

Numerische Mathematik für Elektrotechniker

Interpolation III/Numerische Integration I

Benjamin Berkels

Karl-Heinz Brakhage, Thomas Jankuhn, Christian Löbbert

Institut für Geometrie und Praktische Mathematik
RWTH Aachen

Wintersemester 2019/2020

Zusammenfassung: Darstellung Interpolationspolynom

Die eindeutige Lösung $P_n = P(f \mid x_0, \dots, x_n)$ hat unterschiedliche Darstellungen abhängig von der Basis in Π_n :

1. Lagrange-Basis

$$P_n(x) = \sum_{j=0}^n f(x_j) \ell_{jn}(x), \quad \ell_{jn}(x) = \prod_{\substack{k=0 \\ k \neq j}}^n \frac{x - x_k}{x_j - x_k}.$$

2. monomiale Basis

$$P_n(x) = \sum_{j=0}^n a_j x^j, \quad V_n \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{pmatrix}.$$

3. Newtonsche Basis

$$P_n(x) = \sum_{j=0}^n [x_0, \dots, x_j] f \omega_j(x), \text{ Knotenpolynome } \omega_j \text{ und}$$

$$[x_0, \dots, x_n] f = \frac{[x_1, \dots, x_n] f - [x_0, \dots, x_{n-1}] f}{x_n - x_0}.$$

Zusammenfassung: Knotenwahl

Definiere

$$\omega_{n+1}(x) := \prod_{j=0}^n (x - x_j) \text{ und } M_{n+1}(f) := \max_{x \in [a,b]} \frac{|f^{(n+1)}(x)|}{(n+1)!}$$

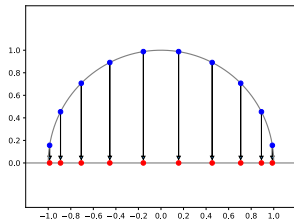
Dann erhält man für die Fehlerschranke

$$|f(x) - P(f|x_0, \dots, x_n)(x)| \leq |\omega_{n+1}(x)| M_{n+1}(f) .$$

Tschebyscheff-Knoten minimieren
den Fehler der Knotenpolynome.

Auf $[-1, 1]$ durch gegeben:

$$x_j = \cos\left(\frac{2j+1}{2n+2}\pi\right), \quad j = 0, \dots, n$$



Zusammenfassung: Interpolationsfehler

- ▶ Sei $a := \min\{x_0, \dots, x_n\}$
und $b := \max\{x_0, \dots, x_n\}$ mit n fest.
- ▶ Die Länge des Intervalls $h := b - a$ sei veränderbar.
- ▶ Falls $x \in I := [a, b]$ liegt, erhält man sofort die grobe Abschätzung

$$|\omega_{n+1}(x)| \leq h^{n+1},$$

und somit

$$\|f - P(f|x_0, \dots, x_n)\|_{L^\infty(I)} \leq \frac{h^{n+1}}{(n+1)!} \|f^{(n+1)}\|_{L^\infty(I)}.$$

- ▶ Der Fehler wird kleiner, wenn die Stützstellen gemäß h zusammenrücken.

Dieser Effekt wird für die numerische Differentiation und Splineinterpolation genutzt.

Zusammenfassung: Fehleranalyse Numerische Differentiation

- Differenzenquotient der zweiten Ableitung

$$\Delta_h := \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}.$$

- Fehler in den Daten

$$\tilde{\Delta}_h := \frac{\tilde{f}(x+h) - 2\tilde{f}(x) + \tilde{f}(x-h)}{h^2}.$$

- Fehler in Δ_h aufgrund von Datenfehler ($|f(y) - \tilde{f}(y)| \leq \epsilon$)

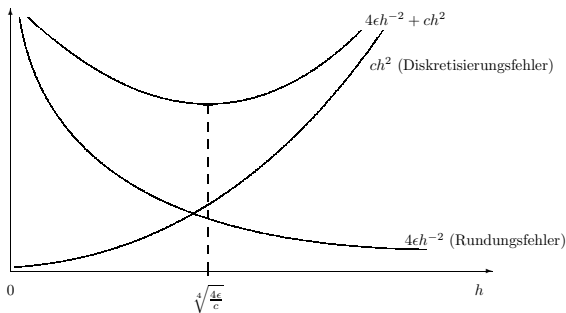
$$\begin{aligned} |\Delta_h - \tilde{\Delta}_h| &= \frac{1}{h^2} \left| \left(f(x+h) - \tilde{f}(x+h) \right) - 2 \left(f(x) - \tilde{f}(x) \right) \right. \\ &\quad \left. + \left(f(x-h) - \tilde{f}(x-h) \right) \right| \leq \frac{4\epsilon}{h^2}. \end{aligned}$$

Gesamtfehler

$$|\tilde{\Delta}_h - f''(x)| \leq |\tilde{\Delta}_h - \Delta_h| + |\Delta_h - f''(x)| \leq \frac{4\epsilon}{h^2} + c h^2$$

Zusammenfassung: Gesamtfehler numerische Differentiation

- Die Schranke wird für $h = \sqrt[4]{4\epsilon/c}$ minimal.
- Bsp: $\epsilon = 10^{-9} \Rightarrow h \approx 10^{-2}$, kleineres h vergrößert Fehler



Merke

Man sollte stets dafür sorgen, dass **Rundungsfehler** einen **kleineren Einfluß** haben als **Diskretisierungsfehler**.

Heute in der Vorlesung

Themen: Dahmen & Reusken Kap 8.6 und 10.1-10.3

Kapitel 8.6: Beispiel einer Splineinterpolation

Kapitel 10: Numerische Integration

- ▶ Einleitung
- ▶ Trapezregel, Simpson-Regel, Fehlerformeln
- ▶ Newton-Cotes-Formeln
- ▶ Gauß-Quadratur

Was Sie mitnehmen sollten:

- ▶ Grundidee der numerischen Integration
- ▶ Wie konstruiert man eine wiederholte Trapez- oder Simpson-Regel?
- ▶ Wie sehen die Fehlerschranken dazu aus?
- ▶ Was sind die Grundideen und wichtige Unterschiede der Newton-Cotes- und Gauß-Methoden?

Beispiel einer Splineinterpolation

Splinefunktionen bilden ein flexibles und effizientes Hilfsmittel, um größere Datenmengen zu interpolieren oder zu approximieren.

Seien

$$a = x_0 < x_1 < \dots < x_n = b$$

Stützstellen und die entsprechenden Daten an diesen Stützstellen:

$$f_0, f_1, \dots, f_n$$

Der Raum der kubischen Splines

$$S_3^n := \left\{ g \in C^2([a, b]) \mid g|_{[x_i, x_{i+1}]} \in \Pi_3, i = 0, 1, \dots, n-1 \right\}$$

Es gilt

$$\dim S_3^n = n + 1 + 2 = \text{Knoten} + \text{Randbedingungen}.$$

Splines

Zur Lösung des Interpolationsproblems wird $S \in S_3^n$ gesucht, so dass

$$S(x_j) = f_j, \quad j = 0, \dots, n.$$

Satz (Kubische Splininterpolation)

Zu den Daten

$$f_0, f_1, \dots, f_n$$

gibt es genau eine Funktion $S \in S_3^n$, so dass

Knoten: $S(x_j) = f_j, \quad j = 0, 1, \dots, n,$

und

Randbedingungen: $S''(a) = S''(b) = 0.$

Splines: Biegeenergieminimierung

Lemma 8.38

Sei g eine beliebige Funktion aus $C^2([a, b])$ mit

$$g(x_j) = f_j, \quad j = 0, 1, \dots, n$$

und

$$g''(a) = g''(b) = 0.$$

Für die eindeutige Lösung S aus den vorigen Satz gilt

$$\int_a^b S''(x)^2 dx \leq \int_a^b g''(x)^2 dx.$$

Diese Eigenschaft bedeutet, dass der kubische Spline S unter allen Funktionen $g \in C^2([a, b])$, die dieselben Anforderungen an die Interpolation erfüllen, **die Biegeenergie minimiert**.

Methode zur Berechnung der gesuchten Lösung S

Wichtig ist die Wahl einer geeigneten Basis (Stichwort **B-Splines**).

Für das folgende Beispiel berechnen wir die Lösung allerdings elementar ohne eine Basis zu konstruieren.

Äquidistante Stützstellen:

$$x_{j+1} - x_j = h, \quad j = 0, 1, \dots, n-1.$$

Bezeichnungen: Biegeenergie der Knoten (gesucht)

$$m_j := S''(x_j), \quad j = 0, 1, \dots, n,$$

und

$$I_j := [x_j, x_{j+1}], \quad j = 0, 1, \dots, n-1.$$

Methode zur Berechnung der gesuchten Lösung S

Wegen $S|_{I_j} \in \Pi_3$ ergibt sich, dass $S''|_{I_j}$ linear ist und damit dass

$$S''|_{I_j}(x) = \frac{(x_{j+1} - x) m_j + (x - x_j) m_{j+1}}{h}.$$

Zweifache Integration zusammen mit den Forderungen

$$S(x_j) = f_j, \quad S(x_{j+1}) = f_{j+1}$$

ergibt (nach längerer Rechnung)

$$\begin{aligned} S|_{I_j}(x) = & \frac{(x_{j+1} - x)^3 m_j + (x - x_j)^3 m_{j+1}}{6h} \\ & + \frac{(x_{j+1} - x) f_j + (x - x_j) f_{j+1}}{h} \\ & - \frac{1}{6} h [(x_{j+1} - x) m_j + (x - x_j) m_{j+1}]. \end{aligned}$$

Man zeigt leicht, dass dieses S obige Bedingungen erfüllt.

Methode zur Berechnung der gesuchten Lösung S

Für dieses stückweise Polynom S gilt

$$S|_{I_j} \in \Pi_3, \quad S \in C([a, b]),$$

$$S''|_{I_j}(x_{j+1}) = m_{j+1} = S''|_{I_{j+1}}(x_{j+1}).$$

- ▶ Stetigkeit folgt aus Interpolationsbedingungen $S(x_j) = f_j$.
- ▶ Stetigkeit der zweiten Ableitung nach Konstruktion.

Es verbleibt die noch unbekannten Größen m_j so zu wählen, dass die erste Ableitung von S in den Stützstellen x_j stetig ist:

$$S'|_{I_{j-1}}(x_j) = S'|_{I_j}(x_j), \quad j = 1, \dots, n-1,$$

Dies führt auf die Bedingungen

$$m_{j-1} + 4m_j + m_{j+1} = \frac{6}{h^2}(f_{j-1} - 2f_j + f_{j+1}), \quad j = 1, \dots, n-1.$$

Methode zur Berechnung der gesuchten Lösung S

Es gilt nach Voraussetzung $S''(x_0) = S''(x_n) = 0$, also

$$m_0 = m_n = 0.$$

Insgesamt ergibt sich folgendes lineare $(n - 1)$ -dimensionale Gleichungssystem:

$$\begin{pmatrix} 4 & 1 & & \\ 1 & 4 & \ddots & 0 \\ & \ddots & \ddots & \ddots \\ & 0 & \ddots & \ddots & 1 \\ & & & 1 & 4 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ \vdots \\ m_{n-1} \end{pmatrix} = \frac{6}{h^2} \begin{pmatrix} f_0 - 2f_1 + f_2 \\ f_1 - 2f_2 + f_3 \\ \vdots \\ \vdots \\ f_{n-2} - 2f_{n-1} + f_n \end{pmatrix}$$

Beispiel 8.40

Daten:

i	0	1	2	3	4	5	6	7
x_i	3	4	5	6	7	8	9	10
$f(x_i)$	2.5	2.0	0.5	0.5	1.5	1.0	1.125	0.0

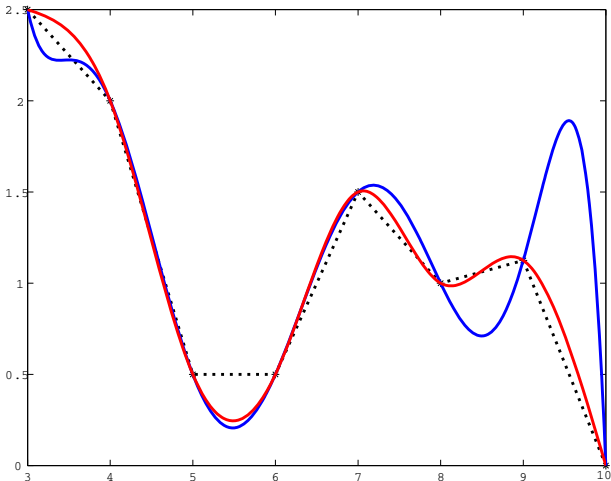
Für die Splineinterpolation $S \in S_3^7$ mit den Endbedingungen $S''(3) = S''(10) = 0$ wird das zugehörige System

$$\begin{pmatrix} 4 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \end{pmatrix} = 6 \begin{pmatrix} -1 \\ 1.5 \\ 1 \\ -1.5 \\ 0.625 \\ -1.25 \end{pmatrix}$$

gelöst.

Beispiel 8.40

Lagrange-Interpolationspolynom P_7 vs Spline-Interpolation $S \in S_3^7$



Fehler bei der kubischen Splineinterpolation

Bemerkung 8.41

Bezeichnet

$$h := \max_{i=0,\dots,n-1} x_{i+1} - x_i$$

den maximalen Knotenabstand, kann man die Abschätzung

$$\|f - S\|_{L_\infty([a,b])} \leq C h^4 \|f^{(4)}\|_{L_\infty([a,b])}$$

für eine von h und f unabhängige Konstante C zeigen.

Höhere Genauigkeit gewinnt man also durch Verringerung der Schrittweite bzw. Erhöhung der Knotenzahl.

Numerische Integration

Kapitel 10:

Numerische Integration

Grundlegende Idee

Die gängige Strategie zur näherungsweise Berechnung des Integrals

$$\int_a^b f(x) dx = ?$$

lässt sich folgendermaßen umreißen:

1. Man unterteile $[a, b]$ in Teilintervalle $[t_{k-1}, t_k]$ z.B. mit $t_j = a + j h$, $j = 0, \dots, n$, $h = (b - a)/n$.
2. **Approximiere** f auf jedem Intervall $[t_{k-1}, t_k]$ durch eine **einfach/exakt zu integrierende Funktion** g_k , und verwende

$$\int_a^b f(x) dx = \sum_{k=1}^n \int_{t_{k-1}}^{t_k} f(x) dx \approx \sum_{k=1}^n \int_{t_{k-1}}^{t_k} g_k(x) dx$$

als Näherung für das exakte Integral.

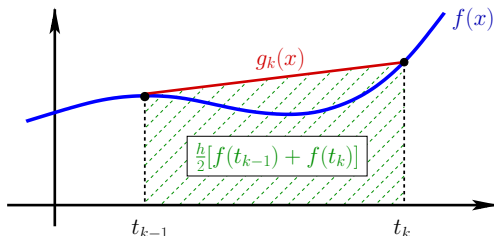
Trapezregel

- ▶ Lineare Interpolation an den Intervallenden von $[t_{k-1}, t_k]$, d.h.

$$g_k(x) = \frac{x - t_{k-1}}{h} f(t_k) + \frac{t_k - x}{h} f(t_{k-1}),$$

- ▶ Das Integral ist damit gegeben durch

$$\int_{t_{k-1}}^{t_k} g_k(x) dx = \frac{h}{2} [f(t_{k-1}) + f(t_k)].$$



Summierte Trapezregel

Wiederholtes Anwenden der Trapezregel auf jedem Teilintervall $[t_{k-1}, t_k]$ als Näherung für $\int_a^b f(x) dx$ liefert die

Summierte Trapezregel

$$T(h) = h \left[\frac{1}{2} f(a) + f(t_1) + \cdots + f(t_{n-1}) + \frac{1}{2} f(b) \right]$$

Für den Verfahrensfehler der Teilintegrale gilt folgende Darstellung:

Lemma 10.1

Sei $f \in C^2([t_{k-1}, t_k])$. Es gilt:

$$\frac{h}{2} [f(t_{k-1}) + f(t_k)] = \int_{t_{k-1}}^{t_k} f(x) dx + \frac{f''(\xi_k)}{12} h^3,$$

für $\xi_k \in [t_{k-1}, t_k]$.

Summierte Trapezregel

Für den Verfahrensfehler des gesamten Integrals $T(h)$ ergibt sich damit die Abschätzung

$$\begin{aligned} \left| T(h) - \int_a^b f(x) dx \right| &= \left| \sum_{k=1}^n \frac{f''(\xi_k)}{12} h^3 \right| \\ &\leq \frac{h^3}{12} \sum_{k=1}^n |f''(\xi_k)| \leq \frac{h^3}{12} n \max_{x \in [a,b]} |f''(x)|. \end{aligned}$$

Mit $n h = b - a$ ergibt sich insgesamt die

Fehlerschranke

$$\left| T(h) - \int_a^b f(x) dx \right| \leq \frac{h^2}{12} (b - a) \max_{x \in [a,b]} |f''(x)|$$

Summierte Trapezregel

Ebenfalls erhält man wegen

$$E(h) := T(h) - \int_a^b f(x) dx = \sum_{k=1}^n \frac{f''(\xi_k)}{12} h^3 = \frac{h^2}{12} \sum_{k=1}^n h f''(\xi_k)$$

und

$$\lim_{h \rightarrow 0} \frac{E(h)}{h^2} = \frac{1}{12} \int_a^b f''(x) dx = \frac{1}{12} (f'(b) - f'(a))$$

die

Fehlerschätzung

$$E(h) \approx \hat{E}(h) := \frac{h^2}{12} (f'(b) - f'(a))$$

Beispiel 10.2

Zur näherungsweise Berechnung von

$$I = \int_0^{\pi/2} x \cos(x) + e^x dx = \frac{\pi}{2} + e^{\frac{1}{2}\pi} - 2$$

mit der Trapezregel ergeben sich die in folgender Tabelle angegebenen Näherungswerte, Verfahrensfehler und Fehlerschätzungen.

n	$T(h)$	$ E(h) = T(h) - I $	$ \hat{E}(h) = \frac{h^2}{12} f'(\frac{\pi}{2}) - f'(0) $
4	4.396928	1.57e-02	1.59e-02
8	4.385239	3.97e-03	3.98e-03
16	4.382268	9.95e-04	9.96e-04
32	4.381523	2.49e-04	2.49e-04

Allgemeine Quadraturformel

- ▶ Für ein typisches Teilintervall $[t_{k-1}, t_k]$ stehe der Einfachheit halber im Folgenden $[c, d]$.
- ▶ Seien nun $x_0, \dots, x_m \in [c, d]$ paarweise verschiedene Punkte.
- ▶ Integration des Interpolationspolynoms liefert die Quadraturformel

$$I_m(f) = \int_c^d P(f|x_0, \dots, x_m)(x) dx.$$

Satz 10.3

Sei $I_m(f)$ wie oben. Für jedes Polynom $Q \in \Pi_m$ gilt

$$I_m(Q) = \int_c^d Q(x) dx.$$

Man sagt, die Quadraturformel ist **exakt vom Grade m** .

Allgemeine Quadraturformel

Wir erhalten damit als Näherung für das exakte Integral

$$\begin{aligned}
 \int_c^d P(f|x_0, \dots, x_m)(x) dx &= \int_c^d \sum_{j=0}^m f(x_j) \ell_{jm}(x) dx \\
 &= \sum_{j=0}^m f(x_j) \int_c^d \ell_{jm}(x) dx \\
 &= \sum_{j=0}^m f(x_j) \int_c^d \prod_{\substack{k=0 \\ k \neq j}}^m \frac{x - x_k}{x_j - x_k} dx \\
 &= h \sum_{j=0}^m f(x_j) \underbrace{\frac{1}{h} \int_c^d \prod_{\substack{k=0 \\ k \neq j}}^m \frac{x - x_k}{x_j - x_k} dx}_{c_j} \\
 &= h \sum_{j=0}^m f(x_j) \cdot c_j
 \end{aligned}$$

Allgemeine Quadraturformel

Lemma 10.4

Es gibt **Gewichte** $c_0, \dots, c_m$, so dass $I_m(f)$ die Form

$$I_m(f) = h \sum_{j=0}^m c_j f(x_j)$$

hat, wobei wieder $h = d - c$.

Die c_j sind durch

$$c_j = \frac{1}{h} \int_c^d \prod_{\substack{k=0 \\ k \neq j}}^m \frac{x - x_k}{x_j - x_k} dx = \frac{1}{h} \int_c^d \ell_{jm}(x) dx$$

gegeben, wobei ℓ_{jm} ($0 \leq j \leq m$) die Lagrange-Fundamentalpolynome zu den Stützstellen $x_0, \dots, x_m$ sind.

Newton-Cotes-Formeln

Wählt man speziell die Stützstellen x_j äquidistant

$$x_0 = c + \frac{1}{2}h =: c + \xi_0 h, \quad \text{wenn } m = 0$$

$$x_j = c + \frac{j}{m}h =: c + \xi_j h, \quad j = 0, \dots, m, \quad \text{wenn } m > 0,$$

erhält man die sog. **Newton-Cotes-Formeln**.

Beispiel

Für $m = 1$ erhalten wir $x_0 = c$, $x_1 = d$ und

$$c_0 = \frac{1}{h} \int_c^d \frac{x - x_1}{x_0 - x_1} dx = \frac{1}{2},$$

$$c_1 = \frac{1}{h} \int_c^d \frac{x - x_0}{x_1 - x_0} dx = \frac{1}{2}.$$

Newton-Cotes-Formeln

Man kann die Quadraturformel in der Form

$$I_m(f) = h \sum_{j=0}^m c_j f(c + \xi_j h)$$

mit **normierten** Stützstellen ξ_j und Gewichten c_j schreiben, die jetzt **unabhängig vom speziellen Intervall** $[c, d]$ sind, z.B.

m		ξ_j	c_j	$I_m(f) - \int_c^d f(x)dx$
0	Mittelpunktsregel	$\frac{1}{2}$	1	$-\frac{1}{24} h^3 f^{(2)}(\xi)$
1	Trapezregel	0, 1	$\frac{1}{2}, \frac{1}{2}$	$\frac{1}{12} h^3 f^{(2)}(\xi)$
2	Simpson-Regel	$0, \frac{1}{2}, 1$	$\frac{1}{6}, \frac{4}{6}, \frac{1}{6}$	$\frac{1}{90} \left(\frac{1}{2}h\right)^5 f^{(4)}(\xi)$
3	$\frac{3}{8}$ -Regel	$0, \frac{1}{3}, \frac{2}{3}, 1$	$\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}$	$\frac{3}{80} \left(\frac{1}{3}h\right)^5 f^{(4)}(\xi)$
4	Milne-Regel	$0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$	$\frac{7}{90}, \frac{32}{90}, \frac{12}{90}, \frac{32}{90}, \frac{7}{90}$	$\frac{8}{945} \left(\frac{1}{4}h\right)^7 f^{(6)}(\xi)$

Summierte Newton-Cotes-Formeln

Beispiel: Summierte Simpson-Regel

$$S(h) = \int_a^b f(x) dx + E(h)$$

mit

$$S(h) = \frac{h}{6} \left[f(t_0) + 4f\left(\frac{t_0 + t_1}{2}\right) + 2f(t_1) + 4f\left(\frac{t_1 + t_2}{2}\right) + 2f(t_2) + \cdots + 2f(t_{n-1}) + 4f\left(\frac{t_{n-1} + t_n}{2}\right) + f(t_n) \right]$$

und Fehlerschranke

$$E(h) = \sum_{k=1}^n \frac{1}{90} \left(\frac{h}{2}\right)^5 f^{(4)}(\xi_k) = \frac{h^4}{2880} \sum_{k=1}^n h f^{(4)}(\xi_k),$$

für $\xi_k \in [t_{k-1}, t_k]$.

Summierte Newton-Cotes-Formeln

Es gilt, wegen $n h = b - a$,

$$|E(h)| \leq \frac{h^4}{2880} (b - a) \|f^{(4)}\|_{\infty},$$

$$E(h) \approx \frac{h^4}{2880} \int_a^b f^{(4)}(x) dx = \frac{h^4}{2880} \left(f^{(3)}(b) - f^{(3)}(a) \right).$$

Beachte

Beim [Aufsummieren](#) der einzelnen Teilintegrale,

$$\int_a^b f(x) dx = \sum_{k=1}^n \int_{t_{k-1}}^{t_k} f(x) dx,$$

geht im Fehler eine h -Potenz verloren.

Beispiel 10.5

Wie in Beispiel 10.2 ergeben sich für die näherungsweise Berechnung von

$$I = \int_0^{\pi/2} x \cos(x) + e^x dx = \frac{\pi}{2} + e^{\frac{1}{2}\pi} - 2$$

die Resultate, die in folgender Tabelle dargestellt sind.

n	$S(h)$	$ E(h) $	$\frac{h^4}{2880} f^{(3)}(\frac{\pi}{2}) - f^{(3)}(0) $
4	4.381343022	6.93e-05	6.92e-05
8	4.381278035	4.33e-06	4.33e-06
16	4.381273978	2.70e-07	2.70e-07
32	4.381273725	1.69e-08	1.69e-08

Gauß-Quadratur

Problem Newton-Cotes: Gewichte ändern ab $m \geq 8$ Vorzeichen
 $\Rightarrow$ **Auslöschung.**

Zielvorgabe

Entwickle für $m \in \mathbb{N}$ eine Formel

$$\sum_{i=0}^m \hat{w}_i f(x_i) = \int_c^d P(f|x_0, \dots, x_m)(x) dx$$

mit:

- ▶ **positiven Gewichten** $\hat{w}_i, i = 0, \dots, m$
- ▶ mit möglichst **hohem Exaktheitsgrad** $n \geq m$, d.h.

$$\int_c^d Q(x) dx = \sum_{i=0}^m \hat{w}_i Q(x_i), \quad \forall Q \in \Pi_n.$$

Gauß-Quadratur: Vorbetrachtung

- ▶ Exaktheitsgrad kann **höchstens $2m + 1$** sein.

Beweis: Angenommen der Exaktheitsgrad wäre $2m + 2$.

Für das Polynom $Q(x) = \prod_{j=0}^m (x - x_j)^2$ gilt $\int_a^b Q(x) dx > 0$, aber $I_{2m+2}(Q) = 0$, da $P(Q|x_0, \dots, x_m) = 0$, Widerspruch.

- ▶ Man kann zeigen: Exaktheitsgrad bei Newton-Cotes-Formeln $I_m(f)$ ist m (m ungerade) oder $m + 1$ (m gerade).

Idee: Nutze Gewichte und **Stützstellen** als freie Parameter!

Heuristik: Für $j = 0, 1, \dots$ ist $\int_c^d x^j dx = \sum_{k=0}^m w_k x_k^j$ eine

nichtlineare Gleichung mit $2m + 2$ freien Parametern w_k, x_k für $k = 0, \dots, m$.

$\Rightarrow$ Problem scheint für $j = 0, \dots, 2m + 1$ lösbar zu sein.

$\Rightarrow$ **Gaußsche Quadraturformeln**

Gauß-Quadratur

Exaktheitsgrad kann **höchstens** $2m + 1$ sein.

$\Rightarrow$ **Gaußsche Quadraturformeln**

Satz 10.6

Sei $m \geq 0$. Es **existieren** Stützstellen $x_0, \dots, x_m \in (c, d)$ und **positive** Gewichte $w_0, \dots, w_m$, so dass mit $h = d - c$

$$h \sum_{i=0}^m w_i f(x_i) = \int_c^d f(x) dx + E_f(h)$$

und

$$E_Q = 0 \quad \text{für alle } Q \in \Pi_{2m+1}.$$

Ferner gilt für passendes $\xi \in [c, d]$

$$|E_f(h)| = \frac{((m+1)!)^4}{((2m+2)!)^3 (2m+3)} h^{2m+3} \left| f^{(2m+2)}(\xi) \right|.$$

Fehlerbetrachtung

- ▶ Betrachte summierte Quadraturformel

$$I_{k,n} \approx \int_a^b f(x) dx = I(f),$$

wobei $[a, b]$ in n Teilintervalle mit Länge $(b - a)/n = h$ unterteilt wird und auf jedem Teilintervall eine **Gauß-Quadratur** mit $k = m + 1$ Stützstellen angewandt wird.

- ▶ Für glatte Funktionen (d.h. $|f^{(2k)}|$ wird nicht allzu groß, wenn k größer wird) wird die Genauigkeit der Gauß-Quadratur im Wesentlichen durch den Faktor $C_{k,h}$ bestimmt:

$$C_{k,h} := \frac{(k!)^4}{((2k)!)^3 (2k+1)} h^{2k+1}.$$

Wir erhalten folgende Werte ...

Numerische Tests

$h = (b - a)/n$	$k = 2$	$k = 4$	$k = 8$
4	2.4e-01	1.5e-04	2.9e-13
2	7.4e-03	2.9e-07	2.2e-18
1	2.3e-04	5.6e-10	1.7e-23
0.5	7.2e-06	1.1e-12	1.3e-28

- Für $I_{2 \cdot k, n}$ und $I_{k, 2 \cdot n}$ wird die Anzahl der Funktionsauswertungen etwa verdoppelt im Vergleich zu $I_{k, n}$.
- In obiger Tabelle kann man sehen, dass man

$$|I - I_{2 \cdot k, n}| \ll |I - I_{k, 2 \cdot n}|$$

erwarten darf.

- In der Praxis wird daher bei der Gauß-Quadratur n in der Regel klein gewählt, oft sogar $n = 1$.

Matlab-Demo

Beispiel 10.7

Wie in Beispiel 10.2 ergeben sich für die Gauß-Quadratur von

$$I = \int_0^{\pi/2} x \cos(x) + e^x dx = \frac{\pi}{2} + e^{\frac{1}{2}} \pi - 2$$

mit $[c, d] = [0, \frac{\pi}{2}]$ (d.h. $n = 1$), die Resultate:

m	$I_{m+1,1}$	$ I_{m+1,1} - I $
1	4.3690643196	1.22e-02
2	4.3813023500	2.86e-05
3	4.3812734352	2.73e-07
4	4.3812737081	3.00e-10

Die Genauigkeit der Gauß-Quadratur mit **5** Funktionswerten ($m = 4; k = 5$) ist besser als die der Simpson-Regel angewandt auf $n = 32$ Teilintervalle (vgl. Beispiel 10.5), wobei insgesamt **65** Funktionwerte benötigt werden.

Beispiel 10.8

Aufgabe: Finden von Gewichten und Stützstellen.

Berechnung der Stützstellen und Gewichte der Gauß-Quadratur für $[c, d] = [-1, 1]$ und $m = 1$.

- Die Gauß-Quadraturformel

$$I_1(f) = 2 (c_0 f(x_0) + c_1 f(x_1))$$

muss für $p \in \Pi_3$ exakt sein, d.h.

$$\int_{-1}^1 p(x) dx = 2 (c_0 p(x_0) + c_1 p(x_1))$$

für $p(x) = x^k, k = 0, 1, 2, 3$.

- Aus

$$\int_{-1}^1 x^k dx = 2 (c_0 x_0^k + c_1 x_1^k), \quad k = 0, 1, 2, 3,$$

erhält man die Gleichungen ...

Beispiel 10.8

erhält man die Gleichungen

$$\begin{aligned} 2 &= 2(c_0 + c_1), & 0 &= 2(c_0 x_0 + c_1 x_1), \\ \frac{2}{3} &= 2(c_0 x_0^2 + c_1 x_1^2), & 0 &= 2(c_0 x_0^3 + c_1 x_1^3). \end{aligned}$$

- Dieses **nichtlineare Gleichungssystem** hat genau zwei Lösungen:

$$\begin{aligned} c_0 = c_1 = \frac{1}{2}, \quad x_0 &= -\frac{1}{3}\sqrt{3}, \quad x_1 = \frac{1}{3}\sqrt{3} \\ c_0 = c_1 = \frac{1}{2}, \quad x_0 &= \frac{1}{3}\sqrt{3}, \quad x_1 = -\frac{1}{3}\sqrt{3} \end{aligned}$$

Daraus ergibt sich

$$\int_{-1}^1 f(x) dx \approx I_1(f) = 2 \left(\frac{1}{2} f\left(-\frac{1}{3}\sqrt{3}\right) + \frac{1}{2} f\left(\frac{1}{3}\sqrt{3}\right) \right)$$

Zusammenfassung

- ▶ Numerische Integration basiert auf **Integration des Interpolationspolynoms** auf einem Teilintervall
- ▶ $I_m(f) = \int_c^d P(f \mid x_0, \dots, x_m)(x) dx, \quad [c, d] : \text{Teilintervall}$
- $\Rightarrow I_m(f) = h \sum_{j=0}^m c_j f(x_j), \quad h = d - c$
 c_j : hängt nur von den Stützstellen ab.
- ▶ Newton-Cotes: **äquidistante** Stützstellen $x_0, \dots, x_m$
- ▶ Gauß-Quadratur: **spezifisch gewählte** Stützstellen

Unterschiede

- ▶ Newton-Cotes: einfacher, Exaktheitsgrad m oder $m + 1$
- ▶ Gauß-Quadratur: Gewichte $c_j > 0$, Exaktheitsgrad $2m + 1$

Verständnisfragen

Es sei $f \in C[a, b]$. Das Integral $I(f) = \int_a^b f(x) dx$ soll numerisch approximiert werden durch eine Quadraturformel

$$I_m(f) = (b - a) \sum_{j=0}^m w_j f(x_j), \quad a \leq x_0 < \dots < x_m \leq b.$$

Weiter sei $I_{m,n}(f)$ die aus $I_m(f)$ konstruierte summierte Quadraturformel auf den Teilintervallen $[t_{j-1}, t_j]$, $j = 1, \dots, n$, mit $t_j = a + jh$, $j = 0, 1, \dots, n$, $h = (b - a)/n$.

Es seien $a = 0$, $b = 1$ und $I_2(f)$ die Simpson-Regel.

Berechnen Sie den Fehler $I(x^3 + 1) - I_2(x^3 + 1)$.

0

W Es seien $I_m^{NC}(f)$ und $I_m^G(f)$ die Newton-Cotes und die Formel der Gauß-Quadratur.

Für $m \geq 1$ gilt, dass der Exaktheitsgrad von $I_m^{NC}(f)$ strikt kleiner ist als der von $I_m^G(f)$.

Verständnisfragen

Es sei $f \in C[a, b]$. Das Integral $I(f) = \int_a^b f(x) dx$ soll numerisch approximiert werden durch eine Quadraturformel

$$I_m(f) = (b - a) \sum_{j=0}^m w_j f(x_j), \quad a \leq x_0 < \dots < x_m \leq b.$$

Weiter sei $I_{m,n}(f)$ die aus $I_m(f)$ konstruierte summierte Quadraturformel auf den Teilintervallen $[t_{j-1}, t_j]$, $j = 1, \dots, n$, mit $t_j = a + jh$, $j = 0, 1, \dots, n$, $h = (b - a)/n$.

w Es sei $P(f | x_0, \dots, x_m)$ das Lagrange-Interpolationspolynom. Bei Gauß-Quadratur gilt

$$I_m(f) = \int_a^b P(f | x_0, \dots, x_m)(x) dx.$$

f Es seien $f \in C^4[a, b]$ und $I_2(f)$ die Simpson-Regel. Es gilt $|I_{2,n}(f) - I(f)| \leq ch^5$, wobei c nicht von n abhängt.

f Bei der Gauß-Quadratur hängen die Stützstellen x_j von der Funktion f ab.