1. NUMERISCHE INTEGRATION

In diesem Kapitel betrachten wir das Problem, für eine gegebene Funktion $f : [a, b] \to \mathbb{R}$ das Integral

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

zu berechnen oder zu approximieren. Hat die Funktion f keine Stammfunktion oder ist f sogar nur durch Werte an einigen wenigen Punkten im Intervall $[a, b]$ gegeben, so muß man sich im Allgemeinen damit begnügen, das Integral näherungsweise zu berechnen.

Die Ziele in diesem Kapitel sind die Herleitung guter Approximationen und deren Fehler- und Konvergenzanalyse sowie die Implementierung in einer Programmiersprache wie C, Fortran oder Matlab.

1.1 Erste Beispiele von Quadraturformeln

Um das Integral über das Intervall $[a, b]$ zu berechnen, ist es häufig nötig, das Intervall in kleinere Teilintervalle $[x_{j-1}, x_j]$ zu zerlegen, wobei $x_0 = a < x_1 < \cdots < x_N = b$ gegeben sind:

$$\int_a^b f(x)dx = \sum_{j=1}^{N} \int_{x_{j-1}}^{x_j} f(x)dx = \sum_{j=1}^{N} h_j \int_0^1 f(x_{j-1} + th_j)dt,$$

wobei $h_j = x_j - x_{j-1}$ die Intervalllängen bezeichnet. Setzen wir $g(t) := f(x_{j-1} + th_j)$, so verbleibt das Problem, eine Approximation an

$$\int_0^1 g(t)dt$$

zu berechnen. Wir betrachten zunächst einige naheliegende Approximationen.

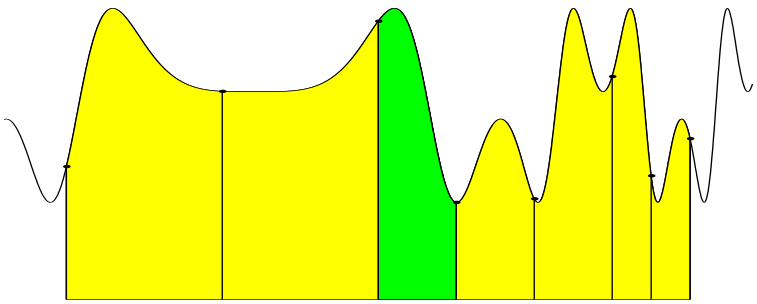


Abb. 1.1: Aufteilung des Integrationsintervalls

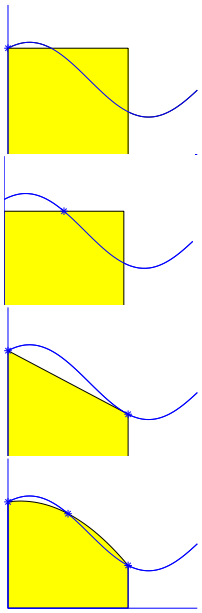
Beispiel.

Rechtecksregel $\int_0^1 g(t)dt \approx g(0)$

Mittelpunktsregel $\int_0^1 g(t)dt \approx g(\frac{1}{2})$

Trapezregel $\int_0^1 g(t)dt \approx \frac{1}{2}\big[g(0)+g(1)\big]$

Simpsonregel $\int_0^1 g(t)dt \approx \frac{1}{6}\big[g(0)+4g(\frac{1}{2})+g(1)\big]$



Die Simpsonregel erhält man dadurch, dass man eine Parabel durch die Punkte $(0, g(0))$, $(1/2, g(1/2))$ und $(1, g(1))$ legt und das Integral durch das Integral über die Parabel approximiert (Übung).

Eine Verallgemeinerung dieser Konstruktion (wähle s äquidistante Punkte $i/(s-1)$, $i = 0, \ldots, s-1$ im Intervall $[0,1]$ und lege durch diese ein Polynom vom Grad $s-1$) führt auf die **Newton-Cotes-Formeln** (Newton 1680, Cotes 1711). $\diamond$

Definition 1.1. *Eine s-stufige Quadraturformel $(b_j, c_j)_{j=1}^s$ ist von der Form*

$$\int_0^1 g(t)dt \approx \sum_{j=1}^{s} b_j g(c_j).$$

Die Koeffizienten b_j heißen **Gewichte** *und die Werte c_j* **Knoten** *der Quadraturformel.*

In obigen Beispielen sind die Gewichte und Knoten wie folgt:

Rechtecksregel	$s=1,$	$c_1 = 0,$	$b_1 = 1$
Mittelpunktsregel	$s=1,$	$c_1 = \frac{1}{2},$	$b_1 = 1$
Trapezregel	$s=2,$	$c_1 = 0,\ c_2 = 1,$	$b_1 = b_2 = \frac{1}{2},$
Simpsonregel	$s=3,$	$c_1 = 0,\ c_2 = \frac{1}{2},\ c_3 = 1,$	$b_2 = \frac{4}{6},\ b_1 = b_3 = \frac{1}{6}$

Die Approximation des Integrals über $[a,b]$ hat die Form

$$\begin{aligned}\int_a^b f(x)dx &= \sum_{j=1}^{N} h_j \int_0^1 f(x_{j-1} + t h_j)dt \\ &\approx \sum_{j=1}^{N} h_j \sum_{i=1}^{s} b_i f(x_{j-1} + c_i h_j).\end{aligned}$$

Bei der Implementierung sollte man darauf achten, mit einer minimalen Anzahl von Funktionsauswertungen auszukommen, da dies in der Regel der wesentliche Aufwand bei der Approximation ist. So kann man etwa bei der Trapez- und der Simpsonregel Funktionswerte an den Grenzen benachbarter Teilintervalle zweimal verwenden.

1.2 Ordnung einer Quadraturformel

In diesem Abschnitt werden wir Kriterien für die Approximationsgüte einer Quadraturformel herleiten.

Definition 1.2. *Eine Quadraturformel* $(b_j, c_j)_{j=1}^s$ *hat* **Ordnung** p *genau dann, wenn sie exakt ist für alle Polynome vom Grad höchstens* $p-1$.

Im Folgenden bezeichne $\mathcal{P}_m$ den Raum aller Polynome vom Grad höchstens m.

Lemma 1.3. *Eine Quadraturformel* $(b_j, c_j)_{j=1}^s$ *hat Ordnung* p *genau dann, wenn*

$$\sum_{j=1}^{s} b_j c_j^{q-1} = \frac{1}{q}, \qquad \text{für} \qquad q = 1, \dots, p.$$

Beweis. Ist $g \in \mathcal{P}_{p-1}$, dann ist nach dem Satz von Taylor

$$g(t) = \sum_{q=1}^{p} t^{q-1} \frac{g^{(q-1)}(0)}{(q-1)!}$$

und somit

$$\sum_{j=1}^{s} b_j g(c_j) = \sum_{q=1}^{p} \frac{g^{(q-1)}(0)}{(q-1)!} \sum_{j=1}^{s} b_j c_j^{q-1}.$$

Andererseits gilt

$$\int_0^1 g(t)dt = \sum_{q=1}^{p} \frac{g^{(q-1)}(0)}{(q-1)!} \underbrace{\int_0^1 t^{q-1} dt}_{1/q}.$$

Daraus folgt die Behauptung. □

Die Beispiele aus dem vorigen Abschnitt haben folgende Ordnung:

Rechtecksregel	$s=1,$	$p=1$
Mittelpunktsregel	$s=1,$	$p=2$
Trapezregel	$s=2,$	$p=2$
Simpsonregel	$s=3,$	$p=4$

Die Simpsonregel hat nicht die Ordnung fünf, da die Bedingung aus Lemma 1.3 für $p=5$ nicht erfüllt ist: $\frac{5}{24} \neq \frac{1}{5}$. In allen Beispielen ist stets $p \geq s$, in zweien sogar $p > s$. Wir werden später in Satz 1.6 zeigen, dass es zu beliebigen s Knoten immer eine Quadraturformel mit $p \geq s$ gibt.

Definition 1.4. *Eine Quadraturformel heißt* **symmetrisch** *genau dann, wenn*

$$c_j = 1 - c_{s+1-j}, \qquad b_j = b_{s+1-j} \qquad \text{für alle } j = 1, \dots, s.$$

Bei einer symmetrischen Quadraturformel sind die Knoten symmetrisch zu $1/2$ und die Gewichte zu Knoten, die denselben Abstand zu $1/2$ haben stimmen überein. Mittelpunkts-, Trapez- und Simpsonregel sind symmetrisch.

Satz 1.5. *Die Ordnung einer symmetrischen Quadraturformel ist gerade.*

Beweis. Wir nehmen an, dass die Quadraturformel die Ordnung $2m-1$ hat, also exakt ist für Polynome vom Grad $\leq 2m-2$, und wählen ein beliebiges $g \in \mathcal{P}_{2m-1}$. Dann werden wir zeigen, dass die Quadraturformel auch für g exakt ist. g können wir folgendermaßen darstellen:

$$g(t) = c\left(t - \frac{1}{2}\right)^{2m-1} + \widetilde{g}(t), \qquad \widetilde{g} \in \mathcal{P}_{2m-2}, \quad c \in \mathbb{R}.$$

Aus Symmetriegründen ist

$$\int_0^1 \left(t - \frac{1}{2}\right)^{2m-1} dt = 0.$$

Nach Voraussetzung ist die Quadraturformel für $\widetilde{g}$ exakt und wegen der Symmetrie der Quadraturformel gilt

$$\sum_{j=1}^{s} b_j \left(c_j - \frac{1}{2}\right)^{2m-1} = \sum_{j=1}^{s} b_{s+1-j} \left(\frac{1}{2} - c_{s+1-j}\right)^{2m-1} = -\sum_{j=1}^{s} b_j \left(c_j - \frac{1}{2}\right)^{2m-1},$$

die Summe ist also Null. Die Quadraturformel integriert die Funktion g damit exakt. □

Satz 1.6. *Seien Knoten $c_1 < \ldots < c_s$ gegeben. Dann existieren eindeutig bestimmte Gewichte $b_1, \ldots, b_s$, so dass die Quadraturformel $(b_j, c_j)_{j=1}^s$ die Ordnung $p \geq s$ hat. Es gilt*

$$b_j = \int_0^1 \ell_j(t)dt,$$

wobei

$$\ell_j(t) = \frac{\prod_{k\neq j}(t - c_k)}{\prod_{k\neq j}(c_j - c_k)}$$

das j-te **Lagrange-Polynom** *ist. Dieses erfüllt*

$$\deg \ell_j = s - 1, \qquad \ell_j(c_m) = \begin{cases} 1, & j = m, \\ 0, & j \neq m. \end{cases}$$

Beweis. Falls $p \geq s$ ist, muss wegen $\deg \ell_j = s - 1$

$$\int_0^1 \ell_k(t)dt = \sum_{j=1}^{s} b_j \ell_k(c_j) = b_k$$

gelten. Sind umgekehrt die b_j wie angegeben definiert, dann ist die Quadraturformel exakt für *alle* Polynome vom Grad $\leq s - 1$, weil $\ell_1, \ldots, \ell_s$ eine Basis von $\mathcal{P}_{s-1}$ bilden und alle Basisfunktionen exakt integriert werden. □

1.3 Integrale mit Gewichtsfunktion

Etwas allgemeiner kann man analog zum vorigen Abschnitt auch Integrale der Form

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

betrachten. Hierbei sei eine positive **Gewichtsfunktion**

$$\omega : (a,b) \to \mathbb{R} \quad \text{stetig}, \qquad \omega(x) > 0 \quad \forall x \in (a,b)$$

gegeben.

Der Einfachheit halber sei in diesem Abschnitt $(a,b) = (0,1)$. Eine Quadraturformel $(b_j, c_j)_{j=1}^s$ ist dann durch

$$\int_0^1 \omega(t) g(t) dt \approx \sum_{j=1}^s b_j g(c_j) \tag{1.1}$$

gegeben und es gilt die folgende Verallgemeinerung von Lemma 1.3

Korollar 1.7. *Eine Quadraturformel $(b_j, c_j)_{j=1}^s$ zur Approximation von (1.1) hat Ordnung p genau dann, wenn*

$$\sum_{j=1}^s b_j c_j^{q-1} = I_{q-1} := \int_0^1 \omega(t) t^{q-1} dt, \text{für} \qquad q = 1, \ldots, p.$$

Beweis. Übung. □

Beispiel. Für die Gewichtsfunktion $\omega(t) = 1/\sqrt{t}$ und die Knoten $c_1 = 0$, $c_2 = 1$ ergeben sich die Ordnungsbedingungen:

$$I_0 = \int_0^1 \frac{1}{\sqrt{t}} \cdot 1 dt = 2,$$
$$I_1 = \int_0^1 \frac{1}{\sqrt{t}} \cdot t dt = \int_0^1 \sqrt{t} dt = \frac{2}{3}.$$

Das führt auf das folgende lineare Gleichungssystem

$$\begin{array}{lccccccc} q=1: & 1 \cdot b_1 & + & b_2 & = & I_0 & = & 2 \\ q=2: & 0 \cdot b_1 & + & b_2 & = & I_1 & = & \frac{2}{3} \end{array}$$

mit der Lösung $b_1 = 4/3$ und $b_2 = 2/3$. Damit ergibt sich die Quadraturformel:

$$\int_0^1 \frac{1}{\sqrt{t}} g(t) dt \approx \frac{4}{3} g(0) + \frac{2}{3} g(1).$$

Wegen

$$b_2 c_2^2 = \frac{2}{3} \neq I_2 = \frac{2}{5}$$

hat die Quadraturformel die Ordnung zwei. ⋄

1.4 Untersuchung des Quadraturfehlers

Um den Fehler zu untersuchen, betrachten wir das Fehlerfunktional

$$E(g) = \int_0^1 g(t) dt - \sum_{j=1}^s b_j g(c_j).$$

Satz 1.8. *Hat die Quadraturformel (mindestens) Ordnung p und ist die Funktion $g : [0,1] \to \mathbb{R}$ (mindestens) p-mal stetig differenzierbar, dann gilt*

$$E(g) = \int_0^1 K_p(\tau) g^{(p)}(\tau) d\tau,$$

wobei

$$K_p(\tau) = E\big(t \mapsto \sigma_{p,\tau}(t)\big), \qquad \sigma_{p,\tau}(t) = \frac{(t-\tau)_+^{p-1}}{(p-1)!} = \begin{cases} \frac{(t-\tau)^{p-1}}{(p-1)!}, & t > \tau \\ 0, & t \le \tau \end{cases}$$

der **Peano-Kern** *ist.*

Beweis. Wir entwickeln g in eine Taylor-Reihe mit Restglied in Integralform:

$$g(t) = \underbrace{g(0) + g'(0)\frac{t}{1!} + \cdots + g^{(p-1)}(0)\frac{t^{p-1}}{(p-1)!}}_{= q(t), \quad \deg q \le p-1} + \underbrace{\int_0^t \frac{(t-\tau)^{p-1}}{(p-1)!} g^{(p)}(\tau) d\tau}_{= \int_0^1 \sigma_{p,\tau}(t) g^{(p)}(\tau) d\tau}.$$

Wegen der Linearität von E folgt

$$E(g) = E(q) + \int_0^1 E\big(t \mapsto \sigma_{p,\tau}(t)\big) g^{(p)}(\tau) d\tau = \int_0^1 K_p(\tau) g^{(p)}(\tau) d\tau,$$

denn nach Voraussetzung ist die Ordnung der Quadraturformel mindestens p, also $E(q) = 0$. □

Korollar 1.9. *Falls $K_p(\tau) \ge 0$ oder $K_p(\tau) \le 0$ für alle $\tau \in (0,1)$ dann gibt es $\tau^* \in [0,1]$, so dass*

$$E(g) = \int_0^1 K_p(\tau) g^{(p)}(\tau) d\tau = g^{(p)}(\tau^*) \int_0^1 K_p(\tau) d\tau.$$

Beweis. Folgt direkt aus dem Mittelwertsatz der Integralrechnung. □

Das Integral

$$d_p := \int_0^1 K_p(\tau) d\tau$$

lässt sich leicht berechnen, indem man $g(\tau) = \tau^p$ in Korollar 1.9 setzt. Dann gilt nämlich $g^{(p)}(\tau^*) = p!$, so dass

$$d_p = \frac{1}{p!}\Big(\int_0^1 t^p dt - \sum_{j=1}^s b_j c_j^p\Big) = \frac{1}{p!}\Big(\frac{1}{p+1} - \sum_{j=1}^s b_j c_j^p\Big).$$

Beispiel 1.1. Es gilt

$$d_2^M = \frac{1}{2}\Big(\frac{1}{3} - \frac{1}{4}\Big) = \frac{1}{24}, \qquad d_2^T = \frac{1}{2}\Big(\frac{1}{3} - \frac{1}{2}\Big) = -\frac{1}{12}$$

für die Mittelpunkts- bzw. die Trapezregel. ◇

Nach Definition von E gilt für den Peano-Kern

$$K_p(\tau) = \frac{(1-\tau)^p}{p!} - \sum_{j=1}^s b_j \frac{(c_j - \tau)_+^{p-1}}{(p-1)!},$$

denn

$$\int_0^1 \frac{(t-\tau)_+^{p-1}}{(p-1)!}dt = \int_\tau^1 \frac{(t-\tau)^{p-1}}{(p-1)!}dt = \frac{(1-\tau)^p}{p!}.$$

Wenden wir Satz 1.8 auf $g(t) = f(x_0+th)$ an, so ergibt sich wegen $g^{(p)}(t) = h^p f^{(p)}(x_0+th)$ für ein Integrationsintervall der Länge h

$$\int_{x_0}^{x_0+h} f(x)dx - h\sum_{j=1}^{s} b_j f(x_0+c_jh) = hE(g)$$
$$= h^{p+1}\int_0^1 K_p(\tau) f^{(p)}(x_0+\tau h)d\tau.$$

Korollar 1.10. *Hat die Quadraturformel (mindestens) Ordnung p und ist die Funktion $f : [x_0, x_0+h] \to \mathbb{R}$ (mindestens) p-mal stetig differenzierbar, dann gilt*

$$\left|\int_{x_0}^{x_0+h} f(x)dx - h\sum_{j=1}^{s} b_j f(x_0+c_jh)\right| \le h^{p+1}\int_0^1 |K_p(\tau)|d\tau \cdot \max_{x\in[x_0,x_0+h]} |f^{(p)}(x)|$$
$$= const\ h^{p+1} \max_{x\in[x_0,x_0+h]} |f^{(p)}(x)|.$$

Für die in Abschnitt 1.1 angegebenen Beispiele gilt:

Rechtecksregel	$\int_0^1 \lvert K_1(\tau)\rvert d\tau = \frac{1}{2}$
Mittelpunktsregel	$\int_0^1 \lvert K_2(\tau)\rvert d\tau = \frac{1}{24}$
Trapezregel	$\int_0^1 \lvert K_2(\tau)\rvert d\tau = \frac{1}{12}$
Simpsonregel	$\int_0^1 \lvert K_4(\tau)\rvert d\tau = \frac{1}{2880}$.

Korollar 1.11. *Hat die Quadraturformel (mindestens) Ordnung p und ist die Funktion $f : [a,b] \to \mathbb{R}$ (mindestens) p-mal stetig differenzierbar, dann gilt*

$$\left|\int_a^b f(x)dx - \sum_{j=1}^{N} h_j \sum_{i=1}^{s} b_i f(x_{j-1}+c_ih_j)\right| \le (b-a)h^p\,\epsilon_p,$$

wobei

$$h = \max_j h_j, \qquad \epsilon_p = \int_0^1 |K_p(\tau)|d\tau \cdot \max_{x\in[a,b]} |f^{(p)}(x)|.$$

Beweis. Wir zerlegen das Intervall $[a,b]$ in N Teilintervalle und erhalten

$$\int_a^b f(x)dx = \sum_{j=1}^{N}\int_{x_{j-1}}^{x_j} f(x)dx.$$

Nach Korollar 1.10 gilt für die Integrale über die Teilintervalle

$$\left|\int_{x_{j-1}}^{x_j} f(x)dx - h_j\sum_{i=1}^{s} b_i f(x_{j-1}+c_ih_j)\right| \le h_j^{p+1}\epsilon_p \le h_j h^p \epsilon_p$$

und die Behauptung folgt direkt aus der Dreiecksungleichung und $\sum_{j=1}^{N} h_j = b-a$. □

1.5 Quadraturformeln mit erhöhter Ordnung

Falls beliebige Knoten $c_1 < \ldots < c_s$ gegeben sind, dann wissen wir aus Satz 1.6, dass dazu eine Quadraturformel der Ordnung $p \geq s$ existiert. Eine naheliegende Frage ist dann, ob man die Knoten c_i so wählen kann, dass sogar $p > s$ gilt. In manchen Fällen ist das möglich, wie wir bei der Mittelpunkts- und der Simpsonregel bereits gesehen haben. Wir werden jetzt zeigen, welche Ordnung man damit höchstens erreichen kann und wie man dazu die Knoten wählen muss.

Das Ziel ist es, eine Quadraturformel mit der Ordnung $p = s + m$ mit $m \geq 1$ zu konstruieren. Das bedeutet, dass alle Polynome f vom Grad $\leq s + m - 1$ exakt integriert werden. Wir dividieren dazu f durch das Knotenpolynom

$$M(t) = \prod_{j=1}^{s} (t - c_j)$$

vom Grad s:

$$f(t) = M(t)g(t) + r(t), \qquad \deg r \leq s - 1.$$

Satz 1.12. *Sei $(b_i, c_i)_{i=1}^s$ eine Quadraturformel der Ordnung $p \geq s$. Dann hat die Quadraturformel die Ordnung $s + m$ genau dann, wenn*

$$\int_0^1 M(t)g(t)dt = 0 \quad \text{für alle } g \in \mathcal{P}_{m-1} \tag{1.2}$$

gilt.

Beweis. Wegen $p \geq s$ und $r \in \mathcal{P}_{s-1}$ gilt

$$\begin{aligned} \int_0^1 f(t)dt &= \int_0^1 M(t)g(t)dt + \int_0^1 r(t)dt \\ &= \int_0^1 M(t)g(t)dt + \sum_{i=1}^{s} b_i r(c_i). \end{aligned}$$

Andererseits folgt aus $M(c_i) = 0$

$$\sum_{i=1}^{s} b_i f(c_i) = \sum_{i=1}^{s} b_i M(c_i)g(c_i) + \sum_{i=1}^{s} b_i r(c_i) = \sum_{i=1}^{s} b_i r(c_i).$$

Also wird f genau dann exakt integriert, wenn (1.2) erfüllt ist. □

Die Bedingung (1.2) besagt, dass M orthogonal ist auf allen Polynomen vom Grad $\leq m-1$ bezüglich des L^2-Skalarprodukts

$$\langle f, g \rangle = \int_0^1 f(t)g(t)dt,$$

d. h. $\langle M, f \rangle = 0$ für alle $f \in \mathcal{P}_{m-1}$.

Satz 1.13. *Die Ordnung einer Quadraturformel mit s Knoten ist höchstens $2s$.*

Beweis. Angenommen, die Ordnung wäre $p > 2s$. Nach Satz 1.12 gilt dann

$$\int_0^1 M(t)g(t)dt = 0 \qquad \text{für alle } g \in \mathcal{P}_s.$$

Insbesondere dürfen wir $g = M$ setzen. Dies ist ein Widerspruch zu

$$\int_0^1 M(t)^2 dt > 0.$$

Die Annahme, $p > 2s$ ist also falsch. □

Bemerkung. Analoge Resultate gelten für Integrale mit Gewichtsfunktion (Übung).

Wir werden in Abschnitt 1.7 Quadraturformeln der maximalen Ordnung $2s$ konstruieren, die sogenannten **Gauß-Quadraturformeln**. Wesentliches Hilfsmittel hierfür sind Orthogonalpolynome.

1.6 Orthogonalpolynome

In diesem Abschnitt werden Orthogonalpolynome bezüglich gewichteter L^2-Skalarprodukte eingeführt und deren Eigenschaften hergeleitet. Dazu sei eine positive **Gewichtsfunktion**

$$\omega : (a, b) \to \mathbb{R} \quad \text{stetig}, \qquad \omega(x) > 0 \quad \forall x \in (a, b)$$

mit endlichen **Momenten**

$$\mu_k := \int_a^b \omega(x)|x|^k dx < \infty \quad \text{für } k = 0, 1, 2, \ldots$$

gegeben. Wir betrachten den Vektorraum

$$V = \{f : [a, b] \to \mathbb{R} \mid f \text{ stetig }, \int_a^b \omega(x)|f(x)|^2 dx < \infty\}.$$

Wegen der Voraussetzung an ω gilt $\mathcal{P}_m \subseteq V$ für alle m. Auf V sei das gewichtete L^2-Skalarprodukt

$$\langle f, g\rangle = \int_a^b \omega(x)f(x)g(x)dx$$

definiert. Dann ist f orthogonal zu g, $f \perp g$ genau dann, wenn $\langle f, g\rangle = 0$.

Satz 1.14. *(Existenz und Eindeutigkeit von Orthogonalpolynomen)*
Es existiert eine, bis auf Skalierung eindeutige, Folge von Polynomen $\phi_0, \phi_1, \ldots$ mit

$$\phi_k \in \mathcal{P}_k \setminus \mathcal{P}_{k-1}, \qquad \langle \phi_k, q\rangle = 0 \text{ für alle } q \in \mathcal{P}_{k-1}.$$

Für beliebig gewählte $\gamma_j \in \mathbb{R} \setminus \{0\}$, $j \geq 0$ können diese Polynome durch folgende Rekursion berechnet werden:

$$\gamma_{k+1}\phi_{k+1}(x) = (x - \alpha_k)\phi_k(x) - \beta_k\phi_{k-1}(x), \quad k \geq 0, \tag{1.3}$$

mit $\phi_0(x) = 1/\gamma_0$, $\phi_{-1}(x) = 0$,

$$\alpha_k = \frac{\langle x\phi_k, \phi_k\rangle}{\langle \phi_k, \phi_k\rangle}, \qquad \beta_k = \gamma_k \frac{\langle \phi_k, \phi_k\rangle}{\langle \phi_{k-1}, \phi_{k-1}\rangle}. \tag{1.4}$$

Beweis. (Gram-Schmidt-Orthogonalisierung)

Wir nehmen an, $\phi_0, \phi_1, \ldots, \phi_k$ seien bereits bekannt und konstruieren daraus ϕ_{k+1}. Nach Voraussetzung ist $\phi_j \in \mathcal{P}_j \setminus \mathcal{P}_{j-1}$, $j = 1, \ldots k$. Daher bilden $\phi_0, \ldots, \phi_k, x\phi_k$ eine Basis von $\mathcal{P}_{k+1}$ und wir können wegen $\gamma_{k+1} \neq 0$

$$\gamma_{k+1}\phi_{k+1}(x) = x\phi_k(x) + \sum_{j=0}^{k} \rho_j \phi_j(x)$$

schreiben. Wegen $\langle \phi_i, \phi_j \rangle = 0$ für $i \neq j$ können wir aus

$$0 = \gamma_{k+1}\langle \phi_{k+1}, \phi_k \rangle = \langle x\phi_k, \phi_k \rangle + \rho_k \langle \phi_k, \phi_k \rangle$$

$\rho_k = -\alpha_k$ schließen. Ferner gilt

$$0 = \gamma_{k+1}\langle \phi_{k+1}, \phi_{k-1} \rangle = \langle x\phi_k, \phi_{k-1} \rangle + \rho_{k-1} \langle \phi_{k-1}, \phi_{k-1} \rangle$$

und aus

$$\begin{aligned} \langle x\phi_k, \phi_{k-1} \rangle &= \langle \phi_k, x\phi_{k-1} \rangle \\ &= \langle \phi_k, \gamma_k \phi_k + q \rangle, \qquad q \in \mathcal{P}_{k-1} \\ &= \gamma_k \langle \phi_k, \phi_k \rangle \end{aligned}$$

folgt

$$\rho_{k-1} = -\frac{\langle x\phi_k, \phi_{k-1} \rangle}{\langle \phi_{k-1}, \phi_{k-1} \rangle} = -\frac{\gamma_k \langle \phi_k, \phi_k \rangle}{\langle \phi_{k-1}, \phi_{k-1} \rangle} = -\beta_k.$$

Für $j \leq k-2$ ist $x\phi_j \in \mathcal{P}_{k-1}$ und daher folgt aus

$$\begin{aligned} 0 &= \gamma_{k+1}\langle \phi_{k+1}, \phi_j \rangle \\ &= \langle x\phi_k, \phi_j \rangle + \rho_j \langle \phi_j, \phi_j \rangle \\ &= \underbrace{\langle \phi_k, x\phi_j \rangle}_{=0} + \rho_j \langle \phi_j, \phi_j \rangle \end{aligned}$$

$\rho_j = 0$ für alle $j \leq k-2$. □

Bemerkungen.

(a) Die meisten klassischen Orthogonalpolynome sind monisch, d. h. sie haben Höchstkoeffizienten eins ($\phi_k(x) = x^k + \ldots$) für alle $k = 0, 1, \ldots$. In diesem Fall ist $\gamma_k = 1$ für alle k. Eine andere häufige Normierung ist so, dass $\langle \phi_k, \phi_k \rangle = 1$ gilt. Die Orthogonalpolynome sind dann sogar Orthonormalpolynome und die Rekursionskoeffizienten vereinfachen sich zu

$$\alpha_k = \langle x\phi_k, \phi_k \rangle, \qquad \beta_k = \gamma_k, \tag{1.5}$$

γ_k so, dass $\langle \phi_k, \phi_k \rangle = 1$.

(b) Die Rekursion (1.3) führt auf den **Lanczos-Algorithmus** zur Konstruktion einer Folge von Orthogonalpolynomen. Diesen Algorithmus werden wir später auch noch für Anwendungen aus der numerischen linearen Algebra kennenlernen. Er kann nämlich zur Lösung großer linearer Gleichungssysteme und für Eigenwertprobleme verwendet werden.

Umgekehrt gilt auch

Satz 1.15. *Durch eine Rekursion der Form (1.3) mit $\beta_k\gamma_k > 0$ für alle $k \geq 1$ wird eine Folge von Orthogonalpolynomen bezüglich eines geeigneten Skalarprodukts auf dem Raum $\mathcal{P}$ aller Polynome definiert.*

Beweis. Wir definieren eine Linearform $\ell : \mathcal{P} \to \mathbb{R}$ durch Vorgabe der Werte auf einer Basis von $\mathcal{P}$:

$$\ell(\phi_k^2) = a_k, \qquad \ell(x\phi_k^2) = b_k, \qquad k \geq 0$$

wobei $a_k, b_k \in \mathbb{R}$ zunächst unbekannt sind. Mit Hilfe der Linearform ℓ ist durch

$$\langle f, g\rangle = \ell(fg)$$

eine symmetrische Bilinearform auf $\mathcal{P} \times \mathcal{P}$ gegeben. Es sei nun $a_0 > 0$ beliebig gewählt. Motiviert durch (1.4) definieren wir rekursiv

$$b_k = a_k\alpha_k, \qquad a_{k+1} = \frac{\beta_{k+1}a_k}{\gamma_{k+1}}, \qquad k \geq 0.$$

Nach Voraussetzung folgt aus $a_0 > 0$ wegen $\beta_k/\gamma_k > 0$ auch $a_k > 0$ für alle $k \geq 1$, also ist $\langle\phi_k, \phi_k\rangle > 0$. Durch Induktion zeigt man nun völlig analog zum Beweis von Satz 1.14, dass $\langle\phi_i, \phi_j\rangle = 0$ für alle $i \neq j$ gilt (Übung).

Man rechnet leicht nach, dass wegen $a_k > 0$ die Bilinearform $\langle\cdot,\cdot\rangle$ tatsächlich ein Skalarprodukt ist. □

Um eine Quadraturformel maximaler Ordnung zu konstruieren, muss man nach Satz 1.12 die Knoten $c_1, \ldots, c_s$ so wählen, dass

$$M(x) = \prod_{j=1}^{s}(x - c_j) = C\phi_s(x), \qquad C \neq 0$$

das Orthogonalpolynom vom Grad s bezüglich $\omega(x) \equiv 1$ und $[a, b] = [0, 1]$ ist. Dies geht natürlich nur, wenn die Nullstellen von ϕ_s auch reell sind. Allgemein gilt:

Satz 1.16. *Sei ϕ_k das k-te Orthogonalpolynom bezüglich des gewichteten L^2-Skalarprodukts $\langle\cdot,\cdot\rangle$. Die Nullstellen von ϕ_k sind reell, einfach und liegen im offenen Intervall (a, b).*

Beweis. Seien $\lambda_1, \ldots, \lambda_r$ die Nullstellen von ϕ_k, die reell sind, im Intervall (a, b) liegen und bei denen ϕ_k das Vorzeichen wechselt. Offensichtlich ist $r \leq k$. Für

$$g(x) = \prod_{j=1}^{r}(x - \lambda_j)$$

gilt

$$0 \neq \int_a^b \omega(x)\phi_k(x)g(x)dx = \langle\phi_k, g\rangle,$$

denn nach Konstruktion wechselt $\phi_k g$ das Vorzeichen in (a, b) nicht. Außerdem ist $\phi_k \perp \mathcal{P}_{k-1}$, also muss $r = \deg g \geq k$ gelten. □

Beispiel. (Klassische Orthogonalpolynome)

Bezeichnung	(a,b)	$\omega(x)$	Name
P_k	$(-1,1)$	1	Legendre-Polynome
T_k	$(-1,1)$	$(1-x^2)^{-1/2}$	Tschebyscheff-Polynome
$P_k^{(\alpha,\beta)}$	$(-1,1)$	$(1-x)^\alpha(1+x)^\beta$	Jacobi-Polynome, $\alpha,\beta>-1$
L_k^α	$(0,\infty)$	$x^\alpha e^{-x}$	Laguerre-Polynome, $\alpha>-1$
H_k	$(-\infty,\infty)$	e^{-x^2}	Hermite-Polynome

Die Legendre-Polynome sind so normiert, dass $P_k(1)=1$ gilt, das k-te Tschebyscheff-Polynom T_k hat Höchstkoeffizient 2^{k-1}, alle anderen Orthogonalpolynome sind monisch. $\diamond$

1.7 Gauß-Quadraturformeln

Satz 1.17. *(Gauß 1814)*
Es existiert eindeutig eine s-stufige Quadraturformel der maximalen Ordnung $2s$. Sie hat die Knoten $c_i=\frac{1}{2}(1+\lambda_i)$, $i=1,\dots,s$, wobei $\lambda_1,\dots,\lambda_s$ die Nullstellen des s-ten Legendre-Polynoms P_s sind. Die Gewichte b_i sind durch Satz 1.6 gegeben.

Beweis. Nach Satz 1.12 hat die Quadraturformel die Ordnung $p=2s$ genau dann, wenn

$$\int_0^1 M(t)g(t)dt=0 \qquad \forall g\in\mathcal{P}_{s-1}, \qquad M(t)=\prod_{i=1}^s(t-c_i).$$

Wir substituieren $x=2t-1$ oder $t=\frac{1}{2}(1+x)$. Dann ist die Bedingung äquivalent zu

$$\int_{-1}^1 M\left(\tfrac{1}{2}(1+x)\right)f(x)dx=0 \qquad \forall f\in\mathcal{P}_{s-1}.$$

Nach Satz 1.14 muss dann $M\left(\frac{1}{2}(1+x)\right)=CP_s(x)$ mit einer Konstanten C gelten, also $c_i=\frac{1}{2}(1+\lambda_i)$. $\square$

Analog erhalten wir, dass die Quadraturformel

$$\int_a^b f(x)\omega(x)dx\approx\sum_{j=1}^s b_i f(\lambda_i)$$

exakt für alle Polynome vom Grad $\le 2s-1$ ist, wenn λ_i, $i=1,\dots,s$, die Nullstellen des s-ten Orthogonalpolynoms ϕ_s bezüglich des gewichteten L^2-Skalarprodukts über (a,b) sind und die Gewichte b_i analog zu den Bedingungen aus Abschnitt 1.1 zu wählen sind.

Satz 1.18. *Die Gewichte von Gauß-Quadraturformeln sind positiv.*

Beweis. Aus Satz 1.6 wissen wir

$$b_j=\int_0^1\ell_j(t)dt, \qquad \ell_j(t)=\frac{\prod_{k\neq j}(t-c_k)}{\prod_{k\neq j}(c_j-c_k)}.$$

Für die Lagrange-Polynome gilt $\deg\ell_j=s-1$, $\ell_j(c_i)=1$ für $i=j$ und $\ell_j(c_i)=0$ für $i\neq j$. Da die Quadraturformel die Ordnung $p=2s$ und $\deg\ell_j^2=2(s-1)<2s$ gilt, haben wir auch

$$0<\int_0^1\ell_j^2(t)dt=\sum_{i=1}^s b_i\ell_j^2(c_i)=b_j. \quad \square$$

Beispiel. Gauß-Quadraturformeln der Ordnung $2, 4, 6$:

(a) $s = 1$: $P_1(x) = x$, $\lambda_1 = 0$, $c_1 = \frac{1}{2}$, $b_1 = 1$: Mittelpunktsregel, $p = 2$

(b) $s = 2$: $P_2(x) = \frac{3}{2}x^2 - \frac{1}{2}$, $\lambda_{1,2} = \pm\frac{\sqrt{3}}{3}$, $c_{1,2} = \frac{1}{2} \mp \frac{\sqrt{3}}{6}$, $b_1 = b_2 = \frac{1}{2}$. Die Quadraturformel

$$\int_0^1 f(t)dt \approx \frac{1}{2} f\left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right) + \frac{1}{2} f\left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)$$

hat die Ordnung 4.

(c) $s = 3$: $P_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x$, $\lambda_2 = 0$, $\lambda_{1,3} = \pm\frac{\sqrt{15}}{5}$, $c_2 = \frac{1}{2}$, $c_{1,3} = \frac{1}{2} \mp \frac{\sqrt{15}}{10}$. Wegen der Symmetrie und der Eindeutigkeit der Gauß-Quadraturformel muss $b_1 = b_3$ sein. Da die Quadraturformel exakt ist für Polynome bis zum Grad 5, erhalten wir durch Integration von 1 und $(t - \frac{1}{2})^2$

$$2b_1 + b_2 = \int_0^1 1dt = 1,$$
$$b_1 \left(\frac{\sqrt{15}}{10}\right)^2 2 = \int_0^1 \left(t - \frac{1}{2}\right)^2 dt = \frac{1}{12}.$$

Daraus folgt $b_1 = b_3 = \frac{5}{18}$, $b_2 = \frac{8}{18}$. Die Quadraturformel

$$\int_0^1 f(t)dt \approx \frac{5}{18} f\left(\frac{1}{2} - \frac{\sqrt{15}}{10}\right) + \frac{8}{18} f\left(\frac{1}{2}\right) + \frac{5}{18} f\left(\frac{1}{2} + \frac{\sqrt{15}}{10}\right)$$

hat die Ordnung 6.

◇

1.8 Numerische Berechnung von Gauß-Quadraturformeln

Um Knoten und Gewichte von Gauß-Quadraturformeln numerisch zu berechnen, verwendet man die enge Beziehung zwischen den Nullstellen von Polynomen ϕ_k, die durch eine 3-Term-Rekursion (1.3) definiert sind, und den Eigenwerten einer geeigneten Tridiagonalmatrix T_n. Es seien hierzu ϕ_k die Orthogonalpolynome bezüglich des gewichteten L^2-Skalarprodukts über (a, b) mit der Normierung $\langle \phi_k, \phi_k \rangle = 1$. Dann gilt nach (1.5) und Satz 1.14

$$x\phi_k(x) = \beta_k \phi_{k-1}(x) + \alpha_k \phi_k(x) + \beta_{k+1}\phi_{k+1}(x), \quad k \geq 0, \tag{1.6}$$

oder in Matrixschreibweise

$$x\Phi_n(x) = T_n \Phi_n(x) + \beta_n \phi_n(x) e_n, \qquad \Phi_n(x) = \begin{bmatrix} \phi_0 & \dots & \phi_{n-1} \end{bmatrix}^T. \tag{1.7}$$

Hierbei bezeichnet e_n den n-ten Einheitsvektor und T_n die symmetrische Tridiagonalmatrix

$$T_n = \begin{bmatrix} \alpha_0 & \beta_1 & & \\ \beta_1 & \alpha_1 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_{n-1} \end{bmatrix}.$$

Nach Voraussetzung ist $\beta_j \neq 0$ für alle j.

Satz 1.19. *$\lambda \in \mathbb{R}$ ist Nullstelle von ϕ_n definiert durch die Rekursion (1.6) mit $\beta_k \neq 0$ für alle $1 \leq k \leq n-1$, genau dann, wenn λ Eigenwert der Tridiagonalmatrix T_n ist. Der Vektor $\Phi_n(\lambda)$ ist Eigenvektor zum Eigenwert λ.*

Beweis. Sei λ Nullstelle von ϕ_n. Dann gilt nach (1.7)

$$\lambda \Phi_n(\lambda) = T_n \Phi_n(\lambda).$$

Wegen $\phi_0 \neq 0$ ist $\Phi_n(\lambda)$ nicht der Nullvektor, also ist $\Phi_n(\lambda)$ Eigenvektor von T_n zum Eigenwert λ. Da nach Satz 1.16 alle Nullstellen von ϕ_n reell und einfach sind, sind dadurch alle Eigenwerte von T_n gegeben. □

Bemerkung. Kombinieren wir diesen Satz mit Satz 1.15 und 1.16, so haben wir auch gezeigt, dass alle Eigenwerte einer symmetrischen Tridiagonalmatrix mit von Null verschiedenen Nebendiagonalelementen einfach sind.

Satz 1.20. *Bezeichnen wir mit q_{1j}, $j = 1, \ldots, s$ die ersten Komponenten der normierten Eigenvektoren von T_s, dann sind die Gewichte der Gauß-Quadraturformel gegeben durch*

$$b_j = \frac{q_{1j}^2}{\phi_0^2}.$$

Hierbei ist $\phi_0^{-2} = \mu_0$, wobei $\mu_0 = \int_a^b \omega(x)dx$ das 0-te Moment ist.

Beweis. Es seien λ_j, $j = 1, \ldots, s$ die Nullstellen von ϕ_s. Da die Gauß-Quadraturformel exakt ist für Polynome vom Grad $\leq 2s-1$ und wegen der Orthonormalität gilt

$$\langle 1, \phi_k \rangle = \int_a^b \phi_k(x)\omega(x)dx = \sum_{j=1}^{s} b_j \phi_k(\lambda_j) = \begin{cases} \mu_0\phi_0, & k = 0, \\ 0, & k = 1, \ldots, s. \end{cases}$$

Definieren wir die Matrix

$$P = (\phi_{i-1}(\lambda_j))_{i,j=1,\ldots,s} = \begin{bmatrix} \Phi_s(\lambda_1) & \ldots & \Phi_s(\lambda_s) \end{bmatrix},$$

dann können wir obige Gleichung als

$$P \begin{bmatrix} b_1 \\ \vdots \\ b_s \end{bmatrix} = \mu_0 \phi_0 e_1$$

schreiben. Q sei die orthogonale Matrix, die die Eigenvektoren von T_s enthält, also $Q^T T_s Q = \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_s)$. Da Eigenvektoren zu einfachen Eigenwerten bis auf Skalierung eindeutig bestimmt sind, gilt nach Satz 1.19

$$q_j = Qe_j = \begin{bmatrix} q_{1j} \\ \vdots \\ q_{sj} \end{bmatrix} = \frac{q_{1j}}{\phi_0} \Phi_s(\lambda_j).$$

Also folgt ferner

$$Q = \frac{1}{\phi_0} PD, \qquad D = \operatorname{diag}(q_{11}, \ldots, q_{1s})$$

und daraus

$$b_j = \mu_0 \phi_0 e_j^T P^{-1} e_1 = \mu_0 e_j^T D Q^T e_1 = \mu_0 q_{1j}^2.$$

Schließlich gilt wegen der Normierung der Orthonormalpolynome ϕ_k

$$1 = \int_a^b \phi_0^2(x)\omega(x)dx = \phi_0^2 \mu_0. \quad \square$$

Wir haben damit gezeigt, dass sich die Gewichte direkt aus den Quadraten der ersten Komponenten der normieren Eigenvektoren von T_s berechnen lassen und die Knoten gerade die Eigenwerte sind. Später werden wir Algorithmen zur Lösung dieses und allgemeinerer Eigenwertprobleme kennenlernen.

1.9 Ein adaptives Programm

In praktischen Anwendungen interessiert uns weniger, ob eine Quadraturformel eine bestimmte Ordnung hat, sondern vielmehr dass das Integral bis auf eine vorgegebene Genauigkeit approximiert wird. Wir wählen dazu eine Quadraturformel aus und versuchen, das Intervall (a, b) so zu zerlegen, dass die Quadraturformel trotz des gegebenenfalls sehr unterschiedlichen Verhaltens des Integranden auf jedem Teilintervall ungefähr den gleichen Fehler produziert. Die Teilintervalle können gegebenenfalls sehr unterschiedliche Größe haben.

Das genaue Ziel dieses Abschnitts ist eine Funktion `integral`, die zu einer gegebenen Funktion f und einer gegebenen Toleranz `tol` automatisch eine (nicht äquidistante) Zerlegung von (a, b) wählt, so dass der Fehler $\leq$ `tol` ist und zwar in folgendem Sinn:

$$\left| \int_a^b f(x)dx - \text{numer. Resultat} \right| \leq \texttt{tol} \int_a^b |f(x)|dx.$$

Da wir den exakten Wert des Integrals nicht kennen, können wir den Fehler nicht berechnen sondern müssen uns mit einer Fehlerschätzung zufrieden geben. Diese sollte ohne großen zusätzlichen Aufwand berechenbar sein, d. h. insbesondere ohne zusätzliche Auswertungen von f. Die Idee ist, zu der gegebenen Quadraturformel $(b_i, c_i)_{i=1}^s$ der Ordnung p eine **eingebettete** Quadraturformel zu suchen. Darunter versteht man eine Quadraturformel $(\widehat{b}_i, c_i)_{i=1}^s$ mit der Ordnung $\widehat{p} < p$. Da die eingebettete Quadraturformel die gleichen Knoten hat, sind keine zusätzlichen f-Auswertungen erforderlich.

Bemerkung. Falls die Quadraturformel $(b_i, c_i)_{i=1}^s$ die Ordnung $p \geq s$ hat, dann ist die Ordnung der eingebetteten Quadraturformel nur $\widehat{p} < s$, denn nach Satz 1.6 wäre sonst $\widehat{b}_i = b_i$ für alle $i = 1, \ldots, s$. Dies führt insbesondere bei Gauß-Quadraturformeln ($p = 2s$) zu einer starken Ordnungsreduktion der eingebetteten Quadraturformel.

Nach Korollar 1.10 gilt

$$\int_{x_0}^{x_0+h} f(x)dx - h\sum_{i=1}^s b_i f(x_0 + c_i h) \sim C h^{p+1},$$
$$\int_{x_0}^{x_0+h} f(x)dx - h\sum_{i=1}^s \widehat{b}_i f(x_0 + c_i h) \sim \widehat{C} h^{\widehat{p}+1}.$$

Wir stellen drei Varianten von Fehlerschätzern vor, die auf verschiedenen heuristischen Überlegungen basieren.

(a) Wegen $\widehat{p} < p$ ist für kleines h

$$\texttt{diff} := h \sum_{i=1}^{s} (\widehat{b}_i - b_i) f(x_0 + c_i h) \sim \widehat{C} h^{\widehat{p}+1}$$

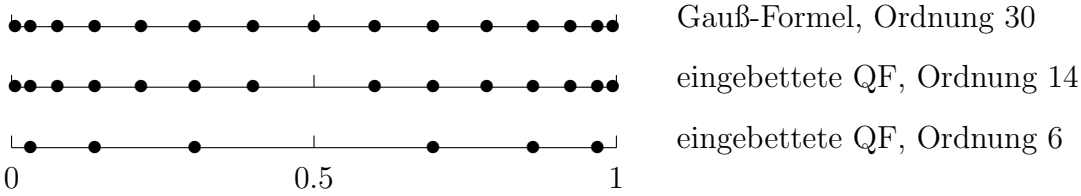
eine Approximation des Fehlers der eingebetteten Quadraturformel. Leider ist auf diese Weise nur eine Fehlerschätzung für die eingebettete, schlechtere Quadraturformel möglich. Eigentlich möchten wir aber den Fehler der besseren Quadraturformel schätzen.

(b) Falls $p = 2s$ und $\widehat{p} = s-1$ ist, dann ist der Fehler von (b_i, c_i) ungefähr $Ch^{2s+1} = Ch(h^s)^2$ und der Fehler von $(\widehat{b}_i, c_i)$ ungefähr $\widehat{C}h^s$, also könnte man

$$\texttt{err}[\texttt{x}_0, \texttt{x}_0 + \texttt{h}] := \texttt{diff}^2$$

setzen.

(c) Hairer (1986) schlägt vor, zu der eingebetten Quadraturformel $(\widehat{b}_i, c_i)$ noch eine weitere, zu dieser Quadraturformel eingebettete Quadraturformel $(\widetilde{b}_i, c_i)$ zu verwenden, zum Beispiel zu einer Gauß-Quadraturformel mit einer ungeraden Anzahl $s = 2m + 1$ von Knoten mit ungeradem m eine zweite symmetrische Quadraturformel durch Weglassen des Knotens $c_{m+1} = \frac{1}{2}$ mit $2m$ Knoten der Ordnung $\widehat{p} = s - 1 = 2m$ und dazu eine weitere symmetrische Quadraturformel mit $m - 2$ oder $m - 1$ Knoten der Ordnung $\widetilde{p} = m - 1$. Für $s = 15$, $m = 7$ würde man folgende Knoten verwenden:



Dann wäre

$$\texttt{diff} := h \sum_{i=1}^{s} (\widehat{b}_i - b_i) f(x_0 + c_i h) \sim \widehat{C} h^{2m+1},$$

$$\widehat{\texttt{diff}} := h \sum_{i=1}^{s} (\widetilde{b}_i - \widehat{b}_i) f(x_0 + c_i h) \sim \widetilde{C} h^{m}.$$

Da der Fehler der Gauß-Quadraturformel ungefähr $Ch^{2s+1} = Ch^{4m+3}$ ist, setzt man schließlich

$$\texttt{err}[\texttt{x}_0, \texttt{x}_0 + \texttt{h}] = \texttt{diff} \left(\frac{\texttt{diff}}{\widehat{\texttt{diff}}} \right)^2 \sim \widehat{C} h^{2m+1} \left(\frac{\widehat{C} h^{2m+1}}{\widetilde{C} h^m} \right)^2 = C' h^{4m+3}.$$

Alternativ zur Verwendung eingebetteter Verfahren kann man auch eine Quadraturformel mit zwei verschiedenen Schrittweiten verwenden. Sei

$$Q_h(f, x_0) := h \sum_{j=1}^{s} b_j f(x_0 + c_j h) \approx \int_{x_0}^{x_0+h} f(x) dx.$$

Wir approximieren

$$\int_{x_0}^{x_0+h} f(x)dx \approx Q_{h/2}(f,x_0) + Q_{h/2}(f,x_0+h/2)$$

und zusätzlich zur Fehlerschätzung

$$\int_{x_0}^{x_0+h} f(x)dx \approx Q_h(f,x_0).$$

Angenommen, die Quadraturformel habe die Ordnung p und es existiert eine Fehlerdarstellung der Form

$$e_h := \int_{x_0}^{x_0+h} f(x)dx - Q_h(f,x_0) = c_p f^{(p)}(x^*)h^{p+1}, \qquad x^* \in [x_0, x_0+h],$$

wie etwa für die Mittelpunkts- und die Trapezregel mit $p=2$ (vgl. Beispiel 1.1). Dann gilt nach dem Zwischenwertsatz

$$\begin{aligned} e_{h/2} &:= \int_{x_0}^{x_0+h} f(x)dx - \big(Q_{h/2}(f,x_0) + Q_{h/2}(f,x_0+\tfrac{h}{2})\big) \\ &= c_p\Big(\frac{h}{2}\Big)^{p+1}\Big(f^{(p)}(x_1^*) + f^{(p)}(x_2^*)\Big) && x_{1,2}^* \in [x_0, x_0+h] \\ &= \frac{1}{2^p}\, c_p\, h^{p+1}\, f^{(p)}(x_3^*), && x_3^* \in [x_0, x_0+h] \end{aligned}$$

und

$$e_h := c_p h^{p+1} f^{(p)}(x_4^*), \qquad x_4^* \in [x_0, x_0+h].$$

Die berechenbare Größe $e_h - e_{h/2}$ erfüllt somit

$$\begin{aligned} e_h - e_{h/2} &= Q_{h/2}(f,x_0) + Q_{h/2}(f,x_0+h/2) - Q_h(f,x_0) \\ &= c_p h^{p+1} f^{(p)}(x_4^*) - \frac{1}{2^p}\, c_p h^{p+1} f^{(p)}(x_3^*). \end{aligned}$$

Nimmt man $x_3^* \approx x_4^*$ an, so gilt

$$e_h - e_{h/2} \approx \Big(1 - \frac{1}{2^p}\Big)\, c_p h^{p+1} f^{(p)}(x_3^*) = (2^p-1)e_{h/2}.$$

Hieraus ergibt sich die folgende Schätzung für den Fehler:

$$\texttt{err}[\texttt{x0}, \texttt{x0}+\texttt{h}] = e_{h/2} \approx \frac{1}{2^p-1}\Big(Q_{h/2}(f,x_0) + Q_{h/2}(f,x_0+h/2) - Q_h(f,x_0)\Big).$$

Der relative Fehler dieser Fehlerschätzung geht unter schwachen Voraussetzungen gegen Null für $h \to 0$.

Für ein Teilintervall $[x_0, x_0+h]$ von $[a,b]$ können wir folgende Näherungen der Integrale mit f und $|f|$ berechnen:

$$\texttt{res}[x_0, x_0+h] = h\sum_{j=1}^{s} b_j f(x_0+c_j h) \approx \int_{x_0}^{x_0+h} f(x)dx,$$

$$\texttt{resabs}[x_0, x_0+h] = h\sum_{j=1}^{s} b_j |f(x_0+c_j h)| \approx \int_{x_0}^{x_0+h} |f(x)|dx.$$

Algorithmus 1.1 Adaptives Quadraturverfahren

function `integral`$(f, a, b, \texttt{tol})$
Initialisiere die Zerlegung $\Delta = \{a, b\}$
Berechne $\rho = \texttt{res}[a, b]$, $\rho_a = \texttt{resabs}[a, b]$ und $\epsilon = |\texttt{err}[a, b]|$
while $\epsilon > \texttt{tol}\,\rho_a$ **do**
  Wähle das Teilintervall $[\alpha, \beta]$ der Zerlegung mit dem größten Fehler:
    $|\texttt{err}[\alpha, \beta]| = \max_I |\texttt{err}(I)|$
  Subtrahiere Fehler und Näherung für dieses Teilintervall vom Gesamtergebnis:
    $\epsilon = \epsilon - |\texttt{err}[\alpha, \beta]|$, $\rho_a = \rho_a - \texttt{resabs}[\alpha, \beta]$, $\rho = \rho - \texttt{res}[\alpha, \beta]$
  Ergänze die Zerlegung um den Punkt $\frac{\alpha+\beta}{2}$, d. h. teile das Intervall $[\alpha, \beta] = I_l \cup I_r$
  Berechne $\texttt{res}(I_l)$, $\texttt{res}(I_r)$ $\texttt{resabs}(I_l)$, $\texttt{resabs}(I_r)$ $\texttt{err}(I_l)$, $\texttt{err}(I_r)$
  Aktualisiere Fehlerschätzung $\epsilon = \epsilon + |\texttt{err}(I_l)| + |\texttt{err}(I_r)|$
  Aktualisiere Näherung $\rho = \rho + \texttt{res}(I_l) + \texttt{res}(I_r)$
  Setze $\rho_a = \rho_a + \texttt{resabs}(I_l) + \texttt{resabs}(I_r)$
end while
Akzeptiere ρ als Approximation an $\int_a^b f(x)dx$.

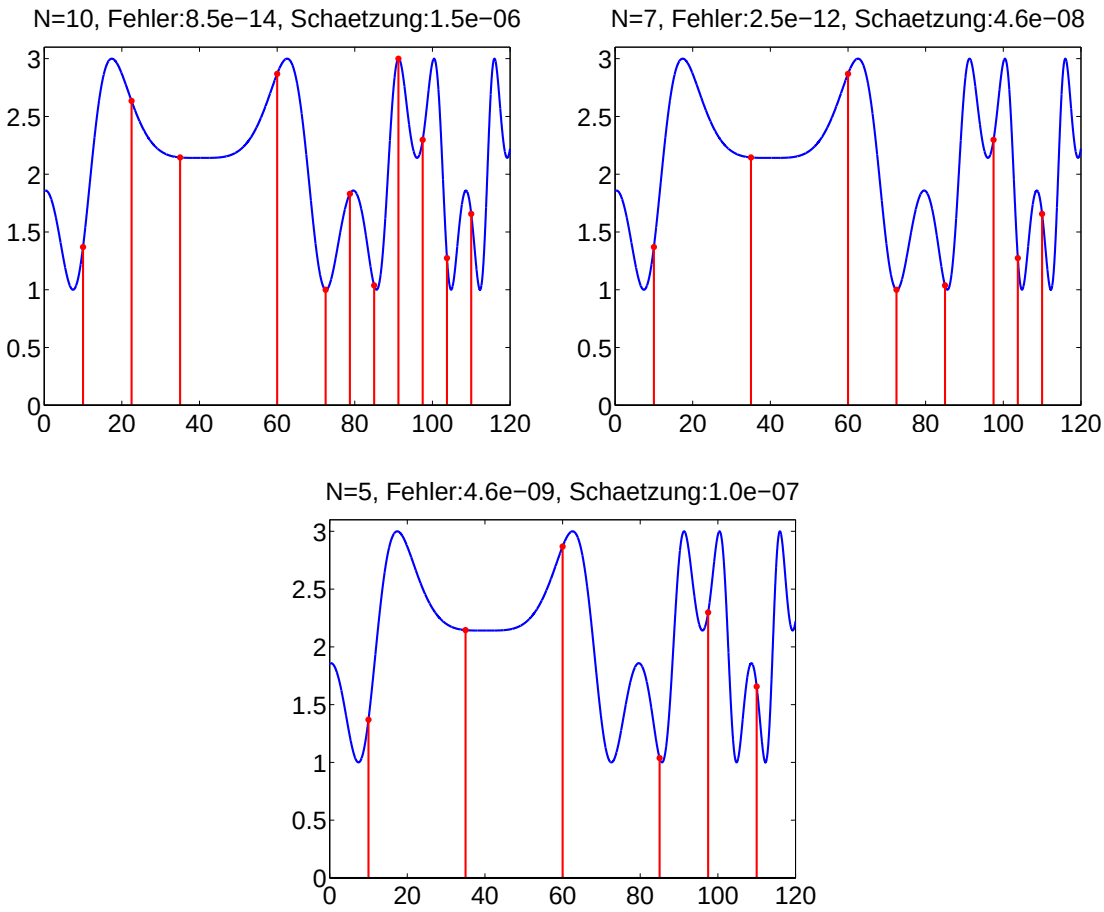


Abb. 1.2: Zerlegungen der verschiedenen Fehlerschätzer (a), (b), (c). Angegeben sind der die Zahl N der Teilintervalle sowie der tatsächliche absolute Fehler und der berechnete Fehlerschätzer.

Ausgehend vom gesamten Intervall $[a, b]$ unterteilt Algorithmus 1.1 Teilintervalle solange, bis der Gesamtfehler kleiner als die vorgegebene Toleranz ist. In diesem Algorithmus wird eine weitere Funktion `gauss` verwendet, die `res`, `resabs` und `err` berechnet.

Beispiel. Abbildung 1.2 zeigt die mit Algorithmus 1.1 berechneten Teilintervalle für die Funktion $f(x) = 2 + \sin(3\cos(0.002(x-40)^2))$ integriert über das Intervall $[10, 110]$ mit Toleranz 10^{-8}. Das obere Bild zeigt, dass der erste Fehlerschätzer (a) für die eingebettete Formel den Fehler deutlich überschätzt und zu viele Intervalle erzeugt. In der Mitte und unten sind die Ergebnisse der Fehlerschätzer (b) und (c) dargestellt. $\diamond$

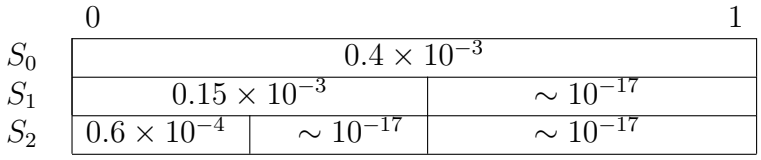
1.10 Konvergenzbeschleunigung mit dem ϵ-Algorithmus

Bisher haben wir nur Integrale ohne Singularitäten untersucht. In diesem Abschnitt werden wir einen Algorithmus vorstellen, mit dem auch bei Singularitäten des Integranden gute Approximationen berechnet werden können.

Wir illustrieren die auftretenden Schwierigkeiten zunächst an einem Beispiel. Approximieren wir

$$S = \int_0^1 \sqrt{x} \log x dx = -\frac{4}{9}$$

mit einer Gauß-Quadraturformel der Ordnung 30 und verwenden zum Beispiel den adaptiven Algorithmus aus dem letzten Abschnitt, dann beobachten wir folgende Fehler auf den eingezeichneten Teilintervallen:



Die Konvergenz der Folge $\{S_n\}_n$ gegen den exakten Wert S ist also sehr langsam. Häufig beobachtet man

$$S_{n+1} - S \approx \rho(S_n - S), \tag{1.8}$$

in unserem Beispiel $\rho \approx 0.4$. Die Idee von Aitken (1926) nutzt dies wie folgt aus: Angenommen, wir kennen bereits drei aufeinanderfolgende Werte S_n, S_{n+1}, S_{n+2}. Dann bestimmen wir s und ρ so, dass

$$\begin{aligned} S_{n+1} - s &= \rho(S_n - s), \\ S_{n+2} - s &= \rho(S_{n+1} - s) \end{aligned}$$

und verwenden $S'_n = s$ als "hoffentlich" bessere Näherung an S. Zur Berechnung von s bilden wir die Differenz der beiden Gleichungen:

$$\Delta S_{n+1} = \rho \Delta S_n, \qquad \Delta S_n = S_{n+1} - S_n.$$

In der zweiten Gleichung ergänzen wir S_{n+1}:

$$S_{n+2} - S_{n+1} + S_{n+1} - s = \rho(S_{n+1} - s) \Longleftrightarrow \Delta S_{n+1} = (\rho - 1)(S_{n+1} - s).$$

Für $\rho \neq 1$ ergibt sich daraus

$$s = S_{n+1} - \frac{\Delta S_{n+1}}{\rho - 1} = S_{n+1} - \frac{\Delta S_{n+1} \cdot \Delta S_n}{\Delta S_{n+1} - \Delta S_n}$$

oder mit der Notation $\Delta^2 S_n = \Delta(\Delta S_n) = \Delta S_{n+1} - \Delta S_n$

$$S'_n = S_{n+1} - \frac{\Delta S_{n+1} \cdot \Delta S_n}{\Delta^2 S_n},$$

die Δ^2-**Regel von Aitken**. Diese liefert zu einer gegebenen Folge $\{S_n\}_n$ eine neue Folge $\{S'_n\}_n$, die schneller konvergiert, falls (1.8) erfüllt ist.

Die Idee von Aitken lässt sich in naheliegender Weise erweitern. Dazu nehmen wir allgemeiner an, dass statt (1.8) nun

$$S_{n+2} - S = \alpha_1(S_{n+1} - S) + \alpha_0(S_n - S)$$

gilt. Aus den Tripeln (S_n, S_{n+1}, S_{n+2}), $(S_{n+1}, S_{n+2}, S_{n+3})$, $(S_{n+2}, S_{n+3}, S_{n+4})$ erhalten wir drei Gleichungen für die Unbekannten α_0, α_1 und s und verwenden dann S''_n als neue Näherung.

Analog können wir $S_n^{(k)}$ mit $S_n^{(k)} = S$ für Folgen mit

$$S_{n+k} - S = \alpha_{k-1}(S_{n+k-1} - S) + \cdots + \alpha_0(S_n - S)$$

konstruieren. Die so definierten $S_n^{(k)}$ lassen sich auf überraschend einfache Weise definieren:

Satz 1.21. *(ϵ-Algorithmus von Wynn, 1956)*
Es sei eine Folge $\{S_n\}_n$ gegeben. Man definiert

$$\begin{aligned} &\epsilon_{-1}^{(n)} = 0, \quad \epsilon_0^{(n)} = S_n, \\ &\epsilon_{k+1}^{(n)} = \epsilon_{k-1}^{(n+1)} + \frac{1}{\epsilon_k^{(n+1)} - \epsilon_k^{(n)}}, \qquad k, n \geq 0. \end{aligned}$$

Dann ist $\epsilon_2^{(n)} = S'_n$, $\epsilon_4^{(n)} = S''_n$, $\ldots$, $\epsilon_{2k}^{(n)} = S_n^{(k)}$.

Beweis. Der Beweis ist nicht trivial. Zunächst zeigt man

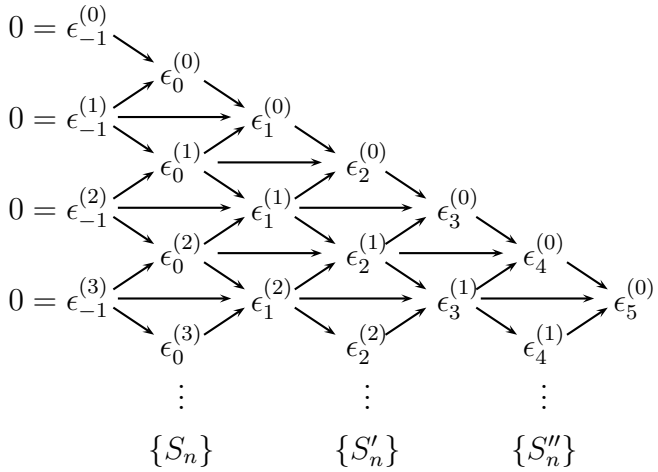
$$S'_n = \frac{\det \begin{bmatrix} S_n & S_{n+1} \\ S_{n+1} & S_{n+2} \end{bmatrix}}{\det \Delta^2 S_n}, \qquad S''_n = \frac{\det \begin{bmatrix} S_n & S_{n+1} & S_{n+2} \\ S_{n+1} & S_{n+2} & S_{n+3} \\ S_{n+2} & S_{n+3} & S_{n+4} \end{bmatrix}}{\det \begin{bmatrix} \Delta^2 S_n & \Delta^2 S_{n+1} \\ \Delta^2 S_{n+1} & \Delta^2 S_{n+2} \end{bmatrix}}, \qquad \ldots$$

Die $S_n^{(k)}$ können allgemein mit Hilfe von Determinanten von **Hankel-Matrizen**, das sind Matrizen, die auf jeder Gegendiagonale konstante Einträge haben, ausgedrückt werden. Die Werte $S_n^{(k)}$ können auch als Auswertung von sogenannten **Padé-Approximationen**, das sind rationale Approximationen, an der Stelle eins interpretiert werden. Aus den Rekursionsformeln für Padé-Approximationen leiten sich dann die Rekursionsformeln des ϵ-Algorithmus ab, siehe zum Beispiel Gragg (1972).

Wir verzichten auf die weiteren Details und zeigen die Behauptung nur für S'_n, also für $k = 1$:

$$\begin{aligned} \epsilon_1^{(n)} &= \frac{1}{S_{n+1} - S_n} = \frac{1}{\Delta S_n} \\ \epsilon_2^{(n)} &= S_{n+1} + \frac{1}{(\Delta S_{n+1})^{-1} - (\Delta S_n)^{-1}} = S_{n+1} + \frac{\Delta S_{n+1} \cdot \Delta S_n}{\Delta S_n - \Delta S_{n+1}} \\ &= S_{n+1} - \frac{\Delta S_{n+1} \cdot \Delta S_n}{\Delta^2 S_n}. \quad \square \end{aligned}$$

Zur Berechnung der Folgen ordnet man die $\epsilon_k^{(n)}$ am besten in einem Schema an:



Beispiel. Wenden wir den ϵ-Algorithmus auf die Näherungen der Trapezregel zur Schrittweitenfolge $\{1/2^k\}$ bei obigem Beispiel an, so ergibt sich folgendes Fehlertableau für die Folgen $\epsilon_{2k}^{(n)}$

$1.2266 \cdot 10^{-1}$				
$4.3028 \cdot 10^{-2}$	$2.7352 \cdot 10^{-3}$			
$1.6272 \cdot 10^{-2}$	$4.0854 \cdot 10^{-4}$	$-1.5652 \cdot 10^{-5}$		
$6.3134 \cdot 10^{-3}$	$4.1457 \cdot 10^{-5}$	$-1.0284 \cdot 10^{-5}$	$-5.4345 \cdot 10^{-5}$	
$2.4651 \cdot 10^{-3}$	$-4.2225 \cdot 10^{-6}$	$-4.8182 \cdot 10^{-6}$	$3.6329 \cdot 10^{-7}$	$5.5136 \cdot 10^{-9}$
$9.6092 \cdot 10^{-4}$	$-4.8107 \cdot 10^{-6}$	$-4.3315 \cdot 10^{-6}$	$2.6562 \cdot 10^{-8}$	$-9.8868 \cdot 10^{-11}$
$3.7279 \cdot 10^{-4}$	$-2.2318 \cdot 10^{-6}$	$7.4004 \cdot 10^{-7}$	$1.9965 \cdot 10^{-9}$	
$1.4379 \cdot 10^{-4}$	$-8.5616 \cdot 10^{-7}$	$7.0805 \cdot 10^{-8}$		
$5.5141 \cdot 10^{-5}$	$-3.0448 \cdot 10^{-7}$			
$2.1030 \cdot 10^{-5}$				

In der ersten Spalte stehen die Fehler der Näherungen des Ausgangsfolge, in unserem Fall der Trapezregel, in der zweiten die der Δ^2-Methode von Aitken. $\diamond$