum_{k=0}^{n} \alpha_{ik} a_k = f \\ &\Longleftrightarrow \quad \sum_{k=0}^{n} a_k \varphi_k(x_i) = f_i \\ &\Longleftrightarrow \quad p(x_i) = f_i \text{ mit } p(x) = \sum_{k=0}^{n} a_k \varphi_k(x) \end{aligned}$$

Existenz und Eindeutigkeit: Es reicht zu zeigen, dass A regulär ist.

Sei also $a = (a_0, \ldots, a_n)^\top$ Lösung von $\sum_{k=0}^{n} a_k \varphi_k(x_i) = 0$ $(i = 0, \ldots, n)$. Dann hat $p(x) = \sum_{k=0}^{n} a_k \varphi_k(x) \in P_n$ die $(n+1)$-Nullstellen $x_0, \ldots, x_n$. Folglich muß $p \equiv 0$ sein, und somit $a_0 = \cdots = a_n = 0$.

Also gilt $Aa = 0 \implies a = 0 \implies A$ injektiv $\implies A$ regulär.
Also ist das Interpolationsproblem eindeutig lösbar $\square$

Bemerkung: Der Beweis von Satz 4.1 erlaubt es, Verfahren zur Lösung des Interpolationsproblems zu konstruieren. Dazu muss man eine Basis $\varphi_0, \ldots, \varphi_n$ von $\mathbb{P}_n$ wählen und das $(n+1) \times (n+1)$ LGS $Aa = f$ lösen.
Wählt man die Monombasis $\varphi_0(x) = 1, \varphi_1(x) = x, \varphi_2(x) = x^2, \ldots, \varphi_n(x) = x^n$ (also $\varphi_i(x) = x^i$), so 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.

Die Darstellung des Interpolationspolynoms $\sum_{k=0}^{n} a_k x^k$ heißt **Normalform**. Diese Darstellung wird in der Praxis nicht verwendet. Andere Darstellungen sind üblicher, auch wenn dadurch das Interpolationspolynom nicht verändert wird!

a) **Lagrange-Form des Interpolationsproblems**

Am einfachsten ist $Aa = f$ zu lösen, falls $A = \mathbb{I}$, d.h. $\varphi_k(x_i) = \delta_{ki}$ $(0 \le k, i \le n)$. Wir erhalten mit $\varphi_k(x_i) = 0$ für $k \neq i$ den Ansatz

$$\varphi_k(x) = c \prod_{\substack{i=0 \\ i\neq k}}^{n} (x - x_i).$$

Aus $\varphi_k(x_k) = 1$ folgt dann

$$c = \left(\prod_{\substack{i=0 \\ i\neq k}}^{n} (x_k - x_i) \right)^{-1}$$

und somit erhalten wir

$$\varphi_k(x) = \prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{(x - x_i)}{x_k - x_i}, \ (k = 0, \ldots, n).$$

Definition 4.2 (Lagrange-Polynome)
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**. *$(l_0^n, \ldots, l_n^n)$ bilden eine Basis von $\mathbb{P}_n$ und $p(x) = \sum_{k=0}^{n} f_k l_k^n(x)$ ist das Interpolationspolynom zu $(x_0, f_0), \ldots, (x_n, f_n)$.*

Für die Lagrange-Polynome gilt $l_i^n(x_j) = \delta_{ij}$.

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 Hinzunahme von weiteren Stützstellen.

b) **Newton-Form des Interpolationsproblems**

Wähle eine Basis von $\mathbb{P}_n$, so dass A eine untere $\triangle$-Matrix wird

$$\varphi_k(x) := \prod_{j=0}^{k-1}(x - x_j),\ (k = 0, \ldots, n) \implies \varphi_k \in \mathbb{P}_k.$$

$$\begin{aligned} \text{etwa} : \varphi_0(x) &= 1 \left(\text{verwende Konvention } \prod_{j=j_0}^{j_n} a_j = 1 \text{ falls } j_n < j_0 \right) \\ \varphi_1(x) &= (x - x_0) \\ \varphi_2(x) &= (x - x_0)(x - x_1) \\ &\vdots \end{aligned}$$

Damit ist A untere $\triangle$-Matrix, da

$$\varphi_k(x_i) = 0 \text{ für } i < k.$$

Definition 4.3 (Newton-Polynome)
Die Polynome

$$N_k^n := \prod_{j=0}^{k-1}(x - x_j)$$

heißen **Newton-Polynome** *und das Interpolationspolynom* $p(x) = \sum_{k=0}^{n} a_k N_k^n(x)$ *heißt in* **Newton-Form**.

$$\begin{aligned} \text{Es gilt} : a_0 &= \tfrac{f_0}{\varphi_0(x_0)} = f_0, \\ a_1 &= \tfrac{(f_1 - \varphi_0(x_1)a_0)}{\varphi_1(x_1)} = \tfrac{f_1 - f_0}{x_1 - x_0} =: f[x_0, x_1], \\ a_2 &= \tfrac{(f_2 - \varphi_0(x_2)a_0 - \varphi_1(x_2)a_1)}{\varphi_2(x_2)} \\ &= \tfrac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0} = \tfrac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} =: f[x_0, x_1, x_2] \end{aligned}$$

Diese Koeffizienten werden berechnet über die **dividierten Differenzen** $f[x_0, \ldots, x_n]$ *(siehe Abschnitt 4.3).*

4.2 Funktionsinterpolation durch Polynome

Gegeben: Stützstellen $x_0, \ldots, x_n$ und f stetig.

Gesucht: Interpolationspolynom zu $(x_0, f(x_0)), \ldots, (x_n, f(x_n))$.

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 $\lambda = \frac{f(x)-p(x)}{\omega(x)} \in \mathbb{R}$ (beachte x fest). Es folgt $\Phi(x) = 0$ und $\Phi(x_i) = 0$ $(i = 0, \ldots, n)$ und somit hat Φ $n+2$ Nullstellen. Nach dem Satz von Rolle folgt weiter: Φ' hat $n+1$ Nullstellen und folglich $\Phi^{(n+1)}$ eine Nullstelle $\xi_x \in (a,b)$ mit

$$0 = \Phi^{(n+1)}(\xi_x) = f^{(n+1)}(\xi_x) - (n+1)! \frac{f(x) - p(x)}{\omega(x)}$$

Also haben wir $(*)$ bewiesen. □

Folgerung 4.5
Seien die Vorraussetzungen von Satz 4.4 erfüllt.
Dann gilt $\|f - p\|_\infty \leq \frac{1}{(n+1)!} \left\|f^{(n+1)}\right\|_\infty \|\omega\|_\infty$ mit dem **Knotenpolynom** $\omega(x) = \prod_{j=0}^{n} (x - x_j)$.

Beweis: Folgt direkt aus Satz 4.4.

Bemerkung Folgerung 4.5 zeigt, dass die Approximation durch Polynominterpolation durch eine geeignete Wahl der Stützstellen optimiert werden kann. **Frage:** Wird der Interpolationsfehler bei wachsender Stützstellenzahl immer kleiner? Das folgende Beispiel zeigt, dass dies i.A. nicht der Fall ist.

Beispiel 4.6 (Runge)

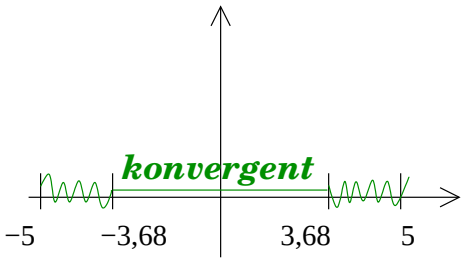


Abbildung 4.3: Interpolation von Funktionen, Beispiel 4.6

Betrachten wir $f(x) = \frac{1}{1+x^2}$, $-5 \leq x \leq 5$ und $x_n^{(n)} := -5 + kh_n$, $0 \leq k \leq n$ mit $h_n := \frac{10}{n}$ (gleichmäßige Stützstellenverteilung). Sei $p_n(x_k^{(n)}) = f(x_k^{(n)})$.
Man kann zeigen (siehe Abb. 4.3), dass es ein $\tilde{x} \approx 3,6$ gibt, so dass für $|x| < \tilde{x}$ Konvergenz vorliegt, während der Interpolationsfehler für $|x| > \tilde{x}$ gegen unendlich geht.

Allgemein gilt:

(a) Für jede stetige Funktion f existiert eine Folge von Stützstellen mit $p_n \longrightarrow f$ gleichmäßig.

(b) Zu jeder Folge von Stützstellen gibt es eine stetige Funktion f, so dass $p_n \not\longrightarrow f$ gleichmäßig.

Optimale Wahl von Stützstellen für das Referenzintervall $[-1, 1]$

Idee: Wähle Stützstellen $x_0, \ldots, x_n \in [-1, 1]$, so dass $\|w\|_{L^\infty([-1,1])}$ minimiert wird.

Bemerkung: Das Knotenpolynom $\omega(t) = \prod_{k=0}^{n} (t - x_k)$ ist ein normiertes Polynom $(n+1)$-tes Grades (d.h. Koeffizient 1 vor x^{n+1}) und die Stützstellen $x_0, \ldots, x_n$ sind die Nullstellen von ω. Wir erreichen also dann eine optimale Wahl der Stützstellen, wenn wir ein normiertes Polynom $(n+1)$-tes Grades finden, dass unter allen normierten Polynomen die ∞-Norm minimiert.

Definition 4.7 (Tschebyschev-Polynome)
Wir definieren die Tschebyschev-Polynome auf [-1,1] durch
$T_0(x) = 1,\ T_1(x) = x,\ T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x),\ \hat{T}_n(x) = 2^{1-n}T_n(x).$

D.h. $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3x$, $\hat{T}_2(x) = x^2 - \frac{1}{2}$, $\hat{T}_3(x) = x^3 - \frac{3}{4}x$.

Satz 4.8
Für $x \in [-1, 1]$ gilt:

$$T_n(x) = \cos\left(n \cos^{-1}(x)\right). \qquad (*)$$

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) $\hat{T}_n \in P_n(-1,1)$ mit Koeffizient 1 vor x^n (normiertes Polynom).

Beweis: Nach Additionstheorem gilt

$$\cos(A+B) = \cos(A)\cos(B) - \sin(A)\sin(B).$$

Also folgt

$$\cos((n+1)\Theta) = \cos(n\Theta)\cos(\Theta) - \sin(n\Theta)\sin(\Theta)$$

und

$$\cos((n-1)\Theta) = \cos(n\Theta)\cos(\Theta) + \sin(n\Theta)\sin(\Theta).$$

Zusammen erhalten wir

$$\cos((n+1)\Theta) + \cos((n-1)\Theta) = 2\cos(n\Theta)\cos(\Theta).$$

Setze $\Theta := \cos^{-1}(x)$, bzw. $x := \cos(\Theta)$, so folgt weiter

$$\cos\left((n+1)\cos^{-1}(x)\right) = 2\cos(n\cos^{-1}(x))x - \cos((n-1)\cos^{-1}(x))$$

d.h. $F_n := \cos\left(n\cos^{-1}(x)\right)$ genügt der Rekursionsformel von Definiton 4.7. Weiter ist $F_0(x) = 1$, $F_1(x) = \cos(\cos^{-1}(x)) = x$ und somit $F_n = T_n$, was $(*)$ beweist.

Die Eigenschaften (i), (ii), (iii) folgen direkt aus $(*)$; (iv) und (v) folgen aus der Rekursionsformel für T_n in Definition 4.7. □

Lemma 4.9
Sei $p \in \mathbb{P}_n$ ein normiertes Polynom auf $[-1,1]$. Dann gilt:

$$\|p\|_\infty = \max_{-1\le x\le 1} |p(x)| \ge 2^{1-n} \quad \text{und} \quad \left\|\hat{T}_n\right\|_\infty = 2^{1-n}.$$

Beweis: Annahme: Es gibt ein normiertes Polynom p mit $|p(x)| < 2^{1-n} \quad \forall\, x \in [-1,1]$.

Sei $x_i = \cos\left(\frac{i\pi}{n}\right)$. Nach Satz 4.8 (ii) folgt dann

$$(-1)^i p(x_i) \le |p(x_i)| < 2^{1-n} = (-1)^i \hat{T}_n(x_i)$$

und hieraus

$$(-1)^i \left(\hat{T}(x_i) - p(x_i)\right) > 0 \quad \text{für} \quad 0 \le i \le n.$$

D.h. das Polynom $\hat{T}_n - p$ wechselt $(n+1)$-Mal das Vorzeichen im Intervall $[-1,1]$. Da $\hat{T}_n$ und p normierte Polynome sind, ist dies ein Widerspruch dazu, dass $\hat{T}_n - p$ vom Grad $n-1$ ist.
Weiter folgt aus $|T_n(x)| \le 1$, dass $|\tilde{T}_n(x)| \le 2^{1-n}$ und mit $\tilde{T}_n(x_i) = 2^{1-n}(-1)^i$ folgt die Behauptung.

Folgerung 4.10 (Optimale Wahl der Stützstellen)
Mit den Stützstellen $x_k = \cos\left(\pi\frac{2k-1}{2(n+1)}\right), \quad k = 1,\dots,n+1$ als die Nullstellen von T_{n+1} gilt, dass das Knotenpolynom gerade $\hat{T}_{n+1}$ ist, d.h. die Maximumsnorm des Knotenpolynoms ist $2^{1-(n+1)}$.

4.3 Dividierte Differenzen

Wiederholung: Die Newton-Verfahren des Interpolationspolynom ist gegeben durch $p(x) = \sum_{k=0}^{n} a_k N_k(x)$ 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 effizienten Berechnung von $a_0, \ldots, a_n$.

Bemerkung: Setze $p_m(x) = \sum_{k=0}^{m} a_k N_k(x)$ für $m \leq n$, dann gilt $p_m(x_j) = f_j \quad (0 \leq j \leq m)$ und $p_m \in \mathbb{P}_m$, da $N_k \in \mathbb{P}_m$ für $0 \leq k \leq m$. D.h. p_m ist **das** Interpolationspolynom in $\mathbb{P}_m$ zu den Daten $(x_0, f_0), \ldots, (x_m, f_m)$. Insbesondere hängt a_k nur ab von $(x_0, f_0), \ldots, (x_k, f_k)$ für $0 \leq k \leq m$. Es wird daher die Schreibweise $f[x_0, \ldots, x_k]$ für a_k benutzt.
Beachte: a_m ist der Koeffizient vor dem x^m im Polynom p_m.

Definition 4.11 (Dividierte Differenzen)
Seien $i_o, \ldots, i_k \in \{0, \ldots, n\}$ 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

(i) Die Polynome $p_{i_0,\ldots,i_k}$ genügen der Rekursionsformel

(1) $\quad p_{i_0,\ldots,i_k}(x) = \frac{(x - x_{i_0}) p_{i_1,\ldots,i_k}(x) - (x - x_{i_k}) p_{i_0,\ldots,i_{k-1}}}{x_{i_k} - x_{i_0}}$.

(ii) Die dividierten Differenzen genügen der Rekursionsformel

(2) $\quad f[x_{i_0}, \ldots, x_{i_k}] = \frac{f[x_{i_1},\ldots,x_{i_k}] - f[x_{i_0},\ldots,x_{i_{k-1}}]}{x_{i_k} - x_{i_0}}$, $f[x_{i_l}] = f_{i_l}$.

(iii) 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.

x_0	$a_0 = f_0$	$a_1 = f[x_0, x_1]$	$a_2 = f[x_0, x_1, x_2]$	$a_3 = f[x_0, x_1, x_2, x_3]$
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			

Dabei ist z.B. $f[x_1, x_2, x_3] = \frac{f[x_2,x_3] - f[x_1,x_2]}{x_3 - x_1}$. 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
Wir betrachten die Daten

x	3	1	5
f	1	-3	2

Die dividierten Differenzen liefern:

$$\begin{array}{cc|l} 3 & 1 & 2 = \frac{-3-1}{1-3} \quad -\frac{3}{8} = \frac{5/4-2}{5-3} \\ 1 & -3 & \frac{5}{4} = \frac{2-(-3)}{5-1} \\ 5 & 2 & \end{array}$$

Das Interpolationspolynom ist also $p(x) = 1 + 2(x-3) - \frac{3}{8}(x-3)(x-1)$.

Fügen wir eine Stützstelle hinzu, so betrachten wir die Daten:

x	3	1	5	6
f	1	-3	2	4

Die dividierten Differenzen liefern durch hinzufuegen einer "Diagonalen":

$$\begin{array}{cc|lll} 3 & 1 & 2 & -\frac{3}{8} & \frac{7}{40} \\ 1 & -3 & \frac{5}{4} & \frac{3}{20} & \\ 5 & 2 & 2 & & \\ 6 & 4 & & & \end{array}$$

Das Interpolationspolynom lautet:
$p(x) = 1 + 2(x-3) - \frac{3}{8}(x-3)(x-1) + \frac{7}{40}(x-3)(x-1)(x-5)$

Beweis: (von Satz 4.12)

(i) Setze $R(x)$ als rechte Seite von (1). Zu zeigen: $p_{i_0,\dots,i_n} = R(x)$.

Notation:
$p_k = p_{i_0,\dots,i_k},\ p_{k-1} = p_{i_0,\dots,i_{k-1}}\ q_k = p_{i_1,\dots,i_k}$

Dann ist

$$R(x) = \frac{(x - x_{i_0})q_k(x) - (x - x_{i_k})p_{k-1}(x)}{x_{i_k} - x_{i_0}}$$

$$\begin{aligned} \Longrightarrow \quad R(x_{i_0}) &= \tfrac{0-(x_{i_0}-x_{i_k})f_{i_0}}{x_{i_k}-x_{i_0}} = f_{i_0}, \\ R(x_{i_k}) &= \tfrac{(x_{i_k}-x_{i_0})f_{i_k}-0}{x_{i_k}-x_{i_0}} = f_{i_k}, \\ R(x_{i_l}) &= \tfrac{(x_{i_l}-x_{i_k})f_{i_l}-(x_{i_l}-x_{i_k})f_{i_l}}{x_{i_k}-x_{i_0}} = f_{i_l} \quad \forall\ 0 < l < k. \end{aligned}$$

Also ist R ist das Interpolationspolynom zu $(x_{i_0}, f_{i_0}), \dots (x_{i_n}, f_{i_n})$. Aufgrund der Eindeutigkeit folgt dann $p_{i_0,\dots,i_k} = R$.

(ii) Aus (i) folgt, dass der Koeffizient vor x^k in $R(x)$ durch $\frac{f[x_{i_1},\dots,x_{i_k}]-f[x_{i_0},\dots,x_{i_{k-1}}]}{x_{i_k}-x_{i_0}}$ gegeben ist. Nach Definition ist dieser Koeffizient gleich $f[x_{i_0}, \dots, x_{i_k}]$, also folgt (ii). □

Satz 4.14 (Weitere Eigenschaften der dividierten Differenz)
Sei $f \in C^0(a,b),\ x_0, \dots, x_n \in (a,b)$ paarweise verschieden und t fest gewählt mit $t \neq x_k\ \forall\ k = 0, \dots, n$.

(i) Wenn p das Interpolationspolynom von f an den Stützstellen $x_0, \ldots, x_n$ ist, so gilt:

$$f(t) - p(t) = f[x_0, \ldots, x_n, t] \prod_{j=0}^{n} (t - x_j)$$

(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 $j = 1, \ldots, n$
  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 Hinzunahme einer weiteren Stützstelle (x_{n+1}, f_{n+1}):

$c_{n+1,0} := f_{n+1}$
Für $j = 1, \ldots, n+1$
  $c_{n+1-j,j} := \frac{c_{n+1-j+1,j-1} - c_{n+1-j,j-1}}{x_{n+1} - x_{n+1-j}}$

Satz 4.16 (Auswertung des Interpolationspolynoms)
1. Fall: Die Koeffizienten $a_0, \ldots, a_n$ seien bekannt. Dann kann folgendes Schema zur Auswertung benutzt werden (**Horner-Schema**):

$$\begin{aligned} p(x) &= \sum_{k=0}^{n} a_k \prod_{j=0}^{k-1} \underbrace{(x - x_j)}_{:=\chi_j} \\ &= (((\cdots((a_n \chi_{n-1} + a_{n-1})\,\chi_{n-2} + a_{n-2})\,\chi_{n-3}\cdots)\,\chi_1 + a_1)\,\chi_0 + a_0) \end{aligned}$$

Algorithmus:
$p := a_n$
Für $k = n-1, \ldots, 0$
  $p := p(x - x_k) + a_k$

2. Fall: Das Interpolationspolynom p soll nur an einer Stelle ausgerechnet werden ohne vorher die Koeffizienten zu berechnen (**Neville-Schema**):

Sei $p_{i_0,\ldots,i_k} \in P_n$ das Interpolationspolynom zu $(x_{i_0}, f_{i_0}), \ldots, (x_{i_k}, f_{i_k})$. Das Neville-Schema verwendet die Rekursion aus 4.12 (i):

$$\begin{array}{cc|ccc} x_0 & f_0 = p_0(x) & p_{0,1}(x) & \cdots & p_{0,1,\ldots,n}(x) \\ x_1 & f_1 = p_1(x) & p_{1,2}(x) & \cdots & \\ \vdots & \vdots & \vdots & & \\ x_n & f_n = p_n(x) & p_{n,n+1}(x) & & \end{array}$$

$p_{0,1,\ldots,n}(x)$ 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** folgt anschließend:

$$\begin{array}{rcl} p(0) & = & \left(\left(\left(\frac{7}{40}(0-5) - \frac{3}{8}\right)(-1) + 2\right)(-3) + 1\right) \\ & = & \left(\left(\frac{5}{4} + 2\right)(-3) + 1\right) \\ & = & \left(-\frac{39}{4} + 1\right) = -\frac{35}{4} \end{array}$$

(ii) Mit dem **Neville-Schema** erhält man

$$\begin{array}{cc|ccc} x_0 & f_0 & & & \\ 3 & 1 & \frac{(0-3)(-3)-(0-1)1}{1-3} = -5 & -\frac{79}{8} & -\frac{35}{4} \\ 1 & -3 & \frac{(0-1)2-(0-5)(-3)}{5-1} = -\frac{17}{4} & -\frac{7}{2} & \\ 5 & 2 & \frac{(0-5)4-(0-6)2}{6-5} = -8 & & \\ 6 & 4 & & & \end{array}$$

4.4 Hermite Interpolation

Gegeben: $x_0, \ldots, x_m$ paarweise verschieden und für jede Stützstelle x_i Werte $c_{ij} \in \mathbb{R}$ für $0 \leq j \leq m_i - 1$.

Gesucht: Ein Polynom p mit $p^{(j)}(x_i) = c_{ij}$.

Die Anzahl der Bedingungen ist $n+1 := m_0 + m_1 + \cdots + m_n$, d.h. es macht Sinn $p \in \mathbb{P}_n$ zu suchen.

Satz 4.18
Es existiert genau ein $p_n \in \mathbb{P}_n$, welches die Bedingungen des Hermite Interpolationspolynoms erfüllt.

Beweis: (analog zu Satz 4.1)

Satz 4.19 (Fehlerdarstellung für Hermite Interpolation)
Seien $f \in C^{n+1}(a,b)$ und $a \leq x_0 < \ldots < x_m \leq b$. Mit $m_0, \ldots, m_m \in \{1, \ldots, n+1\}$ und $n+1 = \sum_{j=0}^{n} m_j$

Sei $p_n \in \mathbb{P}_n$ das Hermite Interpolationspolynom zu den Daten

$$\begin{array}{cccc} (x_0, f(x_0)), & \ldots & , (x_0, f^{(m_0-1)}(x_0)) \\ (x_1, f(x_1)), & \ldots & , (x_1, f^{(m_1-1)}(x_1)) \\ \vdots & & \vdots \\ (x_m, f(x_m)), & \ldots & , (x_m, f^{(m_m-1)}(x_m)) \end{array}$$

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 $\Omega(x) := \prod_{k=0}^{n} (x - x_k)^{m_k}$.
Beweis: (analog zu Satz 4.4)

Beispiel 4.20 (Newton-Form 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:

$$\begin{array}{cc|cc} x_i & f_i & & \\ x_0 & c_{00} & f[x_0, x_0] & f_[x_0, x_0, x_1] \\ x_0 & c_{00} & f[x_0, x_1] & \\ x_1 & c_{10} & & \end{array}$$

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 folgt dann:

$$\begin{array}{cc|c|c} x_0 & c_{00} & c_{01} & \frac{f[x_0,x_1]-f[x_0,x_0]}{x_1-x_0} \\ x_0 & c_{00} & \frac{c_{10}-c_{00}}{(x_1-x_0)} & \\ x_1 & c_{10} & & \end{array}$$

Wir erhalten also im Beispiel $f[x_0, x_0, x_1] = \frac{c_{10}-c_{00}}{(x_1-x_0)^2} - \frac{c_{01}}{x_1-x_0}$ und setzten das Interpolationspolynom in der Newtonform an:

$$p(x) = f[x_0] + f[x_0, x_0](x - x_0) + f[x_0, x_0, x_1](x - x_0)^2.$$

Dieser Ansatz läßt sich verallgemeinern zu:

$$p_n(x) = \sum_{k=0}^{n} f[z_0, \ldots, z_k] \prod_{j=0}^{k-1} (x - z_j)$$

mit

$$\begin{aligned} z_0 = \quad & \cdots = z_{m_0-1} && = x_0, \\ z_{k_0} = \quad & \cdots = z_{m_0-m_1-1} && = x_1, \\ & \vdots \\ & \text{usw.} \end{aligned}$$

Satz 4.21 (Rekursionsformel für dividierte Differenzen)
Sei $i_0, \ldots, i_n \in \{0, \ldots, n\}$ und o.B.d.A $z_{i_0} \leq z_{i_1} \leq \cdots \leq z_{i_k}$. Dann gilt

$$f[z_{i_0}, \ldots, z_{i_k}] = \begin{cases} \frac{f[z_{i_1}, \ldots, z_{i_k}] - f[z_{i_0}, \ldots, z_{i_{k-1}}]}{z_{i_k} - z_{i_0}} & : \quad z_{i_k} \neq z_{i_0} \\ \\ \frac{1}{k!} f^{(k)}(z_{i_0}) & : \quad z_{i_k} = z_{i_0} \end{cases}$$

Bemerkung 4.22

(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 $j = 0, \ldots, m_k - 1$ die Werte $p^{(j)}(x_i)$ vorgeschrieben werden, sind nicht so einfach zu lösen (vergleiche Birkoff-Interpolation in den Übungsaufgaben).

4.5 Richardson Extrapolation

Gegeben: Eine Funktion $a : (0, \infty) \longrightarrow \mathbb{R}$.

Gesucht: $a(0) = \lim\limits_{h \searrow 0} a(h)$.

Idee: Wähle $h_0, \ldots, h_n$, setze $a_k = a(h_k)$ und bestimme das Interpolationspolynom zu $(h_0, a_0), \ldots, (h_n, a_n)$ und approximiere $a(0)$ durch $p(0)$.

Beispiel 4.23

(i) **Regel von L'Hospital**

Berechne $\lim\limits_{x \longrightarrow 0} \frac{\cos(x)-1}{\sin(x)}$, d.h. $a(h) = \frac{\cos(h)-1}{\sin(h)}$.

$$\begin{array}{lll} \text{Setze :} & h_0 = \frac{1}{8} & , \quad a_0 = -6.258151 \cdot 10^{-2} \\ & h_1 = \frac{1}{16} & , \quad a_1 = -3.126018 \cdot 10^{-2} \\ & h_2 = \frac{1}{32} & , \quad a_2 = -1.562627 \cdot 10^{-2} \end{array}$$

$$\Longrightarrow \quad p(0) = -1.02\ldots \cdot 10^{-2}$$

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 Differentiation von $f \in C^1$)

Wähle $a(h) = \frac{f(h)-f(-h)}{2h}$.

Ist f analytisch, so gilt die **asymptotische Entwicklung**

$$a(h) = a(0) + \sum_{i=1}^{\infty} \alpha_{2i} h^{2i} \text{ mit } a(0) = f'(0)$$

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 ($a(h) = a(-h)$) und das Interpolationspolynom solle nur h^{2k}-Terme enthalten.

Sei $f(x) = \sin(x) \quad \Longrightarrow \quad a(h) = \frac{\sin(h)-\sin(-h)}{2h} = \frac{\sin(h)}{h}$, so folgt für $p(x) = q(x^2), q \in \mathbb{P}_1$:

$$\begin{array}{lll} h_0 = \frac{1}{8} & , & a_0 = 0.9973 \\ h_0 = \frac{1}{16} & , & a_0 = 0.99934 \\ h_0 = \frac{1}{32} & , & a_0 = 0.99983 \end{array}$$

$$\Longrightarrow \quad p(0) = 0.999999926.$$

Satz 4.24 (Extrapolationfehler)
Gelte für $a : (0,\infty) \to \mathbb{R}$ die asymptotische Entwicklung

$$a(h) = a(0) + \sum_{j=1}^{n} \alpha_j h^{qj} + a_{n+1}(h) h^{q(n+1)}$$

mit $q > 0$ und $a_{n+1}(h) = \alpha_{n+1} + o(1)$. Dabei seien $\alpha_1, \ldots, \alpha_{n+1} \in \mathbb{R}$ unabhängig von h. Sei $(h_k)_{k\in\mathbb{N}}$ eine monoton fallende Folge, $h_k > 0$ und $\frac{h_{k+1}}{h_k} \le \rho < 1$ für $\rho > 0$ unabhängig von k. Mit $p_n^{(k)} \in \mathbb{P}_n$ 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:

$$\left| a(0) - p_n^{(k)}(0) \right| = O(h_k^{q(n+1)}) \text{ für } k \longrightarrow \infty.$$

Beweis: Setze $z = h^q$, $z_k = h_k^q$. In der Lagrange Darstellung ist das Interpolationspolynom gegeben durch $p_n^{(k)}(z) = \sum_{i=0}^{n} a(h_{k+i}) L_{k,i}^n(z)$ 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 von a folgt

$$\begin{aligned} p_n^{(k)}(0) &= \sum_{i=0}^{n} \left(a(0) + \sum_{j=1}^{n} \alpha_j z_{k+i}^j + \alpha_{n+1} z_{k+i}^{n+1} + o(1) z_{k+i}^{n+1} \right) L_{k,i}^n(0) \\ &= a(0) \sum_{i=0}^{n} L_{k,i}^n(0) + \sum_{j=1}^{n+1} \alpha_j \sum_{i=0}^{n} z_{k+i}^j L_{k,i}^n(0) + o(1) \sum_{i=0}^{n} z_{k+i}^{n+1} L_{k,i}^n(0). \end{aligned}$$

Um die Summanden $z^r L_{k,i}^n(0)$ zu berechnen, verwenden wir die Fehlerdarstellung aus Satz 4.4 mit $f(z) = z^r$, $r = 0, \ldots, n+1$ und dem Interpolationspolynom $q_n^{(k)}$ zu den Daten $(z_k, f(z_k)), \ldots, (z_{k+n}, f(z_{k+n}))$, d.h. $q_n^{(k)}(z) = \sum_{i=0}^{n} f(z_{k+i}) L_{k,i}^n(z)$, bzw. $q_n^{(k)}(0) = \sum_{i=0}^{n} z_{k+i}^r L_{k,i}^n(0)$.

Es gilt $f(0) - q_n^{(k)}(0) = \frac{1}{(k+1)!} f^{(n+1)}(\xi_0) \prod_{i=0}^{n} (0 - z_{k+i})$ und somit folgt

$$\begin{aligned} -\sum_{i=0}^{n} z_{k+i}^r L_{k,i}^n(0) &= \frac{1}{(n+1)!} f^{(n+1)}(\xi_0) (-1)^{n+1} \prod_{i=0}^{n} z_{k+i} - f(0) \\ &= \begin{cases} -1 & : \ r = 0 \\ 0 & : \ r = 1, \ldots, n \\ (-1)^{n+1} \prod_{i=0}^{n} z_{k+i} & : \ r = n+1 \end{cases} \end{aligned}$$

Damit erhalten wir

$$p_n^{(k)}(0) = a(0) + \alpha_{n+1} (-1)^n \prod_{i=0}^{n} z_{k+i} + \sum_{i=0}^{n} o(1) z_{k+i}^{n+1} L_{k,i}^n(0).$$

Es gilt:

$$\begin{aligned} \left| \alpha_{n+1} (-1)^n \prod_{i=0}^{n} z_{k+i} \right| &\le |\alpha_{n+1}| \prod_{i=0}^{n} z_k = |\alpha_{n+1}| \, z_k^{(n+1)} \\ &= |\alpha_{n+1}| \, h_k^{q(n+1)} = O(h_k^{q(n+1)}). \end{aligned}$$

Außerdem gilt $\left| L_{k,i}^n(0) \right| = \prod_{\substack{l=0 \\ l\neq i}}^{n} \frac{1}{\left| \frac{z_{k+i}}{z_{k+l}} - 1 \right|} \le C(\rho, n, q)$, unabhängig von k:

$$\Longrightarrow \left|\sum_{i=0}^{n} o(1)L_{i,k}^{n}(0)z_{k+i}^{n+1}\right| \leq C(\rho,n,q)o(1)z_k^{n+1}$$
$$\leq C(\rho,n,q)o(1)h_k^{q(n+1)} = O(h_k^{q(n+1)}),$$

$$\Longrightarrow \left|p_n^{(k)}(0) - a(0)\right| = O(h_k^{q(n+1)}). \quad \square$$

Algorithmus: (Richardson Extrapolation)
Zur Berechnung von $p_n^{(k)}(0)$ eignet sich das Neville Schema:

$$p_n^{(k)}(0) = p_{n-1}^{(k+1)}(0) + \frac{p_{n-1}^{(k+1)}(0) - p_{n-1}^{(k)}(0)}{\frac{z_k}{z_{k+n}} - 1}$$

mit $p_n^{(k)}(z) = p_{k,k+1,\ldots,k+n}(z)$.
Mit $a_{k,n} := p_n^{(k-n)}(0)$ erhält man als Rekursion für $a_{k,n}$:

$$a_{k,n} = a_{k,n-1} + \frac{a_{k,n-1} - a_{k-1,n-1}}{\left(\frac{h_{k-n}}{h_k}\right)^q - 1}.$$

Als Tableau mit Startwert $a_{k,0} = a(h_k)$ ergibt sich dann

$$\begin{array}{llllll} h_0 & a_{0,0} & & & & \\ h_1 & a_{1,0} & a_{1,1} & & & \\ h_2 & a_{2,0} & a_{2,1} & a_{2,2} & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ h_k & a_{k,0} & a_{k,1} & a_{k,2} & \cdots & a_{k,k} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \end{array}$$

Beispiel 4.25
Berechnung von $e = \lim\limits_{n\to\infty}(1+\frac{1}{n})^n = \lim\limits_{h\to 0}(1+h)^{\frac{1}{h}}$, d.h. $a(h) = (1+h)^{\frac{1}{h}}$.

Wähle $h_k = 2^{-k} \Longrightarrow a_{k,0} = a(h_k) = (1+2^{-k})^{2^k}$.

$\Longrightarrow a_{0,0} = 2,\ a_{1,0} = \frac{9}{4},\ a_{2,0} = \frac{625}{256} \approx 2.44.$

Als Tableau:

		$n=0$	$n=1$	$n=2$	
$k=0:$	$h_0=1$	2			
$k=1:$	$h_1=\frac{1}{2}$	$\frac{9}{4}$	$\frac{5}{2}$		
$k=2:$	$h_2=\frac{1}{4}$	$\frac{625}{256}$	$\frac{337}{128}$	$\frac{257}{96} \approx 2.67708$	

Es folgt also z.B. $a_{22} \approx 2.67708$ als Approximation von $e \approx 2.718281828$. Bereits $a_{5,5}$ liefert $a_{5,5} \approx 2.71827$, während $a_{5,0} \approx 2.6769 \approx a_{22}$.

Nebenrechnung:

$$a_{1,1} = a_{1,0} + \frac{a_{1,0}-a_{0,0}}{\left(\frac{h_0}{h_1}\right)-1} = \frac{9}{4} + \frac{\frac{9}{4}-2}{2-1} = \frac{5}{2}$$
$$a_{2,1} = a_{2,0} + \frac{a_{2,0}-a_{1,0}}{2-1} = \frac{337}{128}$$

Allgemein: $\frac{h_{k-n}}{h_k} = 2^{-k+n+k} = 2^n$

$$a_{2,2} = a_{2,1} + \frac{a_{2,1}-a_{1,1}}{2^2-1} = \frac{257}{96}$$

Aufwand: Die Richardson Extrapolation eignet sich vor allem, falls $a(h)$ sehr teuer zu berechnen ist, etwa falls für den Aufwand $A(h)$ gilt $A(h) = O(1/h)$. In unserem Beispiel folgt dann für $a(1/32)$ der Aufwand $A(h) = 32$, während der Aufwand zur Berechnung von a_{22} gegeben ist durch $A(1) + A(1/2) + A(1/4) = 7$.

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, d.h. $\forall\, x \in \mathbb{R} :\ t_n(x + \omega) = t_n(x)$ und $t_n(x_k) = y_k,\ k = 0, \ldots, n$.

Annahme (O.B.d.A): $\omega = 2\pi$.

Die Fourier Analysis legt nahe, die gesuchte Funktion t_n aus Funktionen der Form

$$1, \cos(x), \cos(2x), \ldots$$
$$\sin(x), \sin(2x), \ldots$$

zusammensetzen.

Ansatz: Suche Koeffizienten (a_k, b_k) mit

$$t_n(x) = \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos(kx) + b_k \sin(kx)) + \frac{\Theta}{2} a_{m+1} \cos((m+1)x),$$

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 der diskreten Fourier Analysis lassen sich kompakter über $\mathbb{C}$ formulieren, wobei die **Eulersche Formel**

$$e^{iz} = \cos(z) + i \cdot \sin(z)$$

benutzt wird.

Definition 4.26 (Trigonometrische Polynome)
Wir definieren den Raum der Trigonometrischen Polynome vom Grad n durch

$$T_n := \left\{ t^* : \ \mathbb{C} \longrightarrow \mathbb{C} \;\middle|\; t^*(z) = \sum_{k=0}^{n} c_k e^{ikz} \right\}$$

Mit $w := e^{iz}$ *gilt* $t^*(z) = \sum_{k=0}^{n} c_k w^k$.

Lemma 4.27

(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 $a_k = c_k + c_{-k},\ b_k = i \cdot (c_k - c_{-k}),\ k = 0, \dots, m$. 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: (Ohne Beweis)

Voraussetzungen und Notationen für diesen Abschnitt

- Äquidistante Stützstellen, d.h. $x_k = \frac{2\pi}{n+1}k,\ k = 0, \dots, n$.
- $w(x) := e^{ix},\ E_k(x) := e^{ikx} = w^k(x)\ (k \in \mathbb{Z})$.
- $\hat{w} := e^{i \cdot \frac{2\pi}{n+1}} \in \mathbb{C},\ w_k := w(x_k) = e^{ik\frac{2\pi}{n+1}} = \hat{w}^k\ (k \in \mathbb{Z})$.

Lemma 4.28

(i) $(E_k)_{k\in\mathbb{Z}}$ bilden ein **Orthonormalsystem**, d.h. $\langle E_k, E_l \rangle = \delta_{kl}$.

(ii) $w_k^{n+1} = 1$, d.h. $w_0, \dots, w_n$ sind die $(n+1)$ Einheitswurzeln und $w_0, \dots, w_n$ sind paarweise verschieden.

(iii) $w_k^l = w_l^k,\ w_{n+1-k}^l = w_{-k}^l,\ w_k^{-l} = \overline{w_k^l}$.

(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 4.29 (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:

$$c_k = \frac{1}{n+1} \sum_{j=0}^{n} y_j e^{-ijx_k} = \frac{1}{n+1} \sum_{j=0}^{n} y_j w_k^{-j}.$$

Beweis: Um die Existenz und Eindeutigkeit zu zeigen, verwenden wir Satz 4.1, der auch im Komplexen gezeigt werden kann. Es existiert daher genau ein $p \in \mathbb{P}_n$ mit $p(x) = \sum_{k=0}^{n} c_k x^k$ 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, verwenden wir Lemma 4.28:

$$\begin{aligned}\sum_{j=0}^{n} y_j w_k^{-j} &= \sum_{j=0}^{n} p(w_j) w_k^{-j} = \sum_{j=0}^{n} \left(\sum_{l=0}^{n} c_l w_j^l \right) w_j^{-k} \\ &= \sum_{l=0}^{n} c_l \left(\sum_{j=0}^{n} w_j^{l-k} \right) = \sum_{l=0}^{n} c_l (n+1)\delta_{lk} = (n+1)c_k.\end{aligned}$$

Also folgt $c_k = \frac{1}{n+1} \sum_{j=0}^{n} y_j w_k^{-j}$. □

Satz 4.30 (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 $y_0, \ldots, y_n \in \mathbb{R}$ 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 $t_n(x_k) = y_k, \ k = 0, \ldots, n$.

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 2.29 gilt:

$$t_n^*(x) = \sum_{k=0}^{n} c_k e^{ikx} \text{ mit } c_k = \frac{1}{n+1} \sum_{j=0}^{n} y_j w_k^{-j}.$$

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 Lemma 4.27 gilt dann:

$$(*) \ \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos(kx) + b_k \sin(kx)) = \sum_{k=-m}^{m} c_k e^{ikx}.$$

Es folgt

$$\begin{aligned} y_l = \sum_{k=0}^{n} c_k w_l^k &= \sum_{k=0}^{m} c_k w_k^l + \sum_{k=1}^{m} c_{-k} w_{n+1-k}^l + \Theta c_{m+1} w_{m+1}^l \\ &= \sum_{k=-m}^{m} c_k w_k^l + \Theta c_{m+1} w_{m+1}^l. \end{aligned}$$

Für n ungerade gilt: $m+1 = \frac{n+1}{2}$ und daher

$$\begin{aligned} w_{m+1}^l &= \cos((m+1)x_l) + i \cdot \sin((m+1)x_l) \\ &= \cos((m+1)x_l) + i \cdot 0, \text{ da } (m+1)x_l = \tfrac{n+1}{2} l \tfrac{2\pi}{n+1} = l\pi. \end{aligned}$$

$$\begin{aligned} \Longrightarrow \quad y_l &= \sum_{k=-m}^{m} c_k w_k^l + \Theta c_{m+1} w_{m+1}^l \\ &\overset{(*)}{=} \frac{a_0}{2} + \sum_{k=1}^{m} \left(a_k \cos(kx_l) + b_k \sin(kx_l)\right) + \frac{\Theta}{2} a_{m+1} \cos((m+1)x_l) \\ &= t_n(x_l). \end{aligned}$$

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 der Koeffizienten folgt aus:

$$c_{-k} = c_{n+1-k} = \frac{1}{n+1} \sum_{j=0}^{n} y_j w_{n+1-k}^{-j} = \frac{1}{n+1} \sum_{j=0}^{n} y_j w_k^j.$$

Wir erhalten:

$$a_k = c_k + c_{-k} = \frac{1}{n+1} \left(\sum_{j=0}^{n} y_j \left(e^{-ijx_k} + e^{ijx_k}\right) \right) = \frac{2}{n+1} \sum_{j=0}^{n} y_j \cos(jx_k),$$

$$b_k = i \cdot (c_k - c_{-k}) = \frac{i}{n+1} \left(\sum_{j=0}^{n} y_j \left(e^{-ijx_k} - e^{ijx_k}\right) \right) = \frac{2}{n+1} \sum_{j=0}^{n} y_j \sin(jx_k). \qquad \square$$

Bemerkung: $t_n(x_l) = t_n^*(x_l)$, aber im Allgemeinen ist $t_n(x) \neq t_n^*(x)$ für $x \neq x_l$ $(l = 0, \ldots, n)$. Es gilt sogar $t_n(x) \neq Re(t_n^*(x))$.

Beispiel 4.31
Gegeben: $n = 2,\ x_0 = 0,\ x_1 = \frac{2}{3}\pi,\ x_2 = \frac{4}{3}\pi.$

Es gilt: $\cos(x_1) = \cos(x_2) = -\frac{1}{2},\ \sin(x_1) = -\sin(x_2) =: \xi,\ 2x_1 = x_2$ und $2x_2 \equiv x_1 \bmod 2\pi$.

$$c_0 = \tfrac{1}{3}\left(y_0 e^{-i0} + y_1 e^{-i0} + y_2 e^{-i0}\right) = \tfrac{1}{3}(y_0 + y_1 + y_2),$$

$$c_1 = \tfrac{1}{3}\left(y_0 e^{-i0} + y_1 e^{-i\frac{2}{3}\pi} + y_2 e^{-i2\frac{2}{3}\pi}\right) = \tfrac{1}{3}y_0 - \tfrac{1}{6}(y_1 + y_2) + i \cdot \tfrac{\xi}{3}(y_1 - y_2),$$

$$c_2 = \tfrac{1}{3}\left(y_0 e^{-i0} + y_1 e^{-i\frac{4}{3}\pi} + y_2 e^{-i2\frac{4}{3}\pi}\right) = \tfrac{1}{3}y_0 - \tfrac{1}{6}(y_1 + y_2) + i \cdot \tfrac{\xi}{3}(y_2 - y_1).$$

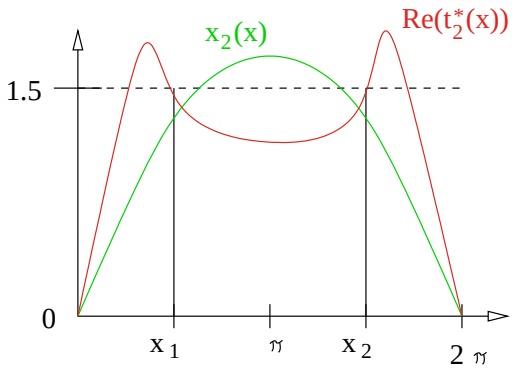


Abbildung 4.4: Beispiel 4.31

Im Reellen: $(m = 1, \Theta = 0)$

$a_0 = \frac{2}{3}(y_0 + y_1 + y_2)\,,\ a_1 = \frac{2}{3}\left(y_0 - \frac{1}{2}y_1 - \frac{1}{2}y_2\right)$

$b_1 = \frac{2\xi}{3}(y_1 - y_2)$

Seien $y_0 = 0,\ y_1 = y_2 = \frac{3}{2}$, so erhalten wir

$$t_2(x) = 1 - \cos(x).$$

Dahingegen erhalten wir als Realteil des komplexen Interpolationspolynoms (siehe auch Abb. 4.4):

$$Re(t_2^*(x)) = 1 - \frac{1}{2}(\cos(x) + \cos(2x))\,.$$

4.6.1 Schnelle Fourier Transformation (FFT)

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.

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. für 2^Q Daten $y_0, \ldots, y_n$.

Sei n ungerade und seien $m = \frac{n-1}{2}$, $l \in \{0, \ldots, n\}$ fest. Dann folgt

$$\begin{aligned} c_l &= \frac{1}{n+1}\sum_{j=0}^{n} y_j w_j^{-l} = \frac{1}{n+1}\left(\sum_{j=0}^{m} y_{2j} w_{2j}^{-l} + \sum_{j=0}^{m} y_{2j+1} w_{2j+1}^{-l}\right) \\ &= \frac{1}{n+1}\left(\sum_{j=0}^{m} y_{2j} w_{2j}^{-l} + \hat{w}^{-l}\left(\sum_{j=0}^{m} y_{2j+1} w_{2j}^{-l}\right)\right), \text{ mit } \hat{w} = e^{i\frac{2\pi}{n+1}}. \end{aligned}$$

Da $n+1 = 2(m+1)$ folgt:

$$c_l = \frac{1}{2}\left(\frac{1}{m+1}\sum_{j=0}^{m} y_{2j} w_{2j}^{-l} + \hat{w}^{-l}\frac{1}{m+1}\sum_{j=0}^{m} y_{2j+1} w_{2j}^{-l}\right).$$

Sei $l_1 \equiv l \bmod (m+1)$, d.h. $l_1 \in \{0,\ldots,m\}$ und $l = \lambda(m+1)+l_1\ \lambda \in \mathbb{N}$. Dann folgt $l = \frac{1}{2}\lambda(n+1)+l$ und somit

$$w_{2j}^{-l} = e^{-il2j\frac{2\pi}{n+1}} = e^{-i\lambda j 2\pi - i\cdot l_1 2j\frac{2\pi}{n+1}} = e^{-i\lambda j 2\pi} w_{2j}^{-l_1} = w_{2j}^{-l_1}$$

$$\Longrightarrow \quad c_l = \frac{1}{2}\left(c_{l_1}^{even} + c_{l_1}^{odd}\hat{w}^{-l}\right)$$

mit

$$\begin{aligned} c_{l_1}^{even} &= \frac{1}{m+1}\sum_{j=0}^{m} y_{2j} w_{2j}^{-l_1}, \\ c_{l_1}^{odd} &= \frac{1}{m+1}\sum_{j=0}^{m} y_{2j+1} w_{2j}^{-l_1},\ l_1 \in \{0,\ldots,\frac{n+1}{2}\}. \end{aligned}$$

Dabei sind $c_{l_1}^{even}$, $c_{l_1}^{odd}$ geraade die Koeffizienten des komplexen trigonometrischen Polynoms zu den Daten $(x_0,y_0),(x_2,y_2),\ldots,(x_{n-1},y_{n-1})$, bzw. zu $(x_0,y_1),(x_2,y_3),\ldots,(x_{n-1},y_n)$.

Idee des Algorithmus:

Stützstellen ($Q = 3$)

$x_0,\ x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6,\ x_7$

$8 \cdot 2$

$x_0,\ x_2,\ x_4,\ x_6$

$4 \cdot 4$

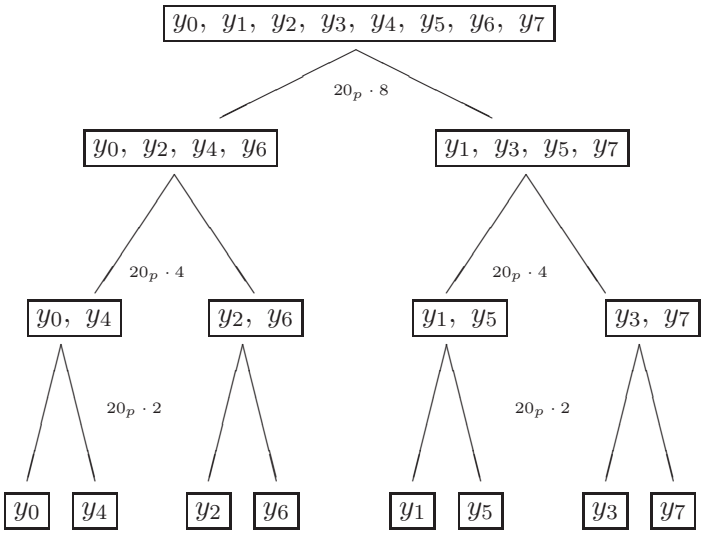
$x_0,\ x_4$

$2 \cdot 8$

Rechenaufwand: 0

$48 = (3 \cdot 2 \cdot (n+1))$

Daten



Allgemein: Pro Level $2(n+1)$ Operationen bei $\log_2(n)$ Levels $\implies$ Anzahl der Operationen zur Berechnung von $c_0, \ldots, c_n$ beträgt $2(n+1)\log_2(n) = O(n \log_2 n)$.

Satz 4.32

Sei $n = 2m+1, m \in \mathbb{N}$ 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 $t_n^{even}(x) = \sum_{j=0}^{m} c_j^{even} e^{ijx}$ 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

$$(*) \quad t_n^*(x) = \frac{1}{2}\left(1 + e^{i\cdot(m+1)x}\right) t_n^{even}(x) + \frac{1}{2}\left(1 - e^{i\cdot(m+1)x}\right) t_n^{odd}\left(x - \frac{\pi}{m+1}\right)$$

und es ist $c_l = \frac{1}{2}\left(c_l^{even} + \hat{w}^{-l} c_l^{odd}\right)$, $c_{l+m+1} = \frac{1}{2}\left(c_l^{even} - \hat{w}^{-l} c_l^{odd}\right)$ mit $l = 0, \ldots, m$ und $\hat{w} = e^{i\frac{\pi}{m+1}}$.

Beweis: Sei r_n die rechte Seite von $(*)$, d.h.

$$\begin{aligned} r_n(x) &= \tfrac{1}{2}\sum_{j=0}^{m}\left[\left(1+e^{i(m+1)x}\right)c_j^{even}e^{ijx}+\left(1-e^{i(m+1)x}\right)c_j^{odd}e^{ij\left(x-\frac{\pi}{m+1}\right)}\right] \\ &= \tfrac{1}{2}\sum_{j=0}^{m}\left[c_j^{even}\left(e^{ijx}+e^{i(j+m+1)x}\right)+c_j^{odd}\left(e^{ijx}-e^{i(j+m+1)x}\right)e^{-ij\frac{\pi}{m+1}}\right] \\ &= \tfrac{1}{2}\sum_{j=0}^{m}\left(c_j^{even}+e^{-ij\frac{\pi}{m+1}}c_j^{odd}\right)e^{ijx}+\tfrac{1}{2}\sum_{j=m+1}^{2m+1}\left(c_{j-(m+1)-l}^{even}-e^{-ij\frac{\pi}{m+1}}c_{j-(m+1)}^{odd}e^{ijx}\right) \\ &= \sum_{j=0}^{n}\hat{c}_j e^{ijx}\in T_n. \end{aligned}$$

Wegen der Eindeutigkeit des Interpolationspolynom folgt $t_n = r_n$, falls r_n die Interpolationsbedingung erfüllt. Für $x_l = \frac{2\pi}{n+1}l$ gilt:

$$e^{i(m+1)x_l} = e^{i\frac{2\pi}{2n+1}(n+1)l} = e^{il\pi} = \begin{cases} 1 & : \ l \text{ gerade} \\ -1 & : \ l \text{ ungerade} \end{cases}$$

$$\Longrightarrow \quad r_n(x_l) \overset{(*)}{=} \begin{cases} t_n^{even}(x_l) & : \ l \text{ gerade} \\ t_n^{odd}\left(x_l-\frac{\pi}{m+1}\right) & : \ l \text{ ungerade} \end{cases}$$

$$= \begin{cases} t_n^{even}(x_l) & : \ l \text{ gerade} \\ t_n^{odd}(x_{l-1}) & : \ l \text{ ungerade} \end{cases}$$

Also $r_n(x_l) = y_l$ und damit $t_n \equiv r_n \implies \hat{c}_j = c_j$.
Da $e^{-ij\frac{\pi}{m+1}} = \hat{w}^{-j}$, folgt die Formel für c_l aus der Definition von $\hat{c}_l$. □

Algorithmus:

Für $q = 0, \dots, Q$ sei $t_k^q(x) = \sum_{j=0}^{2^q-1} c_{k,j}^q e^{ijx}, \ k = 0, \dots, 2^{Q-q} - 1$
das Interpolationspolynom zu $\left(x_{j2^{Q-q}}, y_{2j^{Q-q}+k}\right)_{j=0}^{2^{q-1}}$
Nach Satz 4.32 mit $m = 2^q - 1$ bzw $n = 2^{q+1} - 1$ gilt:

$$\begin{aligned} c_{k,l}^{q+1} &= \frac{1}{2}\left(c_{k,l}^q + e^{-i\frac{2\pi}{2^{q+1}}l} c_{k+2^{Q-q-1},l}^q\right) \ l = 0, \dots, 2^{q-1}, \\ c_{k,l+2^q}^{q+1} &= \frac{1}{2}\left(c_{k,l}^q - e^{-i\frac{2\pi}{2^{q+1}}l} c_{k+2^{Q-q-1},l}^q\right). \end{aligned}$$

Start der Iteration: $\boxed{c_{k,0}^0 = y_k}$.

Speicherbedarf: Für q und $q+1$ müssen Matrizen berechnet werden:

$$C^q = (c_{k,l}^q), \ C^{q+1} = (c_{k,l}^{q+1})$$

$C^q \in \mathbb{C}^{2^{Q-q} \times 2^q}$ 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 $2^{Q-q-1}2^{q+1} = 2^Q = n+1$

Daher sollen die Koeffizienten $c_{k,l}^q, \ c_{k,l}^{q+1}$ in Vektoren der Dimension $n+1$ gespeichert werden

$$C[2^q k + l] := c_{k,l}^q,$$

$$D[2^{q+1} k + l] := c_{k,l}^{q+1}.$$

Es gilt: $e^{-i\frac{2\pi}{2^{q+1}}l} = e^{-il\frac{2\pi}{2^Q}2^{Q-q-1}} =: W[2^{Q-q-1}l]$, wobei der Vektor $W[l] := e^{-i\frac{2\pi}{2^Q}l}, \ l = 0, \dots, 2^Q - 1$ vorab nur einmal berechnet werden muss.

Mit $\hat{w} := e^{-i\frac{2\pi}{2^Q}}$ erhalten wir dann folgenden Algorithmus:

Algorithmus 4.33 (FFT)

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 $l = 0, \ldots, 2^q - 1$

$(*)$
- $u = C[2^q k + l]$
- $v = W[2^{Q-q-1} l] C[2^q (k + 2^{Q-q-1}) + l]$
- $D[2^{q+1} k + l] = \frac{1}{2}(u + v)$
- $D[2^{q+1} k + l + 2^q] = \frac{1}{2}(u - v)$

Für $l = 0, \ldots, 2^Q - 1$

- $C[l] = D[l]$

Aufwand: $(*)$ benötigt 3 Operationen. Anzahl der Durchläufe von $(*)$

$$Q 2^{Q-q-1} 2^q = Q 2^{Q-1} = \log_2(n+1) \frac{n+1}{2}.$$

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. Es existieren 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 stückweise polynomielle Funktionen, d.h.

$$P|_{[x_{i-1}, x_i]} \in \mathbb{P}_r$$

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 $p \in C^q(I)$. Die Parameter (r, q) sind geeignet zu wählen:

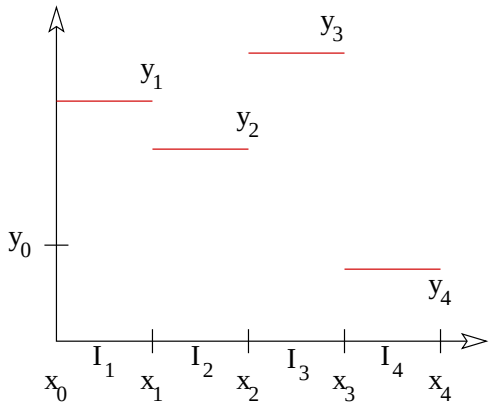


Abbildung 4.5: Beispiel 4.34: Treppenfunktionen

$$P|_{[x_{i-1},x_i]} \in \mathbb{P}_r, \quad p(x) = \begin{cases} p_1(x) & : \quad x \in (x_0, x_1] \\ p_2(x) & : \quad x \in (x_1, x_2] \\ \vdots & \\ p_n(x) & : \quad x \in (x_{n-1}, x_n] \end{cases}$$

$p_i \in \mathbb{P}_r$ hat die 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 der Splines durch

$$S_\Delta^{r,q} := \left\{ P \in C^q(I) \;\middle|\; P|_{I_i} =: p_i \in \mathbb{P}_r \text{ für } 1 \leq i \leq n \right\}$$

Gegeben: Zerlegung Δ, 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 4.34
$\underline{r = 0}$: Die einzig mögliche Interpolation durch stückweise konstante Funktionen ist gegeben durch $P_\Delta(x) = y_i$ für $x \in I_i$ bzw. $p_i(x) = y_i$. Für $q \geq 0$ ist das Problem nicht lösbar.
Abbildung 4.5 zeigt die entstandene Treppenfunktion. In diesem Fall ist P_Δ nicht stetig!

$\underline{r = 1}$: Es soll gelten: $p_i \in \mathbb{P}_1$ und $p_i(x_{i-1}) = y_{i-1},\ p(x_i) = y_i$.
Abbildung 4.6 zeigt die eindeutig bestimmten Funktionen p_i definiert 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 $f \in C^4(I)$

(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 $(x_{i-1}, y_{i-1}), (x_{i1}, f(x_{i1})), (x_{i2}, f(x_{i2})), (x_{i+1}, f(x_{i+1})) \implies p_i \in \mathbb{P}_3$ 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 Hermiteinterpolation 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: Existiert ein $P_\Delta \in S_\Delta^{3,2}$?

Bemerkung: Sei $n > r$, dann ist das Interpolationsproblem in $S_\Delta^{r,q}$ für $q \geq r$ i.a. schlecht gestellt (d.h. 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 Stetigkeitsbedingungen in x_1
$\vdots$
I_n : 2 Interpolationsbedingungen + q Stetigkeitsbedingungen in x_{n-1}

$\implies$ $2n + q(n-1) = n(q+2) - q$ Bedingungen.

Ist $q \geq r$ so folgt: $2n + q(n-1) \geq 2n + r(n-1) = n(r+1) + n - r > n(r+1)$. Für $n - r > 0$ existieren also mehr Bedingungen als Freiheitsgrade und das Problem ist i.A. nicht lösbar.

Spezialfall: $q = r - 1$ (Eigentliche "Spline-Interpolation")

Bedingungen: $n(q+2) - q = n(r+1) - q$, d.h. es müssen noch $q = r - 1$ Freiheitsgrade zusätzlich festgelegt werden.

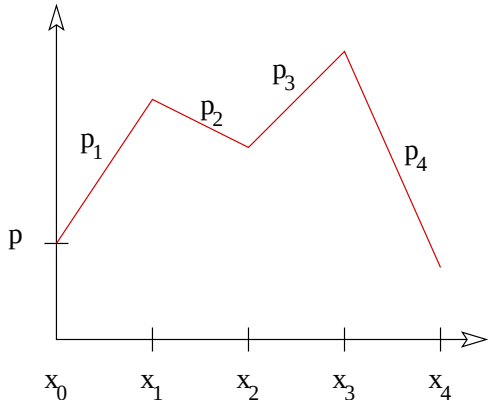


Abbildung 4.6: Beispiel 4.34: Gerade

4.7.1 Kubische Spline-Interpolation

Gegeben: $\Delta = (x_0, \ldots, x_n)$ Zerlegung des Intervalls $I = [a, b]$ und Daten $y_0, \ldots, y_n \in \mathbb{R}$.

Gesucht: $P_\Delta \in S_\Delta^{3,2}$ mit $P_\Delta(x_i) = y_i$ $(0 \le i \le n)$ und eine der Bedingungen a) bis d):

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 von **natürlichen kubischen Splines**.

b) $p'(a) = g_a,\ p'(b) = g_b$ für $g_a, g_b \in \mathbb{R}$ gegeben.

c) P_Δ sei **periodisch fortsetzbar** in $\mathbb{C}^2(\mathbb{R})$, d.h. $y_0 = y_n$ und $p'(a) = p'(b),\ p''(a) = p''(b)$.

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 4.35 (Existenz und Eindeutigkeit)
Zu gegebener Zerlegung Δ und Daten $y_0, \ldots, y_n$ existiert genau ein $P_\Delta \in S_\Delta^{3,2}$ mit $p(x_k) = y_k$, welches eine der Bedingungen a), b), c), oder d) erfüllt. Im Fall c) muss gelten: $y_0 = y_n$.

Beweis: **Idee:** Stelle LGS für die Momente $M_j := P''_\Delta(x_j)$ 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 $a_j, b_j \in \mathbb{R}$:

$$p_j(x) = \frac{1}{6h_j}\left(M_j(x - x_{j-1})^3 + M_{j-1}(x_j - x)^3\right) + b_j\left(x - \frac{x_j + x_{j-1}}{2}\right) + a_j. \quad (*)$$

Aus den Interpolationsbedingungen $p_j(x_{j-1}) = y_{j-1},\ p_j(x_j) = y_j$ folgt:

$$\begin{aligned} y_{j-1} &= \tfrac{1}{6h_j} M_{j-1} h_j^3 - b_j \tfrac{1}{2} h_j + a_j, \\ y_j &= \tfrac{1}{6h_j} M_j h_j^3 + b_j \tfrac{1}{2} h_j + a_j. \end{aligned}$$

Dies 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 $M_0, \ldots, M_n$ ab.

Es bleiben noch die $n - 1$ Bedingungen $p'(x_j) = p'_{j+1}(x_j)$ für $j = 1, \ldots, n-1$:
Aus $(*)$ und $(**)$ folgt: $p'_j(x) = \frac{1}{2h_j}\left(M_j (x - x_{j-1})^2 - M_{j-1}(x_j - x)^2\right) + \frac{1}{h_j}(y_j - y_{j-1}) - \frac{1}{6} h_j (M_j - M_{j-1})$.

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 $j = 1, \ldots, n-1$,
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 $x_{j+1} - x_{j-1} = h_j + h_{j+1}$ folgt:

$$\mu_j M_{j-1} + M_j + \lambda_j M_{j+1} = 3y[x_{j-1}, x_j, x_{x+1}]$$

mit $\mu_j = \frac{h_j}{2(h_j + h_{j+1})},\ \lambda_j = \frac{h_{j+1}}{2(h_j + h_{j+1})}$.

Wir erhalten ein $(n-1) \times (n-1)$ LGS für die (n+1) Momente $M_0, \ldots, M_n$.

Fall a) $M_0 = M_a,\ M_n = M_b$.

Dies führt auf das $(n-1) \times (n-1)$ LGS für $M_1, \ldots, M_{n-1}$ 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 4.36, da $\mu_j + \lambda_j = 1/2 < 1$ und $\lambda_1 < 1,\ \mu_{n-1} < 1$.

Die Fälle b),c),d) führen analog auf einfach strukturierte LGS mit regulären Matrizen, d.h. $p_1, \ldots, p_n$ eindeutig duch $(*), (**)$ festgelegt $\square$

Lemma 4.36
Sei $A \in \mathbb{R}^{n \times n}$ eine **tridiagonale Matrix**, d.h.

$$A = \text{tridiag}(b_i, a_i, c_i) = \begin{pmatrix} a_1 & c_1 & & 0 \\ b_2 & \ddots & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ 0 & & b_n & a_n \end{pmatrix}$$

Es gelte: $|a_1| > |c_1| > 0$ und $|a_n| > |b_n| > 0$ und $|a_i| \geq |b_i| + |c_i|,\ b_i \neq 0,\ c_i \neq 0,\ 2 \leq i \leq n-1$.

Dann gilt:

(i) A ist regulär.

(ii) $A = LR$ mit $L = \text{tridiag}(b_i, \alpha_i, 0)$ und $R = \text{tridiag}(0, 1, \gamma_i)$ 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 4.37
Die Spline-Interpolation mit kubischen Splines und einer der Zusatzbedingungen a), b), c) oder d) kann mit $O(n)$ Operationen gelöst werden.

Beweis: a) folgt aus 4.35, 4.36.
b), c), d): (siehe z.B. Schaback,Werner: Numerische Mathematik, Berlin, Springer, 1992.)

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 dies näherungsweise $\int_I y''(t)^2 \mathrm{d}t$.

Satz 4.38 (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 einer der Bedingungen a), b), oder c):

a) $P''_\Delta(a) = 0,\ P''_\Delta(b) = 0,$

b) $P'_\Delta(a) = g_a,\ P'_\Delta(b) = g_b,$

c) P_Δ periodisch fortsetzbar in $C^2(\mathbb{R})$.

Dann gilt für alle $f \in C^2(a, b)$ mit denselben Interpolationsbedingungen, d.h. mit $f(x_k) = y_k$ und a), b) oder c) und $\int_a^b |f''|^2 \leq \infty$:

$$\int_a^b \left|f''(x)\right|^2 \mathrm{d}x \geq \int_a^b \left|P''_\Delta(x)\right|^2 \mathrm{d}x.$$

Beweis: Zum Beweis dieser Aussage benötigen wir das folgende Lemma.

Lemma 4.39 (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:

$$\int_a^b \left|f'' - P''_\Delta\right|^2 = \int_a^b \left|f''\right|^2 - \int_a^b \left|P''_\Delta\right|^2$$
$$-2\left(\left[(f'(x) - P'_\Delta(x))P''_\Delta(x)\right]_{x=a}^{b} - \sum_{i=1}^{n}\left[(f(x) - P_\Delta(x))P'''_\Delta(x)\right]_{x=x_{i-1}^+}^{x_i^-}\right).$$

Dabei wurden die folgenden Abkürzungen benutzt:

$$[g(x)]_{x=a}^{b} = g(b) - g(a),$$

$$[g(x)]_{x=x_{i-1}^+}^{x_i^-} = \lim_{x \nearrow x_i} g(x) - \lim_{x \searrow x_{i-1}} g(x).$$ Beachte : P'''_Δ ist unstetig!

Beweis: Es ist

$$\begin{aligned}\int_a^b |f''-P''_\Delta|^2 &= \int_a^b |f''|^2 - 2\int_a^b f''P''_\Delta + \int_a^b |P''_\Delta|^2 \\ &= \int_a^b |f''|^2 - \int_a^b |P''_\Delta|^2 - 2\int_a^b (f''-P''_\Delta)P''_\Delta \\ &= \int_a^b |f''|^2 - \int_a^b |P''_\Delta|^2 - 2\sum_{i=1}^n \underbrace{\int_{I_i}(f''-p_i'')p_i''}_{=:A_i}.\end{aligned}$$

Mit partieller Integration folgt für A_i:

$$\begin{aligned}A_i &= \int_{x_{i-1}}^{x_i}(f''-p_i'')p_i'' = [(f'-p_i')p_i'']_{x=x_{i-1}}^{x_i} - \int_{x_{i-1}}^{x_i}(f'-p_i')p_i''' \\ &= [(f'-p_i')p_i'']_{x=x_{i-1}}^{x_i} - [(f-p_i)p_i''']_{x_{i-1}^+}^{x_i^-} + \int_{x_{i-1}}^{x_i}(f-p_i)p_i^{(4)}.\end{aligned}$$

Es ist $p_i^{(4)} \equiv 0$, da $p_i \in P_3$ und

$$\begin{aligned}\sum_{i=1}^n [(f'-p_i')p_i'']_{x=x_{i-1}}^{x_i} &\overset{p_i''\in C^0}{=} \sum_{i=1}^n [(f'-P'_\Delta]P''_\Delta]_{x=x_{i-1}}^{x_i} \\ &= \sum_{i=1}^n [(f'(x_i)-p'_\Delta(x_i))P''_\Delta(x_i) - (f'(x_{i-1}-P'_\Delta(x_{i-1}))P''_\Delta(x_{i-1})] \\ &= (f'(x_n)-P'_\Delta(x_n))P''_\Delta(x_n) - f'(x_0)-P'_\Delta(x_0))P''_\Delta(x_0) \\ &= [(f'(x)-P'_\Delta(x))P''_\Delta(x)]_{x=a}^b.\end{aligned}$$

$$\implies \sum_{i=1}^n A_i = [(f'(x)-P'_\Delta(x))P''_\Delta(x)]_{x=a}^b - \sum_{i=1}^n [(f(x)-p_i(x))p_i'''(x)]_{x=x_{i-1}}^{x_i}.$$

Also folgt die Holladay Identität. □

Beweis: (Fortsetzung des Beweises von Satz 4.38)
In den 3 Fällen a), b), c) verschwindet der Term $2(\cdots)$ in der Holladay Identität $\implies$ $0 \le \int_a^b |f''-P''_\Delta|^2 = \int_a^b |f''|^2 - \int_a^b |P''_\Delta|^2$. □

Satz 4.40 (Fehlerabschätzung)
Sei Δ eine Zerlegung von I mit $h \le Kh_i$ $(1 \le i \le n)$ 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 $l = 0,1,2,3$: $\left|f^{(l)}(x) - P_\Delta^{(l)}(x)\right| \le 2LKh^{4-l}$.
Also insbesondere

$$|f(x) - P_\Delta(x)| \le 2LKh^4.$$

Beweis: (Ohne Beweis. Siehe z.B. Stoer, Bulirsch. Numerische Mathematik 1. Berlin, Springer 2007.)

Basiswahl für den Splineraum $\mathbf{S_\Delta^{r,r-1}}$: B-Splines

Ziel: Konstruktion einer einfachen Basis von $S_\Delta^{r,r-1}$ mit

(a) positiven Basisfunktionen für numerische Stabilität,

(b) möglichst kleinem Träger.

Definition 4.41 (B-Splines)
Sei $(t_i)_{i\in\mathbb{Z}}$ *eine monoton nicht-fallende Folge mit* $\lim\limits_{i\longrightarrow\pm\infty} t_i = \pm\infty$. *Dann sind die* **B-Splines** $B_{i,k} : \mathbb{R} \to \mathbb{R}$ vom Grad $k \in \mathbb{N}$ rekursiv definiert durch

$$B_{i,0}(x) = \begin{cases} 1 & : \quad t_i < x \leq t_{i+1} \\ 0 & : \quad \text{sonst} \end{cases}$$

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

$$\omega_{i,k}(x) = \begin{cases} \frac{x-t_i}{t_{i+k}-t_i} & : \quad t_i < t_{i+k} \\ 0 & : \quad \text{sonst} \end{cases} .$$

Beispiel:
Die Abb. 4.7 zeigt 6 verschiedene Beispiele, die bei den B-Splines auftreten können.

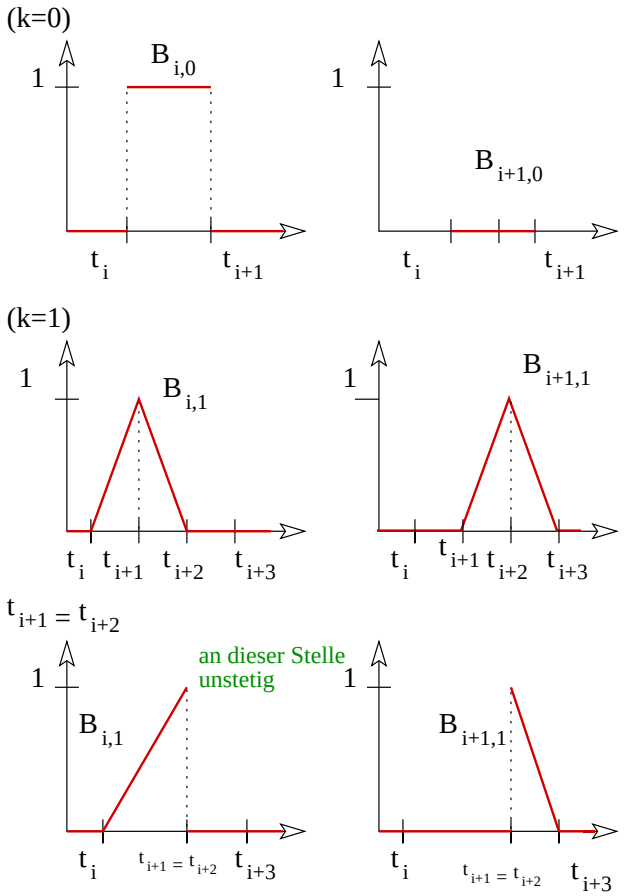


Abbildung 4.7: B-Splines

Satz 4.42 (Eigenschaften der B-Splines)
Sei $(t_i)_{i\in\mathbb{Z}}$ eine monoton nicht-fallende Knotenfolge, wie in Definition 4.41. Dann gilt:

(i) $B_{i,k}|_{[t_j,t_{j+1}]} \in \mathbb{P}_k \ \forall\, i,j \in \mathbb{Z}, k \in \mathbb{N}$,

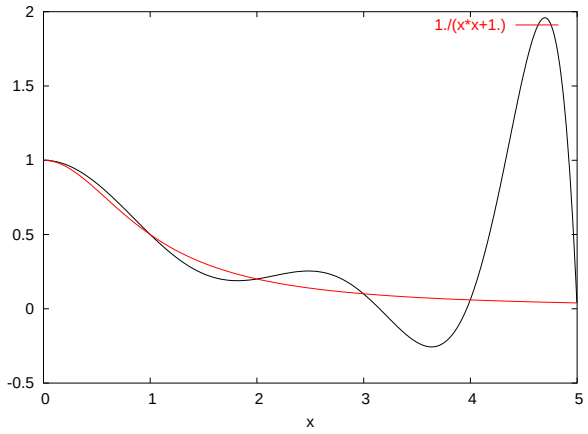
(ii) $supp(B_{i,k}) \subset [t_i, t_{i+k+1}]$, falls $t_i < t_{i+k+1}$ und $B_{i,k} \equiv 0$, falls $t_i = t_{i+k+1}$,

(iii) $B_{i,k} \geq 0$, $\sum\limits_{i\in\mathbb{Z}} B_{i,k}(x) = 1, \ \forall\, x \in \mathbb{R}$ (Zerlegung der 1).

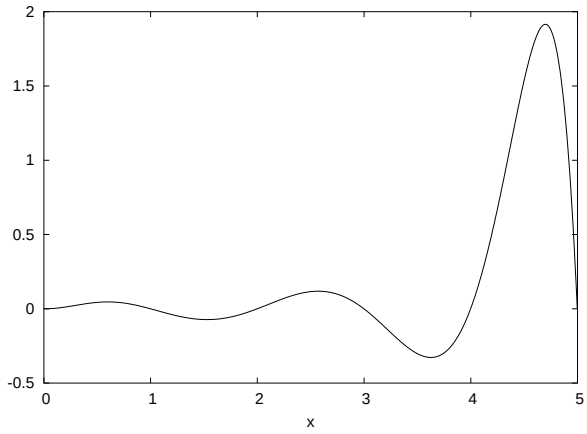
(iv) Falls $\forall\, i \in \mathbb{Z}: \ t_i < t_{i+1}$, dann ist $B_{i,k} \in C^{k-1}$ und $(B_{i,k})_{i\in\mathbb{Z}}$ bildet eine Basis von $S_\Delta^{k,k-1}$.

Beweis: (Ohne Beweis)

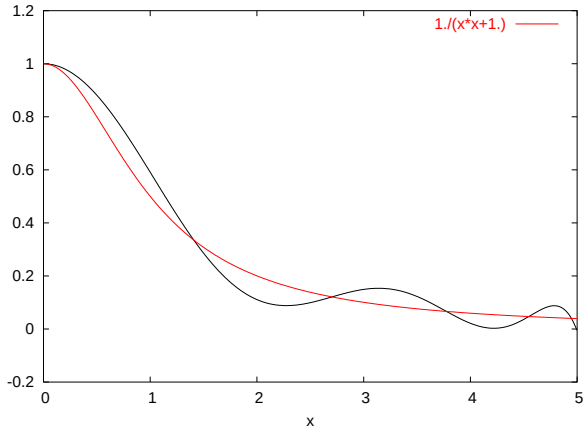
Beispiel Ein Vergleich der angesprochenen Interpolationen für $f(x) = \frac{1}{x^2+1}$ ist in Abb. 4.8 dargestellt. Die Spline-Interpolation ergibt hier das beste Ergebnis.



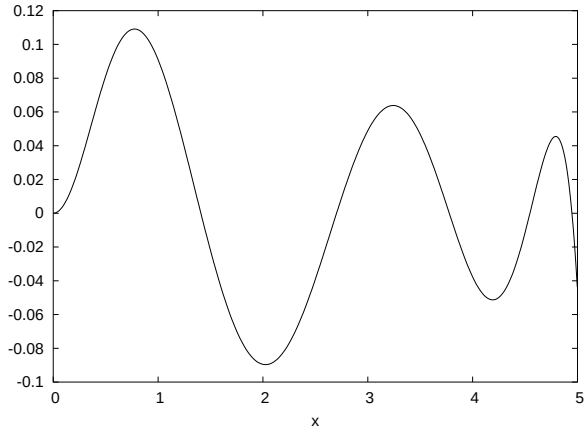
Polynomielle mit gleichmäßig verteilten Stützstellen



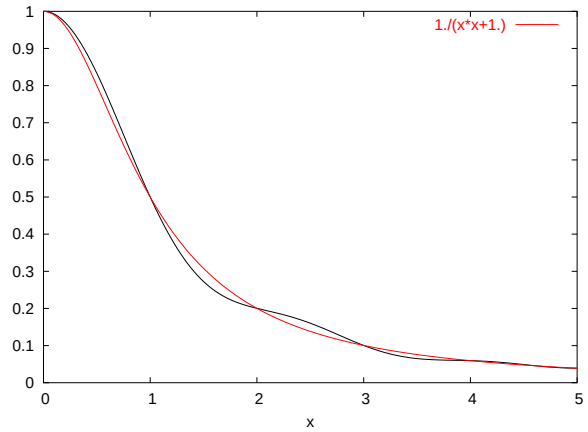
Fehler der Interpolation



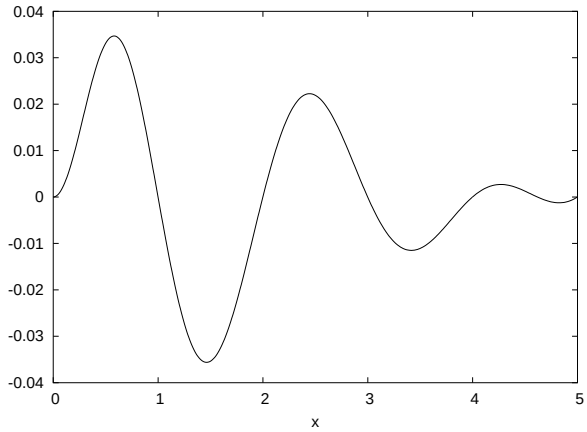
Tschebyschev-Interpolation



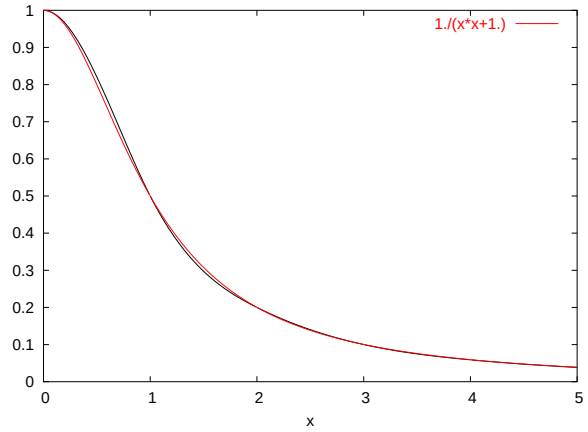
Fehler der Interpolation



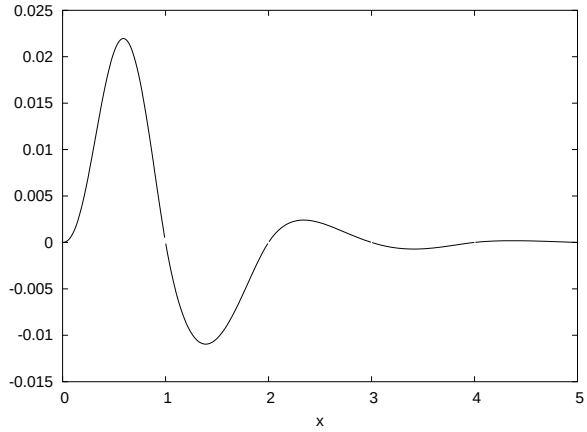
Trigonometrische Interpolation



Fehler der Interpolation



Spline-Interpolation



Fehler der Interpolation

Abbildung 4.8: Unterschiede einiger Interpolationen

Kapitel 5

Numerische Integration

Ziel: Approximation von

$$I(f) := \int_a^b \omega(x) f(x) \mathrm{d}x$$

für $f \in C^k(a,b)$ und für eine gegebene Gewichtsfunktion $\omega \in L^1(a,b)$.

Ansatz: Approximiere $I(f)$ durch eine Summe

$$I_n(f) := \sum_{j=0}^{m} \sum_{l=0}^{m_j - 1} f^{(l)}(x_j) \omega_j^l$$

Definition 5.1 (Quadratur)
Eine Funktional $I_n : C^k(a,b) \longrightarrow \mathbb{R}$ *der Form*

$$I_n(f) := \sum_{j=0}^{m} \sum_{l=0}^{m_j - 1} f^{(l)}(x_j) \omega_j^l$$

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* $m_j \in \{1, \ldots, k+1\}$ *und* $n+1 = \sum_{j=0}^{m} m_j$.

Die Quadratur heißt **exakt** *für* $\mathbb{P}_n$ *(bezüglich* ω*), g.d.w.*

$$I_n(p) = I(p) \ \forall \ p \in \mathbb{P}_n$$

.

$$R(f) = I(f) - I_n(f)$$

ist das zu I_n *gehörende* **Fehlerfunktional**.

Bemerkung: Für die allgemeine Definition der Quadratur vergleiche mit der Definition der Hermite Interpolation. Im folgenden betrachten wir meistens Quadraturen der Form

$$I_n(f) = \sum_{l=0}^{n} \omega_l f(x_l), \ \text{d.h. } m_j = 1.$$

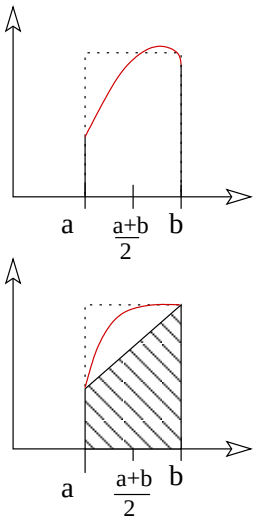


Abbildung 5.1: Beispiel 5.1

Beispiel 5.2 ($\omega \equiv 1$)
Die Abbildung 5.1 verdeutlicht diese Beispiele:

(i) Mittelpunktregel: $I_0(f) = (b-a) f\left(\frac{a+b}{2}\right)$.

(ii) Trapezregel: $I_1(f) = \frac{b-a}{2}\left(f(a) + f(b)\right)$.

(iii) Simpsonregel: $I_2(f) = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$.

Satz 5.3
Gegeben seien $\omega \in L^1(a,b)$ und paarweise disjunkte Stützstellen $x_0, \ldots, x_n$. Dann existiert genau eine Quadraturformel der Form

$$I_n(f) = \sum_{j=0}^{n} \omega_j f(x_j),$$

welche exakt ist auf $\mathbb{P}_n$. Dabei sind die Gewichte gegeben durch

$$\omega_j := \int_a^b \omega(x) L_j^n(x) \mathrm{d}x,$$

wobei $L_j^n(x) = \prod\limits_{\substack{l=0 \\ l \neq j}}^{n} \frac{(x-x_l)}{(x_j - x_l)}$ die Lagrange Polynome sind.

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 $l = 0, \ldots, n$; da I_n, I linear und L_l^n Basis von $\mathbb{P}_n$
$\iff$ $\int\limits_a^b \omega(x) L_l^n(x) = \sum\limits_{j=0}^{n} \omega_j L_l^n(x_j) = \omega_l$, da $L_l^n(x_j) = \delta_{lj}$. □

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:

$$I_n(f) = \sum_{l=0}^{n} \omega_l f(x_l) = \sum_{l=0}^{n} \int_a^b \omega(x) L_l^n(x) f(x_l) \mathrm{d}x$$
$$= \int_a^b \omega(x) \underbrace{\sum_{l=0}^{n} L_l^n(x) f(x_l)}_{p_n(x)} \mathrm{d}x = \int_a^b \omega(x) p_n(x) \mathrm{d}x = I(p_n).$$

Definition 5.4
Eine Quadraturformel $I_n(f) = \sum_{l=0}^{n} \omega_l f(x_l)$ *zu gegebenen Stützstellen* $a \le x_0 < x_1 < \ldots < x_n \le b$ *und Gewichtsfunktion* $\omega \in L^1(a,b)$ *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) $x_j - a = b - x_{n-j}$ $(0 \le j \le n)$ (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 $\omega_{n-j} = \omega_j$ $(0 \le j \le n)$, d.h. die Interpolationsquadratur ist symmetrisch. Falls n gerade ist, so ist I_n exakt auf $\mathbb{P}_{n+1}$.

Beweis: 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 $\tilde{I}_n = I_n$ und folglich $\omega_{n-j} = \omega_j$.

Sei nun $n = 2m$ und damit $x_m = \frac{a+b}{2}$ wegen i). 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

$$c := \frac{f'(x_m) - p_n'(x_m)}{\prod\limits_{\substack{l=0 \\ l \ne m}}^{n} (x_m - x_l)}$$

und $N(x) = \prod_{l=0}^{n} (x - x_l) \in \mathbb{P}_{n+1}$ definiere

$$\tilde{q}_{n+1}(x) := p_n(x) + cN(x).$$

Dann ist $\tilde{q}_{n+1} \in \mathbb{P}_{n+1}$ und $\tilde{q}_{n+1}(x_l) = p_n(x_l) + cN(x_l) = f(x_l) + 0$. Weiter folgt

$$\tilde{q}_{n+1}'(x_m) = p_n'(x_m) + c \prod_{\substack{l=0 \\ l \ne m}}^{n} (x_m - x_l) = f'(x_m).$$

Wegen der Eindeutigkeit der Hermite Interpolation gilt daher $q_{n+1} = \tilde{q}_{n+1}$.

Es gilt wegen i): $N(x) = \prod_{l=0}^{m-1} (x - x_l) \left(x - \frac{a+b}{2}\right) \prod_{l=0}^{m-1} (x - (a+b-x_l))$ und somit folgt $N(a+b-x) = (-1)^{n+1} N(x) = -N(x)$.

Wegen ii) gilt damit:

$$\begin{aligned}\int_a^b \omega(x)N(x)\mathrm{d}x &= \int_a^{x_m} \omega(x)N(x)\mathrm{d}x + \int_{x_m}^b \omega(x)N(x)\mathrm{d}x \\ &\overset{t=a+b-x}{=} \int_a^{x_m} \omega(x)N(x)\mathrm{d}x - \int_{x_m}^a \omega(a+b-t)N(a+b-t)\mathrm{d}t \\ &= \int_a^{x_m} \omega(x)N(x)\mathrm{d}x + \int_a^{x_m} \omega(t)(-N(t))\mathrm{d}t = 0.\end{aligned}$$

Wir erhalten: $\int_a^b q_{n+1}(x)\omega(x)\mathrm{d}x = \int_a^b p_n(x)\omega(x)\mathrm{d}x = I(p_n) = I_n(f)$, da p_n Interpolationspolynom zu f.

Sei nun $f \in \mathbb{P}_{n+1} \implies f = q_{n+1}$ und daher $I_n(f) = I(q_{n+1}) = I(f)$. □

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) $|R_n(f)| \le \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}(b-a)^{n+2}$ für alle $f \in C^{n+1}(a,b)$, falls n ungerade ist,

(ii) $|R_n(f)| \le \frac{\|f^{(n+2)}\|_\infty}{(n+2)!}(b-a)^{n+3}$ 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 $(x_i, f(x_i))$, $i = 0, \dots, n$ ist.

$$\begin{aligned}\implies |R_n(f)| &= \left|\int_a^b (f - p_n)\right| \le \int_a^b |f(x) - p_n(x)|\,\mathrm{d}x \\ &\overset{\text{Satz 4.4}}{=} \int_a^b \left|\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\prod_{k=0}^n (x - x_k)\right|\mathrm{d}x \\ &\le \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}(b-a)^{n+2}\end{aligned}$$

(ii) Aus dem Beweis vom Satz 5.5 folgt: $I_n(f) = I(q_{n+1})$. Dann folgt die Behauptung mit Satz 4.19 □

Bemerkung:

(a) Die Abschätzungen lassen sich leicht verallgemeinern auf den Fall $\omega \in L^1(a,b)$.

(b) Die Abschätzung $\left|\prod_{k=0}^n (x - x_k)\right| \le (b-a)^{n+1}$ kann für gegebene $x_0, \dots, x_n$ deutlich verbessert werden zu $\left|\prod_{k=0}^n (x - x_k)\right| \le K(b-a)^{n+1}$ 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

$$I_n(f) := \sum_{k=0}^{n} \omega_k f(x_k)$$

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 $|R_n(f)| \le K \left\|\hat{f}^{(m)}\right\|_\infty 2^{m+1}$, so gilt für R_n zu I_n: $|R_n(f)| = K \left\|f^{(m)}\right\|_\infty (b-a)^{m+1}$.

Beweis: Sei $p \in \mathbb{P}_n$ 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

$$\begin{aligned} I(p) = \int_a^b p(x)\mathrm{d}x &= \int_{-1}^{1} p(x(t))x'(t)\mathrm{d}t \\ &= \frac{(b-a)^2}{2}\int_{-1}^{1} \hat{p}(t)\mathrm{d}t = \frac{b-a}{2}\hat{I}_n(\hat{p}) \\ &= \sum_{k=0}^{n} \frac{b-a}{2}\hat{\omega}_k - \hat{p}(t_k) \\ &= \sum_{k=0}^{n} \omega_k p(x_k) = I_n(p). \end{aligned}$$

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 $\hat{f}'(t) = x'(t)f'(x(t)) = \frac{b-a}{2}f'(x(t))$ und weiter $\hat{f}^{(m)}(t) = \left(\frac{b-a}{2}\right)^m f^{(m)}(x(t))$.
Also folgt

$$\left\|\hat{f}^{(m)}\right\|_\infty = 2^{-m}(b-a)^m \left\|f^{(m)}(x(t))\right\|_\infty .$$

$$\begin{aligned} \Longrightarrow \quad |R_n(f)| &= \left|\int_a^b f(x)\mathrm{d}x - I_n(f)\right| = \left|\frac{b-a}{2}\left[\int_{-1}^{1}\hat{f}(t)\mathrm{d}t - \hat{I}_n(\hat{f})\right]\right| \\ &= \frac{b-a}{2}\left|\hat{R}_n(\hat{f})\right| \le \frac{b-a}{2}\left\|\hat{f}^{(m)}\right\|_\infty K 2^{m+1} \\ &= (b-a)^{m+1}\left\|f^{(m)}\right\|_\infty K. \quad \square \end{aligned}$$

Bemerkung:

(a) Es reicht also aus I.Q.en auf $[-1,1]$ zu konstruieren. Zu $-1 \le t_0 < \ldots < t_n \le 1$ wird durch

$$\hat{\omega}_j := \int_{-1}^{1} \prod_{\substack{k=0 \\ k\neq j}}^{n} \frac{t-t_k}{t_j-t_k}\mathrm{d}t$$

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.

(b) Da für jede I.Q. $I_n(1) = (b-a)$ gilt, muss $\sum_{k=0}^{n} \omega_k = (b-a)$ gelten.

(c) Wie bei der Polynominterpolation treten Probleme für große Werte von n auf, wie z.B. negative Gewichte. 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 $\left|\hat{R}_n(\hat{f})\right| \leq K \left\|\hat{f}^{(m)}\right\|_\infty 2^{m+1}$. Zu $a < b,\ N \in \mathbb{N}$ setze $a_l := a + lH,\ l = 0, \ldots, N$ mit $H := \frac{b-a}{N}$.
Dann ist

$$I_h(f) := \frac{H}{2} \sum_{l=1}^{N} \sum_{k=0}^{n} \hat{\omega}_k f\left(\frac{H}{2}(t_k - 1) + a + lH\right)$$

eine Quadraturformel mit der Abschätzung
$|R_h(f)| := |I(f) - I_h(f)| \leq K \left\|f^{(m)}\right\|_\infty (b-a)H^m$.

Beweis: Wir wenden Satz 5.7 auf $[a_{l-1}, a_l]$ an:

$$\begin{aligned} \Longrightarrow \quad I_n^l(f) &:= \sum_{k=0}^{n} \tfrac{a_l - a_{l-1}}{2} \hat{\omega}_k f\left(\tfrac{a_l - a_{l-1}}{2} t_k + \tfrac{a_l + a_{l-1}}{2}\right) \\ &= \tfrac{H}{2} \sum_{k=0}^{n} \hat{w}_k f\left(\tfrac{H}{2}(t_k - 1) + a + lH\right). \end{aligned}$$

Also gilt $I_h(f) = \sum_{l=1}^{N} I_n^l(f)$ und es folgt:

$$\begin{aligned} |R_h(f)| &\leq \sum_{l=1}^{N} \left|R_n^l(f)\right| \overset{\text{Satz 5.7}}{\leq} K \left\|f^{(m)}\right\|_\infty \sum_{l=1}^{N} (a_l - a_{l-1})^{m+1} \\ &= K \left\|f^{(m)}\right\|_\infty \underbrace{NH}_{=(b-a)} H^m. \quad \square \end{aligned}$$

5.1 Newton-Cotes Formeln

- Die Newton-Cotes Formeln sind I.Q.en mit äquidistanten Stützstellen $x_k = a + kh,\ h = \frac{b-a}{n}$.
- Als offene Newton-Cotes Formeln bezeichnet man I.Q.en zu äquidistanten Stützstellen $x_k = a + (k+1)h,\ h = \frac{b-a}{n+2}$, d.h. die Randpunkte a, b sind keine Stützstellen.

(a) $n = 1$ (Trapezregel)

$x_0 = a,\ x_1 = b,\ \omega_0 = \int_a^b \frac{x-b}{a-b}\mathrm{d}x = \frac{b-a}{2},\ \omega_1 = \frac{b-a}{2}$,
$T(f) = I_1(f) = \frac{b-a}{2}(f(a) + f(b))$ (vgl. Abb. 5.1).

$$\begin{aligned} |R_n(f)| &\leq \tfrac{\|f''\|_\infty}{2} \int_a^b |x-a|\,|x-b| \\ &= \tfrac{\|f''\|_\infty}{2} \tfrac{(b-a)^3}{6} = \tfrac{\|f''\|_\infty}{12}(b-a)^3. \end{aligned}$$

(b) $n = 2$ (Simpson-Regel)

$$x_0 = a,\ x_1 = \tfrac{a+b}{2},\ x_2 = b,\ \omega_0 = \omega_2 = \tfrac{b-a}{6},\ \omega_1 = \tfrac{2(b-a)}{3},$$
$$S(f) = I_2(f) = \tfrac{b-a}{6}\left(f(a) + 4f\left(\tfrac{a+b}{2}\right) + f(b)\right),$$
$$|R_5(f)| \leq \tfrac{\|f^{(4)}\|_\infty}{2880}(b-a)^5.$$

Zusammengesetze Newton-Cotes Formeln

(a) **Zusammengesetze Trapezregel** (Satz 5.8, $n = 1,\ h = H$)

$$T_h(f) = \tfrac{h}{2}\sum_{l=1}^{N}[f(a+lh-h) + f(a+lh)] = \tfrac{h}{2}\left(f(a) + 2\sum_{l=1}^{N-1} f(a+lh) + f(b)\right),$$
$$|R_h(f)| \leq \tfrac{\|f''\|_\infty}{12}(b-a)h^2.$$

(b) **Zusammengesetze Simpson-Regel** (Satz 5.8, $n = 2,\ h = \frac{H}{2}, x_i := a + ih$)

$$S_h(f) = \tfrac{h}{3}\left(f(a) + 2\sum_{l=1}^{N-1} f(x_{2l}) + 4\sum_{l=1}^{N} f(x_{2l-1}) + f(b)\right),$$
$$|R_n(f)| \leq \tfrac{\|f^{(4)}\|_\infty}{180}(b-a)h^4.$$

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 $\mathbb{P}_m$ mit möglichst großem m exakt ist. Dies ist **nicht** möglich für $m = 2n + 2$ (Gegenbeispiel konstruierbar). 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 exakt ist auf $\mathbb{P}_{2n+1}$.*

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

$$\int_a^b \omega(x) p_{n+1}(x) q(x) \mathrm{d}x = Q_n(p_{n+1}q) = \sum_{k=0}^{n} \omega_k \underbrace{p_{n+1}(x_k)}_{=0 \text{ nach Def.}} q(x_k) = 0.$$

„(ii) $\Longrightarrow$ (i)“: Sei $p \in \mathbb{P}_{2n+1}$. Mit Polynomdivision gilt: $p = qp_{n+1} + r$ mit $q, r \in P_n$. Damit folgt

$$\begin{aligned} \int_a^b \omega(x) p(x) \mathrm{d}x &= \int_a^b \omega(x) \Big(\underbrace{q(x) p_{n+1}(x)}_{=0} + r(x) \Big) \mathrm{d}x \\ &\overset{\text{Vor. (ii)}}{=} 0 + \int_a^b \omega(x) r(x) \mathrm{d}x \\ &= 0 + Q_n(r) \\ &= Q_n(p_{n+1}q) + Q_n(r) \\ &= Q_n(p). \quad \square \end{aligned}$$

Definition 5.11

(i) Eine Funktion $\omega \in L^1(a,b)$ heißt **zulässige Gewichtsfunktion**, *falls gilt $\omega \geq 0$ und $\int_a^b w(x)\mathrm{d}x > 0$.*

(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.

Satz 5.12
Sei ω eine zulässige Gewichtsfunktion. Dann liefert die durch das Gram-Schmidtsche Orthogonalisierungsverfahren definierte Folge $(p_n)_{n\in\mathbb{N}}$

$$p_{n+1}(x) = x^{n+1} - \sum_{i=0}^{n} \frac{\langle x^{n+1}, p_i \rangle_\omega}{\langle p_i, p_i \rangle_\omega} p_i(x), p_0 = 1$$

das eindeutig bestimmte normierte 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** von $\mathbb{P}_{n+1}$ 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 $\mathbb{P}_n$. Setze

$$\mathbb{P}_n^\perp := \left\{ p \in \mathbb{P}_{n+1} \;\middle|\; \langle p, q \rangle_\omega = 0 \quad \forall\, q \in \mathbb{P}_n \right\} \implies \dim(\mathbb{P}_n^\perp) = 1.$$

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 $\langle p_{n+1}, p_k \rangle_\omega = 0 \quad \forall\, 0 \le k \le n$.
Also folgt $\langle p_{n+1}, q \rangle_\omega = 0 \quad \forall\, q \in \mathbb{P}_n$ □

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, $0 \le j \le k$ alle solche Nullstellen sind.

Zu zeigen: $k = n$ und somit $q = p_{n+1}$.

Annahme: $k < n$: Nach Definition hat $p := p_{n+1} q$ kein Vorzeichenwechsel in (a, b). Da $k < n$, folgt $q \in \mathbb{P}_n$ und somit $\langle p_{n+1}, q \rangle_\omega = 0 \implies \omega p_{n+1} q = 0$ (fast überall) und somit $\omega = 0$ (fast überall). Dies ist ein Widerspruch zur Definition von ω. □

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

$$L_j(x) := \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{(x - x_k)}{(x_j - x_k)}$$

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-Quadratur als Interpolationsquadratur)
Seien 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 die Nullstellen von p_{n+1}. Dann gilt:

$$Q_n(f) = Q_n(p) = I(p).$$

Beweis: (Übungsaufgabe)

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: (Übungsaufgabe)

Bemerkung: Die G.Q.en sind für stetige Funktionen auf kompakten Intervallen konvergent bei Graderhöhung, d.h. $|I(f) - Q_n(f)| \overset{n}{\longrightarrow} \overset{\infty}{} 0$.

Beispiel 5.17

1. **Gauß-Legendre-Quadratur**
   $\omega(x) = 1$, $[-1, 1]$.
   Es gilt $p_n(x) = \frac{(2n)!}{2^n (n!)^2} P_n(x)$, wobei $P_n(x)$ die **Legendre-Polynome** sind mit $P_0(x) = 1,\ P_1(x) = x$, und
   $$P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x).$$
   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 $n = 1$: $Q_1(f) = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)$ („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-Quadratur**

   $\omega(x) = \sqrt{1-x^2}^{-1},\ [-1,1]$.
   $p_n(x) = \frac{1}{2^{n-1}} T_n(x)$ und T_n die Tschebyscheff-Polyome 1. Art mit

   $T_0(x) = 1,\ T_1(x) = x$ und

   $$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x),$$

   $\Longrightarrow\ T_n(x) = \cos(n \arccos(x))$.

   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. **Gauß-Laguerre-Quadratur**

   $\omega(x) = e^{-x},\ [0,\infty)$.
   $p_n(x) = (-1)^n L_n(x)$ und L_n Laguerre-Polynome mit

   $L_0(x) = 1,\ L_1(x) = 1 - x$ und

   $$L_{n+1}(x) = (1 + 2n - x)L_n(x) - n^2 L_{n-1}(x).$$

   Fehler: $I(f) - Q_n(f) = \frac{n+1}{2} \frac{(n!)^2}{(2n+1)!} f^{(2n+2)}(\xi)$.

4. **Gauß-Hermite-Quadratur**

   $\omega(x) = e^{-x^2},\ (-\infty,\infty)$.
   $p_n(x) = 2^n H_n(x)$ und H_n die Hermite Polynome mit
   $H_0(x) = 1,\ H_1(x) = 2x$ und

   $$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x).$$

   Fehler: $I(f) - Q_n(f) = \frac{\sqrt{\pi} n!}{2^{n+1}(2n+1)!} f^{(2n+2)}(\xi)$.

5. **Gauß-Jacobi-Quadratur**

   $\alpha, \beta > -1,\ \omega(x) = (1-x)^\alpha (1+x)^\beta, [-1,1]$.
   $p_n(x) = J_n(x, \alpha, \beta)$ und $J_n(x, \alpha, \beta)$ sind die Jacobi-Polynome, definiert durch

   $$J_n(x, \alpha, \beta) := \frac{1}{2^n n! \omega(x)} \frac{d^k}{dx^n} \left((x^2-1)^n \omega(x)\right).$$

Definition 5.18 (Zusammengesetze Gauß-Quadraturen)
Zu $a < b,\ N \in \mathbb{N}$ setze $a_l := a + lH,\ l = 0, \ldots, N$ mit $H := \frac{b-a}{N}$. Sei $Q_n^l(f), n \in \mathbb{N}$ eine Gauß-Quadratur auf $[a_{l-1}, a_l]$, dann ist durch

$$Q_h(f) := \sum_{l=1}^{N} Q_n^l(f)$$

eine zusammengesetzte Gauß-Quadratur definiert.

Beispiel: (Zusammengesetze 2-Punkt G.Q. mit $n = 1$, $\omega = 1$)
Setze $h = \frac{b-a}{N}$, $a_l = a + lh$ für $l = 0, \ldots, N$. Dann ist die zusammengesetze 2-Punkt G.Q. gegeben durch

$$Q_h(f) = \frac{h}{2} \sum_{j=0}^{N-1} \left(f(a_j + h') + f(a_{j+1} - h') \right)$$

mit $h' = \frac{h}{2} \left(1 - \frac{1}{\sqrt{3}} \right)$.

5.3 Romberg Verfahren

Idee: Anwendung der Richardson Extrapolation auf eine zusammengesetze Quadraturformel, d.h.

$$a(h) = T_h(f)$$

wobei $h_k = h_0 2^{-k}$ gewählt wird (Romberg Folge). 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. Um dies zu beweisen, führen wir zunächst die Bernoulli Polynome ein.

Definition 5.19 (Bernoulli Polynome/Zahlen)
Die durch $B_0(t) = 1$ *und* $\frac{\partial}{\partial x} B_k(t) = B_{k-1}(t)$, $\int_0^1 B_k(t)\mathrm{d}t = 0, k \geq 1$, *definierten Polynome heißen* **Bernoulli Polynome**. *Es ist also*

$$B_0(t) = 1, \quad B_1(t) = t - \frac{1}{2}, \quad B_2(t) = \frac{1}{2}t^2 - \frac{1}{2}t + \frac{1}{12}, \quad \dots$$

Die **Bernoulli Zahlen** *sind gegeben durch*

$$B_k := k! \cdot B_k(0).$$

Lemma 5.20 (Eigenschaften der Bernoulli Polynome)
Für die Bernoulli Polynome gilt:

(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 $k \geq 1$.

Beweis: (ohne Beweis)

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:

$$T_h(f) = \int_a^b f(x)\mathrm{d}x - \sum_{k=1}^{m-1} h^{2k} \frac{B_{2k}}{(2k)!} \left(f^{(2k-1)}(b) - f^{(2k-1)}(a) \right) + O(h^{2m}).$$

Beweis: Sei $\varphi \in C^{2m}(0,1)$ beliebig. Dann gilt mit $B_1' = B_0,\ B_1(0) = \frac{1}{2},\ B_1(0) = -\frac{1}{2}$:

$$\begin{aligned}
\int_0^1 \varphi(t)\mathrm{d}t &= \int_0^1 B_0(t)\varphi(t)\mathrm{d}t \\
&= [B_1(t)\varphi(t)]_{t=0}^1 - \int_0^1 B_1(t)\varphi'(t)\mathrm{d}t \\
&= \tfrac{1}{2}(\varphi(1)+\varphi(0)) - [B_2(t)\varphi'(t)]_{t=0}^1 + \int_0^1 B_2(t)\varphi''(t)\mathrm{d}t \\
&\overset{5.20.i}{=} \tfrac{1}{2}(\varphi(1)+\varphi(0)) - B_2(0)(\varphi'(1)-\varphi'(0)) + \underbrace{\left[B_3(t)\varphi''(t)\right]_{t=0}^1}_{=0\ 5.20.iii} - \int_0^1 B_3(t)\varphi'''(t)\mathrm{d}t \\
&= \cdots \\
&= \tfrac{1}{2}(\varphi(1)-\varphi(0)) - \sum_{k=1}^{m-1} B_{2k}(0)\left(\varphi^{(2k-1)}(1) - \varphi^{(2k-1)}(0)\right) + \int_0^1 B_{2m}(t)\varphi^{(2m)}(t)\mathrm{d}t
\end{aligned}$$

Setze $\varphi_j(t) := hf(x_{j-1}+th), \quad 1 \le j \le n$, dann gilt:

- $\int_0^1 \varphi_j(t)\mathrm{d}t = \int_{x_{j-1}}^{x_j} f(x)\mathrm{d}x$,
- $\varphi_j^{(k-1)}(t) = h^k f^{(k-1)}(x_{j-1}+th)$,
- $\varphi_j(1) = hf(x_j) = \varphi_{j+1}(0)$,
- $\varphi_j^{(2k-1)}(1) = \varphi_{j+1}^{(2k-1)}(0)$.

Daher gilt:

$$\begin{aligned}
\int_a^b f(x)\mathrm{d}x &= \sum_{j=1}^n \int_{x_{j-1}}^{x_j} f(x)\mathrm{d}x = \sum_{j=1}^n \int_0^1 \varphi_j(t)\mathrm{d}t \\
&= \sum_{j=1}^n \tfrac{1}{2}(\varphi_j(0)+\varphi_j(1)) - \sum_{j=1}^n \sum_{k=1}^{m-1} B_{2k}(0)\left(\varphi_j^{(2k-1)}(1) - \varphi_j^{(2k-1)}(0)\right) \\
&\quad + \sum_{j=1}^n \int_0^1 B_{2m}(t)\varphi_j^{(2m)}(t)\mathrm{d}t \\
&= \sum_{j=1}^n \tfrac{h}{2}(f(x_j)+f(x_{j-1})) - \sum_{k=1}^{m-1} B_{2k}(0)\left(\varphi_k^{(2k-1)}(1) - \varphi_k^{(2k-1)}(0)\right) \\
&\quad + \sum_{j=1}^n \int_0^1 B_{2m}(t)h^{2m+1}f^{(2m)}(x_{j-1}+th)\mathrm{d}t \\
&= T_h(f) - \sum_{k=0}^{m-1} B_{2k}(0)\left(f^{(2k-1)}(b) - f^{(2k-1)}(a)\right)h^{2k} \\
&\quad + h^{2m}\left[h\sum_{j=1}^m \int_0^1 B_{2m}(t)f^{(2m)}(x_{j-1}+(h))\mathrm{d}t\right].
\end{aligned}$$

Der letzte Term ist $O(h^{2m})$, falls $[\cdot]$ durch eine Konstante unabhängig von h abgeschätzt werden kann. Wir erhalten

$$\begin{aligned}
\left|h\sum_{j=1}^n \int_0^1 B_{2m}(t)f^{(2m)}(x_{j-1}(h))\mathrm{d}t\right| &\le h\sum_{j=1}^n \|B_{2m}\|_\infty \|f(2m)\|_\infty \\
&= n\cdot h\cdot \|B_{2m}\|_\infty \cdot \left\|f^{(2m)}\right\|_\infty \\
&= (b-a)\|B_{2m}\|_\infty \cdot \left\|f^{(2m)}\right\|_\infty = \text{konstant.}
\end{aligned}$$

□

Bemerkung: Die Summenformel zeigt die assymptotische Entwicklung und dass die Trapezregel auch ohne Extrapolation sehr gut für die Integration periodischer Funktionen geeignet ist.

5.4 Fehlerdarstellung nach Peano

Ziel: Wir wollen das Fehlerfunktional R_n, definiert durch $R_n(f) = I(f) - I_n(f)$ als Integral darstellen, um einen abstrakten Zugang zu Fehlerdarstellungen zu erhalten.

Definition 5.22
Ein lineares Funktional $R : C^{k+1}(a,b) \longrightarrow \mathbb{R}$ *heißt* **zulässig**, *falls es entweder aus einer Auswertung* $f^{(\nu)}(x_0)$, $x_0 \in [a,b]$, $0 \le \nu \le k$, *aus einem gewichteten Integral* $\int_{a_0}^{b_0} \omega(x) f^{(\nu)}(x)\mathrm{d}x$, $a_0, b_0 \in [a,b]$, $0 \le \nu \le k$ *oder aus einer endlichen Linearkombination solcher Funktionale besteht.*

Beispiel 5.23

(i) Fehlerfunktionale von Quadraturformeln sind zulässig, $R_n(f) = I(f) - I_n(f)$.

(ii) Fehlerfunktionale 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 $v(t) := R(K(t,\cdot))$ ist eine Abbildung von $[a,b] \longrightarrow \mathbb{R}$. Analog wird durch $w(x) := \int_a^b K(t,x)u(t)\mathrm{d}t$ eine Funktion $w \in C^{k+1}(a,b)$ definiert und es ist $R\left(\int_a^b K(t,\cdot)u(t)\mathrm{d}t\right) = R(w) \in \mathbb{R}$.

Lemma 5.24
Für ein nach Definition 5.21 zulässiges Funktional gilt die Vertauschungsregel
$R\left(\int_a^b K(t,\cdot)u(t)\mathrm{d}t\right) = \int_a^b R(K(t,\cdot))u(t)\mathrm{d}t$ für alle $K \in C^{k+1}\left((a,b)^2\right)$, $u \in C^0(a,b)$.

Beweis: Seien $w(x) = \int_a^b K(t,x)u(t)\mathrm{d}t$ und $v(t) = R(K(,\cdot))$.

Zu zeigen: $R(w) = \int_a^b v(t)u(t)\mathrm{d}t$.

Wegen der Linearität des Integrals reicht es 2 Fälle zu untersuchen:

(i) $R(f) = f^{(\nu)}(x_0)$,

(ii) $R(f) = \int_{a_0}^{b_0} \omega(x) f^{(\nu)}\mathrm{d}x$.

Zu (i): $R(w) = \frac{d^\nu}{dx^\nu} \int_a^b K(t,x)u(t)\mathrm{d}t = \int_a^b K^{(\nu)}(t,x)u(t)\mathrm{d}t = \int_a^b v(t)u(t)\mathrm{d}t$.

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 $f \in C^{k+1}(a,b)$:

$$f(x) = (P_k f)(x) + \left(K_k(\cdot, x), f^{(k+1)}\right)$$

mit $(P_k f)(x) = \sum_{j=0}^{k} f^{(j)}(a)\frac{(x-a)^j}{j!} \in \mathbb{P}_k$ und $K_k(t,x) = \frac{1}{k!}(x-t)_+^k$.

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 (Fehlerdarstellung nach Peano)
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 gilt 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: $R(f) \overset{5.25}{=} R((P_k f) + (K_k(\cdot, x), f^{(k+1)})) \overset{R \text{ linear}}{=} R(P_k f) + R\left(\int_a^b K_k(t,\cdot) f^{(k+1)}(t)\mathrm{d}t\right)$
$\overset{\text{Vor. } 5.24}{=} \int_a^b R(K_k(t,\cdot)) f^{(k+1)}(t)\mathrm{d}t = \int_a^b K(t) f^{(k+1)}(t)\mathrm{d}t.$ □

Folgerung 5.27
Seien die Voraussetzungen von Satz 5.26 erfüllt, so gilt:
Hat der Peano Kern $K(t)$ für $f \in [a,b]$ 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:

$$\begin{aligned} R(f) &= \int_a^b K(t) f^{(n+1)}(t)\mathrm{d}t \\ &\overset{\text{MWS}}{=} f^{(n+1)}(\xi) \int_a^b K(t)\mathrm{d}t \quad \text{da } K \text{ kein Vorzeichenwechsel hat.} \end{aligned}$$

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 (Anwendung auf die Simpsonregel)
Es ist

$$R(f) := R_3(f) = \frac{1}{3}f(-1) + \frac{4}{3}f(0) + \frac{1}{3}f(1) - \int_{-1}^{1} f(x)\mathrm{d}x.$$

Es gilt $R(p) = 0 \quad \forall\, p \in \mathbb{P}_3$, d.h. $k = n = 3, a = -1, b = 1$ in Satz 5.26. Also folgt

$$R(f) = \int_{-1}^{1} K(t) f^{(4)}(t)\mathrm{d}t$$

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: $\quad (-1-t)_+^3 = 0, \qquad (1-t)_+^3 = (1-t)^3$

$$(-t)_+^3 = \begin{cases} -t^3 & : \ -1 \le t \le 0 \\ 0 & : \ 0 \le t \le 1 \end{cases}, \qquad \int_{-1}^{1}(x-t)_+^3\mathrm{d}x = \int_t^1 (x-t)^3\mathrm{d}x = \tfrac{1}{4}(1-t)^4.$$

$$\Longrightarrow K(t) = \begin{cases} \frac{1}{72}(1-t)^3(1+3t) & : \ 0 \le t \le 1 \\ K(-t) & : \ -1 \le t < 0 \end{cases}$$

$$\Longrightarrow K(t) \ge 0 \text{ für } t \in [-1,1].$$

Mit Folgerung 5.27 gilt also

$$\begin{aligned} R(f) &= f^{(4)}(\xi)\frac{1}{24}R(x^4) = f^{(4)}(\xi)\frac{1}{24}\left(\frac{1}{3}\cdot 1 + \frac{4}{3}\cdot 0 + \frac{1}{3}\cdot 1 - \int_{-1}^{1} x^4\mathrm{d}x\right) \\ &= f^{(4)}(\xi)\frac{1}{90}. \end{aligned}$$

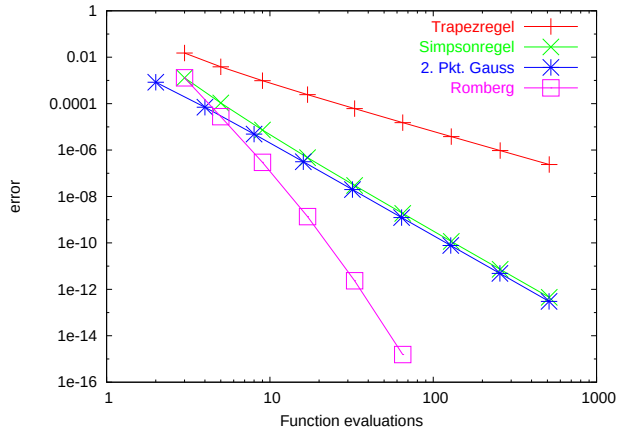
Definition 5.29 (Experimentelle Konvergenzordnung (*EOC*))
Sei $f \in C^k(a,b)$ und $I : C^k(a,b) \longrightarrow \mathbb{R}$ ein Funktional, I_h eine Quadraturformel, die I auf einer Zerlegung der Feinheit h approximiert. Gelte $h_1 > h_2$.

Die experimentelle Konvergenz *$EOC\,(e_{h_1 \longrightarrow h_2})$ (engl. **e**xperimental **o**rder of **c**onvergence) für den Fehler $e_h := |I(f) - I_h(f)|$ ist definiert durch*

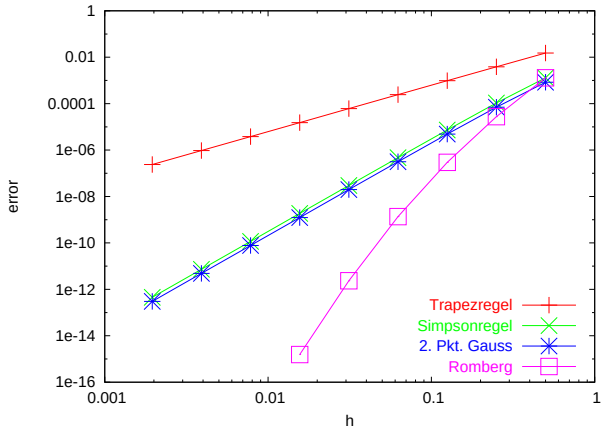
$$EOC\,(e_{h_1 \longrightarrow h_2}) := \frac{\log\left(\frac{e_{h_1}}{e_{h_2}}\right)}{\log\left(\frac{h_1}{h_2}\right)}.$$

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.

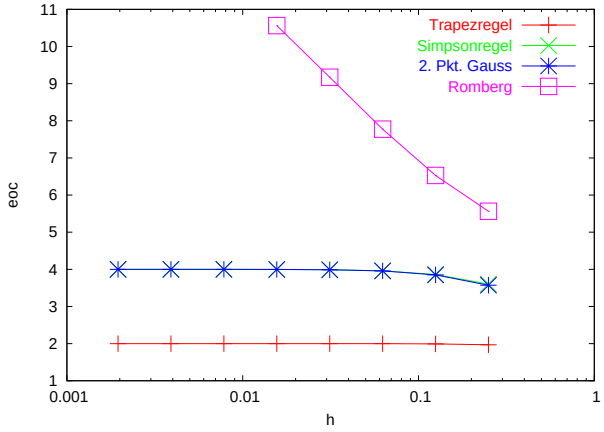
Beispiel 5.30 (Fehler der Approximierung der Integration)
Gegeben seien $I = [0,1]$ und $f(x) := \frac{1}{x+1}$, $g(x) := \frac{3}{2}\sqrt{x}$. Es gilt $\int_0^1 \frac{1}{x+1}\mathrm{d}x = \ln(2)$, $\int_0^1 \frac{3}{2}\sqrt{x}\mathrm{d}x = 1$.
Die Abbildung 5.2 ziegt das Verhalten des Approximationsfehlers von 4 Verfahren: Trapezregel (rot), Simposon-Regel (grün), zwei-Punkt Quadratur (blau) und Romberg Verfahren (lila). ***Typ 1*** ist der Fehler im Vergleich zu der Anzahl der Funktionauswertungen, im Prinzip ein Maß für den Berechnungsaufwand. ***Typ 2*** ist der Fehler im Vergleich zu h, d.h. zu der Unterteilung bei den zusammengesetzen Quadraturen. ***Typ 3*** ist die EOC im Verhältnis zu h.



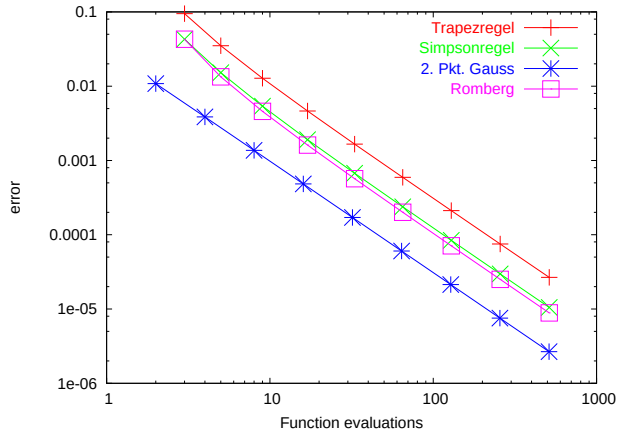
f Typ 1



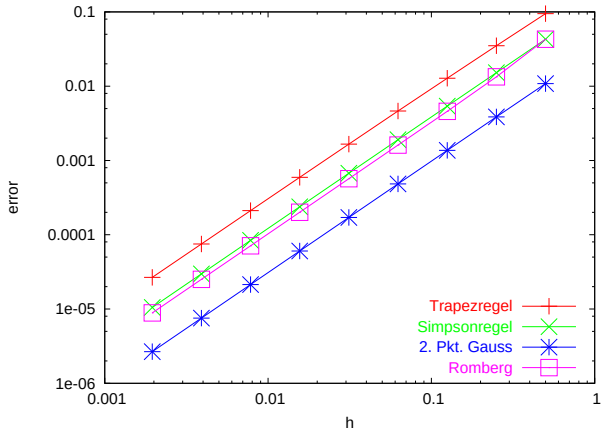
f Typ 2



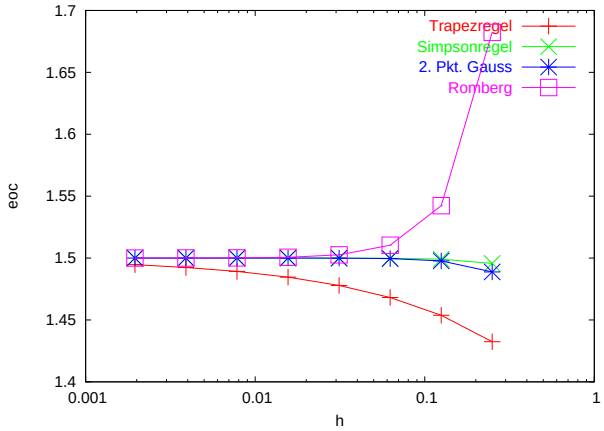
f Typ 3



g Typ 1



g Typ 2



g Typ 3

Abbildung 5.2: Fehler der Quadraturen

Index