

Numerische Verfahren

Jens-Peter M. Zemke
zemke@tu-harburg.de

Institut für Numerische Simulation
Technische Universität Hamburg-Harburg

15.04.2008



Numerische Integration

Konstruktion von Quadraturformeln

Fehler von Quadraturformeln

Quadraturformeln von Gauß

Adaptive Quadratur

Numerische Differentiation

Numerische Integration

In vielen Fällen ist es nicht möglich, ein gegebenes Integral

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

in geschlossener Form auszuwerten;

Numerische Integration

In vielen Fällen ist es nicht möglich, ein gegebenes Integral

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

in geschlossener Form auszuwerten; z.B. ist für das in der Statistik häufig auftretende Integral

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-t^2/2} dt$$

keine elementare Stammfunktion angebar.

Numerische Integration

In vielen Fällen ist es nicht möglich, ein gegebenes Integral

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

in geschlossener Form auszuwerten; z.B. ist für das in der Statistik häufig auftretende Integral

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-t^2/2} dt$$

keine elementare Stammfunktion angebar.

In diesem Fall ist man auf numerische Verfahren zur Integration, sog. **Quadraturverfahren**, angewiesen. Wir wollen in diesem Abschnitt einige wichtige Verfahren zur näherungsweise Berechnung bestimmter Integrale besprechen.

Konstruktion von Quadraturformeln

Wir betrachten das bestimmte Integral

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

mit einer gegebenen integrierbaren Funktion $f : [a, b] \rightarrow \mathbb{R}$.

Konstruktion von Quadraturformeln

Wir betrachten das bestimmte Integral

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

mit einer gegebenen integrierbaren Funktion $f : [a, b] \rightarrow \mathbb{R}$.

Mit der **Variablentransformation** $x = a + t(b - a)$ erhält man

$$\int_a^b f(x) dx = (b - a) \int_0^1 f(a + t(b - a)) dt,$$

so dass man sich ohne Beschränkung der Allgemeinheit auf das Intervall $[a, b] = [0, 1]$ beschränken kann.

Konstruktion von Quadraturformeln

Eine naheliegende Idee zur Konstruktion von Quadraturformeln ist es, zu $n + 1$ verschiedene Knoten $x_0, x_1, \dots, x_n \in [0, 1]$ das Interpolationspolynom p von f zu bestimmen und als Näherung für das Integral von f das Integral über das Interpolationspolynom p zu wählen,

$$\int_0^1 f(x) dx \approx \int_0^1 p(x) dx =: Q(f).$$

Konstruktion von Quadraturformeln

Eine naheliegende Idee zur Konstruktion von Quadraturformeln ist es, zu $n + 1$ verschiedene Knoten $x_0, x_1, \dots, x_n \in [0, 1]$ das Interpolationspolynom p von f zu bestimmen und als Näherung für das Integral von f das Integral über das Interpolationspolynom p zu wählen,

$$\int_0^1 f(x) dx \approx \int_0^1 p(x) dx =: Q(f).$$

Mit

$$\ell_j(x) := \prod_{\substack{i=0 \\ i \neq j}}^n (x - x_i) \bigg/ \prod_{\substack{i=0 \\ i \neq j}}^n (x_j - x_i), \quad j = 0, \dots, n,$$

gilt nach der [Lagrangeschen Interpolationsformel](#)

$$p(x) = \sum_{j=0}^n f(x_j) \ell_j(x),$$

Konstruktion von Quadraturformeln

und daher erhält man

$$Q(f) = \int_0^1 \sum_{j=0}^n f(x_j) \ell_j(x) dx = \sum_{j=0}^n f(x_j) \int_0^1 \ell_j(x) dx =: \sum_{j=0}^n \alpha_j f(x_j).$$

Konstruktion von Quadraturformeln

und daher erhält man

$$Q(f) = \int_0^1 \sum_{j=0}^n f(x_j) \ell_j(x) dx = \sum_{j=0}^n f(x_j) \int_0^1 \ell_j(x) dx =: \sum_{j=0}^n \alpha_j f(x_j).$$

Dabei hängen die **Gewichte**

$$\alpha_j := \int_0^1 \ell_j(x) dx$$

nur von den gewählten Knoten x_0, \dots, x_n ab und sind unabhängig vom aktuellen Integranden f . Sie können also ein für alle mal berechnet werden und in Tafeln oder Dateien bereitgestellt werden.

Konstruktion von Quadraturformeln

Beispiel 3.1: Für $n = 0$ und $x_0 = 0.5$ gilt $\ell_0(x) \equiv 1$ und

$$\alpha_0 = \int_0^1 \ell_0(x) dx = 1.$$

Konstruktion von Quadraturformeln

Beispiel 3.1: Für $n = 0$ und $x_0 = 0.5$ gilt $\ell_0(x) \equiv 1$ und

$$\alpha_0 = \int_0^1 \ell_0(x) dx = 1.$$

Die entstehende Quadraturformel

$$\int_0^1 f(x) dx \approx f(0.5) =: R(f),$$

Konstruktion von Quadraturformeln

Beispiel 3.1: Für $n = 0$ und $x_0 = 0.5$ gilt $\ell_0(x) \equiv 1$ und

$$\alpha_0 = \int_0^1 \ell_0(x) dx = 1.$$

Die entstehende Quadraturformel

$$\int_0^1 f(x) dx \approx f(0.5) =: R(f),$$

bzw. die Quadraturformel für das allgemeine Intervall,

$$\int_a^b f(x) dx \approx (b - a) \cdot f\left(\frac{a + b}{2}\right) =: R(f),$$

Konstruktion von Quadraturformeln

Beispiel 3.1: Für $n = 0$ und $x_0 = 0.5$ gilt $\ell_0(x) \equiv 1$ und

$$\alpha_0 = \int_0^1 \ell_0(x) dx = 1.$$

Die entstehende Quadraturformel

$$\int_0^1 f(x) dx \approx f(0.5) =: R(f),$$

bzw. die Quadraturformel für das allgemeine Intervall,

$$\int_a^b f(x) dx \approx (b - a) \cdot f\left(\frac{a + b}{2}\right) =: R(f),$$

heißt **Rechteckregel** oder auch **Mittelpunktregel**. □

Konstruktion von Quadraturformeln

Beispiel 3.2: Für $n = 1$, $x_0 = 0$ und $x_1 = 1$ ist

$$\ell_0(x) = 1 - x \quad \text{und} \quad \ell_1(x) = x.$$

Konstruktion von Quadraturformeln

Beispiel 3.2: Für $n = 1$, $x_0 = 0$ und $x_1 = 1$ ist

$$\ell_0(x) = 1 - x \quad \text{und} \quad \ell_1(x) = x.$$

Durch Integration erhält man $\alpha_0 = \alpha_1 = 0.5$ und damit die **Trapezregel**

$$\int_0^1 f(x) dx \approx \frac{f(0) + f(1)}{2} =: T(f),$$

Konstruktion von Quadraturformeln

Beispiel 3.2: Für $n = 1$, $x_0 = 0$ und $x_1 = 1$ ist

$$\ell_0(x) = 1 - x \quad \text{und} \quad \ell_1(x) = x.$$

Durch Integration erhält man $\alpha_0 = \alpha_1 = 0.5$ und damit die **Trapezregel**

$$\int_0^1 f(x) dx \approx \frac{f(0) + f(1)}{2} =: T(f),$$

bzw. die **Trapezregel** für das allgemeine Intervall

$$\int_a^b f(x) dx \approx (b - a) \cdot \frac{f(a) + f(b)}{2} =: T(f). \quad \square$$

Konstruktion von Quadraturformeln

Beispiel 3.3: Für $n = 2$,

$$x_0 = 0, \quad x_1 = 0.5 \quad \text{und} \quad x_2 = 1$$

erhält man analog zu der Herangehensweise in den beiden vorhergehenden Beispielen die **Simpson-Regel**

$$\int_0^1 f(x) dx \approx \frac{1}{6} \cdot (f(0) + 4f(0.5) + f(1)) =: S(f)$$

Konstruktion von Quadraturformeln

Beispiel 3.3: Für $n = 2$,

$$x_0 = 0, \quad x_1 = 0.5 \quad \text{und} \quad x_2 = 1$$

erhält man analog zu der Herangehensweise in den beiden vorhergehenden Beispielen die **Simpson-Regel**

$$\int_0^1 f(x) dx \approx \frac{1}{6} \cdot (f(0) + 4f(0.5) + f(1)) =: S(f)$$

bzw.

$$\int_a^b f(x) dx \approx \frac{b-a}{6} \cdot \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) =: S(f). \quad \square$$

Konstruktion von Quadraturformeln

Beispiel 3.3: Für $n = 2$,

$$x_0 = 0, \quad x_1 = 0.5 \quad \text{und} \quad x_2 = 1$$

erhält man analog zu der Herangehensweise in den beiden vorhergehenden Beispielen die **Simpson-Regel**

$$\int_0^1 f(x) dx \approx \frac{1}{6} \cdot (f(0) + 4f(0.5) + f(1)) =: S(f)$$

bzw.

$$\int_a^b f(x) dx \approx \frac{b-a}{6} \cdot \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) =: S(f). \quad \square$$

Die Simpson-Regel ist in der deutschsprachigen Literatur auch als **Keplersche Fassregel** bekannt.

Konstruktion von Quadraturformeln

Beispiel 3.4: Für $n = 4$ und äquidistante Knoten

$$x_j = a + j(b - a)/4, \quad j = 0, 1, \dots, 4,$$

erhält man die **Milne-Regel**

$$\int_a^b f(x) dx \approx \frac{b - a}{90} \cdot (7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)) =: M(f). \quad \square$$

Konstruktion von Quadraturformeln

Es ist naheliegend (und dies ist auch die historisch älteste Wahl), die Knoten äquidistant im Intervall $[a, b]$ zu wählen. Berücksichtigt man dabei die Intervallenden, wählt man also

$$x_j = a + j \cdot \frac{b - a}{n}, \quad j = 0, \dots, n,$$

Konstruktion von Quadraturformeln

Es ist naheliegend (und dies ist auch die historisch älteste Wahl), die Knoten äquidistant im Intervall $[a, b]$ zu wählen. Berücksichtigt man dabei die Intervallenden, wählt man also

$$x_j = a + j \cdot \frac{b - a}{n}, \quad j = 0, \dots, n,$$

so erhält man die **abgeschlossenen Newton-Cotes-Formeln**

$$\int_0^1 f(x) dx \approx \sum_{j=0}^n \alpha_j^{(n)} f\left(\frac{j}{n}\right)$$

Konstruktion von Quadraturformeln

Es ist naheliegend (und dies ist auch die historisch älteste Wahl), die Knoten äquidistant im Intervall $[a, b]$ zu wählen. Berücksichtigt man dabei die Intervallenden, wählt man also

$$x_j = a + j \cdot \frac{b-a}{n}, \quad j = 0, \dots, n,$$

so erhält man die **abgeschlossenen Newton-Cotes-Formeln**

$$\int_0^1 f(x) dx \approx \sum_{j=0}^n \alpha_j^{(n)} f\left(\frac{j}{n}\right)$$

bzw.

$$\int_a^b f(x) dx \approx (b-a) \sum_{j=0}^n \alpha_j^{(n)} f\left(a + j \cdot \frac{b-a}{n}\right) =: \text{ANC}_n(f).$$

Konstruktion von Quadraturformeln

abgeschlossene Newton-Cotes-Formeln

n	$\alpha_j^{(n)}$					Name
1	1/2	1/2				Trapezregel
2	1/6	4/6	1/6			Simpson-Regel
3	1/8	3/8	3/8	1/8		3/8-Regel
4	7/90	32/90	12/90	32/90	7/90	Milne-Regel

Konstruktion von Quadraturformeln

Berücksichtigt man die Intervallenden nicht, wählt man also

$$x_j = a + (j + 1) \cdot \frac{b - a}{n + 2}, \quad j = 0, \dots, n,$$

so erhält man die **offenen Newton-Cotes-Formeln**

Konstruktion von Quadraturformeln

Berücksichtigt man die Intervallenden nicht, wählt man also

$$x_j = a + (j + 1) \cdot \frac{b - a}{n + 2}, \quad j = 0, \dots, n,$$

so erhält man die **offenen Newton-Cotes-Formeln**

$$\int_0^1 f(x) dx \approx \sum_{j=0}^n \beta_j^{(n)} f\left(\frac{j+1}{n+2}\right)$$

Konstruktion von Quadraturformeln

Berücksichtigt man die Intervallenden nicht, wählt man also

$$x_j = a + (j + 1) \cdot \frac{b - a}{n + 2}, \quad j = 0, \dots, n,$$

so erhält man die **offenen Newton-Cotes-Formeln**

$$\int_0^1 f(x) dx \approx \sum_{j=0}^n \beta_j^{(n)} f\left(\frac{j+1}{n+2}\right)$$

bzw.

$$\int_a^b f(x) dx \approx (b - a) \sum_{j=0}^n \beta_j^{(n)} f\left(a + (j + 1) \frac{b - a}{n + 2}\right) =: \text{ONC}_n(f).$$

Konstruktion von Quadraturformeln

offene Newton-Cotes-Formeln

n	$\beta_j^{(n)}$				
0	1				
1	1/2	1/2			
2	2/3	-1/3	2/3		
3	11/24	1/24	1/24	11/24	
4	11/20	-14/20	26/20	-14/20	11/20

Konstruktion von Quadraturformeln

Bemerkung 3.5: Die **Mittelpunktregel** ist die **offene** Newton-Cotes-Formel für den Fall $n = 0$.

Konstruktion von Quadraturformeln

Bemerkung 3.5: Die **Mittelpunktregel** ist die **offene** Newton-Cotes-Formel für den Fall $n = 0$.

Die **Trapezregel**, die **Simpson-** und die **Milne-Regel** sind die **abgeschlossenen** Newton-Cotes-Formeln für die Fälle $n = 1$, $n = 2$ und $n = 4$. \square

Konstruktion von Quadraturformeln

Bemerkung 3.6: **Offene** Formeln (d.h. solche Formeln, bei denen die Intervallenden keine Knoten sind,) verwendet man, wenn die Auswertung des Integranden an den Randpunkten schwierig ist (z.B. der Grenzwert eines Quotienten, für den Zähler und Nenner gegen 0 gehen, oder wenn gar eine Singularität von f am Rand vorliegt).

Konstruktion von Quadraturformeln

Bemerkung 3.6: **Offene** Formeln (d.h. solche Formeln, bei denen die Intervallenden keine Knoten sind,) verwendet man, wenn die Auswertung des Integranden an den Randpunkten schwierig ist (z.B. der Grenzwert eines Quotienten, für den Zähler und Nenner gegen 0 gehen, oder wenn gar eine Singularität von f am Rand vorliegt).

Offene **Newton-Cotes-Formeln** werden heute (mit Ausnahme der Mittelpunkregel) **kaum noch verwendet**, da sie wesentlich schlechtere Fehlerordnungen haben als die Gauß Formeln, die ebenfalls offen sind. □

Konstruktion von Quadraturformeln

Die Gewichte der Newton-Cotes-Formeln **wachsen** rasch an.

Konstruktion von Quadraturformeln

Die Gewichte der Newton-Cotes-Formeln **wachsen** rasch an.

Für die abgeschlossenen Formeln treten für $n \geq 8$ **wechselnde Vorzeichen** auf, für die offenen Formeln sogar schon für $n \geq 2$.

Konstruktion von Quadraturformeln

Die Gewichte der Newton-Cotes-Formeln **wachsen** rasch an.

Für die abgeschlossenen Formeln treten für $n \geq 8$ **wechselnde Vorzeichen** auf, für die offenen Formeln sogar schon für $n \geq 2$.

Diese Formeln sind also **anfällig für Rundungsfehler**.

Konstruktion von Quadraturformeln

Die Gewichte der Newton-Cotes-Formeln **wachsen** rasch an.

Für die abgeschlossenen Formeln treten für $n \geq 8$ **wechselnde Vorzeichen** auf, für die offenen Formeln sogar schon für $n \geq 2$.

Diese Formeln sind also **anfällig für Rundungsfehler**.

Man benutzt die Newton-Cotes-Formeln daher nur für kleine n auf Teilintervallen von $[a, b]$ und summiert auf.

Konstruktion von Quadraturformeln

Die Gewichte der Newton-Cotes-Formeln **wachsen** rasch an.

Für die abgeschlossenen Formeln treten für $n \geq 8$ **wechselnde Vorzeichen** auf, für die offenen Formeln sogar schon für $n \geq 2$.

Diese Formeln sind also **anfällig für Rundungsfehler**.

Man benutzt die Newton-Cotes-Formeln daher nur für kleine n auf Teilintervallen von $[a, b]$ und summiert auf.

Man erhält somit die **summierten** oder **zusammengesetzten Newton-Cotes-Formeln**.

Konstruktion von Quadraturformeln

Beispiele hierfür sind mit

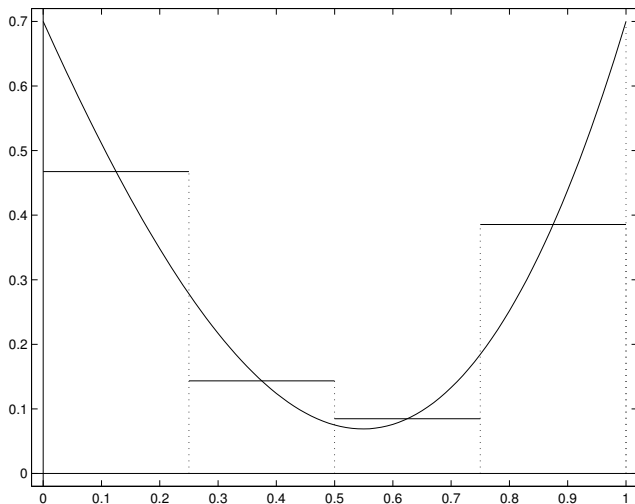
$$h := \frac{b-a}{m} \quad \text{und} \quad x_j := a + j \cdot h, \quad j = 0, \dots, m,$$

die **summierte Rechteckregel** (summierte Mittelpunkregel)

$$\int_a^b f(x) dx \approx h \sum_{j=1}^m f(x_j - h/2) =: R_h(f), \quad (3.1)$$

Konstruktion von Quadraturformeln

Summierte Mittelpunkregel



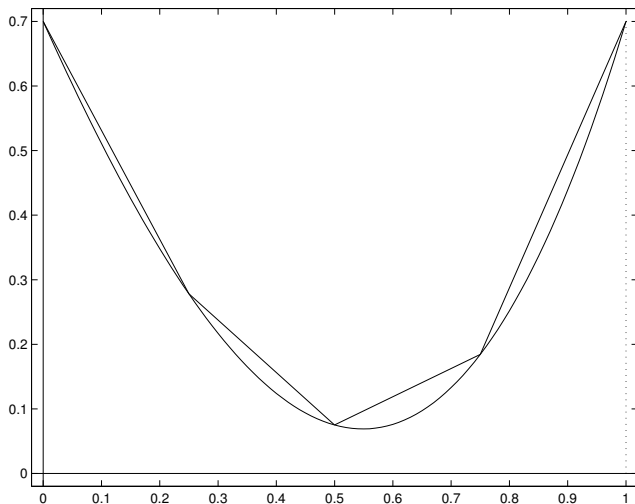
Konstruktion von Quadraturformeln

die **summierte Trapezregel**

$$\int_a^b f(x) dx \approx h \left(\frac{1}{2} f(a) + \sum_{i=1}^{m-1} f(x_i) + \frac{1}{2} f(b) \right) =: T_h(f), \quad (3.2)$$

Konstruktion von Quadraturformeln

Summierte Trapezregel



Konstruktion von Quadraturformeln

und für $m = 2k$ die **summierte Simpson-Regel**

$$\begin{aligned} \int_a^b f(x) dx &\approx \frac{2h}{6} (f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 4f(x_{2k-1}) + f(x_{2k})) \\ &= \frac{2h}{6} \left(f(a) + 4 \sum_{i=1}^k f(x_{2i-1}) + 2 \sum_{i=1}^{k-1} f(x_{2i}) + f(b) \right) =: S_h(f). \end{aligned}$$

Konstruktion von Quadraturformeln

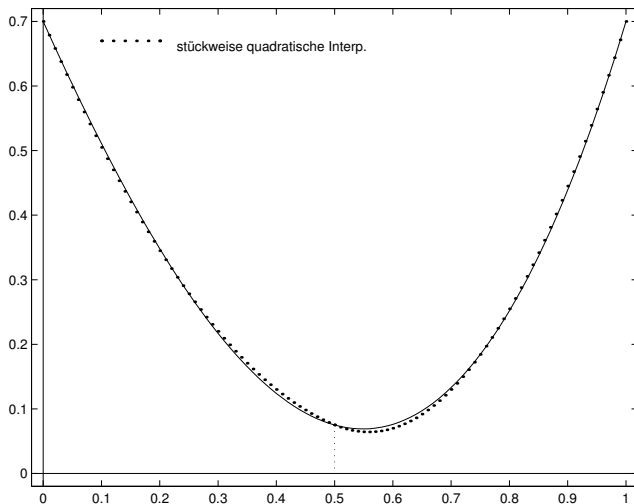
und für $m = 2k$ die **summierte Simpson-Regel**

$$\begin{aligned} \int_a^b f(x) dx &\approx \frac{2h}{6} (f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 4f(x_{2k-1}) + f(x_{2k})) \\ &= \frac{2h}{6} \left(f(a) + 4 \sum_{i=1}^k f(x_{2i-1}) + 2 \sum_{i=1}^{k-1} f(x_{2i}) + f(b) \right) =: S_h(f). \end{aligned}$$

Beachten Sie, dass die Simpson-Regel jeweils auf zwei benachbarte Teilintervalle der Gesamtlänge $2h$ angewandt wird.

Konstruktion von Quadraturformeln

Summierte Simpson-Regel



Fehler von Quadraturformeln

Wir untersuchen nun den Fehler von Quadraturformeln.

Fehler von Quadraturformeln

Wir untersuchen nun den Fehler von Quadraturformeln.

Wir betrachten das Integral

$$I := \int_0^1 f(x) dx$$

Fehler von Quadraturformeln

Wir untersuchen nun den **Fehler von Quadraturformeln**.

Wir betrachten das Integral

$$I := \int_0^1 f(x) dx$$

und eine zugehörige Quadraturformel mit den Knoten $x_0, \dots, x_n \in [0, 1]$ und den Gewichten $w_0, \dots, w_n \in \mathbb{R}$

$$Q(f) := \sum_{i=0}^n w_i f(x_i)$$

für das Referenzintervall $[0, 1]$.

Fehler von Quadraturformeln

Definition 3.7: Die Quadraturformel $Q(f)$ hat die **Fehlerordnung** m , wenn für den **Fehler**

$$E(f) := \int_0^1 f(x) dx - \sum_{i=0}^n w_i f(x_i)$$

gilt:

- ▶ $E(p) = 0$ für alle Polynome $p \in \Pi_{m-1}$,
- ▶ $E(p) \neq 0$ für ein $p \in \Pi_m$.

Fehler von Quadraturformeln

Definition 3.7: Die Quadraturformel $Q(f)$ hat die **Fehlerordnung** m , wenn für den **Fehler**

$$E(f) := \int_0^1 f(x) dx - \sum_{i=0}^n w_i f(x_i)$$

gilt:

- ▶ $E(p) = 0$ für alle Polynome $p \in \Pi_{m-1}$,
- ▶ $E(p) \neq 0$ für ein $p \in \Pi_m$.

Bemerkung 3.8: Wegen der **Linearität** des Fehlers ist klar, dass Q genau dann die Fehlerordnung m hat, wenn $E(x^j) = 0$ für $j = 0, \dots, m-1$ und $E(x^m) \neq 0$ gilt. □

Fehler von Quadraturformeln

Bemerkung 3.9: Die Konstruktion liefert, dass die Newton-Cotes-Formeln **wenigstens** die Fehlerordnung $n + 1$ haben. Für die Trapezregel ist dies die genaue Fehlerordnung, denn

$$T(x^2) = \frac{1}{2} \neq \int_0^1 x^2 dx = \frac{1}{3}.$$

Fehler von Quadraturformeln

Bemerkung 3.9: Die Konstruktion liefert, dass die Newton-Cotes-Formeln **wenigstens** die Fehlerordnung $n + 1$ haben. Für die Trapezregel ist dies die genaue Fehlerordnung, denn

$$T(x^2) = \frac{1}{2} \neq \int_0^1 x^2 dx = \frac{1}{3}.$$

Für die Simpson-Regel gilt

$$S(x^3) = \frac{1}{6} \left(0 + 4 \cdot \frac{1}{8} + 1 \right) = \frac{1}{4} = \int_0^1 x^3 dx, \quad S(x^4) = \frac{5}{24} \neq \int_0^1 x^4 dx = \frac{1}{5},$$

so dass die Simpson-Regel sogar die Ordnung 4 hat. □

Fehler von Quadraturformeln

Satz 3.10

Es sei Q eine Quadraturformel der Fehlerordnung $m \geq 1$. Dann hat für $f \in C^m[0, 1]$ der Fehler von Q die Darstellung

$$E(f) = \int_0^1 f(x) dx - Q(f) = \int_0^1 K(x) f^{(m)}(x) dx \quad (3.3)$$

mit

$$K(x) = \frac{1}{(m-1)!} \left(\frac{1}{m} (1-x)^m - \sum_{i=0}^n w_i (x_i - x)_+^{m-1} \right).$$

Fehler von Quadraturformeln

Satz 3.10

Es sei Q eine Quadraturformel der Fehlerordnung $m \geq 1$. Dann hat für $f \in C^m[0, 1]$ der Fehler von Q die Darstellung

$$E(f) = \int_0^1 f(x) dx - Q(f) = \int_0^1 K(x) f^{(m)}(x) dx \quad (3.3)$$

mit

$$K(x) = \frac{1}{(m-1)!} \left(\frac{1}{m} (1-x)^m - \sum_{i=0}^n w_i (x_i - x)_+^{m-1} \right).$$

Beweis.

Siehe dazu das Skript „Grundlagen der Numerischen Mathematik“ von Heinrich Voß, Abschnitt 3.2. □

Fehler von Quadraturformeln

Satz 3.10

Es sei Q eine Quadraturformel der Fehlerordnung $m \geq 1$. Dann hat für $f \in C^m[0, 1]$ der Fehler von Q die Darstellung

$$E(f) = \int_0^1 f(x) dx - Q(f) = \int_0^1 K(x) f^{(m)}(x) dx \quad (3.3)$$

mit

$$K(x) = \frac{1}{(m-1)!} \left(\frac{1}{m} (1-x)^m - \sum_{i=0}^n w_i (x_i - x)_+^{m-1} \right).$$

Beweis.

Siehe dazu das Skript „Grundlagen der Numerischen Mathematik“ von Heinrich Voß, Abschnitt 3.2. □

Definition 3.11: Die Funktion K in der Fehlerdarstellung (3.3) heißt der **Peano-Kern** der Quadraturformel Q .

Fehler von Quadraturformeln

Beispiel 3.12: Für die Trapezregel gilt $x_0 = 0$, $x_1 = 1$, $w_0 = w_1 = 0.5$, $m = 2$, und daher ist der **Peano-Kern der Trapezregel** gegeben als

$$K_T(x) = \frac{1}{2}(1-x)^2 - \frac{1}{2}(1-x) = -\frac{1}{2}x(1-x). \quad \square$$

Fehler von Quadraturformeln

Beispiel 3.12: Für die Trapezregel gilt $x_0 = 0$, $x_1 = 1$, $w_0 = w_1 = 0.5$, $m = 2$, und daher ist der **Peano-Kern der Trapezregel** gegeben als

$$K_T(x) = \frac{1}{2}(1-x)^2 - \frac{1}{2}(1-x) = -\frac{1}{2}x(1-x). \quad \square$$

Beispiel 3.13: Für die Simpson-Regel gilt $x_0 = 0$, $x_1 = 0.5$, $x_2 = 1$, $w_0 = w_2 = 1/6$, $w_1 = 2/3$ und $m = 4$, und daher ist der **Peano-Kern der Simpson-Regel** gegeben als

$$K_S(x) = \frac{1}{3!} \left(\frac{1}{4}(1-x)^4 - \frac{1}{6}(1-x)^3 - \frac{2}{3} \left(\frac{1}{2} - x \right)_+^3 \right). \quad \square$$

Fehler von Quadraturformeln

Aus Satz 3.10 erhält man die folgende **Abschätzung für den Fehler**

$$|E(f)| \leq \|f^{(m)}\|_{\infty} \int_0^1 |K(x)| dx =: \tilde{c}_m \|f^{(m)}\|_{\infty}. \quad (3.4)$$

Fehler von Quadraturformeln

Aus Satz 3.10 erhält man die folgende **Abschätzung für den Fehler**

$$|E(f)| \leq \|f^{(m)}\|_{\infty} \int_0^1 |K(x)| dx =: \tilde{c}_m \|f^{(m)}\|_{\infty}. \quad (3.4)$$

In vielen Fällen wechselt der Peano-Kern $K(x)$ das Vorzeichen auf dem Intervall $[0, 1]$ nicht. Dann folgt aus (3.3) mit dem Mittelwertsatz der Integralrechnung mit einem $\xi \in (0, 1)$

$$E(f) = f^{(m)}(\xi) \int_0^1 K(x) dx =: c_m f^{(m)}(\xi) \quad (3.5)$$

für eine Quadraturformel der Ordnung m .

Fehler von Quadraturformeln

Aus Satz 3.10 erhält man die folgende **Abschätzung für den Fehler**

$$|E(f)| \leq \|f^{(m)}\|_{\infty} \int_0^1 |K(x)| dx =: \tilde{c}_m \|f^{(m)}\|_{\infty}. \quad (3.4)$$

In vielen Fällen wechselt der Peano-Kern $K(x)$ das Vorzeichen auf dem Intervall $[0, 1]$ nicht. Dann folgt aus (3.3) mit dem Mittelwertsatz der Integralrechnung mit einem $\xi \in (0, 1)$

$$E(f) = f^{(m)}(\xi) \int_0^1 K(x) dx =: c_m f^{(m)}(\xi) \quad (3.5)$$

für eine Quadraturformel der Ordnung m .

Definition 3.14: Die Konstante c_m heißt **Fehlerkonstante** des Verfahrens Q .

Fehler von Quadraturformeln

Für die Trapezregel gilt

$$K_T(x) = -\frac{1}{2}x(1-x) \quad \text{für alle } x \in [0, 1].$$

Fehler von Quadraturformeln

Für die Trapezregel gilt

$$K_T(x) = -\frac{1}{2}x(1-x) \quad \text{für alle } x \in [0, 1].$$

Da T die Ordnung 2 hat, gilt

$$E_T(f) = -\frac{1}{2}f''(\xi) \int_0^1 x(1-x) dx = -\frac{1}{12}f''(\xi),$$

und die Fehlerkonstante der Trapezregel ist somit $c_2 = -1/12$.

Fehler von Quadraturformeln

Eine elementare Rechnung zeigt, dass auch für die Simpson-Regel

$$K_S(x) = \frac{1}{3!} \left(\frac{1}{4} (1-x)^4 - \frac{1}{6} (1-x)^3 - \frac{2}{3} \left(\frac{1}{2} - x \right)_+^3 \right) \leq 0$$

für alle $x \in [0, 1]$ gilt.

Fehler von Quadraturformeln

Eine elementare Rechnung zeigt, dass auch für die Simpson-Regel

$$K_S(x) = \frac{1}{3!} \left(\frac{1}{4} (1-x)^4 - \frac{1}{6} (1-x)^3 - \frac{2}{3} \left(\frac{1}{2} - x \right)_+^3 \right) \leq 0$$

für alle $x \in [0, 1]$ gilt.

Durch Integration von K von 0 bis 1 erhält man für den Fehler wegen $m = 4$

$$E_S(f) = -\frac{f^{(4)}(\xi)}{2880}$$

für ein $\xi \in (0, 1)$, d.h. die **Fehlerkonstante der Simpson-Regel** ist gegeben durch $c_4 = -1/2880$.

Fehler von Quadraturformeln

Bemerkung 3.15: Man kann die Fehlerkonstante der Simpson-Regel und auch anderer Formeln ohne Integration (sogar ohne Kenntnis) des Peano-Kerns bestimmen.

Fehler von Quadraturformeln

Bemerkung 3.15: Man kann die Fehlerkonstante der Simpson-Regel und auch anderer Formeln ohne Integration (sogar ohne Kenntnis) des Peano-Kerns bestimmen.

Ist bekannt, dass der Peano-Kern einer Formel der Ordnung m sein Vorzeichen in $[0, 1]$ nicht wechselt, der Fehler also eine Darstellung (3.5) hat, so hat man nur $E(x^m)$ zu berechnen.

Fehler von Quadraturformeln

Bemerkung 3.15: Man kann die Fehlerkonstante der Simpson-Regel und auch anderer Formeln ohne Integration (sogar ohne Kenntnis) des Peano-Kerns bestimmen.

Ist bekannt, dass der Peano-Kern einer Formel der Ordnung m sein Vorzeichen in $[0, 1]$ nicht wechselt, der Fehler also eine Darstellung (3.5) hat, so hat man nur $E(x^m)$ zu berechnen.

Es ist nämlich

$$\frac{d^m}{dx^m}(x^m) = m!$$

und daher

$$E(x^m) = m! \cdot c_m,$$

und hieraus erhält man c_m . □

Fehler von Quadraturformeln

Beispiel 3.16: Im Fall der Simpson-Regel ist

$$E(x^4) = \int_0^1 x^4 dx - \frac{1}{6} \left((0)^4 + 4 \cdot (1/2)^4 + 1^4 \right) = \frac{1}{5} - \frac{1}{6} \cdot \frac{5}{4} = -\frac{1}{120},$$

und daher gilt

$$c_4 = \frac{1}{4!} \cdot \left(-\frac{1}{120} \right) = -\frac{1}{2880}.$$

□

Fehler von Quadraturformeln

Beispiel 3.16: Im Fall der Simpson-Regel ist

$$E(x^4) = \int_0^1 x^4 dx - \frac{1}{6} \left((0)^4 + 4 \cdot (1/2)^4 + 1^4 \right) = \frac{1}{5} - \frac{1}{6} \cdot \frac{5}{4} = -\frac{1}{120},$$

und daher gilt

$$c_4 = \frac{1}{4!} \cdot \left(-\frac{1}{120} \right) = -\frac{1}{2880}.$$

□

Da uns hauptsächlich die **summierten** Quadraturformeln interessieren, gehen wir als nächstes auf deren Fehleranalyse ein.

Fehler von Quadraturformeln

Beispiel 3.16: Im Fall der Simpson-Regel ist

$$E(x^4) = \int_0^1 x^4 dx - \frac{1}{6} \left((0)^4 + 4 \cdot (1/2)^4 + 1^4 \right) = \frac{1}{5} - \frac{1}{6} \cdot \frac{5}{4} = -\frac{1}{120},$$

und daher gilt

$$c_4 = \frac{1}{4!} \cdot \left(-\frac{1}{120} \right) = -\frac{1}{2880}.$$

□

Da uns hauptsächlich die **summierten** Quadraturformeln interessieren, gehen wir als nächstes auf deren Fehleranalyse ein.

Im Zusammenhang mit den summierten Formeln werden auch die Begriffe **Fehlerordnung** und **Fehlerkonstante** endlich ihren tieferen Sinn offenbaren.

Fehler von Quadraturformeln

Wir betrachten nun das Integral von f über ein Intervall der Länge h :

$$\int_{\alpha}^{\alpha+h} f(x) dx.$$

Fehler von Quadraturformeln

Wir betrachten nun das Integral von f über ein Intervall der Länge h :

$$\int_{\alpha}^{\alpha+h} f(x) dx.$$

Mit der Variablentransformation $x =: \alpha + ht$ geht dieses über in

$$\int_{\alpha}^{\alpha+h} f(x) dx = h \int_0^1 g(t) dt, \quad g(t) := f(\alpha + ht) = f(x),$$

Fehler von Quadraturformeln

Wir betrachten nun das Integral von f über ein Intervall der Länge h :

$$\int_{\alpha}^{\alpha+h} f(x) dx.$$

Mit der Variablentransformation $x =: \alpha + ht$ geht dieses über in

$$\int_{\alpha}^{\alpha+h} f(x) dx = h \int_0^1 g(t) dt, \quad g(t) := f(\alpha + ht) = f(x),$$

das wir mit der Quadraturformel

$$Q(g) = \sum_{i=0}^n w_i g(x_i)$$

Fehler von Quadraturformeln

Wir betrachten nun das Integral von f über ein Intervall der Länge h :

$$\int_{\alpha}^{\alpha+h} f(x) dx.$$

Mit der Variablentransformation $x =: \alpha + ht$ geht dieses über in

$$\int_{\alpha}^{\alpha+h} f(x) dx = h \int_0^1 g(t) dt, \quad g(t) := f(\alpha + ht) = f(x),$$

das wir mit der Quadraturformel

$$Q(g) = \sum_{i=0}^n w_i g(x_i)$$

behandeln, d.h.

$$Q_{[\alpha, \alpha+h]}(f) := h \sum_{i=0}^n w_i f(\alpha + hx_i).$$

Fehler von Quadraturformeln

Für den Fehler gilt

$$\begin{aligned} E_{[\alpha, \alpha+h]}(f) &:= \int_{\alpha}^{\alpha+h} f(x) dx - Q_{[\alpha, \alpha+h]}(f) \\ &= h \left(\int_0^1 g(t) dt - Q(g) \right) = h E(g). \end{aligned}$$

Fehler von Quadraturformeln

Für den Fehler gilt

$$\begin{aligned} E_{[\alpha, \alpha+h]}(f) &:= \int_{\alpha}^{\alpha+h} f(x) dx - Q_{[\alpha, \alpha+h]}(f) \\ &= h \left(\int_0^1 g(t) dt - Q(g) \right) = h E(g). \end{aligned}$$

Besitzt der Fehler $E(g)$ eine Darstellung (3.4), so folgt wegen

$$\begin{aligned} \frac{d^m g}{dt^m} &= \frac{d^m f}{dx^m} \cdot \left(\frac{dx}{dt} \right)^m = h^m \frac{d^m f}{dx^m} \\ |E_{[\alpha, \alpha+h]}(f)| &\leq h^{m+1} \cdot \tilde{c}_m \cdot \max_{\alpha \leq x \leq \alpha+h} |f^{(m)}(x)|. \end{aligned} \quad (3.6)$$

Fehler von Quadraturformeln

Für den Fehler gilt

$$\begin{aligned} E_{[\alpha, \alpha+h]}(f) &:= \int_{\alpha}^{\alpha+h} f(x) dx - Q_{[\alpha, \alpha+h]}(f) \\ &= h \left(\int_0^1 g(t) dt - Q(g) \right) = h E(g). \end{aligned}$$

Besitzt der Fehler $E(g)$ eine Darstellung (3.4), so folgt wegen

$$\frac{d^m g}{dt^m} = \frac{d^m f}{dx^m} \cdot \left(\frac{dx}{dt} \right)^m = h^m \frac{d^m f}{dx^m}$$

$$|E_{[\alpha, \alpha+h]}(f)| \leq h^{m+1} \cdot \tilde{c}_m \cdot \max_{\alpha \leq x \leq \alpha+h} |f^{(m)}(x)|. \quad (3.6)$$

Gilt eine Darstellung (3.5), so erhält man genauso

$$E_{[\alpha, \alpha+h]}(f) = h^{m+1} \cdot c_m \cdot f^{(m)}(\eta), \quad \eta \in [\alpha, \alpha + h]. \quad (3.7)$$

Fehler von Quadraturformeln

Wir haben bereits erwähnt, dass die Genauigkeit einer Näherung für ein Integral nicht durch die Erhöhung der Ordnung der benutzten Quadraturformel verbessert wird, sondern dass das Intervall $[a, b]$ in n Teilintervalle der Länge h zerlegt wird, und dass in jedem Teilintervall eine Quadraturformel kleiner Ordnung verwendet wird.

Fehler von Quadraturformeln

Wir haben bereits erwähnt, dass die Genauigkeit einer Näherung für ein Integral nicht durch die Erhöhung der Ordnung der benutzten Quadraturformel verbessert wird, sondern dass das Intervall $[a, b]$ in n Teilintervalle der Länge h zerlegt wird, und dass in jedem Teilintervall eine Quadraturformel kleiner Ordnung verwendet wird.

Für die summierte Quadraturformel

$$Q_h(f) := \sum_{i=1}^n Q_{[a+(i-1)h, a+ih]}(f)$$

erhält man aus (3.6) (und genauso aus (3.7))

Fehler von Quadraturformeln

$$\begin{aligned} & \left| \int_a^b f(x) dx - Q_h(f) \right| \\ &= \left| \sum_{i=1}^n \left(\int_{a+(i-1)h}^{a+ih} f(x) dx - Q_{[a+(i-1)h, a+ih]}(f) \right) \right| \\ &\leq \sum_{i=1}^n |E_{[a+(i-1)h, a+ih]}(f)| \\ &\leq \sum_{i=1}^n h^{m+1} \cdot \tilde{c}_m \cdot \max\{|f^{(m)}(x)| : a + (i-1)h \leq x \leq a + ih\} \\ &\leq n \cdot h \cdot h^m \cdot \tilde{c}_m \cdot \|f^{(m)}\|_\infty = h^m (b-a) \tilde{c}_m \cdot \|f^{(m)}\|_\infty. \end{aligned}$$

Fehler von Quadraturformeln

$$\begin{aligned}
 & \left| \int_a^b f(x) dx - Q_h(f) \right| \\
 &= \left| \sum_{i=1}^n \left(\int_{a+(i-1)h}^{a+ih} f(x) dx - Q_{[a+(i-1)h, a+ih]}(f) \right) \right| \\
 &\leq \sum_{i=1}^n |E_{[a+(i-1)h, a+ih]}(f)| \\
 &\leq \sum_{i=1}^n h^{m+1} \cdot \tilde{c}_m \cdot \max\{|f^{(m)}(x)| : a + (i-1)h \leq x \leq a + ih\} \\
 &\leq n \cdot h \cdot h^m \cdot \tilde{c}_m \cdot \|f^{(m)}\|_\infty = h^m (b-a) \tilde{c}_m \cdot \|f^{(m)}\|_\infty.
 \end{aligned}$$

Man verliert also für die summierte Formel eine Potenz in h .

Fehler von Quadraturformeln

Insbesondere erhält man für die **summierte Trapezregel** den Fehler

$$\left| \int_a^b f(x) dx - T_h(f) \right| \leq \frac{h^2}{12} (b - a) \cdot \|f''\|_\infty \quad (3.8)$$

Fehler von Quadraturformeln

Insbesondere erhält man für die **summierte Trapezregel** den Fehler

$$\left| \int_a^b f(x) dx - T_h(f) \right| \leq \frac{h^2}{12} (b-a) \cdot \|f''\|_\infty \quad (3.8)$$

und für die **summierte Simpson-Regel**

$$\left| \int_a^b f(x) dx - S_h(f) \right| \leq \frac{h^4}{2880} (b-a) \cdot \|f^{(4)}\|_\infty.$$

Quadraturformeln von Gauß

Wir haben bisher die Knoten der Quadraturformeln (äquidistant) **vorgegeben**.

Quadraturformeln von Gauß

Wir haben bisher die Knoten der Quadraturformeln (äquidistant) **vorgegeben**.

Die Fehlerordnung war dann (wenigstens) gleich der Anzahl der Knoten (im Falle der Simpson-Regel bei 3 Knoten 4).

Quadraturformeln von Gauß

Wir haben bisher die Knoten der Quadraturformeln (äquidistant) **vorgegeben**.

Die Fehlerordnung war dann (wenigstens) gleich der Anzahl der Knoten (im Falle der Simpson-Regel bei 3 Knoten 4).

Wir fragen nun, wie weit wir durch **Wahl der Knoten und der Gewichte** die Fehlerordnung **erhöhen** können.

Quadraturformeln von Gauß

Beispiel 3.17: Wir betrachten die Quadraturformel

$$G_1(f) := w_1 f(x_1)$$

mit **einem** Knoten x_1 für das Integral

$$\int_0^1 f(x) dx$$

Quadraturformeln von Gauß

Beispiel 3.17: Wir betrachten die Quadraturformel

$$G_1(f) := w_1 f(x_1)$$

mit **einem** Knoten x_1 für das Integral

$$\int_0^1 f(x) dx$$

und bestimmen $x_1 \in [0, 1]$ und $w_1 \in \mathbb{R}$ so, dass Polynome möglichst hohen Grades exakt integriert werden:

$$\int_0^1 x^0 dx = 1 = w_1 x_1^0 = w_1 \quad \Rightarrow \quad w_1 = 1,$$

$$\int_0^1 x^1 dx = 0.5 = w_1 x_1^1 = x_1 \quad \Rightarrow \quad x_1 = 0.5.$$

Quadraturformeln von Gauß

Durch diese beiden Gleichungen ist also die Quadraturformel

$$G_1(f) = f(0.5)$$

bereits festgelegt. Man erhält die **Mittelpunktregel**.

Quadraturformeln von Gauß

Durch diese beiden Gleichungen ist also die Quadraturformel

$$G_1(f) = f(0.5)$$

bereits festgelegt. Man erhält die **Mittelpunktregel**.

Wegen

$$\int_0^1 x^2 dx = \frac{1}{3} \neq 1 \cdot (0.5)^2$$

hat sie die Fehlerordnung 2. □

Quadraturformeln von Gauß

Beispiel 3.18: Für die Quadraturformel

$$G_2(f) = w_1 f(x_1) + w_2 f(x_2)$$

mit **zwei** Knoten $x_1, x_2 \in [0, 1]$ erhält man die Bestimmungsgleichungen

$$\int_0^1 x^0 dx = 1 = w_1 + w_2,$$

$$\int_0^1 x^1 dx = \frac{1}{2} = w_1 x_1 + w_2 x_2,$$

$$\int_0^1 x^2 dx = \frac{1}{3} = w_1 x_1^2 + w_2 x_2^2,$$

$$\int_0^1 x^3 dx = \frac{1}{4} = w_1 x_1^3 + w_2 x_2^3,$$

Quadraturformeln von Gauß

mit der (bis auf Vertauschung von x_1 und x_2) **eindeutigen Lösung**

$$w_1 = w_2 = \frac{1}{2}, \quad x_1 = \frac{1}{2} \left(1 - \frac{1}{\sqrt{3}} \right), \quad x_2 = \frac{1}{2} \left(1 + \frac{1}{\sqrt{3}} \right).$$

Quadraturformeln von Gauß

mit der (bis auf Vertauschung von x_1 und x_2) **eindeutigen Lösung**

$$w_1 = w_2 = \frac{1}{2}, \quad x_1 = \frac{1}{2} \left(1 - \frac{1}{\sqrt{3}} \right), \quad x_2 = \frac{1}{2} \left(1 + \frac{1}{\sqrt{3}} \right).$$

Wegen

$$\int_0^1 x^4 dx = \frac{1}{5} \neq \frac{7}{36} = w_1 x_1^4 + w_2 x_2^4$$

hat die gefundene Formel G_2 die Fehlerordnung 4. □

Quadraturformeln von Gauß

mit der (bis auf Vertauschung von x_1 und x_2) **eindeutigen Lösung**

$$w_1 = w_2 = \frac{1}{2}, \quad x_1 = \frac{1}{2} \left(1 - \frac{1}{\sqrt{3}} \right), \quad x_2 = \frac{1}{2} \left(1 + \frac{1}{\sqrt{3}} \right).$$

Wegen

$$\int_0^1 x^4 dx = \frac{1}{5} \neq \frac{7}{36} = w_1 x_1^4 + w_2 x_2^4$$

hat die gefundene Formel G_2 die Fehlerordnung 4. □

Prinzipiell kann man so fortfahren und Quadraturformeln immer höherer Ordnung konstruieren. Man erhält dann nichtlineare Gleichungssysteme, die immer **unübersichtlicher** werden.

Quadraturformeln von Gauß

In Verallgemeinerung unserer bisherigen Überlegungen betrachten wir gleich

$$I(f) = \int_a^b w(x)f(x) dx$$

mit einer **positiven Gewichtsfunktion** $w \in C[a, b]$.

Quadraturformeln von Gauß

In Verallgemeinerung unserer bisherigen Überlegungen betrachten wir gleich

$$I(f) = \int_a^b w(x)f(x) dx$$

mit einer **positiven Gewichtsfunktion** $w \in C[a, b]$.

Man kann zeigen, dass es zu jedem $n \in \mathbb{N}$ und jeder positiven Gewichtsfunktion $w \in C[a, b]$ eindeutig bestimmte Gewichte $w_i > 0$ und Knoten $x_i \in (a, b)$, $i = 1, \dots, n$, gibt, so dass die Quadraturformel

$$G_n(f) := \sum_{i=1}^n w_i f(x_i) \tag{3.9}$$

die Fehlerordnung $2n$ besitzt und dass durch keine Wahl von Knoten und Gewichten eine höhere Fehlerordnung erreichbar ist. Diese Formel heisst **Gaußsche Quadraturformel**.

Quadraturformeln von Gauß

Dass mit n Knoten nicht die Fehlerordnung $2n + 1$ erreicht werden kann, sieht man so ein.

Quadraturformeln von Gauß

Dass mit n Knoten nicht die Fehlerordnung $2n + 1$ erreicht werden kann, sieht man so ein.

Besäße (3.9) die Fehlerordnung $2n + 1$, so würde insbesondere das Polynom

$$p(x) := \prod_{j=1}^n (x - x_j)^2 \in \Pi_{2n}$$

exakt integriert.

Quadraturformeln von Gauß

Dass mit n Knoten nicht die Fehlerordnung $2n + 1$ erreicht werden kann, sieht man so ein.

Besäße (3.9) die Fehlerordnung $2n + 1$, so würde insbesondere das Polynom

$$p(x) := \prod_{j=1}^n (x - x_j)^2 \in \Pi_{2n}$$

exakt integriert.

Wegen $p \geq 0$ und $p \not\equiv 0$ gilt aber

$$\int_a^b w(x)p(x) dx > 0, \quad \text{während} \quad G_n(p) = 0 \quad \text{ist.}$$

Quadraturformeln von Gauß

Wir geben einige Gewichte und Knoten für den Spezialfall $w(x) \equiv 1$ und (ohne Beschränkung der Allgemeinheit) $[a, b] = [-1, 1]$ an.

n	w_i	x_i
1	$w_1 = 2$	$x_1 = 0$
2	$w_1 = w_2 = 1$	$x_2 = -x_1 = \frac{1}{\sqrt{3}}$
3	$w_1 = w_3 = \frac{5}{9}, w_2 = \frac{8}{9}$	$x_3 = -x_1 = \sqrt{\frac{3}{5}}, x_2 = 0$

Weitere Werte findet man in Abramowitz und Stegun, pp. 916 ff, und in Piessens et al., pp. 19 ff.

Quadraturformeln von Gauß

Für das Integral

$$I := \int_0^1 e^x \sin(5x) dx$$

erhält man mit den Gauß-Quadraturformeln G_n die Fehler

n	$G_n - I$
1	$1.04 \cdot 10^{+0}$
2	$-1.97 \cdot 10^{-1}$
3	$1.18 \cdot 10^{-2}$
4	$-3.04 \cdot 10^{-4}$
5	$3.73 \cdot 10^{-6}$

Quadraturformeln von Gauß

Bemerkung 3.19: Auch für unbeschränkte Integrationsintervalle kann man Gaußsche Quadraturformeln entwickeln.

Quadraturformeln von Gauß

Bemerkung 3.19: Auch für unbeschränkte Integrationsintervalle kann man Gaußsche Quadraturformeln entwickeln. So erhält man für Integrale der Gestalt

$$\int_0^{\infty} e^{-x} f(x) dx$$

(mit den Nullstellen der **Laguerre-Polynome** als Knoten) die **Gauß-Laguerre-Quadraturformeln**

Quadraturformeln von Gauß

Bemerkung 3.19: Auch für unbeschränkte Integrationsintervalle kann man Gaußsche Quadraturformeln entwickeln. So erhält man für Integrale der Gestalt

$$\int_0^{\infty} e^{-x} f(x) dx$$

(mit den Nullstellen der **Laguerre-Polynome** als Knoten) die **Gauß-Laguerre-Quadraturformeln**, und für Integrale der Gestalt

$$\int_{-\infty}^{\infty} e^{-x^2} f(x) dx$$

(mit den Nullstellen der **Hermite-Polynome** als Knoten) die **Gauß-Hermite-Quadraturformeln**.

Quadraturformeln von Gauß

Bemerkung 3.19: Auch für unbeschränkte Integrationsintervalle kann man Gaußsche Quadraturformeln entwickeln. So erhält man für Integrale der Gestalt

$$\int_0^{\infty} e^{-x} f(x) dx$$

(mit den Nullstellen der **Laguerre-Polynome** als Knoten) die **Gauß-Laguerre-Quadraturformeln**, und für Integrale der Gestalt

$$\int_{-\infty}^{\infty} e^{-x^2} f(x) dx$$

(mit den Nullstellen der **Hermite-Polynome** als Knoten) die **Gauß-Hermite-Quadraturformeln**.

Knoten und Gewichte der Gauß-Laguerre-Formeln (für $n \leq 15$) findet man in Abramowitz und Stegun auf Seite 923 und für die Gauß-Hermite-Formeln (für $n \leq 20$) auf Seite 924. □

Quadraturformeln von Gauß

Beispiel 3.20: Für das Integral

$$\int_0^{\infty} \frac{x}{1+e^x} dx = \int_0^{\infty} e^{-x} \frac{x}{1+e^{-x}} dx = \frac{\pi^2}{12} \approx 0.8224670334241132$$

enthält die Tabelle die Näherungen mit den [Gauß-Laguerre-Quadraturformeln](#) und die Fehler. Man sieht, dass man auch mit wenigen Knoten zu sehr guten Näherungen gelangt. □

n	Q_n	Fehler
1	0.7310585786300049	$9.14 \cdot 10^{-2}$
2	0.8052717896130982	$1.72 \cdot 10^{-2}$
3	0.8238172597250991	$-1.35 \cdot 10^{-3}$
4	0.8236994602380588	$-1.23 \cdot 10^{-3}$
5	0.8226695411616926	$-2.03 \cdot 10^{-4}$
6	0.8224050273750929	$6.20 \cdot 10^{-5}$

Adaptive Quadratur

Wir haben summierte Quadraturformeln nur mit **konstanter** Schrittweite $h > 0$ betrachtet.

Adaptive Quadratur

Wir haben summierte Quadraturformeln nur mit **konstanter** Schrittweite $h > 0$ betrachtet.

Es ist klar, dass man dabei für Funktionen mit unterschiedlichem Verhalten in verschiedenen Teilen des Integrationsintervalls entweder bei zu großem h ein ungenaues Ergebnis erhält oder bei zu kleinem h Arbeit in den Bereichen verschwendet, in denen die Funktion „gutartig“ ist.

Adaptive Quadratur

Wir haben summierte Quadraturformeln nur mit **konstanter** Schrittweite $h > 0$ betrachtet.

Es ist klar, dass man dabei für Funktionen mit unterschiedlichem Verhalten in verschiedenen Teilen des Integrationsintervalls entweder bei zu großem h ein ungenaues Ergebnis erhält oder bei zu kleinem h Arbeit in den Bereichen verschwendet, in denen die Funktion „gutartig“ ist.

Einen intelligenteren Ansatz verfolgt man mit der sogenannten **adaptiven Quadratur**.

Adaptive Quadratur

Wir haben summierte Quadraturformeln nur mit **konstanter** Schrittweite $h > 0$ betrachtet.

Es ist klar, dass man dabei für Funktionen mit unterschiedlichem Verhalten in verschiedenen Teilen des Integrationsintervalls entweder bei zu großem gewähltem h ein ungenaues Ergebnis erhält oder bei zu kleinem h Arbeit in den Bereichen verschwendet, in denen die Funktion „gutartig“ ist.

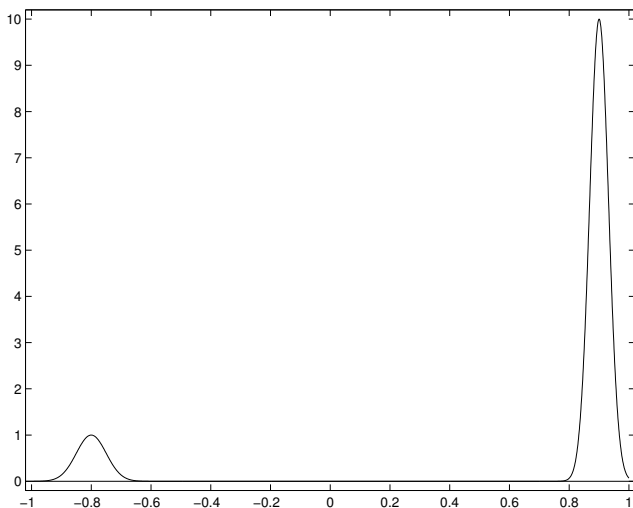
Einen intelligenteren Ansatz verfolgt man mit der sogenannten **adaptiven Quadratur**.

Beispiel 3.21: (siehe dazu die folgende Abbildung)

$$f(x) = \exp\left(-200(x + 0.8)^2\right) + 10 \cdot \exp\left(-500(x - 0.9)^2\right), \quad -1 \leq x \leq 1. \quad \square$$

Adaptive Quadratur

Abbildung zu Beispiel 3.21



Adaptive Quadratur

Wir entwickeln nun eine Vorgehensweise, mit der bei gegebenem Integranden, gegebenen Grenzen a und b und gegebener Genauigkeitsschranke $\varepsilon > 0$ eine Quadraturformel

$$I(f) = \sum_{j=0}^k w_j f(x_j)$$

erzeugt wird,

Adaptive Quadratur

Wir entwickeln nun eine Vorgehensweise, mit der bei gegebenem Integranden, gegebenen Grenzen a und b und gegebener Genauigkeitsschranke $\varepsilon > 0$ eine Quadraturformel

$$I(f) = \sum_{j=0}^k w_j f(x_j)$$

erzeugt wird, für die gilt

$$\left| \int_a^b f(x) dx - I(f) \right| \lesssim \varepsilon,$$

Adaptive Quadratur

Wir entwickeln nun eine Vorgehensweise, mit der bei gegebenem Integranden, gegebenen Grenzen a und b und gegebener Genauigkeitsschranke $\varepsilon > 0$ eine Quadraturformel

$$I(f) = \sum_{j=0}^k w_j f(x_j)$$

erzeugt wird, für die gilt

$$\left| \int_a^b f(x) dx - I(f) \right| \lesssim \varepsilon,$$

wobei das Symbol „ \lesssim “ besagt, dass die Ungleichung nur **asymptotisch**, d.h., für „feine“ Zerlegungen $a \leq x_0 < x_1 < \dots < x_k \leq b$ von $[a, b]$ gilt.

Adaptive Quadratur

Die Knoten x_j und Gewichte w_j werden **adaptiv** durch das Verfahren in Abhängigkeit von f und ε erzeugt.

Adaptive Quadratur

Die Knoten x_j und Gewichte w_j werden **adaptiv** durch das Verfahren in Abhängigkeit von f und ε erzeugt.

Es sei

$$Q(f) = \sum_{i=1}^n w_i f(t_i)$$

eine **Quadraturformel der Ordnung m** für das Referenzintervall $[-1, 1]$.

Adaptive Quadratur

Die Knoten x_j und Gewichte w_j werden **adaptiv** durch das Verfahren in Abhängigkeit von f und ε erzeugt.

Es sei

$$Q(f) = \sum_{i=1}^n w_i f(t_i)$$

eine **Quadraturformel der Ordnung m** für das Referenzintervall $[-1, 1]$.

Es sei das Integral bis zum Punkt $x_j \in [a, b)$ schon näherungsweise bestimmt und es sei $x_{j+1} \in (x_j, b]$ gegeben. Dann berechnen wir die **zwei** Näherungen

$$Q_{[x_j, x_{j+1}]}(f) = \frac{x_{j+1} - x_j}{2} \sum_{i=1}^n w_i f\left(\frac{x_{j+1} + x_j}{2} + t_i \frac{x_{j+1} - x_j}{2}\right)$$

Adaptive Quadratur

und (mittels eines Doppelschrittes)

$$\begin{aligned}\tilde{Q}_{[x_j, x_{j+1}]}(f) &:= \frac{x_{j+\frac{1}{2}} - x_j}{2} \sum_{i=1}^n w_i f\left(\frac{x_{j+\frac{1}{2}} + x_j}{2} + t_i \frac{x_{j+\frac{1}{2}} - x_j}{2}\right) \\ &+ \frac{x_{j+1} - x_{j+\frac{1}{2}}}{2} \sum_{i=1}^n w_i f\left(\frac{x_{j+1} + x_{j+\frac{1}{2}}}{2} + t_i \frac{x_{j+1} - x_{j+\frac{1}{2}}}{2}\right)\end{aligned}$$

für

$$\int_{x_j}^{x_{j+1}} f(x) dx,$$

wobei $x_{j+\frac{1}{2}} := 0.5(x_j + x_{j+1})$ gesetzt ist

Adaptive Quadratur

und (mittels **eines Doppelschrittes**)

$$\begin{aligned} \tilde{Q}_{[x_j, x_{j+1}]}(f) &:= \frac{x_{j+\frac{1}{2}} - x_j}{2} \sum_{i=1}^n w_i f\left(\frac{x_{j+\frac{1}{2}} + x_j}{2} + t_i \frac{x_{j+\frac{1}{2}} - x_j}{2}\right) \\ &+ \frac{x_{j+1} - x_{j+\frac{1}{2}}}{2} \sum_{i=1}^n w_i f\left(\frac{x_{j+1} + x_{j+\frac{1}{2}}}{2} + t_i \frac{x_{j+1} - x_{j+\frac{1}{2}}}{2}\right) \end{aligned}$$

für

$$\int_{x_j}^{x_{j+1}} f(x) dx,$$

wobei $x_{j+\frac{1}{2}} := 0.5(x_j + x_{j+1})$ gesetzt ist und hiermit eine **dritte Formel**

$$\hat{Q}_{[x_j, x_{j+1}]} := \frac{1}{2^m - 1} \left(2^m \cdot \tilde{Q}_{[x_j, x_{j+1}]}(f) - Q_{[x_j, x_{j+1}]}(f) \right).$$

Adaptive Quadratur

Dann ist mit Q und \tilde{Q} auch \hat{Q} eine Quadraturformel von mindestens der Ordnung m .

Adaptive Quadratur

Dann ist mit Q und \tilde{Q} auch \hat{Q} eine Quadraturformel von mindestens der Ordnung m .

Man kann leicht mit Hilfe der Fehlerdarstellung aus Satz 3.10 zeigen, dass durch \hat{Q} sogar Polynome vom Grade m exakt integriert werden, \hat{Q} also wenigstens die **Ordnung $m + 1$** hat.

Adaptive Quadratur

Dann ist mit Q und \tilde{Q} auch \hat{Q} eine Quadraturformel von mindestens der Ordnung m .

Man kann leicht mit Hilfe der Fehlerdarstellung aus Satz 3.10 zeigen, dass durch \hat{Q} sogar Polynome vom Grade m exakt integriert werden, \hat{Q} also wenigstens die **Ordnung $m + 1$** hat.

Wir benutzen nun \hat{Q} , um den **Fehler von \tilde{Q}** (der genaueren der beiden Ausgangsformeln) zu schätzen.

Adaptive Quadratur

Dann ist mit Q und \tilde{Q} auch \hat{Q} eine Quadraturformel von mindestens der Ordnung m .

Man kann leicht mit Hilfe der Fehlerdarstellung aus Satz 3.10 zeigen, dass durch \hat{Q} sogar Polynome vom Grade m exakt integriert werden, \hat{Q} also wenigstens die **Ordnung $m + 1$** hat.

Wir benutzen nun \hat{Q} , um den **Fehler von \tilde{Q}** (der genaueren der beiden Ausgangsformeln) zu schätzen.

Wie sich dabei herausstellt, benötigen wir nach der Herleitung niemals wieder die Formel \hat{Q} und haben ein Verfahren konstruiert, das **allein auf** den beiden Formeln Q und \tilde{Q} basiert.

Adaptive Quadratur

Es gilt mit $h := x_{j+1} - x_j$

$$\begin{aligned}\tilde{E}_{[x_j, x_{j+1}]}(f) &:= \int_{x_j}^{x_{j+1}} f(x) dx - \tilde{Q}_{[x_j, x_{j+1}]}(f) \\ &= \hat{Q}_{[x_j, x_{j+1}]}(f) - \tilde{Q}_{[x_j, x_{j+1}]}(f) + O(h^{m+2}) \\ &= \frac{1}{2^m - 1} \left(2^m \tilde{Q}_{[x_j, x_{j+1}]}(f) - Q_{[x_j, x_{j+1}]}(f) \right) \\ &\quad - \tilde{Q}_{[x_j, x_{j+1}]}(f) + O(h^{m+2}) \\ &= \frac{1}{2^m - 1} \left(\tilde{Q}_{[x_j, x_{j+1}]}(f) - Q_{[x_j, x_{j+1}]}(f) \right) + O(h^{m+2}).\end{aligned}$$

Adaptive Quadratur

Da andererseits $\tilde{E}_{[x_j, x_{j+1}]}(f) = O(h^{m+1})$ gilt, können wir für kleine h den Summanden $O(h^{m+2})$ vernachlässigen

Adaptive Quadratur

Da andererseits $\tilde{E}_{[x_j, x_{j+1}]}(f) = O(h^{m+1})$ gilt, können wir für kleine h den Summanden $O(h^{m+2})$ vernachlässigen und erhalten die **Fehlerschätzung**

$$\tilde{E}_{[x_j, x_{j+1}]}(f) \approx \frac{1}{2^m - 1} \left(\tilde{Q}_{[x_j, x_{j+1}]}(f) - Q_{[x_j, x_{j+1}]}(f) \right). \quad (3.10)$$

Adaptive Quadratur

Da andererseits $\tilde{E}_{[x_j, x_{j+1}]}(f) = O(h^{m+1})$ gilt, können wir für kleine h den Summanden $O(h^{m+2})$ vernachlässigen und erhalten die **Fehlerschätzung**

$$\tilde{E}_{[x_j, x_{j+1}]}(f) \approx \frac{1}{2^m - 1} \left(\tilde{Q}_{[x_j, x_{j+1}]}(f) - Q_{[x_j, x_{j+1}]}(f) \right). \quad (3.10)$$

Wir benutzen (3.10), um das Intervall $[a, b]$ durch Bisektion zu zerlegen in $a = x_0 < x_1 < \dots < x_k = b$, so dass für $j = 1, \dots, k$ gilt

$$\left| \tilde{Q}_{[x_{j-1}, x_j]}(f) - Q_{[x_{j-1}, x_j]}(f) \right| \leq \frac{2^m - 1}{b - a} (x_j - x_{j-1}) \varepsilon. \quad (3.11)$$

Adaptive Quadratur

Da andererseits $\tilde{E}_{[x_j, x_{j+1}]}(f) = O(h^{m+1})$ gilt, können wir für kleine h den Summanden $O(h^{m+2})$ vernachlässigen und erhalten die **Fehlerschätzung**

$$\tilde{E}_{[x_j, x_{j+1}]}(f) \approx \frac{1}{2^m - 1} \left(\tilde{Q}_{[x_j, x_{j+1}]}(f) - Q_{[x_j, x_{j+1}]}(f) \right). \quad (3.10)$$

Wir benutzen (3.10), um das Intervall $[a, b]$ durch Bisektion zu zerlegen in $a = x_0 < x_1 < \dots < x_k = b$, so dass für $j = 1, \dots, k$ gilt

$$\left| \tilde{Q}_{[x_{j-1}, x_j]}(f) - Q_{[x_{j-1}, x_j]}(f) \right| \leq \frac{2^m - 1}{b - a} (x_j - x_{j-1}) \varepsilon. \quad (3.11)$$

Dann folgt für die summierte Quadraturformel

$$Q(f) := \sum_{j=1}^k \tilde{Q}_{[x_{j-1}, x_j]}(f)$$

Adaptive Quadratur

aus (3.10)

$$\begin{aligned} E(f) &= \int_a^b f(x) dx - Q(f) = \sum_{j=1}^k \tilde{E}_{[x_{j-1}, x_j]}(f) \\ &\lesssim \frac{1}{2^m - 1} \sum_{j=1}^k \left| \tilde{Q}_{[x_{j-1}, x_j]}(f) - Q_{[x_{j-1}, x_j]}(f) \right| \\ &\leq \frac{1}{b-a} \sum_{j=1}^k (x_j - x_{j-1}) \varepsilon = \varepsilon. \end{aligned}$$

Adaptive Quadratur

Mit der Fehlerschätzung kann man nun auf folgende Weise ein Integral mit einer gewünschten (asymptotischen) Genauigkeit adaptiv berechnen.

Adaptive Quadratur

Mit der Fehlerschätzung kann man nun auf folgende Weise ein Integral mit einer gewünschten (asymptotischen) Genauigkeit adaptiv berechnen.

Es sei das Integral

$$\int_a^{x_j} f(x) dx$$

schon mit der gewünschten Genauigkeit bestimmt.

Adaptive Quadratur

Mit der Fehlerschätzung kann man nun auf folgende Weise ein Integral mit einer gewünschten (asymptotischen) Genauigkeit adaptiv berechnen.

Es sei das Integral

$$\int_a^{x_j} f(x) dx$$

schon mit der gewünschten Genauigkeit bestimmt. Es seien mit der Schrittweite h bereits die Näherungen

$$Q_{[x_j, x_j+h]}(f) \quad \text{und} \quad \tilde{Q}_{[x_j, x_j+h]}(f)$$

berechnet.

Adaptive Quadratur

Mit der Fehlerschätzung kann man nun auf folgende Weise ein Integral mit einer gewünschten (asymptotischen) Genauigkeit adaptiv berechnen.

Es sei das Integral

$$\int_a^{x_j} f(x) dx$$

schon mit der gewünschten Genauigkeit bestimmt. Es seien mit der Schrittweite h bereits die Näherungen

$$Q_{[x_j, x_j+h]}(f) \quad \text{und} \quad \tilde{Q}_{[x_j, x_j+h]}(f)$$

berechnet. Dann kann man das Erfülltsein der **lokalen Fehlerschranke** (3.11) prüfen.

Adaptive Quadratur

Mit der Fehlerschätzung kann man nun auf folgende Weise ein Integral mit einer gewünschten (asymptotischen) Genauigkeit adaptiv berechnen.

Es sei das Integral

$$\int_a^{x_j} f(x) dx$$

schon mit der gewünschten Genauigkeit bestimmt. Es seien mit der Schrittweite h bereits die Näherungen

$$Q_{[x_j, x_j+h]}(f) \quad \text{und} \quad \tilde{Q}_{[x_j, x_j+h]}(f)$$

berechnet. Dann kann man das Erfülltsein der **lokalen Fehlerschranke** (3.11) prüfen.

Ist dies der Fall, so geht man zu dem neuen Intervall $[x_{j+1}, x_{j+1} + h_{\text{neu}}]$, $x_{j+1} := x_j + h$, über.

Adaptive Quadratur

Mit der Fehlerschätzung kann man nun auf folgende Weise ein Integral mit einer gewünschten (asymptotischen) Genauigkeit adaptiv berechnen.

Es sei das Integral

$$\int_a^{x_j} f(x) dx$$

schon mit der gewünschten Genauigkeit bestimmt. Es seien mit der Schrittweite h bereits die Näherungen

$$Q_{[x_j, x_j+h]}(f) \quad \text{und} \quad \tilde{Q}_{[x_j, x_j+h]}(f)$$

berechnet. Dann kann man das Erfülltsein der **lokalen Fehlerschranke** (3.11) prüfen.

Ist dies der Fall, so geht man zu dem neuen Intervall $[x_{j+1}, x_{j+1} + h_{\text{neu}}]$, $x_{j+1} := x_j + h$, über. Sonst wiederholt man den Schritt mit einer verkleinerten Schrittweite h_{neu} .

Adaptive Quadratur

Die **neue Schrittweite** kann man so bestimmen: Es ist

$$E_{[x_j, x_{j+h}]} \approx \frac{1}{2^m - 1} |Q_{[x_j, x_{j+h}]}(f) - \tilde{Q}_{[x_j, x_{j+h}]}(f)| \approx Ch^{m+1},$$

Adaptive Quadratur

Die **neue Schrittweite** kann man so bestimmen: Es ist

$$E_{[x_j, x_j+h]} \approx \frac{1}{2^m - 1} |Q_{[x_j, x_j+h]}(f) - \tilde{Q}_{[x_j, x_j+h]}(f)| \approx Ch^{m+1},$$

d.h.,

$$C \approx \frac{h^{-m-1}}{2^m - 1} |Q_{[x_j, x_j+h]}(f) - \tilde{Q}_{[x_j, x_j+h]}(f)|,$$

Adaptive Quadratur

Die **neue Schrittweite** kann man so bestimmen: Es ist

$$E_{[x_j, x_j+h]} \approx \frac{1}{2^m - 1} |Q_{[x_j, x_j+h]}(f) - \tilde{Q}_{[x_j, x_j+h]}(f)| \approx Ch^{m+1},$$

d.h.,

$$C \approx \frac{h^{-m-1}}{2^m - 1} |Q_{[x_j, x_j+h]}(f) - \tilde{Q}_{[x_j, x_j+h]}(f)|,$$

und daher kann man erwarten, dass mit

$$\varepsilon \frac{h_{\text{neu}}}{b-a} = Ch_{\text{neu}}^{m+1} = \frac{h^{-m-1}}{2^m - 1} |Q_{[x_j, x_j+h]}(f) - \tilde{Q}_{[x_j, x_j+h]}(f)| h_{\text{neu}}^{m+1},$$

Adaptive Quadratur

Die **neue Schrittweite** kann man so bestimmen: Es ist

$$E_{[x_j, x_j+h]} \approx \frac{1}{2^m - 1} |Q_{[x_j, x_j+h]}(f) - \tilde{Q}_{[x_j, x_j+h]}(f)| \approx Ch^{m+1},$$

d.h.,

$$C \approx \frac{h^{-m-1}}{2^m - 1} |Q_{[x_j, x_j+h]}(f) - \tilde{Q}_{[x_j, x_j+h]}(f)|,$$

und daher kann man erwarten, dass mit

$$\varepsilon \frac{h_{\text{neu}}}{b-a} = Ch_{\text{neu}}^{m+1} = \frac{h^{-m-1}}{2^m - 1} |Q_{[x_j, x_j+h]}(f) - \tilde{Q}_{[x_j, x_j+h]}(f)| h_{\text{neu}}^{m+1},$$

d.h. mit

$$h_{\text{neu}} = h \left(\frac{(2^m - 1)h\varepsilon}{(b-a)|Q_{[x_j, x_j+h]}(f) - \tilde{Q}_{[x_j, x_j+h]}(f)|} \right)^{1/m}$$

die lokale Fehlerschranke (3.11) eingehalten wird.

Adaptive Quadratur

Algorithmus 3.22: (Adaptive Gauß-Quadratur)

```
function int=adap_gauss3(f,a,b,tol);
factor=63*tol/(b-a);  h=0.1*(b-a);  int=0;
while a < b
    q =gauss3(f,a,a+h);
    qs=gauss3(f,a,a+0.5*h)+gauss3(f,a+0.5*h,a+h);
    h_neu=0.9*h*(h*factor/abs(qs-q))^(1/6);
    if abs(q-q_s) > h*factor;
        h=h_neu;
    else
        int=int+qs+(qs-q)/63;
        a=a+h;
        h=min(h_neu,b-a);
    end
end
```

Adaptive Quadratur

Beispiel 3.23: Für das Beispiel 3.21 erhält man hiermit die Ergebnisse aus der folgenden Tabelle. Wie erwartet wird der tatsächliche Fehler (jedenfalls für $\varepsilon \leq 10^{-4}$) **deutlich kleiner** als die vorgegebene Toleranz ε . \square

ε	Fehler	Funktionsauswertungen
10^{-3}	$8.78 \cdot 10^{-2}$	54
10^{-4}	$2.99 \cdot 10^{-5}$	153
10^{-5}	$2.63 \cdot 10^{-7}$	270
10^{-6}	$8.22 \cdot 10^{-9}$	351
10^{-7}	$2.10 \cdot 10^{-9}$	531
10^{-8}	$8.02 \cdot 10^{-12}$	747

Adaptive Quadratur

Verwendet man wie in einem adaptiven Verfahren wie soeben hergeleitet **dieselbe Gauß-Formel für Q und \tilde{Q}** , so kann man die an den Knoten von Q berechneten Funktionswerte bei der Auswertung von \tilde{Q} **nicht wiederverwenden**.

Adaptive Quadratur

Verwendet man wie in einem adaptiven Verfahren wie soeben hergeleitet **dieselbe Gauß-Formel für Q und \tilde{Q}** , so kann man die an den Knoten von Q berechneten Funktionswerte bei der Auswertung von \tilde{Q} **nicht wiederverwenden**.

Von **Kronrod** wurde 1965 die folgende Vorgehensweise vorgeschlagen, die diesen Nachteil nicht hat:

Adaptive Quadratur

Wir gehen aus von einer Gauß-Formel

$$G_n(f) = \sum_{i=1}^n w_i f(x_i)$$

der Ordnung $2n$ mit den Knoten $x_1, \dots, x_n \in (-1, 1)$

Adaptive Quadratur

Wir gehen aus von einer Gauß-Formel

$$G_n(f) = \sum_{i=1}^n w_i f(x_i)$$

der Ordnung $2n$ mit den Knoten $x_1, \dots, x_n \in (-1, 1)$ und bestimmen $n + 1$ **weitere Knoten** $y_0, \dots, y_n \in (-1, 1)$ und **Gewichte** α_i, β_i , so dass die Quadraturformel

$$K_n(f) := \sum_{i=1}^n \alpha_i f(x_i) + \sum_{i=0}^n \beta_i f(y_i)$$

möglichst hohe Ordnung besitzt.

Adaptive Quadratur

Wir gehen aus von einer Gauß-Formel

$$G_n(f) = \sum_{i=1}^n w_i f(x_i)$$

der Ordnung $2n$ mit den Knoten $x_1, \dots, x_n \in (-1, 1)$ und bestimmen $n + 1$ **weitere Knoten** $y_0, \dots, y_n \in (-1, 1)$ und **Gewichte** α_i, β_i , so dass die Quadraturformel

$$K_n(f) := \sum_{i=1}^n \alpha_i f(x_i) + \sum_{i=0}^n \beta_i f(y_i)$$

möglichst hohe Ordnung besitzt.

Die Quadraturformel K_n heißt die zu G_n gehörige **Kronrod-Formel**.

Adaptive Quadratur

Beispiel 3.24: Ausgehend von G_2 ,

$$G_2(f) = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(+\frac{1}{\sqrt{3}}\right),$$

machen wir für K_2 den Ansatz

$$K_2(f) = \alpha_1 f\left(-\frac{1}{\sqrt{3}}\right) + \alpha_2 f\left(+\frac{1}{\sqrt{3}}\right) + \beta_0 f(y_0) + \beta_1 f(y_1) + \beta_2 f(y_2)$$

und bestimmen die 8 Unbekannten

$$\alpha_1, \alpha_2, \quad \beta_0, \beta_1, \beta_2, \quad y_0, y_1, y_2$$

so, dass die Funktionen x^j , $j = 0, 1, 2, \dots, m$, für möglichst großes m durch K_2 exakt integriert werden.

Adaptive Quadratur

Man kann zeigen, dass die Kronrod-Formeln symmetrisch sind (hier: $\alpha_1 = \alpha_2$, $\beta_0 = \beta_2$, $y_0 = -y_2$, $y_1 = 0$).

Adaptive Quadratur

Man kann zeigen, dass die Kronrod-Formeln symmetrisch sind (hier: $\alpha_1 = \alpha_2$, $\beta_0 = \beta_2$, $y_0 = -y_2$, $y_1 = 0$).

Unter Ausnutzung dieser Symmetrie folgt

$$K_2(x^{2j+1}) = 0 = \int_{-1}^1 x^{2j+1} dx \quad \text{für alle } j = 0, 1, 2, \dots$$

Adaptive Quadratur

Man kann zeigen, dass die Kronrod-Formeln symmetrisch sind (hier: $\alpha_1 = \alpha_2$, $\beta_0 = \beta_2$, $y_0 = -y_2$, $y_1 = 0$).

Unter Ausnutzung dieser Symmetrie folgt

$$K_2(x^{2j+1}) = 0 = \int_{-1}^1 x^{2j+1} dx \quad \text{für alle } j = 0, 1, 2, \dots$$

Für die geraden Potenzen ergibt sich das nichtlineare Gleichungssystem

$$x^0: 2\alpha_1 + 2\beta_0 + \beta_1 = 2,$$

$$x^2: \frac{2}{3}\alpha_1 + 2\beta_0 y_0^2 = \frac{2}{3},$$

$$x^4: \frac{2}{9}\alpha_1 + 2\beta_0 y_0^4 = \frac{2}{9},$$

$$x^6: \frac{2}{27}\alpha_1 + 2\beta_0 y_0^6 = \frac{2}{27},$$

Adaptive Quadratur

mit der **eindeutigen Lösung**

$$y_0 = \sqrt{\frac{6}{7}}, \quad \alpha_1 = \frac{243}{495}, \quad \beta_0 = \frac{98}{495}, \quad \beta_1 = \frac{308}{495},$$

Adaptive Quadratur

mit der **eindeutigen Lösung**

$$y_0 = \sqrt{\frac{6}{7}}, \quad \alpha_1 = \frac{243}{495}, \quad \beta_0 = \frac{98}{495}, \quad \beta_1 = \frac{308}{495},$$

d.h.

$$K_2(f) = \frac{243}{495} G_2(f) + \frac{1}{495} \left(98 \left(f\left(-\sqrt{\frac{6}{7}}\right) + f\left(+\sqrt{\frac{6}{7}}\right) \right) + 308 f(0) \right).$$

Adaptive Quadratur

mit der **eindeutigen Lösung**

$$y_0 = \sqrt{\frac{6}{7}}, \quad \alpha_1 = \frac{243}{495}, \quad \beta_0 = \frac{98}{495}, \quad \beta_1 = \frac{308}{495},$$

d.h.

$$K_2(f) = \frac{243}{495} G_2(f) + \frac{1}{495} \left(98 \left(f\left(-\sqrt{\frac{6}{7}}\right) + f\left(+\sqrt{\frac{6}{7}}\right) \right) + 308f(0) \right).$$

Nach Konstruktion hat diese Formel mindestens die **Ordnung 8** und durch Berechnung von $E(x^8)$ sieht man, dass die Ordnung genau 8 ist. □

Adaptive Quadratur

Man kann zeigen, dass zu einer Gauß-Formel G_n mit n Punkten stets die Kronrod-Formel mit $2n + 1$ Punkten konstruiert werden kann. Ihre **Ordnung** ist $3n + 2$, falls n gerade ist, und sogar $3n + 3$, falls n ungerade ist.

Adaptive Quadratur

Man kann zeigen, dass zu einer Gauß-Formel G_n mit n Punkten stets die Kronrod-Formel mit $2n + 1$ Punkten konstruiert werden kann. Ihre **Ordnung** ist $3n + 2$, falls n gerade ist, und sogar $3n + 3$, falls n ungerade ist.

Die Kronrod Formel K_n kann man nun auf folgende Weise nutzen, um den **Fehler** E_n der zugehörigen Gauß-Formel **zu schätzen**.

Adaptive Quadratur

Man kann zeigen, dass zu einer Gauß-Formel G_n mit n Punkten stets die Kronrod-Formel mit $2n + 1$ Punkten konstruiert werden kann. Ihre **Ordnung** ist $3n + 2$, falls n gerade ist, und sogar $3n + 3$, falls n ungerade ist.

Die Kronrod Formel K_n kann man nun auf folgende Weise nutzen, um den **Fehler** E_n der zugehörigen Gauß-Formel **zu schätzen**.

Es gilt für jedes Intervall $[x_j, x_{j+1}] \subset [-1, 1]$ und $h := x_{j+1} - x_j$

$$\int_{x_j}^{x_{j+1}} f(x) dx = K_n(f) + h \cdot h^{3n+2} \cdot c_{3n+2} \cdot f^{(3n+2)}(\eta),$$

Adaptive Quadratur

Man kann zeigen, dass zu einer Gauß-Formel G_n mit n Punkten stets die Kronrod-Formel mit $2n + 1$ Punkten konstruiert werden kann. Ihre **Ordnung** ist $3n + 2$, falls n gerade ist, und sogar $3n + 3$, falls n ungerade ist.

Die Kronrod Formel K_n kann man nun auf folgende Weise nutzen, um den **Fehler** E_n der zugehörigen Gauß-Formel **zu schätzen**.

Es gilt für jedes Intervall $[x_j, x_{j+1}] \subset [-1, 1]$ und $h := x_{j+1} - x_j$

$$\int_{x_j}^{x_{j+1}} f(x) dx = K_n(f) + h \cdot h^{3n+2} \cdot c_{3n+2} \cdot f^{(3n+2)}(\eta),$$

und daher folgt

$$\begin{aligned} E_n &:= \int_{x_j}^{x_{j+1}} f(x) dx - G_n(f) \\ &= K_n(f) - G_n(f) + h^{3n+3} \cdot c_{3n+2} \cdot f^{(3n+2)}(\eta). \end{aligned}$$

Adaptive Quadratur

Da E_n proportional zu h^{2n+1} ist, können wir für kleine h den letzten Summanden vernachlässigen und erhalten

$$E_n \approx K_n(f) - G_n(f).$$

Adaptive Quadratur

Da E_n proportional zu h^{2n+1} ist, können wir für kleine h den letzten Summanden vernachlässigen und erhalten

$$E_n \approx K_n(f) - G_n(f).$$

In dem folgenden Algorithmus schätzen wir hiermit den Fehler der ungenaueren **Gauß-Formel**, verwenden aber **als Näherung** für das Integral

$$\int_{x_j}^{x_{j+1}} f(x) dx$$

den mit der **Kronrod-Formel** ermittelten Wert.

Adaptive Quadratur

Da E_n proportional zu h^{2n+1} ist, können wir für kleine h den letzten Summanden vernachlässigen und erhalten

$$E_n \approx K_n(f) - G_n(f).$$

In dem folgenden Algorithmus schätzen wir hiermit den Fehler der ungenaueren **Gauß-Formel**, verwenden aber **als Näherung** für das Integral

$$\int_{x_j}^{x_{j+1}} f(x) dx$$

den mit der **Kronrod-Formel** ermittelten Wert.

Dies führt dazu, dass der Fehler wesentlich **unterhalb** der geforderten Toleranz liegt.

Adaptive Quadratur

Algorithmus 3.25: (Adaptive Kronrod Quadratur)

```
function int=adap_kronrod(f,a,b,tol);
h=0.1*(b-a); int=0; eps=tol/(b-a);
while a < b
    [int1,int2]=kronrod(f,a,a+h);
    h_neu=0.9*h*(h*eps/abs(int1-int2))^(1/4);
    if abs(int1-int2) > h*eps;
        h=h_neu;
    else
        int=int+int2;
        a=a+h;
        h=min(h_neu,b-a);
    end
end
```

Adaptive Quadratur

Beispiel 3.26: Für den Algorithmus 3.25 erhält man hiermit die Ergebnisse aus der folgenden Tabelle. Wie erwartet wird der tatsächliche Fehler wiederum deutlich kleiner als die vorgegebene Toleranz ε . □

ε	Fehler	Funktionsauswertungen
10^{-1}	$4.79 \cdot 10^{-5}$	70
10^{-2}	$3.51 \cdot 10^{-7}$	150
10^{-3}	$1.44 \cdot 10^{-8}$	255
10^{-4}	$6.67 \cdot 10^{-11}$	435

Adaptive Quadratur

Bemerkung 3.27: In dem Handbuch von Piessens et al. des Software-Pakets QUADPACK finden sich die Knoten und Gewichte für Gauß-Formeln und die zugehörigen Kronrod-Formeln (bis hin zur 30-Punkt Gauß- und 61-Punkt Kronrod-Formel) und weiteres Material zu adaptiven Quadratur-Methoden.

Adaptive Quadratur

Bemerkung 3.27: In dem Handbuch von Piessens et al. des Software-Pakets QUADPACK finden sich die Knoten und Gewichte für Gauß-Formeln und die zugehörigen Kronrod-Formeln (bis hin zur 30-Punkt Gauß- und 61-Punkt Kronrod-Formel) und weiteres Material zu adaptiven Quadratur-Methoden.

Die FORTRAN 77 Subroutines von QUADPACK können als **Public Domain Software** bezogen werden von

<http://www.netlib.org/quadpack>

Numerische Differentiation

Wir betrachten eine Funktion $f : [a, b] \rightarrow \mathbb{R}$, von der nur die Funktionswerte $y_j := f(x_j)$ an diskreten Punkten $a \leq x_1 < x_2 < \dots < x_n \leq b$ bekannt sind.

Numerische Differentiation

Wir betrachten eine Funktion $f : [a, b] \rightarrow \mathbb{R}$, von der nur die Funktionswerte $y_j := f(x_j)$ an diskreten Punkten $a \leq x_1 < x_2 < \dots < x_n \leq b$ bekannt sind.

Aufgabe ist es, aus diesen diskreten Daten eine **Näherung für** den Wert einer Ableitung $f^{(m)}(x)$, $m \geq 1$, an einer Stelle x zu ermitteln.

Numerische Differentiation

Wir betrachten eine Funktion $f : [a, b] \rightarrow \mathbb{R}$, von der nur die Funktionswerte $y_j := f(x_j)$ an diskreten Punkten $a \leq x_1 < x_2 < \dots < x_n \leq b$ bekannt sind.

Aufgabe ist es, aus diesen diskreten Daten eine **Näherung für** den Wert einer Ableitung $f^{(m)}(x)$, $m \geq 1$, an einer Stelle x zu ermitteln.

Ähnlich wie bei der numerischen Integration **interpolieren** wir hierzu an einigen der gegebenen Daten (x_j, y_j) in der Nähe des Punktes x und wählen die m te Ableitung der interpolierenden Funktion an der Stelle x als Näherung für $f^{(m)}(x)$.

Numerische Differentiation

Wir betrachten eine Funktion $f : [a, b] \rightarrow \mathbb{R}$, von der nur die Funktionswerte $y_j := f(x_j)$ an diskreten Punkten $a \leq x_1 < x_2 < \dots < x_n \leq b$ bekannt sind.

Aufgabe ist es, aus diesen diskreten Daten eine **Näherung für** den Wert einer Ableitung $f^{(m)}(x)$, $m \geq 1$, an einer Stelle x zu ermitteln.

Ähnlich wie bei der numerischen Integration **interpolieren** wir hierzu an einigen der gegebenen Daten (x_j, y_j) in der Nähe des Punktes x und wählen die m te Ableitung der interpolierenden Funktion an der Stelle x als Näherung für $f^{(m)}(x)$.

Gebräuchlich ist die Interpolation mit **Polynomen** und mit **Splines**.

Numerische Differentiation

Wir betrachten eine Funktion $f : [a, b] \rightarrow \mathbb{R}$, von der nur die Funktionswerte $y_j := f(x_j)$ an diskreten Punkten $a \leq x_1 < x_2 < \dots < x_n \leq b$ bekannt sind.

Aufgabe ist es, aus diesen diskreten Daten eine **Näherung für** den Wert einer Ableitung $f^{(m)}(x)$, $m \geq 1$, an einer Stelle x zu ermitteln.

Ähnlich wie bei der numerischen Integration **interpolieren** wir hierzu an einigen der gegebenen Daten (x_j, y_j) in der Nähe des Punktes x und wählen die m te Ableitung der interpolierenden Funktion an der Stelle x als Näherung für $f^{(m)}(x)$.

Gebräuchlich ist die Interpolation mit **Polynomen** und mit **Splines**.

Wir beschränken uns hier auf die Interpolation mit Polynomen und betrachten nur den Fall, dass die Stelle x , an der die Ableitung approximiert werden soll, ein Knoten ist.

Numerische Differentiation

Wir leiten Formeln für die Approximation der **ersten Ableitung** her.

Numerische Differentiation

Wir leiten Formeln für die Approximation der **ersten Ableitung** her.

Beispiel 3.28: Interpoliert man f linear mit den Daten (x_j, y_j) und (x_{j+1}, y_{j+1}) , so erhält man

$$p(x) = y_j + \frac{y_{j+1} - y_j}{x_{j+1} - x_j}(x - x_j),$$

Numerische Differentiation

Wir leiten Formeln für die Approximation der **ersten Ableitung** her.

Beispiel 3.28: Interpoliert man f linear mit den Daten (x_j, y_j) und (x_{j+1}, y_{j+1}) , so erhält man

$$p(x) = y_j + \frac{y_{j+1} - y_j}{x_{j+1} - x_j}(x - x_j),$$

und als Näherung für die Ableitung

$$f'(x_j) \approx p'(x_j) = \frac{y_{j+1} - y_j}{x_{j+1} - x_j}. \quad (3.12)$$

Numerische Differentiation

Wir leiten Formeln für die Approximation der **ersten Ableitung** her.

Beispiel 3.28: Interpoliert man f linear mit den Daten (x_j, y_j) und (x_{j+1}, y_{j+1}) , so erhält man

$$p(x) = y_j + \frac{y_{j+1} - y_j}{x_{j+1} - x_j}(x - x_j),$$

und als Näherung für die Ableitung

$$f'(x_j) \approx p'(x_j) = \frac{y_{j+1} - y_j}{x_{j+1} - x_j}. \quad (3.12)$$

Dieser Ausdruck heißt der **vorwärtsgenommene Differenzenquotient**.

Numerische Differentiation

Interpoliert man die Daten (x_{j-1}, y_{j-1}) und (x_j, y_j) linear, so erhält man genauso die Approximation

$$f'(x_j) \approx \frac{y_j - y_{j-1}}{x_j - x_{j-1}} \quad (3.13)$$

durch den rückwärtsgenommenen Differenzenquotienten.

Numerische Differentiation

Interpoliert man die Daten (x_{j-1}, y_{j-1}) und (x_j, y_j) linear, so erhält man genauso die Approximation

$$f'(x_j) \approx \frac{y_j - y_{j-1}}{x_j - x_{j-1}} \quad (3.13)$$

durch den rückwärtsgenommenen Differenzenquotienten.

Interpoliert man f quadratisch mit den Knoten (x_{j+k}, y_{j+k}) , $k = -1, 0, 1$, so erhält man

$$p(x) = y_j + [x_{j-1}, x_j](x - x_j) + [x_{j+1}, x_{j-1}, x_j](x - x_{j-1})(x - x_j)$$

mit der Ableitung

$$p'(x) = [x_{j-1}, x_j] + [x_{j+1}, x_{j-1}, x_j](2x - x_{j-1} - x_j).$$

Numerische Differentiation

Einsetzen von $x = x_j$ liefert nach kurzer Rechnung

$$f'(x_j) \approx \frac{x_{j+1} - x_j}{x_{j+1} - x_{j-1}} [x_{j-1}, x_j] + \frac{x_j - x_{j-1}}{x_{j+1} - x_{j-1}} [x_j, x_{j+1}]. \quad (3.14)$$

Numerische Differentiation

Einsetzen von $x = x_j$ liefert nach kurzer Rechnung

$$f'(x_j) \approx \frac{x_{j+1} - x_j}{x_{j+1} - x_{j-1}} [x_{j-1}, x_j] + \frac{x_j - x_{j-1}}{x_{j+1} - x_{j-1}} [x_j, x_{j+1}]. \quad (3.14)$$

Ist speziell $x_{j+1} - x_j = x_j - x_{j-1} =: h$, also die Knoten äquidistant, so ist (3.14) der **zentrale Differenzenquotient**

$$f'(x_j) \approx \frac{1}{2} [x_{j-1}, x_j] + \frac{1}{2} [x_j, x_{j+1}] = \frac{y_{j+1} - y_{j-1}}{2h}. \quad (3.15)$$

Numerische Differentiation

Sind nur Funktionswerte von f auf einer Seite von x_j bekannt, so verwendet man **einseitige Differenzenapproximationen**.

Numerische Differentiation

Sind nur Funktionswerte von f auf einer Seite von x_j bekannt, so verwendet man **einseitige Differenzenapproximationen**.

Z. B. erhält man mit $y_{j+k} = f(x_{j+k})$, $k = 0, 1, 2$, die Näherung

$$f'(x_j) \approx \frac{x_{j+2} + x_{j+1} - 2x_j}{x_{j+2} - x_j} [x_j, x_{j+1}] - \frac{x_{j+1} - x_j}{x_{j+2} - x_j} [x_{j+1}, x_{j+2}], \quad (3.16)$$

Numerische Differentiation

Sind nur Funktionswerte von f auf einer Seite von x_j bekannt, so verwendet man **einseitige Differenzenapproximationen**.

Z. B. erhält man mit $y_{j+k} = f(x_{j+k})$, $k = 0, 1, 2$, die Näherung

$$f'(x_j) \approx \frac{x_{j+2} + x_{j+1} - 2x_j}{x_{j+2} - x_j} [x_j, x_{j+1}] - \frac{x_{j+1} - x_j}{x_{j+2} - x_j} [x_{j+1}, x_{j+2}], \quad (3.16)$$

und im äquidistanten Fall

$$f'(x_j) \approx \frac{3}{2} [x_j, x_{j+1}] - \frac{1}{2} [x_{j+1}, x_{j+2}] = \frac{-y_{j+2} + 4y_{j+1} - 3y_j}{2h}. \quad (3.17)$$

Numerische Differentiation

Genauso erhält man mit 5 bzw. 7 äquidistanten Knoten die **zentralen Differenzenapproximationen**

$$f'(x_j) \approx \frac{1}{12h}(y_{j-2} - 8y_{j-1} + 8y_{j+1} - y_{j+2}) \quad (3.18)$$

Numerische Differentiation

Genauso erhält man mit 5 bzw. 7 äquidistanten Knoten die **zentralen Differenzenapproximationen**

$$f'(x_j) \approx \frac{1}{12h}(y_{j-2} - 8y_{j-1} + 8y_{j+1} - y_{j+2}) \quad (3.18)$$

und

$$f'(x_j) \approx \frac{1}{60h}(-y_{j-3} + 9y_{j-2} - 45y_{j-1} + 45y_{j+1} - 9y_{j+2} + y_{j+3}). \quad (3.19)$$

□

Numerische Differentiation

Den Fehler einer Differenzenformel kann man mit Hilfe des **Taylorischen Satzes** bestimmen.

Numerische Differentiation

Den Fehler einer Differenzenformel kann man mit Hilfe des **Taylorischen Satzes** bestimmen. Für $f \in C^2$ gilt mit einem $\xi \in (x_j, x_j + h)$

$$f(x_j + h) = f(x_j) + f'(x_j)h + \frac{h^2}{2}f''(\xi),$$

Numerische Differentiation

Den Fehler einer Differenzenformel kann man mit Hilfe des **Taylorischen Satzes** bestimmen. Für $f \in C^2$ gilt mit einem $\xi \in (x_j, x_j + h)$

$$f(x_j + h) = f(x_j) + f'(x_j)h + \frac{h^2}{2}f''(\xi),$$

d.h.

$$f'(x_j) = \frac{y_{j+1} - y_j}{h} - \frac{h}{2}f''(\xi),$$

Numerische Differentiation

Den Fehler einer Differenzenformel kann man mit Hilfe des **Taylorischen Satzes** bestimmen. Für $f \in C^2$ gilt mit einem $\xi \in (x_j, x_j + h)$

$$f(x_j + h) = f(x_j) + f'(x_j)h + \frac{h^2}{2}f''(\xi),$$

d.h.

$$f'(x_j) = \frac{y_{j+1} - y_j}{h} - \frac{h}{2}f''(\xi),$$

und genauso mit einem $\eta \in (x_j - h, x_j)$

$$f'(x_j) = \frac{y_j - y_{j-1}}{h} + \frac{h}{2}f''(\eta).$$

Numerische Differentiation

Den Fehler einer Differenzenformel kann man mit Hilfe des **Taylorischen Satzes** bestimmen. Für $f \in C^2$ gilt mit einem $\xi \in (x_j, x_j + h)$

$$f(x_j + h) = f(x_j) + f'(x_j)h + \frac{h^2}{2}f''(\xi),$$

d.h.

$$f'(x_j) = \frac{y_{j+1} - y_j}{h} - \frac{h}{2}f''(\xi),$$

und genauso mit einem $\eta \in (x_j - h, x_j)$

$$f'(x_j) = \frac{y_j - y_{j-1}}{h} + \frac{h}{2}f''(\eta).$$

Es gilt also für den Fehler sowohl des vorwärtsgenommenen als auch des rückwärtsgenommenen Differenzenquotienten die **Asymptotik $O(h)$** .

Numerische Differentiation

Definition 3.29: Eine Differenzenapproximation $D_r f(x; h)$ für die Ableitung $f^{(r)}(x)$ mit der Schrittweite h besitzt die **Fehlerordnung** p , falls gilt

$$D_r f(x; h) - f^{(r)}(x) = O(h^p).$$

Numerische Differentiation

Definition 3.29: Eine Differenzenapproximation $D_r f(x; h)$ für die Ableitung $f^{(r)}(x)$ mit der Schrittweite h besitzt die **Fehlerordnung** p , falls gilt

$$D_r f(x; h) - f^{(r)}(x) = O(h^p).$$

Vorwärts- und rückwärtsgenommene Differenzenquotienten zur Approximation von $f'(x)$ besitzen also die Fehlerordnung 1.

Numerische Differentiation

Für $f \in C^4$ gilt

$$f(x_j \pm h) = f(x_j) \pm hf'(x_j) + \frac{h^2}{2}f''(x_j) \pm \frac{h^3}{6}f'''(x_j) + O(h^4),$$

Numerische Differentiation

Für $f \in C^4$ gilt

$$f(x_j \pm h) = f(x_j) \pm hf'(x_j) + \frac{h^2}{2}f''(x_j) \pm \frac{h^3}{6}f'''(x_j) + O(h^4),$$

und daher

$$\frac{y_{j+1} - y_{j-1}}{2h} - f'(x_j) = \frac{h^2}{6}f'''(x_j) + O(h^3).$$

Numerische Differentiation

Für $f \in C^4$ gilt

$$f(x_j \pm h) = f(x_j) \pm hf'(x_j) + \frac{h^2}{2}f''(x_j) \pm \frac{h^3}{6}f'''(x_j) + O(h^4),$$

und daher

$$\frac{y_{j+1} - y_{j-1}}{2h} - f'(x_j) = \frac{h^2}{6}f'''(x_j) + O(h^3).$$

Der zentrale Differenzenquotient besitzt also die **Fehlerordnung 2**.

Numerische Differentiation

Für $f \in C^4$ gilt

$$f(x_j \pm h) = f(x_j) \pm hf'(x_j) + \frac{h^2}{2}f''(x_j) \pm \frac{h^3}{6}f'''(x_j) + O(h^4),$$

und daher

$$\frac{y_{j+1} - y_{j-1}}{2h} - f'(x_j) = \frac{h^2}{6}f'''(x_j) + O(h^3).$$

Der zentrale Differenzenquotient besitzt also die **Fehlerordnung 2**.

Genauso erhält man für die Approximationen in (3.18) und (3.19) die **Fehlerordnungen 4 und 6**.

Numerische Differentiation

Bei Rechnung in exakter Arithmetik beschreibt die **Fehlerordnung** das Verhalten des Fehlers für $h \rightarrow 0$.

Numerische Differentiation

Bei Rechnung in exakter Arithmetik beschreibt die **Fehlerordnung** das Verhalten des Fehlers für $h \rightarrow 0$.

Die Differenzenformeln enthalten alle Differenzen von Funktionswerten von f , die bei kleinem h **nahe beieinander** liegen.

Numerische Differentiation

Bei Rechnung in exakter Arithmetik beschreibt die **Fehlerordnung** das Verhalten des Fehlers für $h \rightarrow 0$.

Die Differenzenformeln enthalten alle Differenzen von Funktionswerten von f , die bei kleinem h **nahe beieinander** liegen.

Dies führt beim Rechnen mit endlicher Stellenzahl zu **Auslöschungen**.

Numerische Differentiation

Beispiel 3.30: Wir bestimmen für $f(x) := \cos x$ Näherungen für $f'(1)$ mit dem vorwärtsgenommenen Differenzenquotienten und erhalten die folgenden Fehler der Approximationen für $h_j = 10^{-j}$, $j = 0, 1, \dots, 16$.

j	Fehler	j	Fehler	j	Fehler
0	$1.15 \cdot 10^{-1}$	6	$2.70 \cdot 10^{-7}$	12	$7.81 \cdot 10^{-5}$
1	$2.56 \cdot 10^{-2}$	7	$2.81 \cdot 10^{-8}$	13	$7.81 \cdot 10^{-5}$
2	$2.69 \cdot 10^{-3}$	8	$3.03 \cdot 10^{-9}$	14	$2.29 \cdot 10^{-3}$
3	$2.70 \cdot 10^{-4}$	9	$1.30 \cdot 10^{-7}$	15	$1.58 \cdot 10^{-1}$
4	$2.70 \cdot 10^{-5}$	10	$3.52 \cdot 10^{-7}$	16	$8.41 \cdot 10^{-1}$
5	$2.70 \cdot 10^{-6}$	11	$3.52 \cdot 10^{-7}$		

Numerische Differentiation

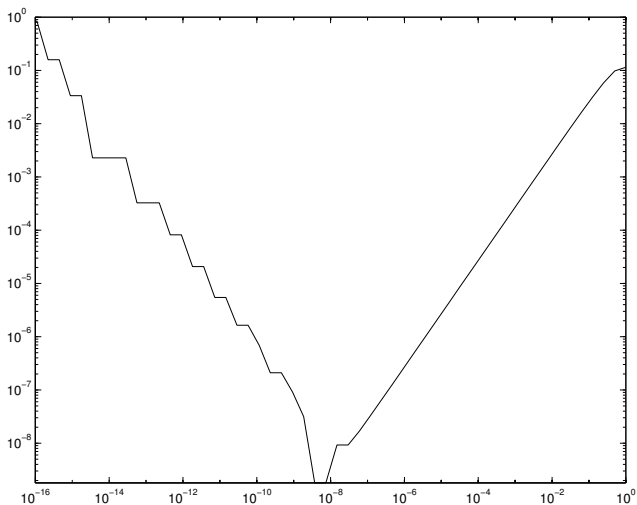
Beispiel 3.30: Wir bestimmen für $f(x) := \cos x$ Näherungen für $f'(1)$ mit dem vorwärtsgenommenen Differenzenquotienten und erhalten die folgenden Fehler der Approximationen für $h_j = 10^{-j}$, $j = 0, 1, \dots, 16$.

j	Fehler	j	Fehler	j	Fehler
0	$1.15 \cdot 10^{-1}$	6	$2.70 \cdot 10^{-7}$	12	$7.81 \cdot 10^{-5}$
1	$2.56 \cdot 10^{-2}$	7	$2.81 \cdot 10^{-8}$	13	$7.81 \cdot 10^{-5}$
2	$2.69 \cdot 10^{-3}$	8	$3.03 \cdot 10^{-9}$	14	$2.29 \cdot 10^{-3}$
3	$2.70 \cdot 10^{-4}$	9	$1.30 \cdot 10^{-7}$	15	$1.58 \cdot 10^{-1}$
4	$2.70 \cdot 10^{-5}$	10	$3.52 \cdot 10^{-7}$	16	$8.41 \cdot 10^{-1}$
5	$2.70 \cdot 10^{-6}$	11	$3.52 \cdot 10^{-7}$		

Der Fehler **fällt** also (wie durch die Fehlerordnung 1 vorausgesagt) zunächst bis $h = 10^{-8}$ linear, **steigt** aber danach durch Auslöschung in der Differenzenformel wieder an.

Numerische Differentiation

Abbildung 5: Differenzenformel Ordnung 1



Numerische Differentiation

Das Verhalten im letzten Beispiel kann man auf folgende Weise erklären.

Numerische Differentiation

Das Verhalten im letzten Beispiel kann man auf folgende Weise erklären. Wir nehmen an, dass der errechnete Funktionswert $\tilde{y}_j \approx y_j$ die Größe

$$\tilde{y}_j = y_j(1 + \delta_j), \quad |\delta_j| \leq K\mathbf{u} \quad (3.20)$$

hat, wobei \mathbf{u} die Maschinengenauigkeit bezeichnet und K eine „kleine“ Konstante ist.

Numerische Differentiation

Das Verhalten im letzten Beispiel kann man auf folgende Weise erklären. Wir nehmen an, dass der errechnete Funktionswert $\tilde{y}_j \approx y_j$ die Größe

$$\tilde{y}_j = y_j(1 + \delta_j), \quad |\delta_j| \leq K\mathbf{u} \quad (3.20)$$

hat, wobei \mathbf{u} die Maschinengenauigkeit bezeichnet und K eine „kleine“ Konstante ist. Dann gilt für den errechneten vorwärtsgenommenen Differenzenquotienten

$$\tilde{D}_1(h) := \text{fl} \left(\frac{\tilde{y}_{j+1} - \tilde{y}_j}{h} \right) = \frac{\tilde{y}_{j+1} - \tilde{y}_j}{h} (1 + \varepsilon_1)(1 + \varepsilon_2), \quad |\varepsilon_1|, |\varepsilon_2| \leq \mathbf{u}.$$

Numerische Differentiation

Das Verhalten im letzten Beispiel kann man auf folgende Weise erklären. Wir nehmen an, dass der errechnete Funktionswert $\tilde{y}_j \approx y_j$ die Größe

$$\tilde{y}_j = y_j(1 + \delta_j), \quad |\delta_j| \leq K\mathbf{u} \quad (3.20)$$

hat, wobei \mathbf{u} die Maschinengenauigkeit bezeichnet und K eine „kleine“ Konstante ist. Dann gilt für den errechneten vorwärtsgenommenen Differenzenquotienten

$$\tilde{D}_1(h) := \text{fl} \left(\frac{\tilde{y}_{j+1} - \tilde{y}_j}{h} \right) = \frac{\tilde{y}_{j+1} - \tilde{y}_j}{h} (1 + \varepsilon_1)(1 + \varepsilon_2), \quad |\varepsilon_1|, |\varepsilon_2| \leq \mathbf{u}.$$

Unter Verwendung von (3.20) erhält man, wenn man Terme der Größenordnung \mathbf{u}^2 vernachlässigt,

$$\tilde{D}_1(h) = \frac{y_{j+1} - y_j}{h} + \frac{y_{j+1}\delta_{j+1} - y_j\delta_j}{h} + \frac{y_{j+1} - y_j}{h}(\varepsilon_1 + \varepsilon_2),$$

Numerische Differentiation

und daher ist der Rundungsfehler bei der Auswertung der Differenzenformel $D_1(h) := (y_{j+1} - y_j)/h$ mit Konstanten C_1, C_2

$$\begin{aligned} |\tilde{D}_1(h) - D_1(h)| &= \left| \frac{y_{j+1}\delta_{j+1} - y_j\delta_j}{h} + \frac{y_{j+1} - y_j}{h}(\varepsilon_1 + \varepsilon_2) \right| \\ &\leq \frac{C_1}{h}\mathbf{u} + C_2\mathbf{u}. \end{aligned} \tag{3.21}$$

Numerische Differentiation

und daher ist der Rundungsfehler bei der Auswertung der Differenzenformel $D_1(h) := (y_{j+1} - y_j)/h$ mit Konstanten C_1, C_2

$$\begin{aligned} |\tilde{D}_1(h) - D_1(h)| &= \left| \frac{y_{j+1}\delta_{j+1} - y_j\delta_j}{h} + \frac{y_{j+1} - y_j}{h}(\varepsilon_1 + \varepsilon_2) \right| \\ &\leq \frac{C_1}{h}\mathbf{u} + C_2\mathbf{u}. \end{aligned} \quad (3.21)$$

Da die vorwärtsgenommene Differenzenformel die Ordnung 1 hat, gibt es eine Konstante C_3 mit

$$|D_1(h) - f'(x_j)| \leq C_3h,$$

Numerische Differentiation

und daher ist der Rundungsfehler bei der Auswertung der Differenzenformel $D_1(h) := (y_{j+1} - y_j)/h$ mit Konstanten C_1, C_2

$$\begin{aligned} |\tilde{D}_1(h) - D_1(h)| &= \left| \frac{y_{j+1}\delta_{j+1} - y_j\delta_j}{h} + \frac{y_{j+1} - y_j}{h}(\varepsilon_1 + \varepsilon_2) \right| \\ &\leq \frac{C_1}{h}\mathbf{u} + C_2\mathbf{u}. \end{aligned} \quad (3.21)$$

Da die vorwärtsgenommene Differenzenformel die Ordnung 1 hat, gibt es eine Konstante C_3 mit

$$|D_1(h) - f'(x_j)| \leq C_3h,$$

und daher folgt für den **Gesamtfehler**

$$|\tilde{D}_1(h) - f'(x_j)| \leq \frac{C_1}{h}\mathbf{u} + C_2\mathbf{u} + C_3h =: \Delta(h). \quad (3.22)$$

Numerische Differentiation

Der Graph dieser Funktion hat die Gestalt der Fehlerfunktion in Abbildung 5.

Numerische Differentiation

Der Graph dieser Funktion hat die Gestalt der Fehlerfunktion in Abbildung 5.

Mit fallendem h fällt die Funktion $\Delta(h)$ bis zum Minimum, das durch

$$\Delta'(h) = -\frac{C_1 \mathbf{u}}{h^2} + C_3 = 0$$

charakterisiert ist, d.h. bis

$$h_{\text{opt}} = \sqrt{\frac{C_1}{C_3}} \cdot \sqrt{\mathbf{u}}, \quad (3.23)$$

und steigt danach wieder.

Numerische Differentiation

Der Graph dieser Funktion hat die Gestalt der Fehlerfunktion in Abbildung 5.

Mit fallendem h fällt die Funktion $\Delta(h)$ bis zum Minimum, das durch

$$\Delta'(h) = -\frac{C_1 \mathbf{u}}{h^2} + C_3 = 0$$

charakterisiert ist, d.h. bis

$$h_{\text{opt}} = \sqrt{\frac{C_1}{C_3}} \cdot \sqrt{\mathbf{u}}, \quad (3.23)$$

und steigt danach wieder.

In MATLAB ist $\mathbf{u} \approx 10^{-16}$, das Minimum des Fehlers muss also in der Größenordnung von 10^{-8} liegen. Abbildung 5, zeigt, dass dies tatsächlich bei Beispiel 3.30 der Fall ist.

Numerische Differentiation

Besitzt die Differenzenformel D_1 die Fehlerordnung m , so bleibt (3.21) richtig, und man erhält entsprechend (3.22) den Gesamtfehler

$$|\tilde{D}_1(h) - f'(x_0)| \leq \frac{C_1}{h} \mathbf{u} + C_2 \mathbf{u} + C_3 h^m =: \Delta(h). \quad (3.24)$$

Numerische Differentiation

Besitzt die Differenzenformel D_1 die **Fehlerordnung** m , so bleibt (3.21) richtig, und man erhält entsprechend (3.22) den Gesamtfehler

$$|\tilde{D}_1(h) - f'(x_0)| \leq \frac{C_1}{h} \mathbf{u} + C_2 \mathbf{u} + C_3 h^m =: \Delta(h). \quad (3.24)$$

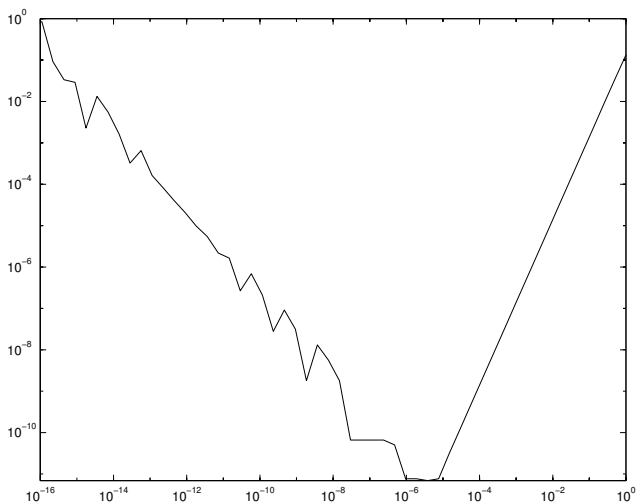
In diesem Fall erhält man als Größenordnung der **optimalen Schrittweite**

$$h_{\text{opt}} = C \mathbf{u}^{1/(m+1)},$$

die ebenfalls durch die Beispiele in Abbildung 6, Abbildung 7 und Abbildung 8 bestätigt werden.

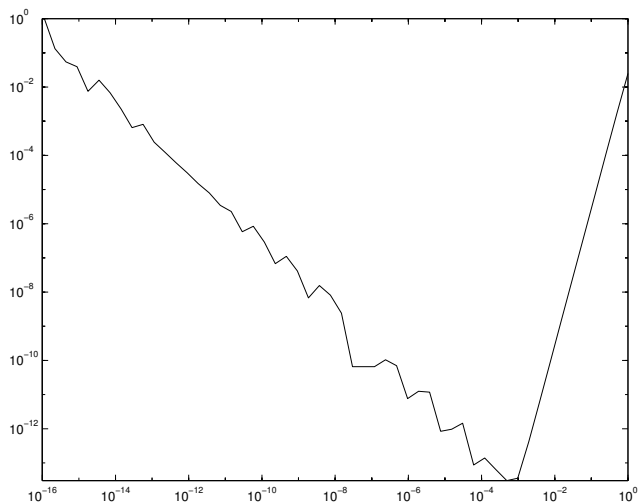
Numerische Differentiation

Abbildung 6: Differenzenformel Ordnung 2 ($\sqrt[3]{10^{-16}} \approx 10^{-5.333}$)



Numerische Differentiation

Abbildung 7: Differenzenformel Ordnung 4 ($\sqrt[5]{10^{-16}} \approx 10^{-3.200}$)



Numerische Differentiation

Abbildung 8: Differenzenformel Ordnung 6 ($\sqrt[7]{10^{-16}} \approx 10^{-2.286}$)

