Übungen zur Vorlesung Numerik I
Sommersemester 2010

PD Dr. Thorsten Hüls
Dipl.-Math. Denny Otten

Übungsblatt 6
20.05.2010

Abgabe: Donnerstag, 27.05.2010, 10:00 Uhr in das Postfach des jeweiligen Tutors.
Mo.-Tutorium: Paul Voigt, paulvoigt@web.de, Postfach 195 in V3-128
Di.-Tutorium: Denny Otten, dotten@math.uni-bielefeld.de, Postfach 44 in V3-128
Mi.-Tutorium: Ingwar Petersen, ipeterse@math.uni-bielefeld.de, Postfach 227 in V3-128

Aufgabe 15: (Numerische Integration, Newton-Cotes-Formeln)

(a) Wir wollen zunächst eine allgemeine Quadraturformel entwickeln.
(2) Bestimmen Sie $w_0, w_1 \in \mathbb{R}$ und $t_0, t_1 \in (-1, 1)$, $t_0 < t_1$ so, dass

$$Q(f) = w_0 f(t_0) + w_1 f(t_1)$$

eine Quadraturformel darstellt, die Polynome vom Grad 3 exakt integriert, also

$$Q(f) = \int_{-1}^{1} f(t)\,dt \quad \text{für } f \in \mathcal{P}_3.$$

(1) Zeigen Sie, dass mit dieser Formel Polynome vom Grad 4 im Allgemeinen nicht exakt integriert werden.

(b) Sei nun speziell die Newton-Cotes-Formel zu $m+1$ Stützstellen für $\int_a^b f(t)\,dt$ gegeben durch

$$Q_m(f) = h \sum_{i=0}^{m} \sigma_i^m f(t_i).$$

Zeigen Sie, dass

(1) – die Koeffizienten symmetrisch sind, d. h. $\sigma_i^m = \sigma_{m-i}^m$ für $i = 0, \ldots, m$.
(1) – Q_m für geradzahlige m Polynome vom Grad $m+1$ exakt integriert.

(1) Stellt Q aus Aufgabenteil (a) eine Newton-Cotes-Formel dar?

(6 Punkte)

Aufgabe 16: (Programmieraufgabe, Adaptives Quadraturverfahren)
Das folgende Integral soll numerisch approximiert werden

$$\int_{-4}^{2} f(t)\,dt, \quad f(t) = -\frac{\pi^3 t}{(t^2+1)^2}\cos\left(\frac{25}{t^2+1}\right) + 5.$$

(a) Implementieren Sie dazu das adaptive Quadraturverfahren, dessen Pseudocode im Skript auf Seite 103 aufgeführt ist. Als Schätzung für das Integral verwenden Sie $W = 30$. Als relative Genauigkeit wählen Sie $\varepsilon = 10^{-2}$ bzw. $\varepsilon = 10^{-7}$. Messen Sie, mit Hilfe von `tic`, `toc` in MATLAB, die Zeit, die Ihr Algorithmus benötigt.
Berechnen Sie eine Stammfunktion F von f und verwenden Sie F zur Ausgabe des Approximationsfehlers

$$|F(2) - F(-4) - I_\varepsilon|,$$

wobei I_ε die Approximation durch die adaptive Quadratur mit Genauigkeit ε bezeichnet. Geben Sie den numerischen Wert des Integrals I_ε aus.

(b) Steigern Sie die Effizienz des zuvor implementierten Algorithmus, indem Sie für die lokalen Variablen t, f, i aus dem Pseudocode Speicher vorreservieren (in MATLAB: $t = \text{zeros}(m)$, sinnvollerweise $m > p$). Wiederholen Sie die Berechnung und Zeitmessung wie im Teil (a).

(c) Verwenden Sie den in der Vorlesung vorgestellten rekursiven Algorithmus:

```
function (I) = integral (x,y)
    Simpson = S(x,y)
    if ( |(T(x,y) - Simpson)/W| ≤ ε (y-x)/(b-a) )
        I = Simpson
    else
        I = integral(x, (x+y)/2) + integral((x+y)/2, y)
```

Messen Sie auch hier die Zeit, die der Algorithmus benötigt.

Kommentieren Sie in den Teilen (a)-(c) Ihre Ergebnisse und Laufzeiten.

(6 Punkte)

Aufgabe 17: (Numerische Integration, Gaußsche Quadraturformeln)
Die Gaußschen Quadraturformeln sind eine spezielle Version der interpolatorischen Quadraturformeln, bei denen die Stützstellen besonders geschickt gewählt werden. Sie besitzen auf $[-1,1]$ die Form

$$Q_m(f) = \sum_{j=1}^{m} w_j f(\rho_j), \quad m \geqslant 1,$$

wobei ρ_j die Nullstellen des m–ten Legendre-Polynoms

$$\mathcal{L}_m(t) = \frac{m!}{(2m)!}\frac{d^m}{dt^m}(t^2-1)^m, \quad m \geqslant 0$$

sind und die Gewichte w_j wie üblich gewählt werden.
SATZ: Gaußsche Quadraturformeln integrieren Polynome sogar bis zum Grad $2m-1$ exakt, das heißt es gilt

$$\forall\, p \in \mathcal{P}_{2m-1}: \quad Q_m(p) = \int_{-1}^{1} p(t)dt.$$

Arbeiten Sie die folgende Beweisskizze des Satzes in allen Einzelheiten aus.

BEWEIS:

(2) (a) Mit Hilfe des Satzes von Rolle sieht man sofort, dass das m–te Legendre-Polynom $\mathcal{L}_m$ genau m paarweise verschiedene Nullstellen

$$\rho_1 < \rho_2 < \ldots < \rho_m$$

in $(-1,1)$ besitzt.

(1) (b) Durch partielle Integration folgt, dass die Polynome $\mathcal{L}_0, \ldots, \mathcal{L}_m$ eine orthogonale Basis von $\mathcal{P}_m$ im folgenden Sinne bilden

$$\int_{-1}^{1} \mathcal{L}_k(t)\mathcal{L}_l(t)dt = 0 \quad \forall\, 0 \leq k < l \leq m.$$

(1) (c) Offensichtlich ist

$$\int_{-1}^{1} \mathcal{L}_m(t)f(t)dt = 0, \; \forall\, f \in \mathcal{P}_{m-1}.$$

(1) (d) Ist nun $p \in \mathcal{P}_{2m-1}$, so lässt sich p trivialerweise schreiben als

$$p(t) = \mathcal{L}_m(t)\,q(t) + r(t),$$

wobei q und r Polynome aus $\mathcal{P}_{m-1}$ sind.

(1) (e) Mit dieser Darstellung von p folgt direkt die behauptete Gleichheit

$$Q_m(p) = \int_{-1}^{1} p(t)dt.$$

(6 Punkte)

Aufgabe 15:

zu (a): 1. Zeige:

$$\exists\, \omega_0, \omega_1 \in \mathbb{R} \wedge \exists\, t_0, t_1 \in]-1,1[\text{ mit } t_0 < t_1\ \forall f \in \mathcal{P}_3 : \begin{cases} ① : Q(f) := \omega_0 \cdot f(t_0) + \omega_1 \cdot f(t_1) \\ \text{und} \\ ② : Q(f) = \int_{-1}^{1} f(t)\,dt \\ \text{(Exaktheit)} \end{cases}$$

2. Zeige: Für $f \in \mathcal{P}_4 \setminus \mathcal{P}_3$ gilt i.A.

$$Q(f) \neq \int_{-1}^{1} f(t)\,dt \quad \text{(Keine Exaktheit)}$$

d.h. die Quadraturformel Q (aus 1.) integriert Polynome vom Grad 4 nicht exakt.

---

zu 1. Vorbemerkung: Die folgenden zwei Aussagen sind äquivalent

ⓘ: Q integriert Polynome $f \in \mathcal{P}_3$ exakt (d.h. $\forall f \in \mathcal{P}_3 : Q(f) = \int_{-1}^{1} f(t)\,dt$)

ⓘⓘ: Q integriert die Monome $e_j(t) := t^j$ $(j = 0,1,2,3)$ exakt (d.h. $\forall j = 0,\dots,3 : Q(e_j) = \int_{-1}^{1} t^j\,dt$)

Dies gilt, da die Integration $\int$ und die Quadraturformel Q lineare Funktionale sind.

> **Beweis:** Da $\int$ und Q lineare Funktionale sind, gilt für ein Polynom $\mathcal{P}_3 \ni f = a_0 e_0 + \dots + a_3 e_3$ mit $a_0, \dots, a_3 \in \mathbb{R}$ und den Monomfunktionen $e_0, \dots, e_3$:
>
> $$Q(f) = Q\Big(\sum_{i=0}^{3} a_i e_i\Big) \underset{\uparrow\ Q \text{ lineares Funktional}}{=} \sum_{i=0}^{3} a_i Q(e_i)$$
>
> $\Vert$ ⓘ $\qquad\qquad\qquad\qquad \Vert$ ⓘⓘ
>
> $$\int_{-1}^{1} f(t)\,dt = \int_{-1}^{1} \sum_{i=0}^{3} a_i e_i(t)\,dt \underset{\uparrow\ \int \text{ lineares Funktional}}{=} \sum_{i=0}^{3} a_i \int_{-1}^{1} e_i(t)\,dt \qquad \blacksquare$$

Daher genügt es, wenn die Exaktheit ② = ⓘ lediglich für die Monome, also ⓘⓘ, erfüllt ist. Für die Monome gilt (aufgrund der Exaktheit):

Ⓐ: $\omega_0 + \omega_1 = 2$

Ⓑ: $\omega_0 t_0 + \omega_1 t_1 = 0$

Ⓒ: $\omega_0 t_0^2 + \omega_1 t_1^2 = \frac{2}{3}$

Ⓓ: $\omega_0 t_0^3 + \omega_1 t_1^3 = 0$

> zu Ⓐ:
>
> $$Q(e_0) \overset{①}{=} \omega_0 \underbrace{e_0(t_0)}_{=1} + \omega_1 \underbrace{e_0(t_1)}_{=1} = \omega_0 + \omega_1$$
>
> $\Vert$ ⓘⓘ (Exaktheit)
>
> $$\int_{-1}^{1} e_0(t)\,dt = \int_{-1}^{1} 1\,dt = [t]_{-1}^{1} = 1 - (-1) = 2$$
>
> zu Ⓑ:
>
> $$Q(e_1) \overset{①}{=} \omega_0 \underbrace{e_1(t_0)}_{=t_0} + \omega_1 \underbrace{e_1(t_1)}_{=t_1} = \omega_0 t_0 + \omega_1 t_1$$
>
> $\Vert$ ⓘⓘ (Exaktheit)
>
> $$\int_{-1}^{1} e_1(t)\,dt = \int_{-1}^{1} t\,dt = \Big[\tfrac{1}{2} t^2\Big]_{-1}^{1} = \tfrac{1}{2} - \tfrac{1}{2} = 0$$
>
> zu Ⓒ:
>
> $$Q(e_2) \overset{①}{=} \omega_0 \underbrace{e_2(t_0)}_{=t_0^2} + \omega_1 \underbrace{e_2(t_1)}_{=t_1^2} = \omega_0 t_0^2 + \omega_1 t_1^2$$
>
> $\Vert$ ⓘⓘ (Exaktheit)
>
> $$\int_{-1}^{1} e_2(t)\,dt = \int_{-1}^{1} t^2\,dt = \Big[\tfrac{1}{3} t^3\Big]_{-1}^{1} = \tfrac{1}{3} - \big(-\tfrac{1}{3}\big) = \tfrac{2}{3}$$
>
> zu Ⓓ:
>
> $$Q(e_3) \overset{①}{=} \omega_0 \underbrace{e_3(t_0)}_{=t_0^3} + \omega_1 \underbrace{e_3(t_1)}_{=t_1^3} = \omega_0 t_0^3 + \omega_1 t_1^3$$
>
> $\Vert$ ⓘⓘ (Exaktheit)
>
> $$\int_{-1}^{1} e_3(t)\,dt = \int_{-1}^{1} t^3\,dt = \Big[\tfrac{1}{4} t^4\Big]_{-1}^{1} = \tfrac{1}{4} - \tfrac{1}{4} = 0$$
>
> Allgemeiner: $\forall j = 0,\dots,3$:
>
> $$Q(e_j) \overset{①}{=} \omega_0 \underbrace{e_j(t_0)}_{=t_0^j} + \omega_1 \underbrace{e_j(t_1)}_{=t_1^j} = \omega_0 t_0^j + \omega_1 t_1^j$$
>
> $\Vert$ ⓘⓘ (Exaktheit)
>
> $$\int_{-1}^{1} e_j(t)\,dt = \int_{-1}^{1} t^j\,dt = \Big[\tfrac{1}{j+1} t^{j+1}\Big]_{-1}^{1} = \tfrac{1}{j+1}\Big[1^{j+1} - (-1)^{j+1}\Big]$$

Hierbei gilt offensichtlich $\omega_0, \omega_1, t_0, t_1 \neq 0$.

zu ω_0: Angenommen $\omega_0 = 0$

$$\overset{A}{\Rightarrow} \omega_1 = 2 \overset{B}{\Rightarrow} t_1 = 0 \overset{C}{\Rightarrow} 0 = \frac{2}{3} \ \lightning$$

zu t_0: Angenommen $t_0 = 0$

$$\overset{B}{\Rightarrow} \omega_1 t_1 = 0 \overset{C}{\Rightarrow} 0 = \frac{2}{3} \ \lightning$$

zu ω_1: Angenommen $\omega_1 = 0$

$$\overset{A}{\Rightarrow} \omega_0 = 2 \overset{B}{\Rightarrow} t_0 = 0 \overset{C}{\Rightarrow} 0 = \frac{2}{3} \ \lightning$$

zu t_1: Angenommen $t_1 = 0$

$$\overset{B}{\Rightarrow} \omega_0 t_0 = 0 \overset{C}{\Rightarrow} 0 = \frac{2}{3} \ \lightning$$

Aus A – D erhalten wir nun

$$t_0 = -\sqrt{\frac{1}{3}}, \quad t_1 = \sqrt{\frac{1}{3}}, \quad \omega_0 = \omega_1 = 1$$

BERECHNUNG:

$$\overset{B}{\Rightarrow} \omega_0 t_0 = -\omega_1 t_1 \overset{D}{\Rightarrow} -\omega_1 t_1 t_0^2 - \omega_0 t_0 t_1^2 = 0$$

$$\Rightarrow -\omega_1 t_0 - \omega_0 t_1 = 0$$

$$\left(\text{„}\cdot \frac{1}{t_0 t_1}\text{“, denn } t_0, t_1 \neq 0\right)$$

$$\overset{B}{\Rightarrow} (\omega_0 + \omega_1)(t_0 + t_1) = 0$$

$$\overset{A}{\Rightarrow} t_0 + t_1 = 0$$

$$\Rightarrow t_0 = -t_1$$

$$\overset{B}{\Rightarrow} \omega_0 = \omega_1$$

$$\left(\text{„}\cdot \frac{1}{t_0}\text{“ bzw. „}\cdot \frac{1}{t_1}\text{“, denn } t_0, t_1 \neq 0\right)$$

$$\overset{A}{\Rightarrow} \omega_0 = \omega_1 = 1$$

$$\overset{C}{\Rightarrow} t_0^2 + t_1^2 = \frac{2}{3}$$

$$\Rightarrow t_0 = -\sqrt{\frac{1}{3}} = -t_1$$

$$(\text{wegen } t_0 = -t_1,\ t_0 < t_1)$$

Die Quadraturformel lautet daher

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

zu 2. Betrachte das Monom $e_4(t) = t^4 \in \mathcal{P}_4 \setminus \mathcal{P}_3$, dann gilt

$$Q(e_4) \neq \int_{-1}^{1} e_4(t)\,dt$$

BEWEIS:

$$Q(e_4) = \omega_0 t_0^4 + \omega_1 t_1^4 = \frac{2}{9} \neq \frac{2}{5} = \left[\frac{1}{5} \cdot t^5\right]_{-1}^{1} = \int_{-1}^{1} t^4\,dt = \int_{-1}^{1} e_4(t)\,dt$$

---

zu (b): 1. Zeige:

$$\sigma_i^m = \sigma_{m-i}^m \quad \forall i = 0, \dots, m$$

2. Zeige:

$$m \in \mathbb{N} \text{ gerade} \Rightarrow \forall f \in \mathcal{P}_{m+1} : Q_m(f) = \int_a^b f(t)\,dt$$

3. Frage: Stellt Q aus Aufgabenteil (a) eine Newton-Cotes-Formel dar?

---

zu 1. Die Koeffizienten σ_i^m sind gegeben durch

$$\sigma_i^m := \int_0^m \prod_{\substack{j=0 \\ j \neq i}}^{m} \frac{(s-j)}{(i-j)}\,ds$$

Es gilt

$$\text{(E)}: \prod_{\substack{j=0\\j\neq i}}^{m}(i-j) = (-1)^{m-i}\cdot i!\cdot(m-i)!$$

$$\text{(F)}: \prod_{\substack{j=0\\j\neq(m-i)}}^{m}((m-i)-j) = (-1)^{i}\cdot i!\cdot(m-i)!$$

$$\text{(G)}: \prod_{\substack{j=0\\j\neq i}}^{m}(s-j) = (-1)^{m}\cdot\prod_{\substack{j=0\\j\neq(m-i)}}^{m}(m-s-j)$$

zu (E):

$$\prod_{\substack{j=0\\j\neq i}}^{m}(i-j) = \underbrace{i\cdot(i-1)\cdot\dots\cdot(i-(i-1))}_{=i!}\cdot\underbrace{(i-(i+1))\cdot\dots\cdot(i-m)}_{m-(i+1)+1=m-i\text{ Faktoren}}$$

$$= (-1)^{m-i}\cdot i!\cdot\underbrace{((i+1)-i)\cdot\dots\cdot(m-i)}_{=(m-i)!}$$

$$= (-1)^{m-i}\cdot i!\cdot(m-i)!$$

zu (F):

$$\prod_{j=0}^{m}((m-i)-j) = \underbrace{(m-i)\cdot((m-i)-1)\cdot\dots\cdot((m-i)-(m-i-1))}_{=(m-i)!}\cdot\underbrace{((m-i)-(m-i+1))\cdot\dots\cdot((m-i)-m)}_{m-(m-i+1)+1=i\text{ Faktoren}}$$

$$= (-1)^{i}\cdot(m-i)!\cdot\underbrace{((m-i+1)-(m-i))\cdot\dots\cdot(m-(m-i))}_{=i!}$$

$$= (-1)^{i}\cdot i!\cdot(m-i)!$$

zu (G):

$$\prod_{\substack{j=0\\j\neq i}}^{m}(s-j) = s\cdot(s-1)\cdot\dots\cdot(s-(i-1))\cdot(s-(i+1))\cdot\dots\cdot(s-m)$$

$$= (-1)^{m}\cdot(-s)\cdot(1-s)\cdot\dots\cdot((i-1)-s)\cdot((i+1)-s)\cdot\dots\cdot(m-s)$$

$$= (-1)^{m}\cdot(m-s-m)\cdot(m-s-(m-1))\cdot\dots\cdot(m-s-(m-(i-1)))\cdot(m-s-(m-(i+1)))\cdot\dots\cdot(m-s)$$

$$= (-1)^{m}\cdot\prod_{\substack{j=0\\j\neq(m-i)}}^{m}(m-s-j)$$

Aus (E) – (G) erhalten wir nun

$$\sigma_i^m = \sigma_{m-i}^m \quad \forall i=0,\dots,m$$

Viel kürzere Alternative: Umnummerierung (Substitution)

$$\sigma_i^m = \int_0^m \prod_{\substack{j=0\\j\neq i}}^{m}\frac{(s-j)}{(i-j)}\,ds = \int_0^m \frac{\prod_{j=0,j\neq i}^{m}(s-j)}{\prod_{j=0,j\neq i}^{m}(i-j)}\,ds$$

$$\underset{\text{(E),(G)}}{=} \frac{(-1)^m}{(-1)^{m-i}}\cdot\int_0^m\frac{\prod_{j=0,j\neq(m-i)}^{m}(m-s-j)}{i!\cdot(m-i)!}\,ds$$

$$\underset{\text{(F)}}{=} \underbrace{\frac{(-1)^m}{(-1)^{m-i}\cdot(-1)^{-i}}}_{=(-1)^{m-m+i+i}=(-1)^{2i}=1\ \forall i=0,\dots,m}\cdot\int_0^m\prod_{\substack{j=0\\j\neq(m-i)}}^{m}\frac{(s-j)}{((m-i)-j)}\,ds = \sigma_{m-i}^m \quad \forall i=0,\dots,m$$

zu 2. Es gilt

$$\text{(H)}: \int_0^m\prod_{j=0}^{m}(s-j)\,ds = 0 \qquad \text{(da } m \text{ gerade)}$$

$$\text{(I)}: p(t) = \sum_{j=0}^{m} a_j\cdot\prod_{i=0}^{j-1}(t-t_i) + a_{m+1}\cdot\prod_{i=0}^{m}(t-t_i)$$

(Newtonsche Darstellung)

$$\text{(J)}: \forall f\in\mathcal{P}_m: Q_m(f) = \int_a^b f(t)\,dt$$

(Exaktheit)

zu (H): Analog zu (G) erhalten wir

$$\prod_{j=0}^{m}(s-j) = (-1)^{m+1}\prod_{j=0}^{m}(m-s-j)$$

$$\overset{m\text{ gerade}}{=} -\prod_{j=0}^{m}(m-s-j)$$

$$= -(-1)^{m}\cdot\prod_{j=0}^{m}(s-j)$$

$$= (-1)^{m+1}\cdot\prod_{j=0}^{m}(s-j)$$

$$\Rightarrow \int_0^m \prod_{j=0}^{m}(s-j)\,ds = (-1)^{m+1}\int_0^m \prod_{j=0}^{m}(s-j)\,ds$$

$$\overset{m\text{ gerade}}{=} -\int_0^m \prod_{j=0}^{m}(s-j)\,ds$$

$$\Rightarrow 2\cdot\int_0^m \prod_{j=0}^{m}(s-j)\,ds = 0$$

zu (I):

$$p(t) \underset{\substack{\uparrow\\ \text{Newtonsche}\\ \text{Darstellung}}}{=} \sum_{j=0}^{m+1} a_j\cdot\prod_{i=0}^{j-1}(t-t_i) = \sum_{j=0}^{m} a_j \prod_{i=0}^{j-1}(t-t_i) + a_{m+1}\cdot\prod_{i=0}^{m}(t-t_i)$$

zu (J): Vgl. Vorlesung

Aus (H) - (J) erhalten wir nun

$$\int_a^b p(t)\,dt =$$

Beweis:

$$\int_a^b p(t)\,dt \underset{\substack{(I),\int\text{ lin.}\\ \text{Funktional}}}{=} \underbrace{\int_a^b \sum_{j=0}^{m} a_j \prod_{i=0}^{j-1}(t-t_i)\,dt}_{\in P_m} + \int_a^b a_{m+1}\prod_{i=0}^{m}(t-\underbrace{t_i}_{=a+\underbrace{\left(\frac{b-a}{m}\right)}_{=h}i})\,dt$$

$$\underset{\substack{(J)\,(\text{Exaktheit})\\ \text{Substitution:}\\ t=a+sh}}{=} Q_m\left(\sum_{j=0}^{m} a_j \prod_{i=0}^{j-1}(t-t_i)\right) + a_{m+1}\cdot h\cdot \underbrace{\int_0^m \prod_{i=0}^{m}(s-i)\,ds}_{\substack{=0\\ (H)\,(\text{da } m \text{ gerade})}}$$

$$\underset{(H)}{=} Q_m\left(\sum_{j=0}^{m} a_j \prod_{i=0}^{j-1}(t-t_i)\right) \quad \forall p\in P_{m+1}$$

zu 3. Die Quadraturformel Q aus Aufgabenteil (a) stellt keine (abgeschlossene) Newton-Cotes-Formel – wie sie in der Vorlesung behandelt wurde – dar, da die Ränder −1 und 1 nicht zu der Stützstellenmenge gehören. Sehr wohl stellt Q jedoch eine (offene) Newton-Cotes-Formel dar, da −1, 1 nicht zu der Stützstellenmenge gehören, alle übrigen Stützstellen in]-1,1[liegen und zueinander äquidistant sind.

AUFGABE 16
(Hauptfunktion)

```
function aufgabe16
    % ------------------
    % | Initialisierung |
    % ------------------
    f = @(t) -pi^3*t/((t^2+1)^2)*cos(25/(t^2+1))+5;
    F = @(t) pi^3/50*sin(25/(t^2+1))+5*t;
    a = -4;             % linke Intervallgrenze
                        % -3
    b = 2;              % rechte Intervallgrenze
                        % 3
    W = 30;             % Schaetzwert fuer das Gesamtintegral
    eps=10.^[-2,-7];    % geforderte relative Genauigkeit
    fehler = @(I) abs(F(b) - F(a) - I);

    % ------------
    % | Berechnung |
    % ------------
    for i=1:2
        % Aufgabenteil (a): (adaptives Quadraturverfahren)
        % (ohne Speichervorreservierung)
        tic;
        [ausgabe(i,1), ts] = adapt_quad(a,b,W,eps(i),f);
        ausgabe(i,4) = toc;
        % Aufgabenteil (b): (adaptives Quadraturverfahren)
        % (mit Speichervorreservierung)
        tic;
        ausgabe(i,2) = adapt_quad_res(a,b,W,eps(i),f);
        ausgabe(i,5) = toc;
        % Aufgabenteil (c): (rekursives adaptives Quadraturverfahren)
        % (rekursiv)
        tic;
        ausgabe(i,3) = adapt_quad_rek(a,b,a,b,W,eps(i),f);
        ausgabe(i,6) = toc;
        ausgabe(i,7) = length(ts);

        subplot(2,1,i)
        fplot(f,[a,b],'b');
        hold on
        plot(ts,zeros(size(ts)),'+r-');
        title(['Adaptive Quadratur mit relativer Genauigkeit epsilon=',num2str↙
(eps(i))]);
        legend('f(t)=-pi^3t/((t^2+1)^2) cos(25/(t^2+1))+5',...
            'Stuetzstellen',...
            'Location','NorthWest');
        xlabel('t');
        ylabel('f(t)');
    end
    I=ausgabe(1:2,1:3);
    ausgabe(:,1:3) =  fehler(ausgabe(:,1:3)); % Berechnung der Fehler

    % ---------
    % | Ausgabe |
    % ---------
    fprintf('\nAdaptive Quadratur\n');
    fprintf('------------------\n');
```

```
    fprintf('Exakter Integralwert : %19.14f\n\n',[fehler(0)]);
    fprintf('zu (a) adaptive Quadratur (ohne Speichervorreservierung)\n');
    fprintf('Relative Genauigkeit : %14.13e %14.13e\n',[eps(1), eps(2)]);
    fprintf('Num. Integralwert    : %19.14f %19.14f\n',[I(1,1), I(2,1)]);
    fprintf('Absolutfehler        : %14.13e %14.13e\n',[ausgabe(1,1), ausgabe
(2,1)]);
    fprintf('Zeit (in Sek.)       : %18.6f   %18.6f\n',[ausgabe(1,4), ausgabe
(2,4)]);
    fprintf('Anz. d. Stuetzstellen: %18.0f   %18.0f\n',[ausgabe(1,7), ausgabe
(2,7)]);
    fprintf('zu (b) adaptive Quadratur (mit Speichervorreservierung)\n');
    fprintf('Relative Genauigkeit : %14.13e %14.13e\n',[eps(1), eps(2)]);
    fprintf('Num. Integralwert    : %19.14f %19.14f\n',[I(1,2), I(2,2)]);
    fprintf('Absolutfehler        : %14.13e %14.13e\n',[ausgabe(1,2), ausgabe
(2,2)]);
    fprintf('Zeit (in Sek.)       : %18.6f   %18.6f\n',[ausgabe(1,5), ausgabe
(2,5)]);
    fprintf('Anz. d. Stuetzstellen: %18.0f   %18.0f\n',[ausgabe(1,7), ausgabe
(2,7)]);
    fprintf('zu (c) adaptive Quadratur (rekursiv)\n');
    fprintf('Relative Genauigkeit : %14.13e %14.13e\n',[eps(1), eps(2)]);
    fprintf('Num. Integralwert    : %19.14f %19.14f\n',[I(1,3), I(2,3)]);
    fprintf('Absolutfehler        : %14.13e %14.13e\n',[ausgabe(1,3), ausgabe
(2,3)]);
    fprintf('Zeit (in Sek.)       : %18.6f   %18.6f\n',[ausgabe(1,6), ausgabe
(2,6)]);
    fprintf('Anz. d. Stuetzstellen: %18.0f   %18.0f\n',[ausgabe(1,7), ausgabe
(2,7)]);
end
```

(a): adaptive Quadratur (ohne Speicher= vorreservierung)

```
function [I,t] = adapt_quad(a,b,W,epsilon,funct)
%ADAPT_QUAD adaptives Quadraturverfahren
%   a       : linke Intervallgrenze
%   b       : rechte Intervallgrenze
%   W       : Schaetzwert fuer das Gesamtintegral
%   epsilon : geforderte relative Genauigkeit
%   funct   : Funktion
%   I       : numerischer Integralwert
%   t       : adaptive Stuetzstellen

    % Initialisierung
    j = 1; k = 2; p = 2; I = 0;
    t(1) = a; t(2) = b;
    f(1) = funct(a); f(2) = funct(b);
    i(1) = 2;

    while 1==1
        % Berechnung
        h = t(k)-t(j);
        T = h*(f(k)+f(j))/2;
        tau = (t(k)+t(j))/2;
        y = funct(tau);
        S = (T+2*h*y)/3;
        r = abs((T-S)/W);

        % Genauigkeit
        if r > epsilon*h/(b-a);
            p = p+1;
            i(p) = k; i(j) = p;
            k = p;
            f(p) = y; t(p) = tau;
        else
            I = I+S;
            if k ~= 2
                j = k;
                k = i(k);
            else
                break;
            end
        end
    end
    t=t(1:p);
end
```

(b): adaptive Quadratur (mit Speicher=vorreservierung

```
function [I,t] = adapt_quad_res(a,b,W,epsilon,funct)
%ADAPT_QUAD adaptives Quadraturverfahren
%   a       : linke Intervallgrenze
%   b       : rechte Intervallgrenze
%   W       : Schaetzwert fuer das Gesamtintegral
%   epsilon : geforderte relative Genauigkeit
%   funct   : Funktion
%   I       : numerischer Integralwert
%   t       : adaptive Stuetzstellen

    % Initialisierung
    t=zeros(1,200000);
    i=zeros(1,200000);
    f=zeros(1,200000);

    j = 1; k = 2; p = 2; I = 0;
    t(1) = a; t(2) = b;
    f(1) = funct(a); f(2) = funct(b);
    i(1) = 2;
    h = 0;

    while 1==1
        % Berechnung
        h = t(k)-t(j);
        tau = (t(k) + t(j))/2; y = funct(tau);
        T = h*(f(k)+f(j))/2;
        S = (T + 2*h*y)/3;
        r = abs((T-S)/W);

        % Genauigkeit
        if r > epsilon*h/(b-a);
            p = p + 1;
            i(p) = k; i(j) = p;
            k = p;
            t(p) = tau; f(p) = y;
        else
            I = I + S;

            if k ~= 2
                j = k;
                k = i(k);
            else
                break;
            end
        end
    end
    t=t(1:p);
end
```

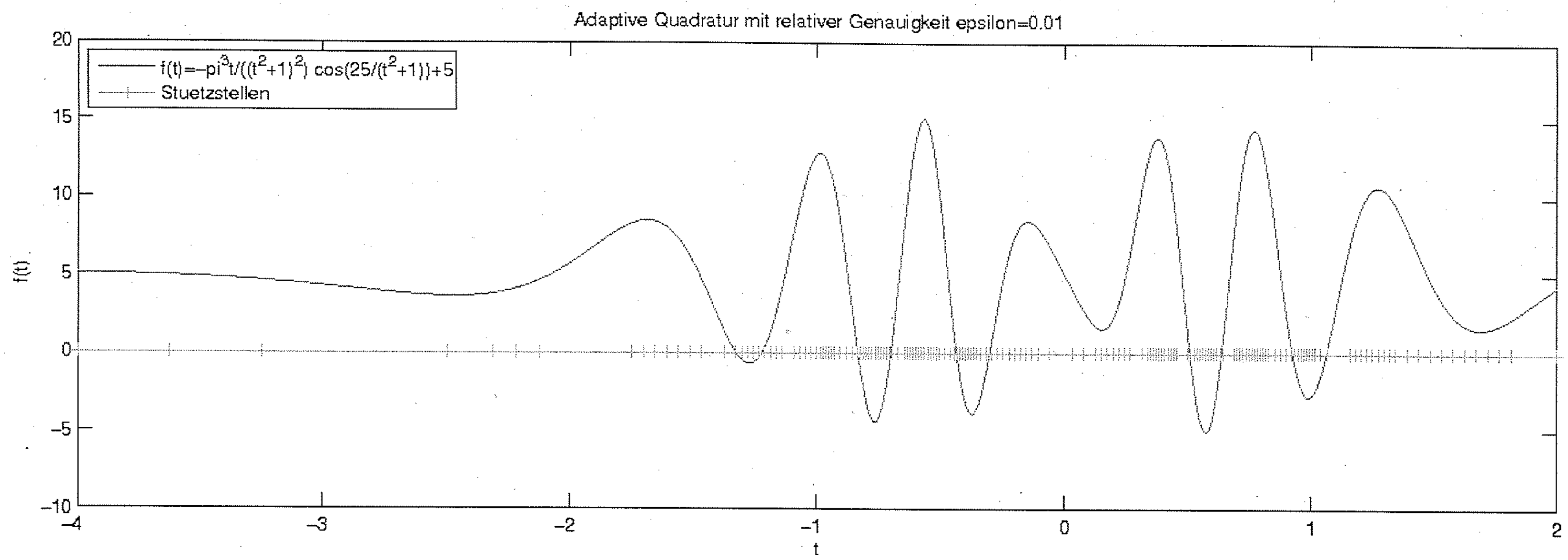
(c): adaptive Quadratur (rekursiv)

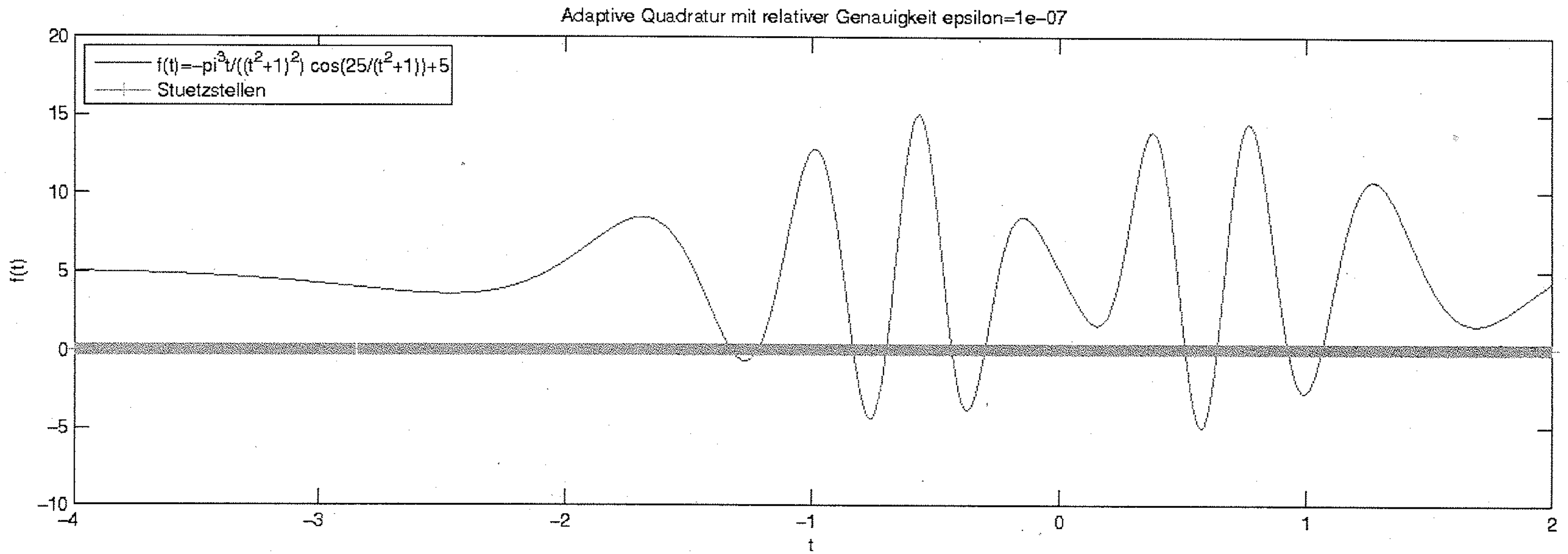
```
function I = adapt_quad_rek(x,y,a,b,W,epsilon,funct)
%ADAPT_QUAD_REK rekursive adaptive Quadratur
%  x        : linke Stuetzstelle
%  y        : rechte Stuetzstelle
%  a        : linke Intervallgrenze
%  b        : rechte Intervallgrenze
%  W        : Schaetzwert fuer das Gesamtintegral
%  epsilon  : geforderte relative Genauigkeit
%  funct    : Funktion
%  I        : numerischer Integralwert

    if abs( (T(x,y,funct)-S(x,y,funct) )/W) <= epsilon*(y-x)/(b-a)
        I = S(x,y,funct);
    else
        I = adapt_quad_rek(x,(x+y)/2,a,b,W,epsilon,funct) ...
            + adapt_quad_rek((x+y)/2,y,a,b,W,epsilon,funct);
    end
end

function z=T(x,y,funct)
%T Trapezregel
%  x      : linke Stuetzstelle
%  y      : rechte Stuetzstelle
%  funct  : Funktion
%  z      : numerischer Integralwert

    z = (y-x)*(funct(x)+funct(y))/2;
end

function z=S(x,y,funct)
%S Simpsonregel
%  x      : linke Stuetzstelle
%  y      : rechte Stuetzstelle
%  funct  : Funktion
%  z      : numerischer Integralwert

    z = (y-x)*(funct(x)+4*funct((x+y)/2)+funct(y))/6;
end
```

(a)-(c): Ausgabe

```
Adaptive Quadratur
------------------
Exakter Integralwert :   28.78833197970165

zu (a) adaptive Quadratur (ohne Speichervorreservierung)
Relative Genauigkeit : 1.0000000000000e-02 1.0000000000000e-07
Num. Integralwert    :   28.78829947304732   28.78833197969336
Absolutfehler        : 3.2506654328301e-05 8.2849282989628e-12
Zeit (in Sek.)       :            0.001690            7.244859
Anz. d. Stuetzstellen:                 169               50239
zu (b) adaptive Quadratur (mit Speichervorreservierung)
Relative Genauigkeit : 1.0000000000000e-02 1.0000000000000e-07
Num. Integralwert    :   28.78829947304732   28.78833197969336
Absolutfehler        : 3.2506654328301e-05 8.2849282989628e-12
Zeit (in Sek.)       :            0.003612            0.143609
Anz. d. Stuetzstellen:                 169               50239
zu (c) adaptive Quadratur (rekursiv)
Relative Genauigkeit : 1.0000000000000e-02 1.0000000000000e-07
Num. Integralwert    :   28.78829947304731   28.78833197969357
Absolutfehler        : 3.2506654338960e-05 8.0824236192711e-12
Zeit (in Sek.)       :            0.017279            5.071415
Anz. d. Stuetzstellen:                 169               50239
```

Plots:


Adaptive Quadratur mit relativer Genauigkeit epsilon=0.01
f(t)=-pi^3t/((t^2+1)^2) cos(25/(t^2+1))+5
Stuetzstellen
f(t)
t
20
15
10
5
0
-5
-10
-4
-3
-2
-1
0
1
2


Adaptive Quadratur mit relativer Genauigkeit epsilon=1e-07
f(t)=-pi^3t/((t^2+1)^2) cos(25/(t^2+1))+5
Stuetzstellen
f(t)
t
20
15
10
5
0
-5
-10
-4
-3
-2
-1
0
1
2

Aufgabe 17

$$Q_m(f) := \sum_{j=1}^{m} \omega_j \cdot f(s_j) \quad , m \geq 1$$

wobei

s_j Nullstellen von $G_m(t) := \frac{m!}{(2m)!} \cdot \frac{d^m}{dt^m}(t^2-1)^m$, $m \geq 0$

(m-TES LEGENDRE-POLYNOM)

$\omega_j := \int_{-1}^{1} L_j(t)\,dt$ Gewichte mit $L_j(t) := \frac{\omega_j(t)}{\omega_j(t_j)}$, $j = 1, \dots, m$

(j-TES LAGRANGES-BASISPOLYNOM)

zz: $\forall p \in \mathcal{P}_{2m-1} : Q_m(p) = \int_{-1}^{1} p(t)\,dt$ (Exaktheit)

LÖSUNG: Definiere

$$q_m(t) := (t^2-1)^m \in \mathcal{P}_{2m}$$

$$\Rightarrow G_m(t) = \frac{m!}{(2m)!} \cdot q_m^{(m)}(t) \in \mathcal{P}_m$$

zu (a):

zz.: $\exists\, s_1, \dots, s_m \in]-1,1[$ mit $s_1 < \dots < s_m$: $G_m(s_j) = 0 \quad \forall j = 1, \dots, m$

Offenbar besitzt das Polynom $q^{(k)} := q_m^{(k)}$ bei -1 und 1 jeweils eine $(m-k)$-fache Nullstelle und k weitere Nullstellen $-1 < \xi_1^{(k)} < \dots < \xi_k^{(k)} < 1$.

BEWEIS: (Induktion über k)

(IV): $q^{(k)} := q_m^{(k)}$ hat (1) in -1 und 1 jeweils eine $(m-k)$-fache Nullstelle und (2) in $]-1,1[$ k weitere Nullstellen $-1 < \xi_1^{(k)} < \dots < \xi_k^{(k)} < 1$

(IA): $(k=0)$

$$q^{(0)}(t) = q_m^{(0)}(t) = (t^2-1)^m = (t-1)^m \cdot (t+1)^m$$

$\Rightarrow -1$ und 1 sind m-fache Nullstellen von $q^{(0)}$

$\Rightarrow$ Es gibt keine weiteren Nullstellen $\xi \in]-1,1[$ von $q^{(0)}$

$q^{(0)} \in \mathcal{P}_{2m}$

(IS): $(k \rightarrow k+1)$. Nach (IV) gilt

$q^{(k+1)} = q_m^{(k+1)}$ hat $(m-(k+1))$-fache Nullstellen bei -1 und 1

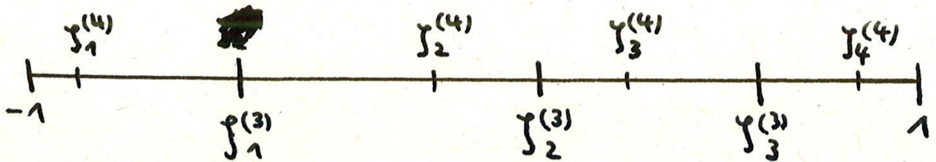
> **SATZ VON ROLLE:** $a, b \in \mathbb{R}$ mit $a < b$, $f : [a,b] \rightarrow \mathbb{R}$ stetig auf $[a,b]$, differenzierbar auf $]a,b[$ und $f(a) = f(b)$.
> $$\Rightarrow \exists\, \xi \in]a,b[: f'(\xi) = 0$$

Sei $j = 0, \dots, k$ beliebig und $\xi_0^{(k)} := -1$, $\xi_{k+1}^{(k)} := 1$. Aus dem Satz von Rolle folgt (mit $a := \xi_j^{(k)}$, $b := \xi_{j+1}^{(k)}$, $q^{(k)} := f$, $q^{(k)}(\xi_j^{(k)}) = q^{(k)}(\xi_{j+1}^{(k)}) = 0$)

$$\exists\, \xi_{j+1}^{(k+1)} \in]\xi_j^{(k)}, \xi_{j+1}^{(k)}[: q^{(k+1)}(\xi_{j+1}^{(k+1)}) = 0 \quad \forall j = 0, \dots, k$$

$$\Rightarrow \exists\, \xi_1^{(k+1)}, \dots, \xi_{k+1}^{(k+1)} \in]-1,1[\text{ mit } \xi_1^{(k+1)} < \dots < \xi_{k+1}^{(k+1)} : q^{(k+1)}(\xi_j^{(k+1)}) = 0$$

$\forall j = 1, \dots, k+1$ ∎



Setze nun $s_j := \xi_j^{(m)}$, $j = 1, \dots, m$, dann gilt $G_m(s_j) = 0 \ \forall j = 1, \dots, m$.

Beachte: $G_m \in \mathcal{P}_m \Rightarrow \exists$ höchstens m-Nullstellen von G_m (d.h. es kann keine weiteren geben) zu (b):

$$\text{zz.}: \int_{-1}^{1} h_k(t) h_l(t)\,dt = 0 \quad \forall 0 \le k < l \le m$$

Nach Teil (a) gilt (vgl. Ⓘⓥ)

$$Ⓐ: \; q_l^{(i)}(1) = q_l^{(i)}(-1) = 0 \quad \forall 0 \le i < l \le m$$

$$Ⓑ: \; q_k^{(i)}(1) = q_k^{(i)}(-1) = 0 \quad \forall 0 \le i < k \le m$$

Aus Ⓐ und Ⓑ folgt

$$\int_{-1}^{1} h_k(t) h_l(t)\,dt = 0 \quad \forall 0 \le k < l \le m$$

BEWEIS:

$$\int_{-1}^{1} h_k(t) h_l(t)\,dt$$

$$= \frac{k!\cdot l!}{(2k)!\cdot(2l)!} \cdot \int_{-1}^{1} q_k^{(k)}(t)\cdot q_l^{(l)}(t)\,dt$$

$$\underset{\substack{k\text{-fache}\\ \text{part. Integ.}}}{=} \frac{k!\cdot l!}{(2k)!\cdot(2l)!} \cdot \Bigg[(-1)^k \cdot \int_{-1}^{1} \underbrace{q_k^{(2k)}(t)}_{\substack{= c \in \mathbb{R}\\ \uparrow\\ q_k \in \mathcal{P}_{2k}}} \cdot q_l^{(l-k)}(t)\,dt + \sum_{j=0}^{k} (-1)^j \cdot \Big[\overbrace{q_k^{(k+j-1)}(1)}^{\substack{Ⓑ\\ =0,\, j=0}} \cdot \overbrace{q_l^{(l-j)}(1)}^{\substack{Ⓐ\\ =0,\, j=1,\dots,k}} - \underbrace{q_k^{(k+j-1)}(-1)}_{\substack{=0,\, j=0\\ Ⓑ}} \cdot \underbrace{q_l^{(l-j)}(-1)}_{\substack{=0,\, j=1,\dots,k\\ Ⓐ}}\Big]\Bigg]$$

$$= \frac{(-1)^k \cdot c \cdot k!\cdot l!}{(2k)!\cdot(2l)!} \cdot \int_{-1}^{1} q_l^{(l-k)}(t)\,dt$$

$$= \frac{(-1)^k \cdot c \cdot k!\cdot l!}{(2k)!\cdot(2l)!} \cdot \Big[\underbrace{q_l^{(l-k-1)}(1)}_{=0,\text{ wg. Ⓐ}} - \underbrace{q_l^{(l-k-1)}(-1)}_{=0,\text{ wg. Ⓐ}}\Big] = 0$$

Bemerke: Damit bildet $\{h_i\}_{i=0,\dots,m}$ eine Basis im euklidischen Vektorraum $\mathcal{P}_m$ mit Skalarprodukt

$$\langle f,g \rangle := \int_{-1}^{1} f(t) g(t)\,dt \quad \forall f,g \in \mathcal{P}_m$$

zu (c):

$$\text{zz}: \int_{-1}^{1} h_m(t) f(t)\,dt = 0 \quad \forall f \in \mathcal{P}_{m-1}$$

BEWEIS: Sei $f \in \mathcal{P}_{m-1}$ beliebig. Da $\{h_i\}_{i=0,\dots,m}$ eine Basis von $\mathcal{P}_m$ sind, gilt:

$$\exists\, \alpha_0,\dots,\alpha_{m-1} \in \mathbb{R}: \; f(t) = \sum_{i=0}^{m-1} \alpha_i h_i(t)$$

Daraus folgt:

$$\int_{-1}^{1} h_m(t)\cdot f(t)\,dt = \int_{-1}^{1} h_m(t) \cdot \sum_{i=0}^{m-1} \alpha_i h_i(t)\,dt$$

$$\underset{\substack{\uparrow\\ \int \text{lineares}\\ \text{Funktional}}}{=} \sum_{i=0}^{m-1} \alpha_i \underbrace{\int_{-1}^{1} h_m(t)\cdot h_i(t)\,dt}_{\substack{=0 \;(\text{da } 0 \le i < m \;\forall i=0,\dots,m-1)\\ (b)}} = 0 \quad \forall f \in \mathcal{P}_{m-1}$$

zu (d):

$$\text{zz}: \forall p \in \mathcal{P}_{2m-1} \;\exists\, q, r \in \mathcal{P}_{m-1}: \; p(t) = h_m(t)\cdot q(t) + r(t)$$

BEWEIS: (Polynomdivision) Aus der Polynomdivision

$$p(t) : h_m(t) = q(t) \;\text{ Rest } r(t)$$

erhalten wir ($\text{grad}(q) \le \text{grad}(p) - \text{grad}(h_m) = (2m-1) - m = m-1$) ein Polynom $p \in \mathcal{P}_{m-1}$ sowie ein Restpolynom $r \in \mathcal{P}_{m-1}$ mit

$$p(t) = h_m(t)\cdot q(t) + r(t)$$

zu (e):

zz: $\forall p \in \mathcal{P}_{2m-1} : Q_m(p) = \int_{-1}^{1} p(t)\,dt$ (Exaktheit)

Dies folgt nun unmittelbar aus

BEWEIS:

$$Q_m(p) \overset{(d)}{=} Q_m(h_m \cdot q + r) \overset{\text{Qm lineares Funktional}}{=} Q_m(h_m \cdot q) + Q_m(r)$$

$$\overset{\text{Def.}}{=} \sum_{j=1}^{m} \omega_j \cdot \underbrace{h_m(\xi_j)}_{\underset{(a)}{=}0\ \forall j=1,\ldots,m} \cdot q(\xi_j) + Q_m(r) \overset{\text{Exaktheit in } \mathcal{P}_m}{=} \int_{-1}^{1} r(t)\,dt$$

$$\underset{\substack{(c) \text{ mit} \\ q \in \mathcal{P}_{m-1}}}{=} \int_{-1}^{1} h_m(t) \cdot q(t)\,dt + \int_{-1}^{1} r(t)\,dt \underset{\int \text{ linea. Funktional}}{\overset{(d)}{=}} \int_{-1}^{1} p(t)\,dt \qquad \forall p \in \mathcal{P}_{2m-1}$$