Einführung in die Numerik (Potschka)

Robin Heinemann

7. Juni 2017

Inhaltsverzeichnis

0 Einführung

Beispiel 0.1 Simulation einer Pendelbewegung
Modellannahmen:

- Masse m an Stange
- keine Reibung
- Stange: Gewicht 0, starr, Länge l

- Auslenkung ϕ

Erste Fehlerquelle: Modellierungsfehler
Modellgleichungen:

$$F_T(\phi) = -m \cdot g \sin\phi$$

Konsistenzcheck:

$$F_T(0) = 0 \qquad \text{(Ruhelage)}$$

$$F_T\left(\frac{\pi}{2}\right) = F_G = -mg$$

Bewegungsgleichungen:

- Weg $s(t)$
- $\frac{\mathrm{d}s}{\mathrm{d}t} =: v(t)$ Geschwindigkeit
- $\frac{\mathrm{d}v}{\mathrm{d}t} =: a(t)$ Beschleunigung

Beziehungen:

- Bogenlänge $s(t) = l\phi(t)$
- 2. Newton'sches Gesetz ($F = ma$)

$$-mg\sin\phi(t) = m\frac{\mathrm{d}}{\mathrm{d}t}v(t) = m\frac{\mathrm{d}^2}{\mathrm{d}t^2}s(t) = ml\frac{\mathrm{d}^2}{\mathrm{d}t^2}\phi(t)$$

$\Longrightarrow$ DGL 2. Ordnung

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\phi(t) = -\frac{g}{l}\sin\phi(t) \quad t \geq 0$$

Für eindeutige Lösung braucht man zwei Anfangsbedingungen:

$$\phi(0) = \phi_0 \quad \frac{\mathrm{d}}{\mathrm{d}t}\phi(0) = u_0$$

Lösung bei kleiner Auslenkung: Linearisiere um $\phi = 0$

$$\sin\phi = \phi - \frac{1}{3!}\phi^3 + \cdots \approx \phi$$

$$\Longrightarrow \frac{\mathrm{d}^2}{\mathrm{d}t^2}\phi(t) = -\frac{g}{l}\phi(t)$$

Für $u_0 = 0$ findet man mit dem Ansatz $\phi(t) = A\cos(\omega t)$:

$$-\omega^2 A\cos(\omega t) = -\frac{g}{l}A\cos(\omega t)$$

die Lösung:

$$\phi(t) = \phi_0 \cos\left(\sqrt{\frac{g}{l}}t\right)$$

Fehlerquelle: Abschneidefehler.
Numerische Lösung:
Setze $u(t) := \frac{\mathrm{d}}{\mathrm{d}t}\phi(t)$

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\phi \\ u\end{pmatrix} = \begin{pmatrix}u \\ -\frac{g}{l}\sin(\phi)\end{pmatrix}$$

Approximation mit Differenzenquotienten

$$\begin{pmatrix}u(t) \\ -\frac{g}{l}\sin\phi(t)\end{pmatrix} = \frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\phi \\ u\end{pmatrix} = \lim_{\Delta t\to 0}\frac{1}{\Delta t}\begin{pmatrix}\phi(t+\Delta t)-\phi(t) \\ u(t+\Delta t)-u(t)\end{pmatrix} \approx \underset{\substack{\downarrow \\ >0,\text{ klein}}}{\frac{1}{\Delta t}}\begin{pmatrix}\phi(t+\Delta t)-\phi(t) \\ u(t+\Delta t)-u(t)\end{pmatrix}$$

Fehlerquelle: Diskretisierungsfehler
Auf Gitter $t_n = n\Delta t$ mit Werten $\phi_n = \phi(n\Delta t), u_n = u(n\Delta t)$:

$$\phi_{n+1} = \phi_n + \Delta t u_n, u_{n+1} = u_n - \Delta t\frac{g}{l}\phi_n$$

Kleinerer Diskretisierungsfehler mit zentralen Differenzen:

$$-\frac{g}{l}\sin\phi(t) = \frac{\mathrm{d}^2}{\mathrm{d}t^2}\phi(t) \approx \frac{\phi(t+\Delta t)-2\phi(t)+\phi(t-\Delta t)}{\Delta t^2}$$

Rekursionsformel:

$$\phi_{n+1} = 2\phi_n - \phi_{n-1} - \Delta t^2\frac{g}{l}\sin\phi_n, n \geq 1$$

mit $\phi_1 = \phi_0 + \Delta t n_0$ (Expliziter Euler)
Letzte Fehlerquelle: Rundungsfehler

1 Fehleranalyse

1.1 Zahldarstellung und Rundungsfehler

Anforderung: Rechnen mit reellen Zahlen auf dem Computer.
Problem: Speicher endlich ($\implies$ endliche Genauigkeit).
Lösung: Gleitkommazahlen, ein **Kompromiss** zwischen:

- Umfang darstellbarer Zahlen
- Genauigkeit
- Geschwindigkeit einfacher Rechenoperationen (+, -, ·, /)

Alternativen:

- Fixkommazahlen
- logarithmische Zahlen
- Rationalzahlen

Definition 1.1 Eine (normalisierte) Gleitkommazahl zur Basis $b \in \mathbb{N}, b \geq 2$, ist eine Zahl $x \in \mathbb{R}$ der Form

$$x = \pm m \cdot b^{\pm e}$$

mit der Mantisse $m = m_1 b^{-1} + m_2 b^{-2} + \ldots \in \mathbb{R}$ und dem Exponenten $e = e_{s-1} b^{s-1} + \cdots + e_0 b^0 \in \mathbb{N}$, wobei $m_i, e_i \in \{0, \ldots, b-1\}$. Für $x \neq 0$ ist die Darstellung durch die Normierungsvorschrift $m \neq 0$ eindeutig. Für $x = 0$ setzt man $m = 0$.

Beispiel 1.2 ($b = 10$)

- m_i: i -te Nachkommastelle der Mantisse
- e: Verschiebt das Komma um e Stellen.

$$0.314 \times 10^1 = 3.14$$
$$0.123 \times 10^6 = 123\,000$$

Auf dem Rechner stehen nur endlich viele Stellen zur Verfügung:
r Ziffern + 1 Vorzeichen für Mantisse m
s Ziffern + 1 Vorzeichen für Exponenten.
Für $x = \pm[m_1 b^{-1} + \cdots + m_r b^{-r}] \cdot b^{\pm[e_{s-1} b^{s-1} + \cdots + e_0 b^0]}$ muss man also nur $(\pm)[m_1 \ldots m_r](\pm)[e_{s-1} \ldots e_0]$ abspeichern. Wählt man $b = 2$, so gilt $m_i, e_i \in \{0, 1\}$ und x kann mit $2 + r + s$ Bits gespeichert werden (Maschinenzahlen). Maschinenzahlen bilden das numerische Gleitkommagitter $A = A(b, r, s)$

Beispiel 1.3 ($b = 2, r = 3, s = 1$)

$$\begin{aligned} m &= \frac{1}{2} + m_2 \frac{1}{4} + m_3 \frac{1}{8} \in \{\frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}\} \\ e &= e_0 \in \{0, 1\} \end{aligned}$$

Da A endlich ist, gibt es eine größte/kleinste darstellbare Zahl:

$$\begin{aligned} x_{\{}min/max\} &= \pm(b-1)[b^{-1} + \cdots + b^{-r}] \cdot b^{(b-1)[b^{s-1} + \cdots + b^0]} \\ &= \pm\left(1 - b^{-r}\right) \cdot b^{(b^s - 1)} \end{aligned}$$

sowie eine kleinste positive/größte negative Zahl:

$$\begin{aligned} x_{posmin/negmax} &= \pm b^{-1} \cdot b^{-(b-1)[b^{s-1} + \cdots + b^0]} \\ &= b^{-b^s} \end{aligned}$$

Das gängigste Format ist das IEEE-Format, das auch hinter dem Python-Datentyp float steht:

$$x = \pm m \cdot 2^{c-1022}$$

Dieser Datentyp ist 64 Bit (8 Byte) groß (doppelte Genauigkeit, double). Davon speichert 1 Bit das Vorzeichen, 52 Bits die Mantisse $m = 2^{-1} + m_2 2^{-2} + \cdots + m_{53} 2^{-53}$ und 11 Bits die Charakteristik $c = c_0 2^0 + \cdots + c_{10} 2^{10}$, mit $m_i, c_i \in \{0, 1\}$. Es gibt folgende spezielle Werte:

- Alle $c_i, m_i = 0 : x = \pm 0$
- Alle $m_i = 0, c_i = 1 : x = \pm\infty$
- Ein $m_i \neq 0$, alle $c_i = 1$: x = NaN (not a number)

Für c bleibt damit ein Bereich von $\{0, \dots, 2046\}$ beziehungsweise $c - 1022 \in \{-1022, \dots, 1024\}$. Damit gilt:

- $x_{max} \approx 2^{1024} \approx 1.8 \times 10^{308}, x_{min} = -x_{max}$
- $x_{posmin} = 2^{-1022} \approx 2.2 \times 10^{-308}, x_{negmax} = -x_{posmin}$

Ausgangsdaten $x \in \mathbb{R}$ einer numerischen Aufgabe und die Zwischenergebnisse einer Rechnung müssen durch Maschinenzahlen dargestellt werden. Für Zahlen des „zulässigen" Bereichs $D = [x_{min}, x_{negmax}] \cup \{0\}[x_{posmin}, x_{max}]$ wird eine Rundungsoperation $\mathrm{rd} : D \to A$ verwendet, die

$$|x - \mathrm{rd}\, x| = \min_{y \in A} |x - y| \forall x \in D$$

erfüllt.

Beispiel 1.4 (Natürliche Rundung im IEEE-Format)

$$\mathrm{rd}(x) = \mathrm{sgn}(x) \cdot \begin{cases} 0, m_1, \dots, m_{53} \cdot 2^e & m_{54} = 0 \\ (0, m_1, \dots, m_{53} + 2^{-53}) \cdot 2^e & m_{54} = 1 \end{cases}$$

Rundungsfehler:

- absolut:
$$|x - \mathrm{rd}(x)| \leq \frac{1}{2} b^{-r} b^e$$
- relativ:
$$\left| \frac{x - \mathrm{rd}(x)}{x} \right| \leq \frac{1}{2} \frac{b^{-r} b^e}{|m| b^e} \leq \frac{1}{2} b^{-r+1}$$

Der relative Fehler ist für $x \in D \setminus \{0\}$ beschränkt durch die „Maschienengenauigkeit"

$$eps = \frac{1}{2} b^{-r+1}$$

Für $x \in D$ ist $\mathrm{rd}(x) = x(1 + \varepsilon), |(|\varepsilon) \leq eps$. Für das IEEE-Format (double)

$$eps = \frac{1}{2} 2^{-52} \approx 10^{-16}$$

Arithmetische Grundoperationen

$$* : \mathbb{R} \times \mathbb{R} \to \mathbb{R}, * \in \{x, -, +, /\}$$

werden auf dem Rechner ersetzt durch Maschinenoperationen:

$$\circledast : A \times A \to A$$

Dies ist normalerweise für $x, y \in A$ und $x * y \in D$ realisiert durch

$$x \circledast y := \mathrm{rd}(x * y) = (x * y)(1 + \varepsilon), |\varepsilon| \leq eps$$

Dazu werden die Operationen maschinenintern (unter Verwendung einer längeren Mantisse) ausgeführt, normalisiert und dann gerundet. Im Fall $x*y \notin D$ gibt es eine Fehlermeldung (overflow, underflow) oder das Ergebnis NaN. Achtung: Das Assoziativ- und Distributivgesetz gilt dann nur näherungsweise. Im Allgemeinen ist für $x, y, z \in A$

$$\begin{aligned}(x \oplus y) \oplus z &\neq x \oplus (y \oplus z) \\ (x \oplus y) \odot z &\neq (x \odot z) \oplus (y \odot z)\end{aligned}$$

Insbesondere gilt für $|y| \leq \frac{|x|}{b} eps$

$$x \oplus y = x$$

Damit ergibt sich eine alternative Charakterisierung der Maschienengenauigkeit: eps ist die kleinste positive Zahl in A, sodass $1 \oplus eps \neq 1$

1.2 Konditionierung numerischer Aufgaben

Eine numerische Aufgabe wird als **gut konditioniert** bezeichnet, wenn eine kleine Störung in den Eingangsdaten (Messfehler, Rundungsfehler) auch nur eine kleine Änderung der Ergebnisse zur Folge hat.

Beispiel 1.5 (Schnittpunkt von Geraden) Zwei Geraden, die sich (annähernd) rechtwinklig treffen sind gut konditioniert.
Zwei Geraden, die sich unter einem stumpfen, oder spitzen Winkel treffen sind schlecht konditioniert.

Beispiel 1.6 (Lineares Gleichungssystem)

$$\begin{pmatrix} 1 & 10^6 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \implies \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 & -10^6 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$$b = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \implies x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$b = \begin{pmatrix} 1 \\ 10^{-3} \end{pmatrix} \implies x = \begin{pmatrix} -999 \\ 10^{-3} \end{pmatrix} \not\approx \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$\implies$ schlecht konditioniert.

Definition 1.7 Eine **numerische Aufgabe** berechnet aus Eingangsgrößen $x_j \in \mathbb{R}, j = 1, \ldots, m$ unter der funktionellen Vorschrift $f(x_1, \ldots, x_m), i = 1, \ldots, n$ Ausgangsgrößen $y_i = f_i(x_1, \ldots, x_m)$

$$y = f(x), f : \mathbb{R}^m \to \mathbb{R}^n$$

Beispiel 1.8 (Lösung eines LGS) $Ay = x, f(x) = A^{-1}x$

Definition 1.9 Fehlerhafte Eingangsgrößen $x_i + \Delta x_i$ (Δx_i: Rundungsfehler, Maschienenfehler) ergeben fehlerhafte Resultate $y_i + \Delta y_i$. Wir bezeichnen $|\Delta y_i|$ als den absoluten Fehler und $\left|\frac{\Delta y_i}{y_i}\right|$ für $y_i \neq 0$ als den relativen Fehler.

1.2.1 Differentielle Fehleranalyse

Annahmen:

- kleine relative Datenfehler $|\Delta x_i| \ll |x_i|$
- f_i stetig partiell differenzierbar nach allen x_i

Dann gilt:

$$\begin{aligned} y_i = f_i(x_i), y_i + \Delta y_i &= f_i(x + \Delta x) \\ \implies \Delta y_i &= f_i(x + \Delta x) - f(x) \end{aligned}$$

Taylorentwicklung

$$= \sum_{j=1}^{m} \frac{\partial f_i}{\partial x_j} \Delta x_j + R_i^f(x, \Delta x)$$

mit einem Restglied R_i^f, das für $|\Delta x| = \max_{j=1,\dots,m} |\Delta x_j| \to 0$ schneller gegen 0 geht als $|\Delta x|$. Wenn f sogar zweimal stetig differenzierbar ist, gilt sogar, dass

$$\left| R_i^f(x, \Delta x) \right| \leq c|\Delta x|^2, c \in \mathbb{R}$$

Definition 1.10 (Landau-Notation) Seien $g, h : \mathbb{R}_+ \to \mathbb{R}, t \to 0^+$. Wir schreiben:

- $g(t) = \mathcal{O}(h(t)) :\iff \exists t_0, c \in \mathbb{R}_+ : \forall t \in (0, t_0] : |g(t)| \leq c|h(t)|$
- $gt = \sigma(ht) :\iff \exists t_0 \in \mathbb{R}_+, c : \mathbb{R}_+ \to \mathbb{R}, \lim_{t \to 0^+} c(t) = 0 : \forall t \in (0, t_0] : |g(t)| \leq c(t)|h(t)|$

Bemerkung 1.11

- Analoge Schreibweise für $t \to \infty$
- $\mathcal{O}$ und σ sind Symbole, keine Funktionen

$$\mathcal{O}(t^2) + \mathcal{O}(t^3) + \mathcal{O}(2t^2) = \mathcal{O}(t^2) \not\Longrightarrow \mathcal{O}(t^3) + \mathcal{O}(2t^2) = 0$$

- $\sigma(t^n)$ ist stärker als $\mathcal{O}(t^n)$: $\sigma(t^n) + \mathcal{O}(t^n) = \mathcal{O}(t^n)$
- $\mathcal{O}(t^{n+1})$ ist stärker als $\sigma(t^n)$: Wähle $c(t) = t$!

Beispiel 1.12 Ist $g(t)$ zweimal stetig differenzierbar, so gilt mit Taylor

$$g(t+\Delta t) = g(t) + \Delta t g'(t) + \frac{1}{2}\Delta t^2 g''(\tau), \tau \in [t, t+\Delta t]$$

$$\implies \frac{1}{\Delta t}(g(t+\Delta t) - g(t)) = g'(t) + \mathcal{O}(\Delta t)$$

Damit folgt dass Δy_i in erster Näherung, das heißt bis auf eine Größe der Ordnung $\mathcal{O}\left(|\Delta x|^2\right)$ gleich

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

ist. Schreibweise

$$\Delta y_i \doteq \sum_{j=1}^{m} \frac{\partial f_i}{\partial x_j}(x)\Delta x_j$$

Für den komponentenweisen relativen Fehler gilt

$$\frac{\Delta y_i}{y_i} \doteq \sum_{j=1}^{m} \frac{\partial f_i}{\partial x_j}(x)\frac{\Delta x_j}{y_i} = \sum_{j=1}^{m} \underbrace{\frac{\partial f_i}{\partial x_j}(x)\frac{x_j}{f_i(x)}}_{=:k_{ij}(x)} \frac{\Delta x_j}{x_j}$$

Vernachlässigt haben wir dabei

$$\left|\frac{R_i^f(x_j, \Delta x)}{y_i}\right| = \mathcal{O}\left(\frac{|\Delta x|^2}{|y_i|}\right)$$

Diese Vernachlässigung ist nur zulässig falls

$$|\Delta x| = \sigma(|y_i|), i = 1, \ldots, n$$

damit

$$\mathcal{O}\left(\frac{|\Delta x|^2}{|y_i|}\right) = \sigma(|\Delta x|)$$

(stärker als $\mathcal{O}(|\Delta x|)$)

Definition 1.13 Die Größen $k_{ij}(x)$ heißen (relative) Konditionszahlen von f im Punkt x. Sie sind Maß dafür, wie sich kleine relative Fehler in den Ausgangsdaten x_j auf das Ergebnis y_i auswirken. Sprechweise:

- $|k_{ij}(x)| \gg 1$: Die Aufgabe $y = f(x)$ ist schlecht konditioniert
- sonst: Die Aufgabe $y = f(x)$ ist gut konditioniert
- $|k_{ij}(x)| < 1$: Fehlerdämpfung

- $|k_{ij}(x)| > 1$: Fehlerverstärkung.

Bemerkung 1.14 Man kann auch Störungen in f betrachten.

Beispiel 1.15 Implizit gegebene Aufgaben. Für $n = m$ sie y die gegebene Eingangsgröße und ein x mit $f(x) = y$ die Ausgabe (zum Beispiel: $f(x) = Ax + b$) Die differentielle Fehleranalyse auf der Umkehrfunktion $x = f^{-1}(y)$ liefert unter geeigneten Annahmen.

$$\frac{\Delta x_i}{x_i} \doteq \sum_{j=1}^{n} k_{ij}^{-1}(y) \frac{\Delta y_j}{y_j}, k_{ij}^{-1} = \frac{\partial f_i^{-1}}{\partial y_j}(y) \frac{y_j}{x_i}$$

Wir definieren die Matrizen

$$K^{-1}(y) = \left(k_{ij}^{-1}\right)_{i,j=1}^{n}, K(x) = (k_{ij}(x))_{i,j=1}^{n}$$

und betrachten deren Produkt:

$$\begin{aligned}
(K^{-1}(y)K(x))_{ij} &= \sum_{l=1}^{n} k_{il}^{-1}(y) k_{lj}(x) \\
&= \sum_{l=1}^{n} \frac{\partial f_i^{-1}}{\partial y_l}(y) \frac{y_l}{x_i} \frac{\partial f_l}{\partial x_j}(x) \frac{x_j}{y_l} \\
&= \frac{x_j}{x_i} \sum_{l=1}^{n} \frac{\partial f_i^{-1}}{\partial y_l} \frac{\partial f_l}{\partial x_j} = \frac{x_j}{x_i} \frac{\mathrm{d}}{\mathrm{d}x_j} \left(f_i^{-1}(f(x))\right) \\
&= \frac{x_j}{x_i} \frac{\mathrm{d}x_i}{\mathrm{d}x_j} = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{sonst} \end{cases}
\end{aligned}$$

K^{-1} ist gerade das Inverse von K.

Wiederholung: Numerische Aufgabe

$$f : x \in \mathbb{R}^m \mapsto y \in \mathbb{R}$$

Konditionszahlen:

$$\frac{\Delta y_i}{y_i} \doteq \sum_{j=1}^{m} k_{ij}(x) \frac{\Delta x_j}{x_j}$$

$$k_{ij}(x) = \frac{\partial f_i}{\partial x_j}(x) \frac{x_j}{f_i(x)}$$

1.2.2 Arithmetische Grundoperationen

Addition: $f(x_1, x_2) = x_1 + x_2, x_1, x_2 \in \mathbb{R} \setminus \{0\}$

$$k_{1j}(x) = \frac{\partial f}{\partial x_j}\frac{x_j}{f} = 1\frac{x_j}{x_1 + x_2} = \frac{1}{1 + \frac{x_{\bar{j}}}{x_j}}$$
$$\bar{j} = \begin{cases} 2 & j = 1 \\ 1 & j = 2 \end{cases}$$

Die Addition ist schlecht konditioniert für $x_1 \approx -x_2$.

Definition 1.16 (Auslöschung) Unter Auslöschung versteht man den Verlust von Genauigkeit bei der Subtraktion von Zahlen gleichen Vorzeichens.

Beispiel 1.17 $b = 10, r = 4, s = 1$

$$\begin{aligned} x_1 &= 0.112\,587 \times 10^2 & \mathrm{rd}(x_1) &= 0.1126 \times 10^2 \\ x_2 &= 0.112\,448 \times 10^2 & \mathrm{rd}(x_1) &= 0.1124 \times 10^2 \\ x_1 + x_2 &= 0.225\,035 \times 10^2 & \mathrm{rd}(x_1) \oplus \mathrm{rd}(x_2) &= 0.2250 \times 10^2 \\ x_1 - x_2 &= 0.129 \times 10^{-1} & \mathrm{rd}(x_1) \ominus \mathrm{rd}(x_2) &= -0.2 \times 10^{-1} & \text{(Großer Fehler)} \end{aligned}$$

Multiplikation: $y = f(x_1, x_2) = x_1 x_2$

$$k_{1j}(x) = \frac{\partial f}{\partial x_j}\frac{x_j}{f} = x_{\bar{j}} \cdot \frac{x_j}{x_1 x_2} = 1$$

$\implies$ gut konditioniert

Beispiel 1.18 (Lösungen quadratischer Gleichungen) Für $p, q \in \mathbb{R}$ betrachte:

$$0 = y^2 - py + q$$
$$y_{1,2} = y_{1,2}(p, q) = \frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}$$

nach Vieta $p = y_1 + y_2, q = y_1 \cdot y_2$

$$\begin{aligned}
1 &= \frac{\mathrm{d}p}{\mathrm{d}p} = \frac{\partial y_1}{\partial p} + \frac{\partial y_2}{\partial p} \\
0 &= \frac{\mathrm{d}q}{\mathrm{d}p} = \frac{\partial y_1}{\partial p} y_2 + y_1 \frac{\partial y_2}{\partial p} \\
\implies (y_2 - y_1) \frac{\partial y_2}{\partial p} &= y_2 \\
\implies \frac{\partial y_2}{\partial p} &= \frac{y_2}{y_2 - y_1} \\
\implies \frac{\partial y_1}{\partial p} &= \frac{y_1}{y_1 - y_2} \\
0 &= \frac{\mathrm{d}p}{\mathrm{d}q} = \frac{\partial y_1}{\partial q} + \frac{\partial y_2}{\partial q} \\
1 &= \frac{\mathrm{d}q}{\mathrm{d}q} = \frac{\partial y_1}{\partial q} y_2 + y_1 \frac{\partial y_2}{\partial q} \\
\implies 1 &= (y_2 - y_1) \frac{\partial y_1}{\partial q} \\
\implies \frac{\partial y_1}{\partial q} &= \frac{1}{y_2 - y_1} \\
\implies \frac{\partial y_2}{\partial q} &= -\frac{1}{y_2 - y_1} \\
k_{11}(x) &= \frac{\partial y_1}{\partial p} \frac{p}{y_1} = \frac{y_1}{y_1 - y_2} \frac{y_1 + y_2}{y_1} = \frac{1 + y_2/y_1}{1 - y_2/y_1} \\
k_{12}(x) &= \frac{\partial y_1}{\partial q} \frac{q}{y_1} = \frac{1}{y_2 - y_1} \frac{y_1 y_2}{y_1} = \frac{1}{1 - y_1/y_2}
\end{aligned}$$

Analog für k_{21}, k_{22}
Die Berechnung von y_1, y_2 ist schlecht konditioniert $y_1 \approx y_2$.
Konkretes Beispiel: $p = 4, q = 33.999, y_{1,2} = 2 \pm 10 \times 10^{-1}$

$$k_{12} = \frac{y_2}{y_2 - y_1} = \frac{2 - 10^{-2}}{-2 \times 10^{-2}} = -99.5$$

$\implies$ 100-fache Fehlerverstärkung.

1.3 Stabilität numerischer Algorithmen

Gegeben: Numerische Aufgabe $f : x \in \mathbb{R}^m \mapsto y \in \mathbb{R}^n$

Definition 1.19 (Verfahren / Algorithmus) Unter einem Verfahren / Algorithmus zur (gegebenenfalls näherungsweise) Berechnung von y aus x verstehen wir eine endliche Folge von elementaren Abbildungen $\varphi^{(k)}$, die durch sukzessiv Anwendung einen Näherungswert $\tilde{y}$ zu y liefern.

$$x = x^{(0)} \mapsto \varphi^{(1)}\left(x^{(0)}\right) = x^{(1)} \mapsto \ldots \mapsto \varphi^{(k)}\left(x^{(k-1)}\right) \mapsto \tilde{y} \to y$$

Im einfachsten Fall sind die $\varphi^{(i)}$ arithmetische Grundoperationen. Bei der Durchführung des Algorithmus auf dem Rechner treten in jedem Schritt Fehler auf (Rundungsfehler, Auswertungsfehler, ...), die sich akkumulieren können.

Definition 1.20 (Algorithmus) Ein Algorithmus heißt stabil, wenn die im Verlauf der Rechnung akkumulierten Fehler den durch die Konditionierung der Aufgabe $y = f(x)$ bedingten unvermeidbaren Problemfehler nicht übersteigen.

Beispiel 1.21 (Lösung quadratischer Gleichungen) Annahme: $0 \neq q < p^2/4$
Für $\left|\frac{y_1}{y_2}\right| \gg 1$, das heißt $q \ll \frac{p^2}{4}$, ist die Aufgabe gut konditioniert. Algorithmus: $u = p^2/4, v = u - q, w = \sqrt{v}$.
Im Fall $p < 0$ wird zur Vermeidung von Auslöschung zunächst $\tilde{y}_2 = p/2 - w$ berechnet.
Fehlerfortpflanzung:

$$w = \sqrt{u-q}\begin{cases} \approx \frac{|p|}{2} & q > 0 \\ > \frac{|p|}{2} & q < 0 \end{cases}$$

$$\begin{aligned} \frac{\Delta y_2}{y_2} &\dot{\leq} \left|\frac{\frac{1}{2}p}{\frac{p}{2} - w}\right|\left|\frac{\Delta p}{p}\right| + \left|\frac{-w}{\frac{p}{2} - w}\right|\left|\frac{\Delta w}{w}\right| \\ &= \underbrace{\left|\frac{1}{1 - \frac{2w}{p}}\right|}_{\leq\frac{1}{2}}\left|\frac{\Delta p}{p}\right| + \underbrace{\left|\frac{1}{1 - \frac{p}{2w}}\right|}_{<1}\left|\frac{\Delta w}{w}\right| \end{aligned}$$

Die zweite Wurzel kann so bestimmt werden:

$$A : \tilde{y}_1 = \frac{p}{2} + w, \quad B : \tilde{y}_1 = \frac{q}{\tilde{y}_2}$$

Für $|q| \ll \frac{p^2}{4}$ ist $w \approx \frac{|p|}{2} \implies$ Auslöschung in Variante A

$$\left|\frac{\Delta y_1}{y_1}\right| \doteq \underbrace{\frac{1}{1 + \frac{2w}{p}}}_{\gg 1}\frac{\Delta p}{p} + \underbrace{\frac{1}{1 + \frac{p}{2w}}}_{\gg 1}\frac{\Delta w}{w}$$

$\implies$ Variante A ist instabil. Variante B ist stabil:

$$\left|\frac{\Delta y_1}{y_1}\right| \dot{\leq} \underbrace{\left|\frac{\Delta q}{q}\right|}_{\leq eps} + \underbrace{\left|\frac{\Delta y_2}{y_2}\right|}_{\approx eps}$$

Regel: Bei der Lösung quadratischer Gleichungen sollten nicht beide Wurzeln aus der Lösungsformel berechnet werden.
Konkretes Beispiel: $p = -4, q = 0.01$ (vierstellige Rechnung)

$$u = 4, v = 3.99, w = 1.9974948\ldots, \tilde{y}_2 = -3.997(4981\ldots)$$

$$\tilde{y}_1 = \begin{cases} \text{exakt:} & -0.9925915\ldots \\ A: & -0.003000 \quad \text{(rel. Fehler: 20\%)} \\ B: & -0.002502 \quad \text{(rel. Fehler: } 1.7 \times 10^{-4}\text{)} \end{cases}$$

Auswertung arithmetischer Ausdrücke
Vorwärtsrundungsfehleranalyse: Akkumulation des Rundungsfehlers ausgehend von Startwert.

Beispiel 1.22 $y = f(x_1, x_2) = x_1^2 - x_2^2 = (x_1 - x_2)(x_1 + x_2)$ Konditionierung:

$$\begin{aligned}\left|\frac{\Delta y}{y}\right| &\dot{\leq} \sum_{i=1}^{2} \left|\frac{\partial f}{\partial x_i}\frac{x_i}{f}\right|\left|\frac{\Delta x_i}{x_i}\right| \\ &= \left|2x_1 \frac{x_1}{x_1^2 - x_2^2}\right|\left|\frac{\Delta x_1}{x_1}\right| + \left|-2x_2\frac{x_2}{x_1^2 - x_2^2}\right|\left|\frac{\Delta x_2}{x_2}\right| \\ &\leq 2\frac{x_1^2 + x_2^2}{|x_1^2 - x_2^2|} eps = 2\left|\frac{\left(\frac{x_1}{x_2}\right)^2 + 1}{\left(\frac{x_1}{x_2}\right)^2 - 1}\right| eps\end{aligned}$$

$\Longrightarrow$ schlecht konditioniert für $\left|\frac{x_1}{x_2}\right| \approx 1$

Algorithmus A	Algorithmus B
$u = x_1 \odot x_1$	$u = x_1 \oplus x_1$
$v = x_2 \odot x_2$	$v = x_1 \ominus x_2$
$\tilde{q} = u \ominus v$	$\tilde{q} = u \odot v$

Sei $x_1, x_2 \in A$. Für Maschinenoperationen $\circledast$ und $a, b \in A$ gilt

$$a \circledast b = (a * b)(1 + \varepsilon), |(|\varepsilon) \leq eps.$$

Algorithmus A:

$$\begin{aligned}u &= x_1^2(1 + \varepsilon_1), v = x_2^2(1 + \varepsilon_2) \\ \tilde{y} &= \big(x_1^2(1 + \varepsilon_1) - x_2^2(1 + \varepsilon_2)\big)(1 + \varepsilon_3) \\ &= \underbrace{x_1^2 - x_2^2}_{y} + x_1^2\varepsilon_1 - x_2^2\varepsilon_2 + \underbrace{(x_1^2 - x_2^2)}_{y}\varepsilon_3, |\varepsilon| \leq eps\end{aligned}$$

$$\Longrightarrow \left|\frac{\Delta y}{y}\right| \dot{\leq} eps\frac{x_1^2 + x_2^2 + |x_1^2 - x_2^2|}{|x_1^2 - x_2^2|} = eps\left(1 + \left|\frac{\left(\frac{x_1}{x_2}\right)^2 + 1}{\left(\frac{x_1}{x_2}\right)^2 - 1}\right|\right)$$

Wegen der Konditionierung des Problems

$$\left|\frac{\Delta y}{y}\right| \leq 2\left|\frac{\left(\frac{x_1}{x_2}\right)^2 + 1}{\left(\frac{x_1}{x_2}\right)^2 - 1}\right| eps$$

ist A stabil. Algorithmus B:

$$u = x_1 \oplus x_2, v = x_1 \ominus x_2, y = u \odot v$$

Rundungsfehleranalyse

$$\begin{aligned} u &= (x_1 + x_2)(1+\varepsilon_1), v = (x_1 - x_2)(1+\varepsilon_2) \\ \tilde{y} &= (x_1 + x_2)(1+\varepsilon_1)(x_1 - x_2)(1+\varepsilon_2)(1+\varepsilon_3) \\ &= \underbrace{x_1^2 - x_2^2}_{y} + \underbrace{(x_1^2 - x_2^2)}_{\varepsilon_1+\varepsilon_2+\varepsilon_3} + \mathcal{O}(eps^3) \end{aligned}$$

$$\implies \left|\frac{\Delta y}{y}\right| \dot{\leq} |(|\varepsilon_1 + \varepsilon_2 + \varepsilon_3) \leq 3eps$$

$\implies$ Algorithmus B ist stabiler als Algorithmus A.

Regel: Bei numerischen Rechnungen sollte man die schlechter konditionierten Operationen möglichst frühzeitug ansetzen.
Wiederholung

- Konditionierung: Eigenschaften einer numerischen Aufgabe
- Stabilität: Eigenschaft eines Verfahrens
  - Auslöschung
- Rundungsfehleranalyse
  - $y = f(x_1, x_2) = x_1^2 - x_2^2 = (x_1 - x_2)(x_1 + x_2)$

Auswertung von Polynomen

$$y = p(x) = a_0 + a_1 x + \cdots + a_n x^n$$

Als Modellfall betrachten wir

$$p(x) = a_1 x + a_2 x^2 = x(a_1 + a_2 x)$$

Zwei Varianten für $\tilde{y} = p(\xi), \xi \in A$
A: $u = \xi \odot \xi, v = a_2 \odot u, w = a_1 \odot \xi, \tilde{y} = v + w$
B: $u = a_2 \odot \xi, v = a_1 \oplus u, \tilde{y} = \xi \odot v$
B spart eine arithmetische Operation.
Rundungsfehleranalyse A:

$$\begin{aligned} u &= \xi^2(1+\varepsilon_1), v = a_2\xi^2(1+\varepsilon_1)(1+\varepsilon_2), w = a_1\xi(1+\varepsilon_3) \\ \tilde{y} &= (a_2\xi_2(1+\varepsilon_1)(1+\varepsilon_2) + a_1\xi(1+\varepsilon_3))(1+\varepsilon_4) \\ &= \underbrace{a_2\xi^2 + a_1\xi}_{y} + \underbrace{(a_2\xi^2 + a_1)}_{y}\varepsilon_4 + a_2\xi^2(\varepsilon_1 + \varepsilon_2) + a_1\xi\varepsilon_3 + \mathcal{O}eps^2 \end{aligned}$$

$$\begin{aligned} \frac{\Delta y}{y} &\doteq \varepsilon_4 \frac{a_2\xi^2(\varepsilon_1\varepsilon_2) + a_1\xi\varepsilon_3}{a_2\xi^2 + a_1\xi} \\ &= \varepsilon_4 + \varepsilon_3 + \frac{a_2\xi^2(\varepsilon_1 + \varepsilon_2 - \varepsilon_3)}{a_2\xi^2 + a_1\xi} \\ &= \varepsilon_3 + \varepsilon_3 + \frac{\xi}{\frac{a_1}{a_2} + \xi}(\varepsilon_1 + \varepsilon_2 - \varepsilon_3) \end{aligned}$$

Variante B:

$$\begin{aligned}
u &= x_2\xi(1+\varepsilon_1), v = (a_1 + a_2\xi(1+\varepsilon_1))(1+\varepsilon_2) \\
\tilde{y} &= \xi \cdot [a_1 + a_2\xi(1+\varepsilon_1)](1+\varepsilon_2)(1+\varepsilon_3) \\
&= \underbrace{\xi(a_1 + a_2\xi)}_{y} + a_1\xi(\varepsilon_2 + \varepsilon_3) + a_2\xi^2(\varepsilon_1 + \varepsilon_2 + \varepsilon_3) + \mathcal{O}(eps^2) \\
\frac{\Delta y}{y} &= \varepsilon_2 + \varepsilon_3 + \frac{a_2\xi^2}{a_1\xi + a_2\xi}\varepsilon_2 = \varepsilon_2 + \varepsilon_3 + \frac{\xi}{\frac{a_1}{x_2} + \xi}\varepsilon_1
\end{aligned}$$

$\implies$ Variante B ist etwas stabiler als A im Fall $\xi \approx -\frac{a_1}{a_2}$ (nahe bei Nullstelle) Allgemein:

$$\begin{aligned}
p(x) &= a_0 + a_1x + \cdots + a_nx^n \\
&= a_0 + x(a_1 + x(\cdots + x(a_{n-1} + a_nx)\ldots))
\end{aligned}$$

Definition 1.23 (Horner-Schema)

$$b_n = a_n, b_k = a_k + \xi b_{k-1}, k = n-1, \ldots, 0$$

liefert den Funktionswert $p(\xi) = b_0$ des Polynoms an der Stelle $x = \xi$.

Regel: Die Auswertung von Polynomen sollte mit dem Horner-Schema erfolgen.

2 Interpolation und Approximation

Grundproblem:
Darstellung und Auswertung von Funktionen.
Aufgabenstellung:

1. Eine Funktion $f(x)$ ist nur auf einer diskreten Menge von Argumenten $x_0, \ldots, x_n$ bekannt und soll rekonstruiert werden (zum Beispiel für Graph Ausgabe)

2. Eine analytisch gegebene Funktion $f(x)$ soll auf dem Rechner so dargestellt werden, dass jederzeit Funktionswerte zu beliebigen Argument x berechnet werden können.

$\rightarrow$ System mit unendlich vielen Freiheitsgraden $y = f(x)$. „Simulation“ durch endlich viele Datensätze in Klassen P von einfach strukturierten Funktionen

- Polynome: $p(x) = a_0 + a_1x + \cdots + a_nx^n$
- rationale Funktionen:
$$r(x) = \frac{a_0 + a_1x + \cdots + a_nx^n}{b_0 + b_1x + \cdots + b_mx^m}$$
- trigonometrische Funktionen
$$t(x) = \frac{1}{2}a_0 + \sum_{k=1}^{n}(a_k\cos(kx) + b_k\sin(kx))$$

- Exponentialsummen
$$e(x) = \sum_{k=1}^{n} a_k \exp(b_k x)$$

Definition 2.1 Geschieht die Zuordnung eines Elementes $g \in P$ zur Funktion f durch Fixieren von Funktionswerten
$$g(x_i) = y_i = f(x_i), i = 0, \dots, n$$
so spricht man von **Interpolation**. Ist g im gewissen Sinne die beste Darstellung von f, zum Beispiel: $\max_{a \leq x \leq b} |f(x) - g(x)|$ minimal für $g \in P$, oder
$\left(\int_a^b |f(x) - g(x)|^2 \mathrm{d}x\right)^{1/2}$ minimal für $g \in P$ so spricht man von **Approximation**. Die Wahl der Konstruktion von $g \in P$ hängt von der zu erfüllenden Aufgabe ab. Offenbar ist die Interpolation eine Approximation mit
$$\max_{i=0,\dots,n} |f(x_i) - g(x_i)|$$
für $g \in P$

Wiederholung: Interpolation und Approximation

- Stützstellen x_i mit Werten $y_i, i = 0, \dots, n$
- Klassen P von Funktion

Polynominterpolation
Wir bezeichnen mit P_n den Vektorraum der Polynome vom Grad $\leq n$:
$$P_n = \{p(x) = a_0 + a_1 x + \cdots + a_n x^n \mid a_i \in \mathbb{R}, i = 0, \dots, n\}$$

Definition 2.2 (Lagrangasche Interpolationsaufgabe) Die Lagrangsche Interpolationsaufgabe besteht darin zu $x + 1$ paarweise verschiedenen Stützstellen (auch Knoten genannt) $x_0, \dots, x_n \in \mathbb{R}$ und gegebenen Knotenwerten $y_0, \dots, y_n \in \mathbb{R}$ ein Polynom $p \in P_n$ zu bestimmen mit der Eigenschaft $p(x_i) = y_i$

Satz 2.3 Die Lagrangsche Interpolationsaufgabe ist eindeutig lösbar.

Beweis **Eindeutigkeit**: Sind $p_1, p_2 \in P_n$ Lösungen, so gilt für $p = p_1 - p_2$, dass
$$p(x_i) = p_1(x_i) - p_2(x_i) = y_i - y_i = 0, i = 0, \dots, n$$
Also hat p $n + 1$ Nullstellen und ist folglich identisch Null. $\implies p_1 = p_2$
Existenz: Wir betrachten die Gleichungen
$$p(x_i) = y_i \qquad i = 0, \dots, n$$

Dies ist ein lineares Gleichungssystem mit $n+1$ Gleichungen und $n+1$ Freiheitsgraden.

$$\begin{pmatrix} x_0^0 & x_0^1 & \dots & x_0^n \\ x_1^0 & x_1^1 & & x_1^n \\ \vdots & & \ddots & \vdots \\ x_n^0 & x_n^1 & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Wegen der Eindeutigkeit von p ist $\ker V = \{0\}$. Mit dem Rangsatz ($\dim \mathbb{R}^{n+1} = \dim \ker V + \dim \operatorname{im} V$) liefert V eine surjektive Abbildung. Damit existiert eine Lösung. □

Zur Konstruktion des Interpolationspolynoms $p \in P_n$ verwenden wir die sogenannten Lagrangschen Basispolynome.

$$L_i^{(n)}(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \in P_n, i = 0, \dots, n$$

Lemma 2.4 $\{L_i^{(n)}, i = 0, \dots, n\}$ ist eine Basis von P_n

Beweis Übung. □

Offensichtlich gilt:

$$L_i^{(n)}(x_k) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Definition 2.5 Das Polynom

$$p(x) = \sum_{i=0}^{n} y_i L_i^{(n)}(x) \in P_n$$

hat die gewünschten Eigenschaften

$$p(x_j) = y_j, j = 0, \dots, n$$

und wird die Lagrangsche Darstellung des (Lagrangschen) Interpolationspolynoms zu dem Stützpunkten $(x_i, y_i), i = 0, \dots, n$ genannt.

Nachteil: Bei Hinzunahme eines weiteren Stützpunktes (x_{n+1}, y_{n+1}) ändern sich die Basispolynome völlig.
Abhilfe: Newtonsche Basispolynome

$$N_0(x) = 1, N_i(x) = (x - x_{i-1}) N_{i-1}(x) = \prod_{j=0}^{i-1} (x - x_j)$$

Für den Ansatz

$$p(x) = \sum_{i=0}^{n} a_i N_i(x)$$

erhält man durch Auswertung von $x_0, \dots, x_n$ das gestaffelte Gleichungssystem

$$\begin{aligned}
y_0 &= p(x_0) = a_0 \\
y_1 &= p(x_1) = a_0 + a_1(x_1 - x_0) \\
&\vdots \\
y_0 &= p(x_0) = a_0 + a_1(x_1 - x_0) + \dots + a_n(x_n - x_0) \cdot \ldots \cdot (x_n - x_{n-1})
\end{aligned}$$

aus dem sich die Koeffizienten a_i rekursiv berechnen lassen. Bei Hinzunahme eines weiteren Stützpunktes (x_{n+1}, y_{n+1}) setzt man den Prozess mit der Basisfunktion N_{n+1} fort. In der Praxis verwendet man folgende stabilere und effizientere Methode

Satz 2.6 (Newtonsche Darstellung) Das Lagrangsche Interpolationspolynom zu den Punkten $(x_0, y_0), \dots, (x_n, y_n)$ lässt sich bezüglich der Newtonschen Polynombasis schreiben in der Form

$$p(x) = \sum_{i=0}^{n} y[x_0, \dots, x_i] N_i(x)$$

Dabei bezeichnen $y[x_0, \dots, x_i]$ die zu den Punkten (x_i, y_i) gehörenden „dividierten Differenzen“, welhce rekursiv definiert sind durch

$$\text{für } j = 0, \dots, n : y[x_j] = y_j$$

$$L\text{für } k = 1, \dots, j : i = k - j : y\,\underbrace{[x_i, \dots, x_{1+k}]}_{k+1} = \frac{y\,\underbrace{[x_{i+1}, \dots, x_{1+k}]}_{k} - y\,\underbrace{[x_i, \dots, x_{x_1+k-1}]}_{k}}{x_{i+k} - x_i}$$

Beweis Es bezeichne $pi, i + k \in P_k$ das Polynom, welches die Punkte $(x_i, y_i), \dots, (x_{i+k}, y_{i+k})$ interpoliert. Speziell ist $p_{0,n} = p$ das gesuchte Interpolationspolynom. Wir zeigen

$$p_{i,i+k}(x) = y[x_i] + y[x_i, x_{i+1}](x - x_i) + \dots + y[x_i, \dots, x_{i+k}](x - x_i) \cdot \ldots \cdot (x - x_{i+k})$$

was für $i = 0$ und $k = n$ den Satz beweist. Der Beweis wird durch Induktion über die Indexdifferenz k geführt. Für $k = 0$ ist $p_{i,i} = y_i = y[x_i], i = 0, \dots, n$. Sei die Behauptung richtig für $k - 1 \geq 0$. Nach Konstruktion gilt für ein $a \in \mathbb{R}$

$$p_{i,i+k}(x) = p_{i,i+k-1}(x) + a(x - x_1) \cdot \ldots \cdot (x - x_{i+k-1}) = 0$$

für $x = x_j, j = i, \dots, i + k - 1$. Zu zeigen: $a = y[x_i, \dots, x_{i+k}]$. Offenbar ist a der Koeffizient von x^k in $p_{0,i+k}$. Nach Induktionsannahme ist also

$$\begin{aligned}
p_{i,i+k-1}(x) &= \dots + y[x_i, \dots, x_{i+k-1}]x^{k-1} \\
p_{i+1,i+k-1}(x) &= \underbrace{\dots}_{\text{Grad } \leq k-2} + y[x_{i+1}, \dots, x_{i+k}]x^{k-1}
\end{aligned}$$

Ansatz:

$$\begin{aligned} q(x) &= \frac{(x-x_i)p_{i+1,i+k}(x)-(x-x_{i+k})p_{i,i+k-1}(x)}{x_{i+k}-x_i} \\ &= p_{i,i+k-1}(x)+\frac{(x-x_i)p_{i+1,i+k}(x)-(x-x_{i+k}+x_{i+k}-xi)p_{i,i+k-1}(x)}{x_{i+k}-x_i} \\ &= p_{i,i+k-1}(x)+(x-x_i)\frac{p_{i+1,i+k}(x)-p_{i,i+k-1}(x)}{x_{i+k}-x_i} \end{aligned}$$

Ex gilt:

$$q(x_i)=y_i, q(x_{i+k})=\frac{(x_{i+k}-x_i)y_{i+k}+0}{x_{i+k}-x_i}=y_{1+k}$$

$$q(x_j)=\frac{(x_j-x_i)y_j-(x_j-x_{i+k})y_j}{x_{i+k}-x_i}=y_j, j=i+1,\dots,i+k-1$$

$\Longrightarrow$ q interpoliert die Stützpunkte $(x_i,y_i),\dots,(x_{i+k},y_{i+k})$ $\Longrightarrow$ $q\equiv p_{i,i+k}$ (Eindeutigkeit des Interpolationspolynoms). Der führende Koeffizient in $p_{i,i+k}(x)$ ist demnach

$$\begin{aligned} q &= \frac{y[x_{i+1},\dots,x_{i+k}]-y[x_i,\dots,x_{i+k-1}]}{x_{i+k}-x_i} \\ &= y[x_i,\dots,x_{i+k}] \end{aligned}$$ □

Korollar 2.7 Sei $\sigma : \{0,\dots,n\} \to \{0,\dots,n\}$ eine beliebige Permutation. Dann gilt für die Stützpunkte $(\tilde{x}_i,\tilde{y}_i)=\big(x_{\sigma(j)},y_{\sigma(j)}\big)$

$$y[\tilde{x}_0,\dots,\tilde{x}_n]=y[x_0,\dots,x_n]$$

Beweis Koeffizient des Monoms x^n ist $y[x_0,\dots,x_n]$ unabhängig von der Reihenfolge. □

Wiederholung: Lagrange-Interpolation:
Gegeben: $(x_i,y_i), i=0,\dots,n$
Suche $p\in P_n : p(x_i)=y_i, i=0,\dots,n$
Lösung:

$$\begin{aligned} p(x) &= \sum_{i=0}^{n} y_i L_i^{(n)}(x) \\ &= L_i^{(n)}(x) \qquad\qquad = \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j}\in P_n \end{aligned}$$

$\Longrightarrow L_i^{(n)}(x_j) = \delta_{ij}$
Andere Darstellung: Newton-Neville

$$N_i(x) = \prod_{j=0}^{n-1}(x - x_j)$$
$$p(x) = \sum_{i=0}^{N} y[x_0, \dots, x_i] D_i(x)$$
$$y[x_i] = q_i$$
$$y[x_i, \dots, x_{i+k}] = \frac{y[x_{i+1}, \dots, x_{i+k}] - y[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

Definition 2.8 Das durch die Rekursion $j = 0, \dots, n, p_{j,j}(x) = y_j$ für $k = 1, \dots, j : i = k - j$

$$p_{i,i+k}(x) = p_{i,i+k-1}(x) + (x - x_i)\frac{p_{i+1,i+k}(x) - p_{i,i+k-1}(x)}{x_{i+k} - x_i}$$

erzeugte Polynom $p_{0,1}$ ist die sogenannte Nevellsche Darstellung des Interpolationspolynom zu den Stützstellen $(x_0, y_0), \dots, (x_n, y_n)$

Schema:

$$\begin{array}{ccccccccccc}
 & k=0 & & k=1 & & k=2 & \dots & k=n-1 & k=n & \\
x_0 & y_0 & \rightarrow & p_{0,1} & \rightarrow & p_{0,2} & \cdots & p_{0,n-1} & \rightarrow & p_{0,n} \\
x_1 & y_1 & \nearrow\rightarrow & p_{1,2} & \nearrow\rightarrow & p_{1,3} & \cdots & p_{1,n} & \nearrow & \\
x_2 & y_2 & \nearrow & & & & & & & \\
\vdots & & \vdots & \iddots & & & & & & \\
x_{n-1} & y_{n-1} & \rightarrow & p_{n-1,n} & & & & & & \\
x_n & y_n & \nearrow & & & & & & &
\end{array}$$

Die Hinzunahme eines weiteren Stützpunktes ist problemlos. Die Auswertung von $p_{0,n}(x)$ an einer Stelle $\xi \neq x_i$ ohne vorige Bestimmung der Koeffizienten der Newton-Darstellung ist damit sehr einfach und numerisch effizient und stabil möglich. Dazu wird im Schema x mit ξ ersetzt.

2.1 Auswertung von Polynomen und deren Ableitungen

Sei $p \in P_n$ gegeben in der Darstellung

$$p(x) = a_0 + a_1 x + \dots + a_n x^n$$

Wiederhohlung: Auswertung von $p(\xi)$ mittels Horner-Schema

$$b_k = \begin{cases} a_n & k = n \\ a_k + \xi b_{k+1} & k = n-1, \dots, 0 \end{cases}$$

$\implies p(\xi) = b_0$.
Zu $p_n = p \in P_n$ und festem ξ wird durch

$$p_{n-1}(x) = b_1 + b_2 x + \cdots + b_n x^{n-1}$$

ein Polynom $p_{n-1} \in P_{n-1}$ definiert. Wegen $a_k = b_k - \xi b_{k+1}, k = 0, \ldots, n-1, a_n = b_n$:

$$\begin{aligned} p_n(x) &= \sum_{k=0}^{n} b_k x^k - \xi \sum_{k=0}^{n-1} b_{k+1} x^k \\ &= b_0 + x \sum_{k=1}^{n} b_k x^{k-1} - \xi \sum_{k=1}^{n} b_k x^{k-1} \\ &= r_0 + (x-\xi) p_{n-1}(x) \quad r_0 = p(\xi) = b_0 \end{aligned}$$

$\implies$ Für eine Nullstelle ξ von p_n leistet das Horner-Schema die Abspaltung des Linearfaktors $(x - \xi)$ vom Polynom p_n. Weiter ist dann für $x \neq \xi$

$$\frac{p_n(x) - p_n(\xi)}{x - \xi} = p_{n-1}(x)$$

$x \to \xi$

$$p_n'(\xi) = p_{n-1}(\xi)$$

Zur Berechnung von $p_n'(\xi)$ wird das Horner-Schema auf p_{n-1} angewendet.

$$p_{n-2} \in P_{n-2}, p_{n-1}(x) = r_1 + (x-\xi) p_{n-2}(x), r_1 = p_{n-1}(\xi)$$

Fortsetzen $\to$ endliche Folge von Polynomen $p_n, p_{n-1}, \ldots, p_0$ mit

$$p_{n-j}(x) = (x-\xi) p_{n-j-1}(x) + r_j, \quad j = 0, \ldots, n$$

$$p_n(x) = r_0 + r_1(x-\xi) + \cdots + r_n(x-\xi)^n$$

Vergleich mit der Taylorentwicklung von p_n um ξ ergibt

$$r_j = \frac{1}{j!} p_n^{(j)}(\xi)$$

Die Koeffizienten von p_{n-j} seien

$$p_{n-j}(x) = a_j^{(j)} + a_{j+1}^{(j)} x + \cdots + a_n^{(j)} x^{n-j}, j = 0, \ldots, n$$

Es gilt die Rekursion:

$$a_k^{(j+1)} = \begin{cases} a_n^{(j)} & k = n \\ a_k^{(j)} + \xi a_{k+1}^{(j+1)} \end{cases}$$

und es gilt

$$p^{(j)}(\xi) = j! a_j^{j+1}, j = 0, \ldots, n$$

Dieses „vollständige Horner-Schema“ kann leicht zur Auswertung von Polynomen in Newton-Darstellung modifiziert werden:

$$p(x) = a_0 + a_1(x - x_0) + \cdots + a_n(x - x_0) \cdot \ldots \cdot (x - x_{n-1})$$

2.2 Interpolation von Funktionen

Stützstellen $x_0, \ldots, x_n \in [a, b]$. Werte gegeben durch Funktion $y_i = f(x_i), i = 0, \ldots, n$
Frage: Wie gut approximiert das Interpolationspolynom $p \in P_n$ die Funktion f auf $[a, b]$?
Bezeichnungen:

- $\overline{(x_0, \ldots, x_n)}$ = kleinstes Intervoll, das alle x_i enthält.
- $C[a, b]$: Vektorraum der über $[a, b]$ stetigen Funktionen
- $C^k[a, b]$: Vektorraum über $[a, b]$ k-mal stetig differenzierbarer Funktionen.

Satz 2.9 (Interpolationsfehler 1) Sei $f \in C^{n+1}[a, b]$. Dann gibt es zu jedem $x \in [a, b]$ ein $\xi_x \in \overline{(x_0, \ldots, x_n, x)}$, sodas gilt

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi x)}{(n+1)!} \prod_{j=0}^{n} (x - x_j)$$

Beweis Für $x \in \{x_0, \ldots, x_n\}$ ist alles klar. Sei $x \in [a, b] \setminus \{x_0, \ldots, x_n\}$. Wir setzen

$$l(t) = \prod_{j=0}^{n} (t - x_j), \quad c(x) = \frac{f(x) - p(x)}{l(x)}$$

Die Funktion

$$F(t) = f(t) - p(t) - c(x) l(t)$$

besitzt dann mindestens die $n+2$ Nullstellen $x_0, \ldots, x_n, x$ in $[a, b]$. Durch wiederhohlte Anwendung des Satzes von Rolle schließt man, dass die Ableitung F^{n+1} eine Nullstelle $\xi_x \in \overline{(x_0, \ldots, x_n, x)}$. Es

$$\begin{aligned} 0 &= F^{(n+1)}(\xi_x) = f^{(n+1)}(\xi) - p^{(n+1)}(\xi) - c(x) l^{(n+1)}(t) \\ &= f^{(n+1)}(\xi) - c(x)(n+1)! \end{aligned}$$

□

Wiederholung:

- Neville-Schema für $p \in P_n$:
$$p(x_i) = y_i, i = 0, \ldots, n$$
- Vollständiges Horner-Schema
- Interpolation von Funktionen $y_i = f(x_i)$

Interpolationsfehler 1: Sei $f \in C^{n+1}[a, b] \implies \forall x \in [a, b] \exists \xi_x \in \overline{(x_0, \ldots, x_n, x)}$:

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{j=0}^{n} (x - x_j)$$

Satz 2.10 (Interpolationsfehler 2) Sei $f \in C^{n+1}[a,b]$. Dass gilt für $x \in [a,b] \setminus \{x_0, \dots, x_n\}$:

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

mit der Notation

$$f[x_i, \dots, x_{i+k}] = y[x_i, \dots, x_{i+k}]$$

und es ist

$$f[x_0, \dots, x_n, x] = \int_0^1 \int_0^{t_1} \dots \int_0^{t_n} f^{(n+1)}(x_0 + t_1(x_1 - x_0) + \dots + t_n(x_n - x_{n-1}) + t(x - x_n)) \mathrm{d}t \mathrm{d}t_n \dots \mathrm{d}t_1$$

Beweis Per Induktion über n.
IA: $n = 0$:

$$f(x) - p_0(x) = f(x) - f(x_0) = \begin{cases} f[x_0, x](x - x_0) \\ (x - x_0) \int_0^1 f'(x_0 + t(x - x_0)) \mathrm{d}t \end{cases}$$

wobei ein

$$\int_0^1 g'(t) \mathrm{d}t = g(1) - g(0)$$

für $g(t) = f(x_0 + t(x - x_0)) \implies g'(t) = f'(t)(x - x_0)$
Sei die Behauptung richtig für $n - 1 \geq 0$. Dann ist

$$\begin{aligned} f(x) - p_n(x) &= f(x) - \sum_{i=0}^{n} f[x_0, \dots, x_n] \prod_{j=0}^{i-1} (x - x_j) \\ &= f(x) - p_{n-1}(x) - f[x_0, \dots, x_n] \prod_{j=0}^{n-1} (x - x_j) \\ &= f[x_0, \dots, x_{n-1}, x] \prod_{j=0}^{n-1} (x - x_j) - f[x_0, \dots, x_n] \prod_{j=0}^{n-1} (x - x_j) \\ &= (f[x_0, \dots, x_{n-1}, x] - f[x_0, \dots, x_n]) \prod_{j=0}^{n-1} (x - x_j) \\ &= \frac{f[x, x_0, \dots, x_n] - f[x_0, \dots, x_n]}{x - x_n} \prod_{j=0}^{n-1} (x - x_j) \\ &= f[x_0, \dots, x_n, x] \prod_{j=0}^{n-1} (x - x_j) \end{aligned}$$

Weiterhin gilt:

$$f[x_0, \dots, x_{n-1}, x] - f[x_0, \dots, x_n] = \int_0^1 \int_0^{t_1} \dots \int_0^{n-1} [f^{(n)}(x_0 + t_1(x_1 - x_0) + \dots + t_n(x - x_{n+1})) - f^{(n)}(x_0$$

Setze $g(t) = f^{(n)}(x_0 + t_1(x_1 - x_0) + \cdots + t_n(x_n - x_{n-1}) + t_{x-x_n})$

$$
\begin{aligned}
&= \int_0^1 \int_0^{t_1} \cdots \int_0^{t_{n-1}} [g(t_n) - g(0)] \mathrm{d}t_n \ldots \mathrm{d}t_1 \\
&= \int_0^1 \int_0^{t_1} \cdots \int_0^{t_{n-1}} \int_0^{t_n} f^{(n+1)}(x_0 + t_1(x_1 - x_0) + \cdots + t_n(x_n - x_{n-1}) + t
\end{aligned}
$$

$$\implies f[x_0, \ldots, x_n, x] = \int_0^1 \int_0^{t_1} \cdots \int_0^{t_n} f^{(n+1)}(\ldots) \mathrm{d}t \mathrm{d}t_n \ldots \mathrm{d}t_1 \qquad \square$$

Die Integraldarstellung der dividierten Differenzen gestattet ihre stetige Fortsetzung für den Fall, das Stützstellen zusammenfallen:

$$f[x_0, \ldots, x_r, x_r, \ldots, x_n] = \lim_{\varepsilon \to 0} f[x_0, \ldots, x_r, x_r + \varepsilon, \ldots, x_n]$$

Im Extremfall $x_0 = x_1 = \cdots = x_n$ wird

$$
\begin{aligned}
f[x_0, \ldots, x_n] &= \int_0^1 \int_0^{t_1} \cdots \int_0^{t_{n-1}} f^{(n)}(x_0) \mathrm{d}t_n \ldots \mathrm{d}t_1 \\
&= \int_0^1 \int_0^{t_1} \cdots \int_0^{t_{n-1}} 1 \mathrm{d}t_n \ldots \mathrm{d}t_1 f^{(n)}(x_0) \\
&= \frac{1}{n!} f^{(n)}(x_0)
\end{aligned}
$$

Damit geht das Newtonsche Interpolationspolynom über in das Taylorpolynom n-ten Grades von f in x_0. Konstruieren wir die Fehlerdarstellung so erhalten wir für ein $\xi_x \in (x_0, \ldots, x_n, x)$

$$
\begin{aligned}
\frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{j=0}^{n} (x - x_j) = f(x) - p(x) \\
= f[x_0, \ldots, x_n, x] \prod_{j=0}^{n} (x - x_j) \\
\implies f[x_0, \ldots, x_n, x] = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}
\end{aligned}
$$

Definition 2.11 (Hermite-Interpolation) Die Hermitesche Interpolationsaufgabe lautet:
Gegeben $x_i, i = 0, \ldots, m$ (paarweise verschieden), $y_i^{(k)}, i = 0, \ldots, m, k = 0, \ldots, \mu_i, \mu \geq 0$.
Gesucht: $p \in P_n, n = m + \sum_{i=0}^{m} \mu_i, p^{(k)}(x_j) = y_i^{(k)}, i = 0, \ldots, m, k = 0, \ldots, \mu_i$, $(\mu_i + 1)$ -fache Stützstellen.

Beispiel 2.12 $x_0 = -1, x_1 = 1, m = 1, y_0^{(0)} = 0, y_1^{(0)} = 1, y_1^{(1)} = 2 \implies \mu_0 = 0, \mu_1 = 1 \implies n = 1 + 0 + 1 = 2 \implies p(x) = x^2$

Analog zur Lagrange-Interpolation:

- Existenz + Eindeutigkeit
- Darstellung des Interpolationsfehlers

Wiederholung: Fehlerdarstellung Lagrange-Interpolation. Sei $f \in C^{n+1}[a,b]$. $\exists \xi_x \in \overline{(x_0, \dots, x_n, x)}$

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{j=0}^{n} (x - x_j)$$

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

$$f[x_0, \dots, x_n, x] = \int_0^1 \int_0^{t_1} \dots \int_0^{t_n} f^{(n+1)}(x_0 + t_1(x_1 - x_0) + \dots + t_n(x_n - x_{n-1}) + t(x_n - x)) \mathrm{d}t \mathrm{d}t_n \dots \mathrm{d}t_1$$

Hermite-Interpolation: Such $p \in P_n, n = m + \sum_{i=0}^{m} \mu_i$

$$p^{(k)}(x_i) = y_i^{(k)}, i = 0, \dots, m, k = 0, \dots, \mu_i$$

2.3 Richardsonsche Extrapolation zum Limes

Gegeben: Numerischer Prozess mit Werten $a(h), h \in \mathbb{R}_+, h \to 0$.
Gesucht: $a(0) = \lim_{h \to 0} a(h)$
Idee: Für $h_i > 0, i = 0, \dots, n$, interpoliere $(h_i, a(h_i))$ und berechne $p_n(0)$

Beispiel 2.13 (Numerische Differentation) Sei $f \in C^\infty[a,b], x \in (a,b)$. Nach Taylor gilt

$$a(h) = \frac{f(x+h) - f(x-h)}{2h} = f'(x) + \sum_{i=1}^{\infty} \frac{f^{(2i+1)}(x)}{(2i)!} h^{2i}$$

Satz 2.14 (Extrapolationsfehler) Für $h \in \mathbb{R}_+$ habe $a(n)$ die Entwicklung

$$a(h) = a_0 + \sum_{j=1}^{n} a_j h^{jq} + a_{n+1}(h) h^{(n+1)q}$$

mit $q > 0$, Koeffizienten a_j und

$$a_{n+1}(h) = a_{n+1} + \wr(1)$$

Die Folge $(h_i)_{k \in \mathbb{N}}$ erfülle

$$0 \leq \frac{h_{k+1}}{h_k} \leq \rho < 1$$

($\implies$ h_k positiv, monoton fallend). Dann gilt für das Interpolationspolynom $p_1^{(k)} \in P_n$ (in h^q) durch

$$\left(h_k^q, a(h_k)\right), \dots, \left(h_{k+n}^q, a(h_{k+n})\right)$$
$$a(0) - p_n^{(k)}(0) = \mathcal{O}\left(h_k^{(n+1)q}\right)$$

$(k \to \infty)$

Beweis Abkürzungen $z = h^q, z_k = h_k^q$. Interpoliere $(z_{k+i}, a(h_{k+i})), i = 0, \dots, n$.

$$\begin{aligned} p_n(z) &= \sum_{i=0}^{n} a(h_{k+i}) L_{k+i}^{(n)} I \\ L_{k+1}^{(n)}(z) &= \prod_{\substack{l=0 \\ l\ neqi}} \frac{z - z_{k+l}}{z_{k+1} - z_{wl}} \end{aligned}$$

Übung:

$$\sum_{i=0}^{n} x_{k+1}^{n}(0) = \begin{cases} 1 & r = 0 \\ 0 & r = 1, \dots, n \\ (-1)^n \prod_{j=0}^{n} z_{k+i} & r = n+1 \end{cases}$$

$$\begin{aligned} p_n(0) &= \sum_{i=0}^{n} \left(a_0 + \sum_{j=1}^{n} a_j z_{k+i}^{j} + a_{n+1}(h_{k+1}) z_{k+i}^{n+1} \right) L_{k+i}^{(n)}(0) \\ &= a_0 \underbrace{\sum_{i=0}^{n} L_{k+1}^{(n)}}_{=1} + \sum_{j=1}^{n} a_j \underbrace{\sum_{i=0}^{n} z_{k+1}^{j} L_{k+i}^{(n)}(0)}_{0} \\ &= +a_{n+1} \underbrace{\sum_{i=0}^{n} z_{k+1}^{n+1} L_{k+1}^{(n)}}_{=(-1)^n \prod_{i=0}^{n} z_{k+i}} + \sum_{i=0}^{n} \wr(1) z_{k+i}^{n+1} L_{k+i}^{(n)}(0) \end{aligned}$$

Da man Landau-Symbole nich ausklammern darf, schätzen wir ab:

$$\begin{aligned} \left| L_{k+i}^{(n)}(0) \right| &= \prod_{\substack{l=0 \\ l \neq i}}^{n} \left| \frac{z_k + l}{z_{k+i} - z_{k+y}} \right| \\ &\le \prod_{l=0}^{i-1} \left| \frac{z_{n+l}}{z_{k+i} - z_{k+l}} \right| \prod_{l=1+i}^{n} \left| \frac{z_{k+i}}{z_{k+i} - z_{k+l}} \right| \\ &= \prod_{l=0}^{i-1} \frac{1}{\left| \frac{z_{k+i}}{z_{k+y}} - 1 \right|} \prod_{l=i+1}^{n} \frac{1}{\left| 1 - \frac{z_{k+l}}{z_{k+i}} \right|} \\ &\le \frac{1}{(1 - \rho^q)^n} \\ \implies p_n(0) &= a_0 + a_{n+1}(-1)^n \prod_{i=0}^{n} z_{k+i} + \wr\left(z_k^{n+1}\right) \\ &= a_0 + \mathcal{O}\left(h_k^{(n+1)q}\right) \end{aligned}$$

$\square$

2.4 Spline-Interpolation

Problem: Oszillationen des Interpolationspolynoms, wenn man Stützstellen nicht geeignet wählen kann. Abhilfe: Stückweise polynomielle Interpolation:

- Zerlegung: $a = x_0 < x_1 < \cdots < x_n = b$
- Teilintervalle: $I_i = [x_{i-1}, x_i], i = 1, \dots, n$
- Feinheit: $h = \max_{i=1,\dots,n} h_i$ mit $h_i = |I_i| = x_i - x_{i-1}$
- Vektorräume stückweise polynomieller Funktionen

$$S_n^{k,r}[a,b] = \{p \in C^r[a,b] \mid p\,|_{I_i} \in P_k(i_i)\}, i = 1, \dots, n$$

Beispiel 2.15 (Stückweise lineare Interpolation) $\implies p \in S_n^{(1,0)}[a,b]$. Fehlerabschätzung:

$$\max_{x \in [a,b]} |f(x) - p(x)| \leq \frac{1}{2} h^2 \max_{x \in [a,b]} \left| f''(x) \right|$$

Beispiel 2.16 (Splines) Zweimal stetig differenzierbare, stückweise kubische Polynome. Motivation: Biegestab. Minimiere Biegeenergie

$$\int_{x_0}^{x_n} s''(x)^2 \mathrm{d}x$$

Definition 2.17 (Kubischer Spline) Eine Funktion $s_n : [a,b] \to \mathbb{R}$ heißt kubischer Spline bezüglich $a = x_0 < x_1 < \cdots < x_n = b$, wenn gilt

1. $s_n \in C^2[a,b]$
2. $s_n \mid I_i \in P_3, i = 1, \dots, n$

Gilt zusätzlich

3. $s_n''(a) = s_n''(b) = 0$ so heißt s_n natürlicher Spline.

Existenz des interpolierenden kubischen Spline zu Knotenwerten $s_n(x_i) = y_i, i = 0, \dots, n$

Satz 2.18 (Spline-Interpolation) Der interpolierende kubische Spline existiert und ist eindeutig bestimmt durch zusätzliche Vorgabe von $s_n''(a), s_n''(b)$

Beweis s hat die Form

$$s(x)\,|_{I_i} = p_i(x), i = 1, \dots, n, p_i \in P_3(I_i)$$

4 Koeffizienten auf jedem der n Intervalle ergeben $4n$ Freiheitsgrade. Zur Bestimmung:

$s(x_i) = y_i, i = 0, \dots, n$	$2n$ Gleichungen
$s' \in C[a,b]$	$n - 1$
$s'' \in C[a,b]$	$n - 1$
Zusatzbedingung für $s_n''(a), s_n''(b)$	2
	$4n$

$\Longrightarrow$ quadratisches lineares Gleichungssystem, $4n \times 4n$

$$N = \{\omega \in C^2[a,b] \mid \omega_{x_i} = 0, i = 0, \ldots, n\}$$

Seien $s_n^{(1)}$ und $s_n^{(2)}$ interpolierende Splines $\Longrightarrow$ $s = s_n^{(1)} - s_n^{(2)} \in N$. Für $\omega \in N$ beliebig:

$$\begin{aligned}
\int_a^b s''(x)\omega''(x)\mathrm{d}x &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} s''(x)\omega''(x)\mathrm{d}x \\
&= \sum_{i=0}^{n-1}[- \int_{x_i}^{x_{i+1}} s^{(3)}\omega'\mathrm{d}x + s''\omega' \mid_{x_i}^{x_{i+1}}] \\
&= \sum_{i=0}^{n-1}[- \int_{x_i}^{x_{i+1}} s^{(4)}\omega\mathrm{d}x - s^{(3)}\omega \mid_{x_i}^{x_{i+1}} + s''\omega' \mid_{x_i}^{x_{i+1}}] \\
&= \sum_{i=0}^{n-1} s''\omega' \mid_{x_i}^{x_{i+1}} = s''(x)\omega'(x) - s''(a)\omega'(a) \\
&= 0
\end{aligned}$$

Speziell für $\omega = s$

$$\int_a^b \left|s''(x)\right|^2 \mathrm{d}x = 0$$

$\Longrightarrow$ s ist linear $0 = s(a) = s(b) = 0$ □

Wiederhohlung: Extrapolation $a(h), h_i > 0, a(0) = \lim_{h \to 0} a(h)$ Fehler: Entwicklung

$$a(h) = a_0 + \sum_{j=1}^{n} a_j h^{a_j}$$

$$0 < \frac{h_{k+1}}{h_k} \leq \rho < 1$$

interpolieren $\left(h_{k+1}^a, a(h_{k+1})\right), i = 1, \ldots, n$

$$\Longrightarrow a(0) - p_i^{(k)}(0) = \mathcal{O}\left(h_k^{(n+1)}\right)$$

Splines: $S_h^{(k,r)}[a,b] = \{p \in C^r[a,b] \mid p\big|_{[x_i,x_{i+1}]} \in P_k[x_i,x_{i+1}]\}$ Splines: $s \in S_k^{(n,x)}[a,b]$. Natürliche kubische Splines: $s''(a) = s''(b) = 0$.

Satz 2.19 Für den interpolierenden natürlichen Spline S_n durch $x_0, \ldots, x_n, y_0, \ldots, y_n$ gilt

$$\int_a^b \left|S'(x)\right|^2 \mathrm{d}x \leq \int_a^b \left|g''(x)\right|^2 \mathrm{d}x$$

bezüglich allen Funktionen $g \in C^2[a,b]$ mit $g(x_i) = y_i, i = 0, \ldots, n$

Beweis Sei $N = \{\omega \in C^2[a,b] \mid \omega(x_i) = 0, i = 0, \dots, n\} \implies \omega = g - I_n \in N$.

$$\begin{aligned}
\implies \int_a^b \left|g''(x)\right|^2 \mathrm{d}x &= \int_a^b \left|S_n''(x) + \omega''(x)\right|^2 \mathrm{d}x \\
&= \int_a^b \left|S_n''(x)\right|^2 \mathrm{d}x + 2\underbrace{\int_a^b S_n''(x)\omega''(x)\mathrm{d}x}_{0} + \underbrace{\int_a^b \left|\omega''(x)\right|^2 \mathrm{d}x}_{\geq 0} \\
&\geq \int_a^b \left|S_n''(x)\right|^2 \mathrm{d}x
\end{aligned}$$

□

Satz 2.20 (Approximationsfehler) Sei $f \in C^4[a,b]$. Erfüllt der interpolierende kubische Spline $S_1''(a) = f''(a) \wedge S_n(b) = f''(b)$ so gilt:

$$\max_{x\in[a,b]} |f(x) - S_n(x)| \leq \frac{1}{2} h^4 \max_{x\in[a,b]} \left|f^{(4)}(x)\right|$$

Ohne Beweis.

2.5 Gauß Approximation

Wir betrachten $C[a,b]$, die Menge der stetigen Funktionen auf $[a,b]$ über dem Zahlkörper $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$, als $\mathbb{K}$ -Vektorraum. Für $f, g \in [a,b]$ erfüllt

$$(f,g) := \int_a^b f(t)\overline{g(t)}\mathrm{d}t$$

die Eigenschaften eines Skalarproduktes:

1. Definitheit:
$$(f,f) = \int_a^b f(t)\overline{f(t)}\mathrm{d}t = \int_a^b |f(t)|^2 \mathrm{d}t \geq 0$$
und $(f,f) = 0 \implies f = 0$
2. $\alpha \in \mathbb{K}, h \in C[a,b]$:
$$(\alpha f + g, h) = \int_a^b (\alpha f(t) + g(t))\overline{h(t)}\mathrm{d}t = \alpha \int_a^b f(t)\overline{h(t)}\mathrm{d}t + \int_a^b g(t)\overline{h(t)}\mathrm{d}t = \alpha(f,h) + (g,h)$$
3. Symmetrie:
$$(f,g) = \int_a^b f(t)\overline{g(t)}\mathrm{d}t = \int_a^b \overline{\overline{f(t)}g(t)}\mathrm{d}t = \int_a^b g(t)\overline{f(t)}\mathrm{d}t = \overline{(g,f)}$$

Durch $\|f\| = \sqrt{(f,f)}$ ist damit eine Norm auf $C[a,b]$ gegeben:

1. Definitheit:
$$\|f\| \geq 0, f = 0 \iff \|f\| = 0$$

2. Sublinearität: Wir benutzen die Cauchy-Schwarz-Ungleichung
$$\begin{aligned}
&|(f,g)| \leq \|f\|\|g\| \\
\implies &\|f+g\|^2 = (f+g, f+g) = (f,f) + (f,g) + (g,f) + (g,g) \\
&= \|f\|^2 + \underbrace{2\Re(f,g)}_{\leq|(f,g)|} + \|g\|^2 \\
&\leq \|f\|^2 + 2\|f\|\|g\| + \|g\|^2 = (\|f\| + \|g\|)^2 \\
\implies &\|f+g\| \leq \|f\| + \|g\| \qquad \text{(Dreiecksungleichung)}
\end{aligned}$$

3. Homogenität:
$$\|\alpha f\| = \sqrt{(\alpha f, \alpha f)} = \sqrt{\alpha\bar{\alpha}(f,f)} = |\alpha|\|f\|$$

Mit diesem Skalarprodukt und dieser Norm ist also $C[a,b]$ ein Prähilbertraum.

Satz 2.21 (Gauß-Approximation) Sei H ein Prähilbertraum und sei $S \subset H$ eine endlichdimensionaler Teilraum. Dann existiert zu jedem $f \in H$ eine eindeutig bestimmte „beste Approximation" $g \in S$
$$\|f-g\| = \min_{\varphi\in S}\|f-\varphi\|$$

Beweis Vorüberlegung: Wenn $g \in S$ eine beste Approximation ist, so hat für $\varphi \in S$ die Hilfsfunktion
$$F_\varphi(t) = \|f-g-t\varphi\|^2, t \in \mathbb{R}$$
bei $t = 0$ ein Minimum. Somit ist
$$\begin{aligned}
0 &= \frac{\mathrm{d}}{\mathrm{d}t}F_\varphi(t)\big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t}[(f-g-t\varphi, f-g-t\varphi)]\big|_{t=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t}[(f-g,f-g) - t(\varphi, f-g) - f(f-g,\varphi) + t^2(\varphi,\varphi)]\big|_{t=0} \\
&= 2\Re(f-g,\varphi)\forall\varphi \in S
\end{aligned}$$

Falls $\mathbb{K} = \mathbb{C}$ ergibt testen mit $i\varphi$
$$0 = \Re(f-g, i\varphi) = -\Re(f-g,\varphi) = \Im(t-g,\varphi) \implies (f-g,\varphi) = 0\forall\varphi \in S$$

Interpretation: Der Fehler $f-g$ ist orthogonal zum Teilraum S. Gilt umgekehrt die letzte Gleichung für ein $g \in S$, so gilt für $\varphi \in S$
$$\|f-g\|^2 = (f-g, f-g) = (f-g, f-\varphi) + \underbrace{(f-g,\varphi)}_{0}$$

Cauchy-Schwarz:

$$\begin{aligned} &\leq \|f-g\|\|f-\varphi\| \\ \implies \|f-g\| \leq \inf_{\varphi\in S}\|f-\varphi\| \end{aligned}$$

$\implies$ g ist Bestapproximation.
Existenz und Eindeutigkeit: Da $n = \dim S < \infty$, besitzt S eine Basis $\{\varphi_1, \ldots, \varphi_n\}$. Jedes $g \in S$ hat eine eindeutige Darstellung

$$g = \sum_{i=1}^{n} \alpha_i \varphi_i$$

$$\implies \left(f - \sum_{i=1}^{n} \alpha_i \varphi, \varphi\right) = (f, \varphi) - \sum_{i=1}^{n} \alpha_i (\varphi_i, \varphi) = 0 \forall \varphi \in S$$

$$\implies \sum_{i=1}^{n} (\varphi_i, \varphi)\alpha_i = (f, \varphi_k), k = 1, \ldots, n$$

Dies ist ein lineares $n\times n$ Gleichungssystem. Notation: $A\alpha = B$ mit $\alpha = (\alpha_1, \ldots, \alpha_n)^T \in \mathbb{K}^n$, $b \in \mathbb{K}^n$, $b_i = (f, \varphi_i)$, $A \in \mathbb{K}^{n\times n}$, $A_{ki} = (\varphi_i, \varphi_k)$. A ist hermitisch wegen $(\varphi_i, \varphi_k) = \overline{(\varphi_k, \varphi_i)}$. Sei $\alpha \in \mathbb{K}^n$ beliebig. Wegen

$$\begin{aligned} \alpha^H A\alpha &= \sum_{k=1}^{n}\sum_{i=1}^{n} \bar{\alpha}_k (\varphi_i, \varphi_k)\alpha_i \\ &= \sum_{k=1}^{n}\sum_{i=1}^{n} (\alpha_i, \varphi_i, \alpha_k, \varphi_k) \\ &= \left(\sum_{i=1}^{n} \alpha_i\varphi_i, \sum_{k=1}^{n} \alpha_k\varphi_k\right) = (g, g) > 0 \end{aligned}$$

für $\alpha \neq 0 (\implies g \neq 0)$ ist A also positiv definit und damit invertierbar $\implies$ mit $\alpha = A^{-1}b$ löst das eindeutig bestimmte Gleichungssystem und g ist die Bestapproximation. □

Das lineare Gleichungssystem besitzt besonders einfache Lösung, wenn die Basis $\{\varphi_1, \ldots, \varphi_n\}$ eine Orthogonalbasis ist, das heist $(\varphi_i, \varphi_j) = \delta_{ij}$

$$\begin{aligned} &\implies \alpha_i = (f, \varphi_i), i = 1, \ldots, n \\ &\implies g = \sum_{i=1}^{n} (f, \varphi_i)\varphi_i \quad \text{ist Bestapproximation} \end{aligned}$$

Lemma 2.22 (Gram-Schmidt-Algorithmus) Zu jeder Basis $\{\psi_1, \ldots, \psi_k\}$ von S lässt sich eine Orthonormalbasis $\{\varphi_1, \ldots, \varphi_n\}$ konstruieren.

$$\tilde{\varphi}_1 = \psi_1, \varphi_1 = \frac{\tilde{\varphi}_1}{\|\tilde{\varphi}_1\|}$$

$$\tilde{\varphi}_k = \psi_k - \sum_{i=1}^{k-1} (\psi_k, \varphi_i), \varphi_k = \frac{\tilde{\varphi}_k}{\|\tilde{\varphi}_k\|}$$

Beweis Per Induktion nach n.
$n = 1$: Da $\psi \neq 0$ gilt $(\varphi_1, \varphi_1) = \frac{|\psi_1|^2}{\|\psi_1\|^2} = 1$.
$n > 1$: Sei $\{\varphi_1, \dots, \varphi_n\}$ eine Orthonormalbasis. Es gilt

$$0 \neq \tilde{\varphi}_n = \psi_n - \sum_{k=1}^{n-1} (\psi_n, \varphi_k)\varphi_k$$

da sonst $\{\psi_1, \dots, \psi_n\}$ linear abhängig wären. Für $i = 1, \dots, n-1$ gilt

$$(\varphi_n, \varphi_1) = (\psi_n, \varphi_i) - \sum_{k=1}^{n-1} (\psi_n, \varphi_k) \underbrace{(\varphi_k, \varphi_i)}_{\delta_{ik}} = 0$$

und $\|\varphi_n\|^2 = 1$ nach Konstruktion. □

Wiederhohlung: Gauß-Approximation, Prähilbertraum H, Teilraum $S \subset H, \dim S = n < \infty$

$$\forall f \in H \exists! g \in S : \|f - g\| \leq \min_{\varphi \in S} \|f - \varphi\|$$

Äquivalent: $e := f - g \perp S \iff (f - g, \varphi) = 0 \forall \varphi \in S$
Orthogonalisiere Basis $\{\psi_1, \dots, \psi_n\}$ von S mit Gram-Schmidt

$$\tilde{\varphi}_i = \begin{cases} \psi_i & i = 1 \\ \psi_i - \sum_{j=1}^{i-1} \frac{(\psi_i, \tilde{\varphi}_j)}{\|\tilde{\varphi}_j\|^2} \tilde{\varphi}_j & i = n, \dots, n \end{cases}$$

Normalisieren: $\varphi_k = \|\tilde{\varphi}_n\|^{-1} \tilde{\varphi}_k$. $(\varphi_1, \dots, \varphi_k)$ Orthogonalbasis $\implies (\varphi_i, \varphi_j) = \delta_{ij}$

$$\implies g = \sum_{k=1}^{n} (f, \varphi_k)\varphi_k$$

Erinnerung:

$$f[x_0, \dots, x_n] = \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0}, \quad f[g_k] = f(k)$$

3 Numerische Integration

Approximation von bestimmten Integralen reeller Funktionen $f \in C[a, b]$ durch Quadraturformeln

$$I(f) = \int_a^b f(x)\mathrm{d}x \approx I^{(n)}(f) = \sum_{i=1}^{n} \alpha_i f(x_i)$$

mit Stützstellen $a \leq x_0 < x_1 < \dots < x_n \leq b$ und Gewichten $\alpha_i \in \mathbb{R}$.

Beispiel 3.1 (Summierte Rechteckregel)

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n-1}(x_{i+1}-x_i)f(x_i)$$

Interpolatorische Quadraturformeln.
Idee: Interpoliere f durch ein Interpolationspolynom auf $[a,b]$!

$$p_n(x) = \sum_{i=0}^{n} f(x_i)L_i^{(n)}(x)$$

$$\implies I^{(n)}(f) = \int_a^b p_n(x)\mathrm{d}x = \sum_{i=0}^{n} f(x_i) \int_a^b \underbrace{L_i^{(n)}(x)\mathrm{d}x}_{=\alpha_i}$$

Quadraturgewichte hängen nur von $a, x_0, \ldots, x_n, b$ ab.

Satz 3.2 (Lagrange-Quadratur) Für interpolatorische Quadraturformeln gilt

$$I(f) - I^{(n)}(f) = \int_a^b f[x_0,\ldots,x_n,x]\prod_{i=0}^{n}(x-x_i)\mathrm{d}x$$

Beweis Restglieddarstellung der Interpolation. □

Definition 3.3 Eine Quadraturformel $I^{(n)}$ wird „von der Ordnung m" genannt, wenn sie alle $p \in P_{m-1}$ exakt integriert. Das heißt

$$\int_a^b p(x)\mathrm{d}x = I^{(n)}(p) \forall p \in P_{m-1}$$

$\implies$ Interpolatorische Quadraturformeln zu $n+1$ Stützstellen sind (mindestens) von der Ordnung $n+1$.

Spezialfall: Äquidistante Stützstellen: Newton-Cotes-Formeln:

1. Abgeschlossene Formeln ($H = \frac{b-a}{n}, x_i = a + iH, a = x_0, b = x_n$)

$$I^{(1)}(f) = \frac{b-a}{2}[f(a)+f(b)] \qquad \text{(Trapezregel)}$$

$$I^{(2)}(f) = \frac{b-a}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right] \qquad \text{(Simpsonregel, Keplersche Fassregel)}$$

$$I^{(3)}(f) = \frac{b-a}{8}[f(a)+3f(a+H)+3f(b-H)+f(b)] \qquad \text{(3/8 Regel)}$$

2. Offene Formeln $\left(H = \frac{b-a}{n+2}, x_i = a + (i+1)H, a < x_0, x_n < b\right)$

$$I^{(0)}(f) = (b-a)f\left(\frac{a+b}{2}\right) \qquad \text{(Mittelpunktregel)}$$
$$I^{(1)}(f) = \frac{(b-a)}{2}(f(a+H) + f(b-H))$$
$$I^{(1)}(f) = \frac{(b-a)}{3}\left(2f(a+H) - f\left(\frac{a+b}{2}\right) + 2f(b-H)\right)$$

Satz 3.4 (Quadraturrestglieder) 1. Trapezregel: Für jedes $f \in C^2[a,b]$ gibt es ein $\xi \in [a,b]$ mit
$$\int_a^b f(x)\mathrm{d}x - \frac{b-a}{2}[f(a) + f(b)] = -\frac{(b-a)^3}{12}f''(\xi)$$

2. Simpson-Regel: Für jedes $f \in C^4[a,b] \exists \xi \in [a,b]$ sodass
$$\int_a^b f(x)\mathrm{d}x - \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] = -\frac{(b-a)^5}{2880}f^{(4)}(\xi)$$

3. Mittelpunktregel: $\forall f \in C^2[a,b] \exists \xi \in [a,b]$ sodass
$$\int_a^b f(x)\mathrm{d}x - (b-a)f\left(\frac{a+b}{2}\right) = \frac{(b-a)^3}{24}f''(\xi)$$

Satz 3.5 (Verallgemeinerter Mittelwertsatz) Sei $f \in C[a,b], g \geq 0$ oder $g \leq 0$ integriebar. Dann $\exists \xi \in [a,b]$, sodass
$$\int_a^b f(x)g(x)\mathrm{d}x = f(\xi)\int_a^b g(x)\mathrm{d}x$$

Beweis (Beweis der Quadraturrestglieder).

1. Für $x \in [a,b]$ ist $(x-a)(x-b) \leq 0$

$$\Longrightarrow I(f) - I^{(1)}(f) = \int_a^b f[x_0, x_1, x]\prod_{i=1}^{1}(x - x_i)\mathrm{d}x$$

Verallgemeinerter Mitterwertsatz: $\exists \xi \in [a,b]$, sodass
$$= \frac{f''(\xi)}{2!}\left(-\frac{1}{6}(b-a)^3\right)$$
$$= -\frac{f''(\xi)}{12}(b-a)^3$$

2.

$$
\begin{aligned}
I(f) - I^{(2)}(f) &= \int_a^b f[a, \frac{a+b}{2}, b, x](x-a)\left(x - \frac{a+b}{2}\right)(x-b) \\
&= \int_a^b \frac{f[a, \frac{a+b}{2}, b, x] - f[\frac{a+b}{2}, a, \frac{a+b}{2}, b]}{x - \frac{a+b}{2}}(x-a)\left(x - \frac{a+b}{2}\right)^2 (x-b)\mathrm{d}x + f[\frac{a+b}{2}, a, \frac{a+b}{2} \\
&= \frac{f^{(4)}(\xi)}{4!}\int_a^b (x-a)\left(x - \frac{a+b}{2}\right)^2 (x-b)\mathrm{d}x \\
&= -\frac{f^{(4)}(\xi)}{2880}(b-a)^5
\end{aligned}
$$

3. analog zu 2. □

Probleme:

- negative Gewichte α_i ab $n = 7$ (geschlossen) und $n = 2$ (offen) $\implies$ Auslöschungsgefahr
- Oszillationen des Lagrange-Interpolanten für äquidistante Gitter (Runge-Phänomen), im Allgemeinen $I^{(n)}(f) \not\to I(f), n \to \infty$

Abhilfe: Summierte Quadraturformeln

$$
I_n^{(n)}(f) = \sum_{i=1}^{N-1} I_{[x_i, x_i+1]}^{(n)}(f), h = \frac{b-a}{N}, x_i = a + ih
$$

Gilt die lokale Fehlerdarstellung:

$$
I_{[x_i, x_{i+1}]}(f) - I_{[x_i, x_{i+1}]}^{(n)}(f) = \omega_n h^{n+2} f^{(m+1)}(\xi_i), \quad \xi_i \in [a, b]
$$

für $m \geq n$ gilt:

$$
\begin{aligned}
I(f) - I_n^{(n)}(f) &= \sum_{i=0}^{N-1} [I_{[x_i, x_{i+1}]}(f) - I_{[x_i, x_{i+1}]}^{(n)}(f)] \\
&= \omega_n h^{m+2} N \underbrace{\sum_{i=0}^{N-1} \frac{f^{(m+1)}(\xi_i)}{N}}_{\in[\min_i f^{(m+1)}(\xi_i), \max_i f^{(m+1)}(\xi_i)]} \\
&= \omega_n h^{m+2} N f^{(m+1)}(\xi) \qquad \text{(für ein } \xi \in [a, b] \text{ (Verallg. Mittelwertsatz))} \\
&= \omega_n h^{(m+1)}(b-a) f^{(m+1)}(\xi)
\end{aligned}
$$

Beispiel 3.6 1. Summierte Trapezregel ($m = 1$)

$$I_h^{(1)} = \sum_{i=0}^{N-1} \frac{x_{i+1} - x_i}{2}[f(x_i) + f(x_{i+1})]$$
$$= \frac{h}{2} f(a) + h \sum_{i=1}^{N-1} f(x_i) + \frac{h}{2} f(b)$$
$$I(f) - I_h^{(n)}(f) = -\frac{b-a}{12} h^2 f''(\xi), \xi \in [a, b]$$

2. Summierte Simpson-Regel ($m = 3$)

$$I_h^{(2)}(f) = \sum_{i=0}^{N-1} \frac{x_{i+1} - x_i}{6}[f(x_i) + 4f\left(\frac{x_i + x_{i+1}}{2}\right) + f(x_{i+1})]$$
$$= \frac{h}{6}[f(a) + 2\sum_{i=1}^{N-1} f(x_i) + 4\sum_{i=0}^{N-1} f\left(\frac{x_i + x_{i+1}}{2}\right) + f(b)]$$
$$I(f) - I_h^{(2)}(f) = -\frac{b-a}{2880} h^4 f^{(4)}(\xi), \xi \in [a, b]$$

3. Summierte Mittelpunktsregel ($m = 1$)

$$I_h^{(0)}(f) = \sum_{i=0}^{N-1} (x_{i+1} - x_i) f\left(\frac{x_i + x_{i+1}}{2}\right) = h \sum_{i=0}^{N-1} f\left(\frac{x_i + x_{i+1}}{2}\right)$$
$$I(f) - I_h^{(0)}(f) = \frac{b-a}{24} h^2 f''(\xi), \quad \xi \in [a, b]$$

Widerholung Quadratur

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n} \alpha_i f(x_i) = I^{(n)f}$$

- Interpolatorische Quadraturregel, Äquidistante Stützstellen
  $\rightarrow$ Newton-Cotes Formeln (abgeschlossen, offen)
- Summierte Formeln $x_i = a + iH, H > 0$

$$I_H^{(n)}(f) = \sum_{i=1}^{n} I_{[x_{i-1}, x_i]}^{(n)}(f)$$

3.1 Gaußsche Quadraturformeln

Frage: Wie wählt man x_i in

$$I^{(n)}(f) = \sum_{i=0}^{N} \alpha_i f(x_i)$$

optimal? Nach Konstruktion ist $I^{(n)}$ mindestens von der Ordnung $n + 1$

Lemma 3.7 Interpolatorische Quadraturformeln sind höchstens von der Ordnung $2n+2$

Beweis Wähle

$$p(x) = \prod_{i=0}^{n} (x - x_i)^2 \in P_{2n+2}$$
$$\implies 0 < \int_a^b p(x)\mathrm{d}x = \sum_{i=0}^{n} \alpha_i \underbrace{p(x_i)}_{0} = 0` \qquad \square$$

Gaußsche Quadraturformen erreichen die Maximalordnung $2n+2$ (exakt für $p \in P_{2n+1}$) Herleitung: Für $x_0, \dots, x_n, x_{n+1}, \dots, x_{2n+1} \in [a,b]$ betrachte $I^{(n)}(t)$ und Î{(2n + 1)}(f)

$$I(f) - I^{(2n+1)}(f) = I(f) - \sum_{i=0}^{2n+1} f[x_0, \dots, x_i]\big|_a^b \int_a^b \prod_{j=0}^{i-1} (x - x_j)\mathrm{d}x$$
$$= I(f) - I^{(n)}(f) - \sum_{i=n+1}^{2n+1} f[x_0, \dots, x_i] \int_a^b \prod_{j=0}^{i-1} (x - x_j)\mathrm{d}x$$

Für $i > n$ gilt

$$\int_a^b \prod_{j=0}^{i-1} (x - x_j)\mathrm{d}x = \int_a^b \underbrace{\prod_{j=0}^{n} (x - x_j)}_{P_{n+1}} \underbrace{\prod_{j=n+1}^{i-1} (x - x_j)}_{\in P_n} \mathrm{d}x$$

Wähle Stützstellen so, dass

$$0 = \int_a^b \prod_{j=0}^{n} (x - x_j) q(x)\mathrm{d}x = \left(\prod_{j=0}^{n} (x - x_j), q \right) \forall q \in P_n$$
$$I(f) - I^{(n)}(f) = I(f) - I^{(2n+1)}(f)$$

$\implies$ $I^{(n)}$ ist exakt für $p \in P_{2n+1}$, das heißt von Ordnung $2n+2$. Mit einem Orthogonalsystem $\{p_0, \dots, p_{n+1}\}$ von P_{n+1} sind die Nullstellen $\lambda_0, \dots, \lambda_n$ von p_{n+1} von Interesse. Frage: Sind die Nullstellen von p_{n+1} reell, einfach und in $[a,b]$?

Satz 3.8 Gegeben sei ein Skalarprodukt auf $C[a,b]$

$$(f,g)_\omega = \int_a^b f(x)g(x)\omega(x)\mathrm{d}x$$

mit integrierbarer Gewichtsfunktion $\omega(x) \geq 0, x \in (a,b)$ mit höchstens endlich vielen Nullstellen. Dann haben die mittels Gram-Schmidt aus \1, x̂1, ...\ bezüglich $(\cdot,\cdot)_\omega$ orthogonalisierten Polynome $\{p_0, p_1, \dots\}$ lauter reelle, einfache Nullstellen in $[a,b]$

Beweis Sei $N_n := \{\lambda \in (a,b) \mid \lambda$ Nullstelle ungerader Vielfachheit von $p_n\}$. Setze

$$q(x) = \begin{cases} 1 & N_n \neq \emptyset \\ \prod_{i=1}^{m}(x-\lambda_i) & N_n = \{\lambda_1, \dots, \lambda_m\}, m > 0 \end{cases}$$

Nach dem Fundamentalsetz der Algebra und wegen $p(x) = x^n - r(x), r \in P_{n-1}$, nach Konstruktion mit Gram-Schmidt (ohne Normalisieren) gilt

$$p_n(x) = \prod_{i=1}^{n}(x-\lambda_i), \lambda_i \in \mathbb{C}, i = 1, \dots, n$$

Ist λ_I nicht reell, so ist $\bar{\lambda}_i$ auch eine Nullstellen von p_N und

$$(x-\lambda_i)x - \bar{\lambda}_i \; = (x-\lambda_I)(x-\lambda_i) \implies |x-\lambda_i|^2 \geq 0$$

$\implies p_n q \in P_{n+m}$ ist reell und hat in $[a,b]$ keinen Vorzeichenwechsel.

$$(p_n, q)_\omega = \int_a^b p_n(x)(x)\omega(x)\mathrm{d}k \neq 0$$

Für $m < n$ ist das ein Wiederspruch zu $p_n \perp p_{n-1} \implies \mu_n = \{\lambda_1, \dots, \lambda_n\}$. Für $[a,b] = [-1,1]$ und $\omega \equiv 1$, das heißt $(\cdot,\cdot)_\omega = (\cdot,\cdot)_2$ sind die p_n mittels $p_n(x) = x^n + \dots$ mormiere Legendre-Polynome L_n(x). Wir wählen also die Nullstellen $\zeta_0, ndots, \lambda_n$ von p_{n+1} beziehungsweise L_{n+2} als Stützstellen einer interpolatorischen Quadraturformel auf $[-1,1]$.

$$I^{(n)}(f) = \sum_{i=9}^{n} \alpha_i f(\lambda_i), \alpha_i = \int_{-1}^{1} \prod_{\substack{j=0 \\ j\neq i}} \frac{x-\lambda_j}{\lambda_i - \lambda_j}\mathrm{d}x$$

□

Satz 3.9 (Gauß-Quadratur) Es gibt genau eine interpolatorische Quadraturformel zu $n+1$ paarweise verschiedenen Stützstellen auf $[-1,b]$ mit Ordnung $2n+2$. Ihre Stützstellen sind gerade die Nullstellen. $\lambda_0, \dots, \lambda_n \in (-1,1)$ das $(n+1)$ - ten Legendre Polynom $L_{n+1} \in P_{n+1}$ und die Gewichte erfüllen

$$\alpha_i = \int_{-1}^{1} \prod_{\substack{j=0 \\ j\neq i}} \left(\frac{x-\lambda_j}{\lambda_i - \lambda_j}\right)^2 \mathrm{d}x > 0, i = 0, \dots, n$$

Für $f \in C^{2n+2}[-1,1]$ besitzt des Restglied die Darstellung

$$R^{(n)} = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \int_{-1}^{1} \prod_{j=0}^{n}(x-\lambda_j)^2 \mathrm{d}x, \xi \in (-1,1)$$

Beweis **Existenz**: Es gilt $p_{n+1} \perp P_n$ Für $\omega = 1$ und $p_n(x) = \prod_{i=0}^{n}(x-\lambda_i) = x^n + \dots$

$$\implies I^{(n)}(f) = I^{(2n+1)}(f)$$

$\implies$ $I^{(n)}$ hat Ordnung $2n+2$. Gewichte:

$$L_i^{(x)}(x) = \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{x-\lambda_j}{\lambda_i-\lambda_j} \in P_n$$

$\implies$ $\left(L_i^{(n)}(x)\right)^2 \in P_{2n}$

$$\implies 0 < \int_{-1}^{1} \left(L_i^{(n)}\right)^2 \mathrm{d}x = \sum_{j=0}^{n} \alpha_j \underbrace{\left(L_i^{(n)}(x_i)\right)}_{\delta_{ij}} = \alpha_i$$

Eindeutigkeit: Sei $\tilde{I}^{(n)}(f) = \sum_{i=0}^{n} \tilde{a}_I f\left(\tilde{\lambda}_i\right)$ ebenfalls der Ordnung $2n+2$. Wie oben folgt $\tilde{\alpha}_i > 0$ mithilfe

$$\tilde{L}_i^{(n)}(x) = \prod_{j=0 j\neq i}^{n} \frac{n-\tilde{\lambda}_j}{\tilde{\lambda}_i-\tilde{\lambda}_j}$$

$$\begin{aligned} 0 &= \int_{-1}^{1} \frac{1}{\tilde{\alpha}_i} \tilde{L}_i^{(n)} p_{n+1}(x)\mathrm{d}x \\ &= \sum_{j=0}^{n} \frac{\tilde{\alpha}_i}{\tilde{\alpha}_i} \underbrace{\tilde{L}_i^{(n)}\left(\tilde{\lambda}_j\right)}_{\delta_{ij}} p_{n+1}\left(\tilde{\lambda}_j\right) = p_{n+1}\left(\tilde{\lambda}_i\right), i = 0, \ldots, n \end{aligned}$$

$\implies$ $\tilde{\lambda}_i = \lambda_i$ und $\tilde{\alpha}_i = \alpha_i, i = 1, \ldots, n$.
Restglied: Für $f \in C^{(2n+2)}[-1,1]$ hat der Hermite-Interpolant $h \in P_{2n+1}$ zu den Bedingungen

$$h(\lambda_i) = f(\lambda_i), h'(\lambda_i) = f'(\lambda_i), i = 0, \ldots, n$$

die Darstellung:

$$f(x) - h(x) = f[\lambda_0, \lambda_0, \ldots, \lambda_n, \lambda_n, x] \prod_{i=0}^{n} (x-\lambda_i)^2$$

$$\begin{aligned} \implies I(f) - I^{(f)} &= I(f) - \underbrace{I^{(n)}(h)}_{=I(h)} - \left(I^{(n)}(f) - I^{(n)}(h)\right) \\ &= I(f-h) - I^{(n)}(f-h) \\ &= \int_{-1}^{1} f[\lambda_0, \lambda_0, \ldots, \lambda_n, \lambda_n] \underbrace{\prod_{i=0}^{n} (x-\lambda_i)^2}_{>0} \mathrm{d}x - \underbrace{\sum_{i=0}^{n} \alpha_i [f(\lambda_i) - h(\lambda_i)]}_{0} \end{aligned}$$

Mit verallgemeinertem Mittelwertsatz folgt:

$$= \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \int_{-1}^{1} \prod_{i=0}^{n} (x-\lambda_i)^2 \mathrm{d}x \qquad \square$$

Die $\lambda_i^{(n)}$ (Nullstellen von p_{n+1}) und die dazugehörigen α_i lassen sich tabellieren. Durch Transformation von $[a, b]$ auf $[-1, 1]$ erhält man eine allgemeine Quadraturformel.

Satz 3.10 (Konvergenz der Gauß-Quadratur) Sei $I^{(n)}(f)$ die $(n+1)$ punktige Grauß-Formel zur Berechnung von $I(f) = \int_{-1}^{1} f(x)\mathrm{d}x$. Für jedes $f \in C[-1, 1]$ konvergiert $I^{(n)}(f) \xrightarrow{n\to\infty} I(f)$

Beweis Es gilt

$$I^{(n)}(f) = \sum_{i=0}^{n} \alpha_i^{(n)} f\left(\lambda_i^{(n)}\right), \alpha_i^{(n)} > 0, \sum_{i=0}^{n} \alpha_i^{(n)} = 2$$

Sei $\varepsilon > 0$. Nach dem Weierstrassschem Approximationssatz gibt es $p_\varepsilon \in P_n$ mit

$$\max_{x\in[-1,1]} |f(x) - p_\varepsilon(x)| \leq \frac{\varepsilon}{4}$$

Für $n > \frac{1}{2}m - 1$ (das heißt $2n + 2 > m$) gilt

$$\left|I(f) - I^{(n)}(f)\right| \leq \underbrace{|I(f - p_\varepsilon)|}_{\leq \frac{\varepsilon}{4}2} + \underbrace{\left|I(p_\varepsilon) - I^{(n)}(p_\varepsilon)\right|}_{0} + \underbrace{\left|I^{(n)}(f - p_\varepsilon)\right|}_{\leq \frac{\varepsilon}{4}2} \leq \varepsilon$$

□

Wiederholung: Gauß-Quadratur

- $n + 1$ Stützstellen, Ordnung $2n + 2$ (optimal)
- x_i Nullstellen des Legendre Polynoms p_{n+1}
- $I^{(n)}(f) \xrightarrow{n\to\infty} I(f)$ für f stetig
- Verallgemeinerung auf gewichtete Integrale

$$\int_a^b f(x)\omega(x)\mathrm{d}x I(f\omega) I_\omega(f)$$

$\implies$ Orthogonalisiere bezüglich

$$(f, g)_\omega = \int_a^b f(x)g(x)\omega(x)\mathrm{d}x$$

3.2 Praktische Aspekte der Quadratur

Ziel: Möglichst hohe Genauigkeit bei möglichst wenig Funktionsauswertungen. Schwierigkeiten:

- Fehlerschätzung: $f^{(k)}$ nur schwer zugänglich für $k > 2 \implies$ a-posteriori Fehlerschätzer.

  Beispiel 3.11 1. Vergleiche $I_n(f)$ und $I_{\frac{n}{2}}(f)$ bei summierten Quadraturformeln
  2. Extrapolationfehler

- Wiederbenutzung bereits berechneter Werte von f
  - schwierig bei Gauß
  - einfach bei Newton-Cotes

4 Lineare Gleichungssystem

Gegeben: $A \in \mathbb{R}^{m\times n} = (a_{ij}), b \in \mathbb{R}^m$. Gesucht: $x \in \mathbb{R}^n$ mit $Ax = b \implies m$ Gleichungen, n Unbekannte. Das lineare Gleichungssystem $Ax = b$ heißt

- unterbestimmt, falls $m < n$
- überbestimmt falls $m > n$
- quadratisch falls $m = n$

Störungsstheorie:

- Konditionierung von quadratischen linearen Gleichungssystemen
- Fehlereinfluss von Datenfehlern und Rundungsfehlern auf Lösung x

Vektor- und Matrizennormen:
Sei $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$. Erinnerung: Eigenschaften einer Norm: $\|\cdot\| : \mathbb{K}^n \to \mathbb{R}$

- Definitheit: $\|x\| > 0 \forall x \in \mathbb{K}^n \setminus \{0\}$
- Positive Homogenität: $\|\alpha x\| = |\alpha|\|x\| \forall x \in \mathbb{K}^n, \alpha \in \mathbb{K}$
- Subadditivität: $\|x+y\| \leq \|x\| + \|y\| \forall x, y \in \mathbb{K}^n$

Beispiel 4.1 Euklidische Norm: (l_2)

$$\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2}$$

Maximumsnorm (l_∞)

$$\|x\|_\infty = \max_{i=1,\dots,n} |x_i|$$

l_1 -Norm:

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

l_p -Norm, $p \geq 1, p < \infty$

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

Betrachte Vektorraum der $n \times n$ -Matrizen $A \in \mathbb{K}^{n\times n}$

Definition 4.2 Eine Norm $\|\cdot\|$ auf $\mathbb{K}^{n\times n}$ heißt verträglich wit einer Vektornorm $\|\cdot\|$ auf $\mathbb{K}^n$, wenn gilt:

$$\|Ax\| \leq \|A\|\|x\| \forall x \in \mathbb{K}^n, A \in \mathbb{K}^{n\times n}$$

Sie heißt Matrizennorm, wenn sie submultiplikativ ist

$$\|AB\| \leq \|A\|\|B\| \forall A, B \in \mathbb{K}^{n\times n}$$

Beispiel 4.3 Die Frobeniusnorm

$$\|A\|_{Fr} = \left(\sum_{i,j=1}^{n} |a_{ij}|^2 \right)^{1/2}$$

ist eine mit $\|\cdot\|_2$ verträgliche Matrizennorn.
Die natürliche Matrizennorm

$$\|A\| = \sup_{x \in \mathbb{K}^n \setminus \{0\}} \frac{\|Ax\|}{\|x\|} = \sup_{\substack{x \in \mathbb{K}^n \\ \|x\|=1}} \|Ax\|$$

ist eine mit $\|\cdot\|$ verträgliche Matrizennorm (Übung!). Es gilt

$$\|\mathbb{I}\| = \sup_{\substack{x \in \mathbb{K}^n \\ \|x\|=1}} \|\mathbb{I}x\| = 1$$

Lemma 4.4 Die natürlichen Matrizennormen zu $\|\cdot\|_\infty$ und $\|\cdot\|_1$ sind die „maximale Zeilen-/Spaltensumme“:

$$\|A\|_\infty = \max_{j=1,\dots,n} \sum_{k=1}^{n} |a_{jk}|$$

$$\|A\|_1 = \max_{k=1,\dots,n} \sum_{j=1}^{n} |a_{jk}|$$

Beweis Skript. □

Betrachte: $Ax = b$ und Störung

$$\underbrace{(A + \delta A)}_{\tilde{A}} \underbrace{(x + \delta x)}_{\tilde{x}} = \underbrace{b + \delta b}_{\tilde{b}}$$

Satz 4.5 (Neumann-Reihe) Gilt $\|A\| < 1$, so

$$\mathbb{I} - \mathbb{A} \sum_{k=0}^{\infty} A^k = \mathbb{I}$$

Beweis Für die Partialsummen gilt

$$(\mathbb{I} - A) \sum_{k=0}^{n} A^k = \mathbb{I} - A + A - A^2 + A^2 \cdots - A^{n+1} \xrightarrow{n \to \infty} \mathbb{I}$$

wegen $\|A^k\| \leq \|A\|^k \xrightarrow{k \to \infty} 0$. □