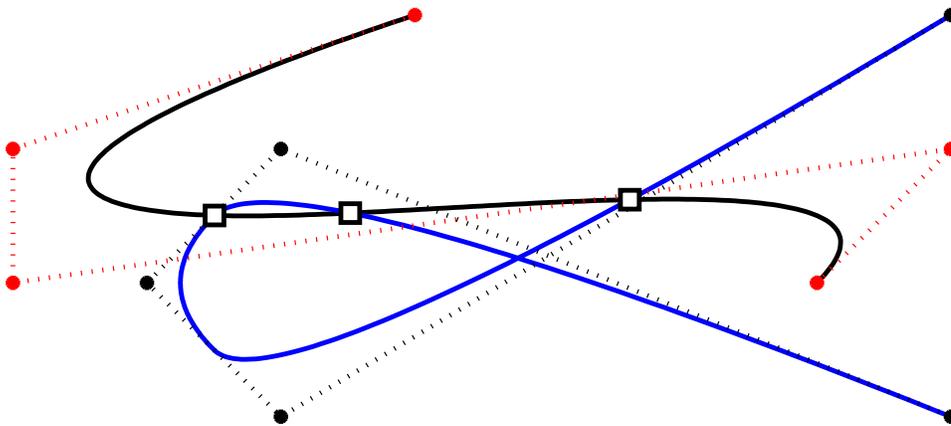

Stefan Funken

Numerik II
(Einführung in die Numerische Analysis)



Danksagung. Herrn Moritz Reinhard bin ich für das sorgfältige Lesen des Manuskripts zu besonderem Dank verpflichtet. Seine zahlreichen Kommentare, Vorschläge, Korrekturen und Hinweise haben die Qualität des Textes wesentlich verbessert. Insbesondere möchte ich ihm für die Ausarbeitung der Aufgaben und zugehöriger Lösungen danken.

Ganz besonderer Dank gebührt auch Frau Petra Hildebrand, die meine handschriftlichen Aufzeichnungen in \LaTeX umgesetzt und zahlreiche Grafiken erstellt hat.

Die aufmerksame Leserin oder Leser sei ermuntert, mir Hinweise auf Druckfehler, sprachliche Unzulänglichkeiten oder inhaltliche Flüchtigkeiten per Email zukommen zu lassen (`stefan.funken@uni-ulm.de`).

Copyright. Alle Rechte, insbesondere das Recht auf Vervielfältigung und Verbreitung sowie der Übersetzung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form ohne schriftliche Genehmigung des Autors reproduziert oder unter Verwendung elektronischer Systeme oder auf anderen Wegen verarbeitet, vervielfältigt oder verbreitet werden.

Stand. Ulm, April 2009.

Inhaltsverzeichnis

1	Nichtlineare Gleichungen	1
1.1	Bisektionsmethode	3
1.2	Regula-Falsi ¹	5
1.3	Die Sekantenmethode	7
1.4	Das Verfahren von <i>Newton</i>	9
1.5	Das <i>Broyden</i> -Verfahren	19
2	Orthogonalpolynome und Gauß-Quadratur	23
2.1	Allgemeine Charakteristika	23
2.2	<i>Tschebyscheff</i> -Polynome	33
2.3	<i>Legendre</i> -Polynome	34
2.4	<i>Jacobi</i> -Polynome	36
2.5	Nullstellen von Orthogonalpolynomen	37
2.6	Radau- und Lobatto-Formeln	43
2.7	Kronrod Formeln	44
2.8	Konkrete Auswertung einer Linearkombination von Orthogonalpolynomen	44
	2.8.Vorwärtsrekursion	45
	2.8.Rückwärtsrekursion	45
3	Interpolation	47
3.1	Klassische Polynom-Interpolation	47
3.2	<i>Hermite</i> -Interpolation und dividierte Differenzen	49
3.3	<i>Tschebyscheff</i> -Interpolation	58
3.4	Rationale Interpolation	60
3.5	<i>Padé</i> -Approximation	63
4	Splines	67
4.1	Kubische Spline-Interpolation	68
4.2	Punktauswertung kubischer Splines	75

4.3	Parametrisierte Kurven und Flächen	78
4.4	Bernstein-Polynome und Bézier-Kurven	80
4.5	B-Splines	88
4.6	Rationale B-Splines	98
4.7	Grundlegende Algorithmen	100
4.8	B-Spline Interpolation	105
5	Quadraturtheorie	107
5.1	Mittelpunkt- und Trapezregel	107
5.2	<i>Newton-Cotes-Formeln</i>	109
5.3	Schwierigkeiten bei der Quadratur	114
	5.3.1 Unstetige Integranden	114
	5.3.2 Singuläre Integrale	114
5.4	Adaptive Quadratur	116
5.5	Extrapolation	117
5.6	Numerische Quadratur von stark oszillierenden Integranden	118
A	Lineare Differenzgleichung	123
A.1	Inhomogene lineare Differenzgleichungen	126
B	Aufgaben und Lösungen	129
C	Aufgaben und Lösungen	153
	Literaturverzeichnis	176
	Stichwortverzeichnis	178

1 NICHTLINEARE GLEICHUNGEN

Nachdem wir uns in der Vorlesung Numerik I mit dem Lösen linearer Gleichungssysteme beschäftigt haben, wenden wir uns nun nichtlinearen Gleichungen (in einer oder mehreren Variablen) zu. Diese Gleichungen treten häufig als Teilaufgabe bei der Behandlung komplexerer Probleme auf.

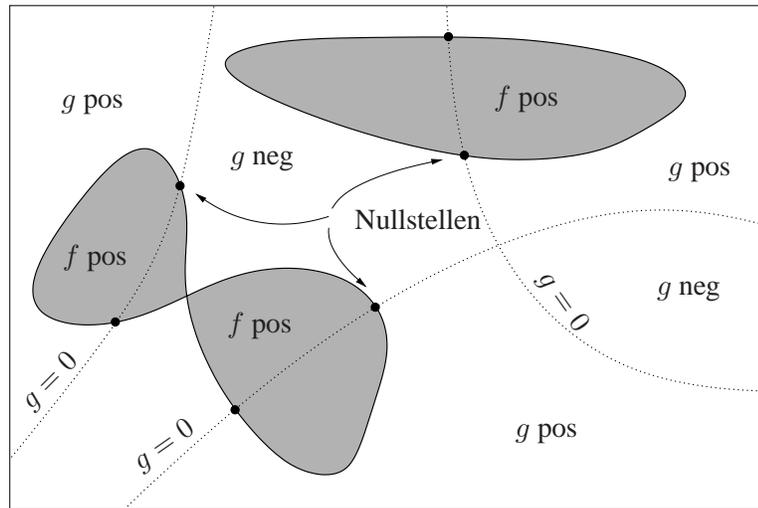


Abb. 1.1: Lösung von zwei Gleichungen in zwei Unbekannten. Die durchgezogenen Linien sind Niveaulinien zu $f(x, y) = 0$, die gestrichelten Linien zu $g(x, y) = 0$. Die gesuchten Lösungen sind die Schnittpunkte der völlig unabhängigen Nulllinien. Die Anzahl der Nullstellen ist im Allgemeinen a-priori nicht bekannt.

Ist ein System von n nichtlinearen Gleichungen in n Unbekannten gegeben, d.h. mit einer stetigen, nichtlinearen Funktion

$$\tilde{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n,$$

so können wir das gegebene Problem $\tilde{f}(x) = b$ in eine **Nullstellenaufgabe** transformieren, so dass Lösungen $x \in \mathbb{R}^n$ der Gleichung

$$f(x) := \tilde{f}(x) - b = 0 \tag{1.1}$$

zu bestimmen sind. Mehrere Modellbeispiele, bei denen nichtlineare Gleichungen auftreten, wollen wir hier vorstellen.

Beispiel 1.0.1 (Zinssatz bei einem Kredit) Wie hoch darf der Zinssatz sein, wenn man einen Kredit über 10.000 Euro in 10 Jahren abzahlen möchte und man höchstens 400 Euro monatlich aufbringen kann? Oder allgemeiner eine Kreditsumme K_0 in n Jahren mit monatlichen Raten R getilgt habe möchte?

Der Einfachheit halber rechnen wir den jährlichen Zinssatz p in einen monatlichen Zinssatz m um, d.h. verzinst man monatlich mit einem Prozentsatz m oder jährlich mit p , so erhält man am Ende des Jahres jeweils das Gleiche. Die Beziehung zwischen p und m ist also $(1 + m)^{12} = 1 + p$.

Für den Restbetrag K_i des Kredits nach $i \in \mathbb{N}$ Monaten gilt:

$$\begin{aligned} K_i &= K_{i-1}(1+m) - R \\ K_{i+1} &= K_i(1+m) - R = [K_{i-1}(1+m) - R](1+m) - R \\ &= K_{i-1}(1+m)^2 - R[(1+m) + 1] \\ &\vdots \\ K_n &= K_0(1+m)^n - R[(1+m)^{n-1} + \dots + (1+m) + 1] \\ &= K_0(1+m)^n - R[(1+m)^n - 1]/m \\ &= (K_0 - R/m)(1+m)^n + R/m \end{aligned}$$

Zu gegebener Kreditsumme K_0 , monatlichem Zinssatz m und Tilgungsdauer (n Monate) berechnet sich die Rate R , um den Kredit vollständig zu tilgen, indem wir $K_n = 0$ setzen, d.h.

$$R = K_0 m (1+m)^n / [(1+m)^n - 1].$$

Auch die Tilgungsdauer lässt sich analytisch darstellen

$$n = \frac{\log(R/(R - mK_0))}{\log(1+m)}.$$

(Wie ist hier das Ergebnis zu interpretieren, falls n nicht ganzzahlig ist?) Wie gewinnt man jedoch m zu gegebenen $K_0 > 0$, $n \in \mathbb{N}$, $K_0 > R > K_0/n$ aus der Gleichung

$$(mK_0 - R)(1+m)^n + R = 0?$$

Es ist offensichtlich, dass $0 < m < 1$ gilt und $f(m) := (mK_0 - R)(1+m)^n + R$ nur eine Nullstelle in $(0, 1)$ hat. Aber wie kann man diese einfach, schnell und numerisch stabil bestimmen?

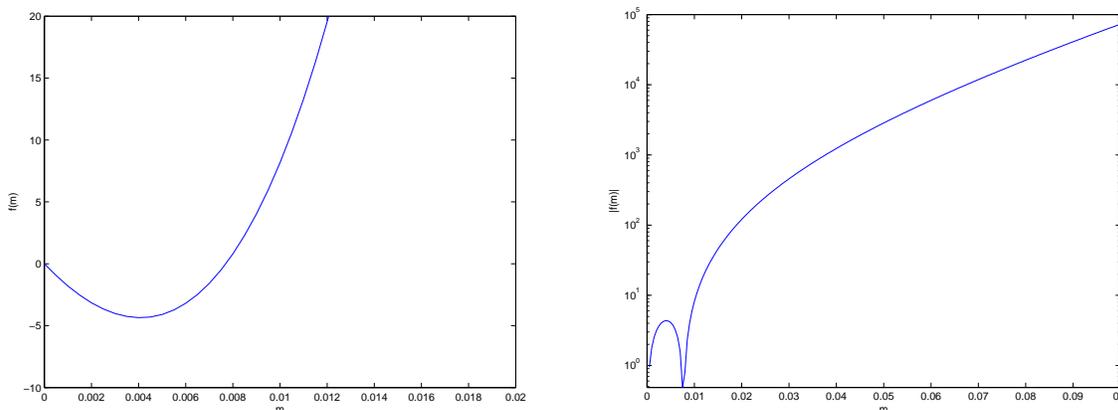


Abb. 1.2: Der Graph von $f(m) := (mK_0 - R)(1+m)^n + R$ (links, y-Achse linear skaliert, $x \in [0, 0.02]$) bzw. $|f(m)|$ (rechts, logarithmisch-skaliert, $x \in [0, 0.1]$) für $K_0 = 10000$, $R = 250$ und $n = 48$. Die Funktion f hat eine Nullstelle bei 0 und bei ≈ 0.008 .

Beispiel 1.0.2 (Nullstellen von Orthogonalpolynomen) Die **Legendre-Polynome** $P_n \in \mathbb{P}_n$ ($n = 0, 1, 2, \dots$) erfüllen die Drei-Term-Rekursion

$$P_0(x) = 1, P_1(x) = x, (n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), n \in \mathbb{N}.$$

Die ersten Legendre-Polynome lauten

$$\begin{aligned} P_0 &= 1 & P_3 &= \frac{1}{2}(5x^3 - 3x) \\ P_1 &= x & P_4 &= \frac{1}{8}(35x^4 - 30x^2 + 3) \\ P_2 &= \frac{1}{2}(3x^2 - 1) & P_5 &= \frac{1}{8}(63x^5 - 70x^3 + 15x). \end{aligned}$$

Die Nullstellen der Legendre-Polynome liegen in $(-1, 1)$ und die Nullstellen von P_n trennen die Nullstellen von P_{n+1} ($n \in \mathbb{N}$). Mit Hilfe dieser Eigenschaft lassen sich sukzessive Intervalle finden, in denen Nullstellen liegen, z.B. P_1 hat die Nullstelle $\xi_1^{(1)} = 0$ und somit liegen die Nullstellen $\xi_1^{(2)}, \xi_2^{(2)}$ von P_2 in $(-1, 0)$ und $(0, 1)$, die Nullstellen $\xi_1^{(3)}, \xi_2^{(3)}, \xi_3^{(3)}$ von P_3 in $(-1, \xi_1^{(2)})$, $(\xi_1^{(2)}, \xi_2^{(2)})$ und $(\xi_2^{(2)}, 1)$. Diese Eigenschaft der Legendre-Polynome gilt auch für weitere Orthogonalpolynome (siehe Satz 2.1.27 in Kapitel 2). Die Berechnung der Nullstellen ist u.a. wichtig im Zusammenhang mit Gauß-Quadraturformeln.

Beispiel 1.0.3 (Extremalstellen von skalaren Funktionen) Die mehrdimensionale Erweiterung der Rosenbrock¹-Funktion

$$f(x) = \sum_{i=1}^{n-1} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2] \quad (x \in \mathbb{R}^n)$$

hat für $n \geq 4$ mindestens ein lokales Minimum in der Umgebung von $(x_1, x_2, \dots, x_n) = (-1, 1, \dots, 1)$ neben dem globalen Minimum $(x_1, \dots, x_n) = (1, \dots, 1)$. Diese Extremalstellen erfüllen notwendigerweise die Gleichung $\nabla f = 0$. Wie kann man nun für $n \geq 4$ ein solches lokales Minimum bestimmen? Dies bedeutet, ein $x \in \mathbb{R}^n \setminus \{(1, \dots, 1)\}$ ist zu bestimmen mit

$$\begin{aligned} g(x) &= (g_1(x), \dots, g_n(x))^T = 0 \\ g_1(x) &:= \frac{\partial f}{\partial x_1} = -2(1 - x_1) - 400x_1(x_2 - x_1^2), \\ g_i(x) &:= \frac{\partial f}{\partial x_i} = -2(1 - x_i) - 400x_i(x_{i+1} - x_i^2) + 200(x_i - x_{i-1}^2) \quad (i = 2, \dots, n-1), \\ g_n(x) &:= \frac{\partial f}{\partial x_n} = 200(x_n - x_{n-1}^2). \end{aligned}$$

Betrachten wir dazu zunächst die Situation in einer Raumdimension.

1.1 BISEKTIONSMETHODE

Herleitung des Algorithmus: Diese Methode, motiviert durch Überlegungen aus der reellen ein-dimensionalen Analysis (siehe Intervallschachtelung), löst (1.1) dadurch, dass sie die Lösung durch systematisch kleiner werdende Intervalle einschließt. Man geht von der Annahme aus, es sei ein Intervall $I := [a, b]$ bekannt mit $f(a) \cdot f(b) < 0$.

Aus Stetigkeitsgründen existiert eine Lösung x^* im Inneren von I mit $f(x^*) = 0$.

Durch den Mittelpunkt $m = \frac{1}{2}(a + b)$ des Intervalls I wird der Funktionswert $f(m)$ bestimmt. Ist $f(m) \neq 0$ entscheidet nun das Vorzeichen, in welchem der Teilintervalle $[a, m]$, $[m, b]$ die gesuchte Lösung x^* liegt. Wir erhalten damit folgenden Algorithmus:

MATLAB-Funktion: BisektionsMethode.m

¹Howard Harry Rosenbrock, 16. Dez. 1920 - 21 Okt. 2010

```

1 function Nullstelle = BisektionsMethode(a,b,func,epsilon)
2 % Initialisierung
3 temp=[]
4 fa = func(a); fb = func(b);
5 while abs(a-b) > epsilon
6     m = (a+b)/2;
7     fm = func(m);
8     if fm == 0
9         Nullstelle = m;
10        return
11    elseif fa*fm < 0
12        b = m;
13        fb = fm;
14    else
15        a = m;
16        fa = fm;
17    end
18    temp = [temp;m];
19 end
20 % Lösung
21 Nullstelle = m;

```

MATLAB-Beispiel:

Testen wir nun die Bisektionsmethode anhand von Bsp. 1.0.1. Wie hoch darf der Zinssatz sein, wenn man einen Kredit über 10.000\$ in 48 Monaten zurückzahlen möchte, aber nur 250\$ monatlich aufbringen kann?

```

>> n = 48;
>> K0 = 10000;
>> R = 250;
>> f = @(m) (m*K0-R)*(1+m)^n+R;
>> m = Bisektionsmethode(eps,1,f,1e-7)
m =
    0.00770145654678
>> p = 100 * ((1+m)^12-1)
p =
    9.64343564476941

```

Nach der Umrechnung in den jährlichen Zinssatz sehen wir, dass wir uns den Kredit nur erlauben könnten, wenn der Zinssatz niedriger als 9.65% ist.

Definition 1.1.1 (Konvergenzgeschwindigkeit) Sei (x_k) eine reellwertige konvergente Folge mit $x_k \in \mathbb{R}^n$ ($k \in \mathbb{N}$) und Grenzwert $x \in \mathbb{R}^n$. Man bezeichnet (x_k) als **linear konvergent**, wenn es ein $0 < \rho < 1$ gibt mit

$$\|x - x_{k+1}\| \leq \rho \|x - x_k\| \quad (k = 0, 1, \dots).$$

Die Zahl ρ wird **Konvergenzfaktor** (oder auch **Kontraktionsrate**) genannt. Gibt es eine gegen Null konvergente Folge (ρ_k) mit

$$\|x - x_{k+1}\| \leq \rho_k \|x - x_k\| \quad (k = 0, 1, \dots),$$

so heißt (x_k) **superlinear konvergent**. Gibt es ein $\rho > 0$ und ein $1 < q \in \mathbb{R}$ mit

$$\|x - x_{k+1}\| \leq \rho \|x - x_k\|^q \quad (k = 0, 1, \dots),$$

so heißt (x_k) konvergent mit **Konvergenzordnung** q . Für $q = 2$ spricht man auch von **quadratischer Konvergenz**.

Bemerkung 1.1.2 (Konvergenz des Bisektionsverfahrens) Betrachten wir den Mittelpunkt x_k des Intervalls nach der k -ten Intervallhalbierung als Näherung an x^* , so gilt die a-priori Fehlerabschätzung

$$|x_k - x^*| \leq \frac{b-a}{2^{k+1}} \quad (k = 0, 1, 2, \dots).$$

Da die Fehlerschranke wie eine geometrische Folge abnimmt, beträgt die „Konvergenzordnung“ 1.

1.2 REGULA-FALSI¹

Herleitung des Algorithmus Wiederum gehen wir davon aus, dass ein Intervall $I := [a, b]$ mit $f(a) \cdot f(b) < 0$ bekannt sei. Anstatt nun I durch Hinzufügen eines Mittelpunkts in zwei Intervalle zu zerlegen, wählen wir eine zusätzliche Stelle ξ , die Nullstelle der Geraden durch $(a, f(a))$ und $(b, f(b))$. Mit den beiden Intervallen $[a, \xi]$, $[\xi, b]$ verfahren wir nun analog zur Methode der Intervallhalbierung, wobei gilt:

$$\xi = \frac{af(b) - bf(a)}{f(b) - f(a)}. \quad (1.2)$$

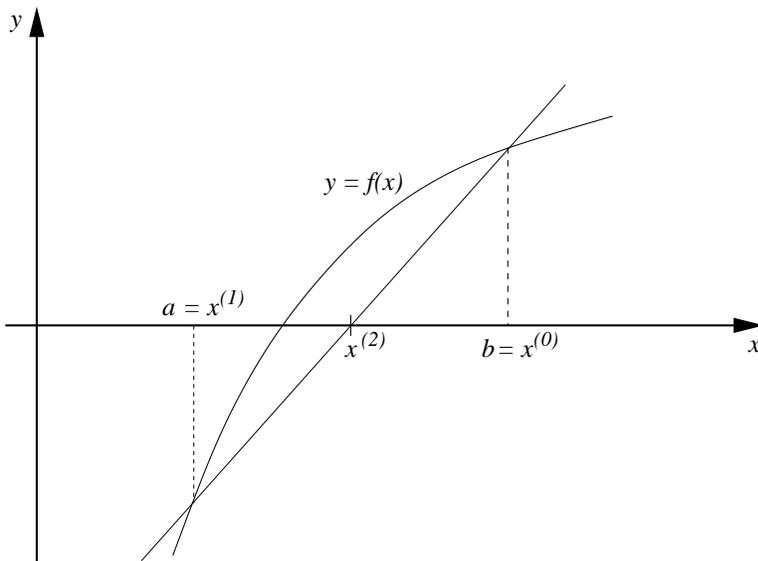


Abb. 1.3: Die geometrische Interpretation der Regula-Falsi-Methode.

Als Algorithmus ergibt sich somit:

MATLAB-Funktion: RegulaFalsi.m

```
1 function Nullstelle = RegulaFalsi(a,b,func,epsilon)
2 fa = func(a); fb = func(b); % Initialisierung
3 m = (a*fb-b*fa)/(fb-fa); fm = func(m);
4 while abs(fm) > epsilon
5     if fa*fm < 0
6         b = m;
```

¹lat.: „Regel des falschen Ansatzes“

```

7     fb = fm;
8     else
9         a = m;
10        fa = fm;
11    end
12    m = (a*fb-b*fa)/(fb-fa);
13    fm = func(m);
14    end
15    Nullstelle = m; % Lösung

```

Satz 1.2.1 Es sei x^* einzige Nullstelle von f in $I := [a, b]$, $f \in C^3(I)$ und $f'(x^*) \cdot f''(x^*) \neq 0$. Dann beträgt die Konvergenzordnung der Regula-Falsi-Methode $p = 1$.

Beweis. Es bezeichne (x_k) die sich aus obigem Algorithmus ?? ergebende Folge von „Mittelpunkten“, d.h. $x_k \in (a_k, b_k)$ ($k \in \mathbb{N}$). Es seien

$$\varepsilon_k^a := a_k - x^*, \quad \varepsilon_k^b := b_k - x^*.$$

Aus (1.2) und der Voraussetzung $f(x^*) = 0$ folgt

$$\begin{aligned} \varepsilon_k := x_k - x^* &= \frac{(a_k - x^*)f(b_k) - (b_k - x^*)f(a_k)}{f(b_k) - f(a_k)} \\ &= \frac{\varepsilon_k^a f(x^* + \varepsilon_k^b) - \varepsilon_k^b f(x^* + \varepsilon_k^a)}{f(x^* + \varepsilon_k^b) - f(x^* + \varepsilon_k^a)}. \end{aligned}$$

Da $f \in C^3(I)$ nach Voraussetzung gilt, liefert die Taylor²-Entwicklung:

$$\begin{aligned} \varepsilon_k &= \frac{\varepsilon_k^a \{ \varepsilon_k^b f'(x^*) + \frac{1}{2}(\varepsilon_k^b)^2 f''(x^*) + \dots \} - \varepsilon_k^b \{ \varepsilon_k^a f'(x^*) + \frac{1}{2}(\varepsilon_k^a)^2 f''(x^*) + \dots \}}{\{ \varepsilon_k^b f'(x^*) + \frac{1}{2}(\varepsilon_k^b)^2 f''(x^*) + \dots \} - \{ \varepsilon_k^a f'(x^*) + \frac{1}{2}(\varepsilon_k^a)^2 f''(x^*) + \dots \}} \\ &= \frac{\frac{1}{2} \varepsilon_k^a \varepsilon_k^b (\varepsilon_k^b - \varepsilon_k^a) f''(x^*) + \dots}{(\varepsilon_k^b - \varepsilon_k^a) \{ f'(x^*) + \frac{1}{2}(\varepsilon_k^b + \varepsilon_k^a) f''(x^*) + \dots \}} \\ &= \frac{\frac{1}{2} \varepsilon_k^a \varepsilon_k^b f''(x^*) + \dots}{f'(x^*) + \dots} \end{aligned}$$

Nach der Herleitung des Algorithmus gilt $a_k < x^* < b_k$. Demzufolge besitzen ε_k^a und ε_k^b entgegengesetztes Vorzeichen und es gilt weiter $\varepsilon_k^b - \varepsilon_k^a > 0$. Für hinreichend kleine $|\varepsilon_k^a|$ und $|\varepsilon_k^b|$ folgt damit

$$\varepsilon_k \approx \frac{f''(x^*)}{2f'(x^*)} \varepsilon_k^a \varepsilon_k^b. \quad (1.3)$$

Mit $f''(x^*) \neq 0$ gilt aber weiterhin, dass f in einer hinreichend kleinen Umgebung von x^* konvex oder konkav ist, und folglich bleibt ein Intervallende fest und wird im Verfahren nur umbenannt. Deshalb ist ε_k nur direkt proportional zu einem der beiden vorhergehenden ε_k^a oder ε_k^b . Die asymptotische Fehlerkonstante C ist für eine konkave Funktion gegeben durch

$$C = \left| \frac{f''(x^*)}{2f'(x^*)} \varepsilon_k^a \right|, \quad \varepsilon_k^a = \varepsilon_{k-1}^a = \dots = \varepsilon_1^a = x_1 - x^*,$$

und damit konvergiert die Folge (x_k) linear. Analoges gilt für konvexe Funktionen mit ε_k^b anstatt ε_k^a . \square

²Taylor, Brook (1685-1731)

MATLAB-Beispiel:

```

Testen wir nun das Regula-Falsi- >> n = 48;
Verfahren anhand Bsp. 1.0.1 K0 = >> K0 = 10000;
10000, n = 48 und R = 250. >> R = 250;
>> f = @(m) (m*K0-R)*(1+m)^n+R;
>> m = RegulaFalsi(1e-5,0.1,f,1e-7)
m =
    0.00770147244890

```

Bei einem genaueren Vergleich mit dem Bisektionsverfahren stellt man bei diesem Beispiel eine langsamere Konvergenz des Regula-Falsi-Verfahrens fest.

1.3 DIE SEKANTENMETHODE

Als Modifikation des Regula-Falsi-Verfahrens verzichtet man bei der Sekantenmethode darauf, die Lösung durch zwei Näherungswerte einzuschließen.

Zu zwei vorgegebenen Näherungswerten x_0 und x_1 , welche die Lösung nicht notwendigerweise einzuschließen brauchen, bestimmt man x_2 als Nullstelle der Sekante zu $(x_0, f(x_0))$ und $(x_1, f(x_1))$. Ungeachtet der Vorzeichen bestimmt man aus x_1, x_2 eine nächste Näherung x_3 .

Die Iterationsvorschrift der Sekantenmethode ergibt sich damit zu

$$x_{k+1} = x_k - f(x_k) \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \quad (k = 1, 2, \dots). \quad (1.4)$$

Dieses zweistufige Iterationsverfahren setzt natürlich $f(x_k) \neq f(x_{k-1})$ voraus und gehört nicht zur Klasse der oben betrachteten Iterationsverfahren.

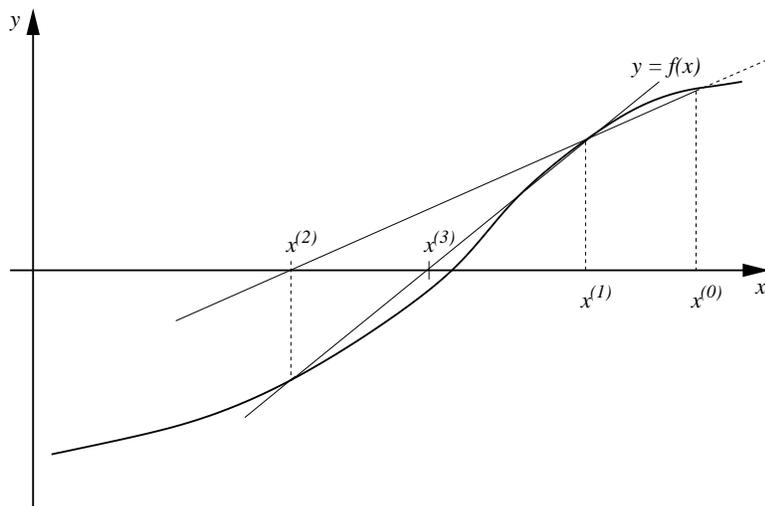


Abb. 1.4: Die geometrische Interpretation der Sekantenmethode.

Satz 1.3.1 Falls $f'(x^*) \cdot f''(x^*) \neq 0$ gilt, ist die Konvergenzordnung der Sekantenmethode $p = \frac{1}{2}(1 + \sqrt{5})$.

Beweis. Wir betrachten die Sekantenmethode als Modifikation des Regula-Falsi-Verfahrens. Für hinreichend kleine Fehler $|\varepsilon_{k-1}|$ und $|\varepsilon_k|$ bleibt (1.3) für die Sekantenmethode gültig, bzw. geht

über in

$$\varepsilon_{k+1} \approx \frac{f''(x^*)}{2f'(x^*)} \varepsilon_k \varepsilon_{k-1}. \quad (1.5)$$

Es besteht jedoch der Unterschied, dass bei Erhöhung von k sich beide Werte ε_{k-1} und ε_k ändern. Mit der Konstanten $C := |f''(x^*)/(2f'(x^*))|$ gilt für hinreichend großes k

$$|\varepsilon_{k+1}| \approx C |\varepsilon_k| \cdot |\varepsilon_{k-1}|.$$

Nach unserer Definition der Konvergenzordnung versuchen wir nun diese **Differenzengleichung** mit dem Ansatz

$$|\varepsilon_k| = \rho |\varepsilon_{k-1}|^p, \quad \rho > 0, \quad p \geq 1$$

zu lösen. Einsetzen ergibt

$$(|\varepsilon_{k+1}| =) \quad \rho |\varepsilon_k|^p = \rho \cdot \rho^p |\varepsilon_{k-1}|^{p^2} \stackrel{!}{=} C \cdot \rho |\varepsilon_{k-1}|^{p+1} \quad (= C |\varepsilon_k| \cdot |\varepsilon_{k-1}|).$$

Diese letzte Gleichung kann aber für alle (hinreichend großen) k nur dann gelten, falls

$$\rho^p = C \quad \text{und} \quad p^2 = p + 1$$

erfüllt sind. Die positive Lösung $p = \frac{1}{2}(1 + \sqrt{5})$ der quadratischen Gleichung ist deshalb die Konvergenzordnung der Sekantenmethode. Die asymptotische Fehlerkonstante ist $\rho = C^{1/p} = C^{0.618}$. \square

MATLAB-Funktion: SekantenMethode.m

```

1 function x1 = SekantenMethode(x0,x1,func,epsilon)
2 % Initialisierung
3 fx0 = func(x0); fx1 = func(x1);
4 while abs(fx1) > epsilon && abs(fx0-fx1) > epsilon
5     tmp = x1;
6     x1 = x1 - fx1 *(x1-x0)/(fx1-fx0);
7     x0 = tmp;
8     fx0 = fx1;
9     fx1 = func(x1);
10 end

```

MATLAB-Beispiel:

Testen wir nun abschließend die Sekantenmethode an Bsp. 1.0.1 mit $K_0 = 10000$, $n = 48$ und $R = 250$.

```

>> n = 48;
>> K0 = 10000;
>> R = 250;
>> f = @(m) (m*K0-R)*(1+m)^n+R;
>> m = SekantenMethode(1e-2,0.1,f,1e-7)
m =
    0.00770147248822

```

Im Vergleich zu den oben gemachten Tests ist hier das Startintervall kleiner zu wählen.

1.4 DAS VERFAHREN VON *Newton*

Herleitung des Algorithmus Ist die gegebene Funktion $f(x)$ der zu lösenden Gleichung $f(x) = 0$ stetig differenzierbar, so wird im Verfahren von *Newton*³ die Funktion $f(x)$ im Näherungswert x_k linearisiert und der iterierte Wert x_{k+1} als Nullstelle der Tangente in x_k definiert (vgl. Abbildung 1.5).

Aus der Tangentengleichung

$$y(x) = f(x_k) + (x - x_k)f'(x_k)$$

ergibt sich die Iterationsvorschrift

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \quad (1.6)$$

Bemerkung 1.4.1 Die Methode von *Newton* gehört zur Klasse der Fixpunktiterationen mit der

³Newton, Isaac (1642-1727)

gilt und $f \in C^3(I)$ vorausgesetzt wurde, existiert ein $C \in \mathbb{R}$ mit $M_k \leq C$ (für $k = 0, 1, 2, \dots$) und somit ist das *Newton*-Verfahren quadratisch konvergent. \square

Betrachten wir nun den allgemeinen Fall von Systemen nichtlinearer Gleichungen ($n \geq 2$). Die *Jacobi*-Matrix oder Funktionalmatrix der Abbildung $f = (f_1, \dots, f_n) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ist gegeben durch

$$J_f(x) := \begin{pmatrix} \nabla f_1 \\ \vdots \\ \nabla f_n \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}.$$

Die *Taylor*-Entwicklung von f um einen Startwert $x^{(0)}$ ergibt in diesem Fall

$$0 = f(x^*) = \underbrace{f(x^{(0)}) + J_f(x^{(0)})(x^* - x^{(0)})}_{=: \bar{f}(x)} + o(\|x^* - x^{(0)}\|) \quad \text{für } x^* \rightarrow x^{(0)}.$$

Die Nullstelle $x^{(1)}$ der linearisierten Abbildung $\bar{f}(x)$ ist jedoch gerade

$$x^{(1)} = x^{(0)} - J_f(x^{(0)})^{-1} f(x^{(0)}),$$

falls $\det(J_f(x^{(0)})) \neq 0$ erfüllt ist.

Dies motiviert die *Newton*-Iteration

$$J_f(x^{(k)}) s^{(k)} = -f(x^{(k)}) \quad (1.7)$$

$$x^{(k+1)} = x^{(k)} + s^{(k)}. \quad (1.8)$$

MATLAB-Funktion: NewtonSimple.m

```

1 function [x,nit] = NewtonSimple(x,f,Df,tol,maxit,param)
2 nit = 0;
3 fx = f(x,param);
4 while norm(fx) > tol && nit <= maxit
5     nit = nit+1;
6     x = x-Df(x,param)\fx;
7     fx = f(x,param);
8 end
9 nit

```

Bemerkung 1.4.3 Wir haben die numerische Lösung eines nichtlinearen Gleichungssystems auf die numerische Lösung einer Folge von linearen Gleichungssystemen übertragen.

Beispiel 1.4.4 Vergleichen wir nun die Bisektionsmethode, das Sekantenverfahren und das *Newton*-Verfahren angewandt auf das Problem aus Bsp. 1.0.1 mit $K_0 = 10000$, $n = 48$ und $R = 250$ sowie Anfangsbedingungen $a = \text{eps}$, $b = 1$ für die Bisektionsmethode, $x_0 = 1/2$, $x_1 = 1$ für das Sekantenverfahren und $x_0 = 0.1$ für das *Newton*-Verfahren, so erhalten wir die in Abb. 1.4 dargestellte Konvergenz.

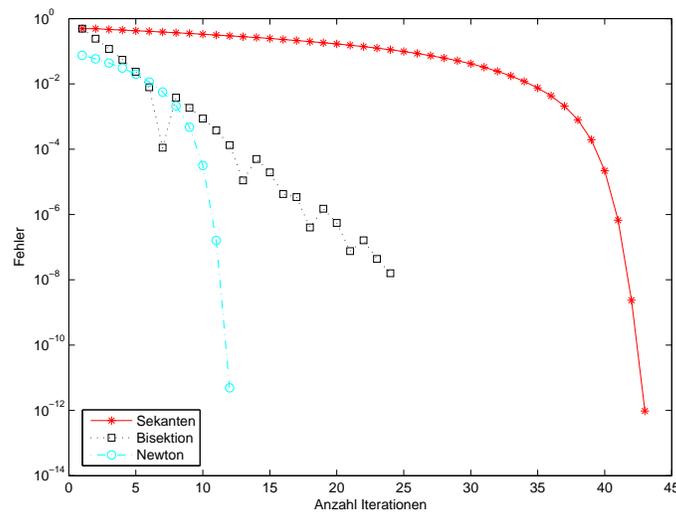


Abb. 1.6: Konvergenz von Bisektions-, Sekanten- und Newton-Verfahren angewandt auf $f(m) = (m \cdot K_0 - R) \cdot (1 + m)^n + R$ aus Bsp. 1.0.1 mit $K_0 = 10000$, $n = 48$ und $R = 250$.

Bemerkung 1.4.5 (Invarianzeigenschaft) Sei $A \in \mathbb{R}^{n \times n}$ einer reguläre Matrix. Offensichtlich ist das Problem der Lösung von $f(x) = 0$ äquivalent zu dem Problem

$$g(x) := Af(x) = 0.$$

Zugleich gilt auch

$$\begin{aligned} J_g(x)^{-1}g(x) &= (AJ_f(x))^{-1}Af(x) = J_f^{-1}(x)A^{-1}Af(x) \\ &= J_f^{-1}(x)f(x), \end{aligned}$$

d.h. sowohl das zu lösende Problem $f(x) = 0$ als auch das Newton-Verfahren sind affin-invariant.

Wir liefern nun einen Konvergenzsatz, der eine geringere Regularität von f voraussetzt.

Satz 1.4.6 Sei $D \subset \mathbb{R}^n$ offen und konvex, $f : D \rightarrow \mathbb{R}^n$ eine stetig partiell differenzierbare Funktion mit invertierbarer Jacobi-Matrix $J(x)$ für alle $x \in D$. Es gelte ferner für ein $\omega \geq 0$ die folgende Lipschitz-Bedingung:

$$\|J^{-1}(x)(J(x+sv) - J(x))v\| \leq s\omega\|v\|^2$$

für alle $s \in [0, 1]$, $x \in D$ und $v \in \mathbb{R}^n$ mit $x+v \in D$. Weiterhin existiere eine Lösung $x^* \in D$ und ein Startwert $x^{(0)} \in D$ derart, dass

$$\rho := \|x^* - x^{(0)}\| < \frac{2}{\omega} \quad \text{und} \quad \mathcal{U}_\rho(x^*) \subseteq D.$$

Dann bleibt die durch das Newton-Verfahren definierte Folge $(x^{(k)})$ für $k > 0$ in der offenen Umgebung $\mathcal{U}_\rho(x^*)$ und konvergiert gegen x^* , d.h.

$$\|x^{(k)} - x^*\| < \rho \quad \text{für } k > 0$$

und

$$\lim_{k \rightarrow \infty} x^{(k)} = x^*.$$

Die Konvergenzgeschwindigkeit läßt sich abschätzen durch

$$\|x^{(k+1)} - x^*\| \leq \frac{\omega}{2} \|x^{(k)} - x^*\|^2 \quad \text{für } k = 0, 1, 2, \dots$$

und darüber hinaus ist die Lösung x^* eindeutig in $\mathcal{U}_{2/\omega}(x^*)$.

Beweis. Als ersten Schritt des Beweises leiten wir aus der *Lipschitz*-Bedingung für $x, y \in D$ her, dass gilt:

$$\|J^{-1}(x)(f(y) - f(x) - J(x)(y - x))\| \leq \frac{\omega}{2} \|y - x\|^2. \quad (1.9)$$

Aus $\frac{\partial}{\partial s} f_j(x + s(y - x)) = \nabla f_j(x + s(y - x)) \cdot (y - x)$ folgt die *Lagrange*-Form des Mittelwertsatzes der Integralrechnung

$$\int_0^1 \left(J(x + s(y - x)) - J(x) \right) (y - x) ds = f(y) - f(x) - J(x)(y - x).$$

Da auch $J^{-1}(x)$ unabhängig von s ist, erhalten wir für die linke Seite von (1.9)

$$\begin{aligned} \left\| J^{-1}(x) \left(f(y) - f(x) - J(x)(y - x) \right) \right\| &= \left\| \int_0^1 J^{-1}(x) \left(J(x + s(y - x)) - J(x) \right) (y - x) ds \right\| \\ &\leq \int_0^1 \left\| J^{-1}(x) \left(J(x + s(y - x)) - J(x) \right) (y - x) \right\| ds \\ &\leq \int_0^1 s\omega \|y - x\|^2 ds = \frac{\omega}{2} \|y - x\|^2. \end{aligned}$$

Nun erhalten wir für die Iterationsvorschrift

$$x^{(k+1)} = x^{(k)} - J^{-1}(x^{(k)})f(x^{(k)})$$

sowie $f(x^*) = 0$

$$\begin{aligned} x^{(k+1)} - x^* &= x^{(k)} - J^{-1}(x^{(k)})f(x^{(k)}) - x^* \\ &= x^{(k)} - x^* - J^{-1}(x^{(k)})(f(x^{(k)}) - f(x^*)) \\ &= J^{-1}(x^{(k)}) \left[f(x^*) - f(x^{(k)}) - J(x^{(k)})(x^* - x^{(k)}) \right]. \end{aligned}$$

Mit (1.9) erhalten wir nun

$$\|x^{(k+1)} - x^*\| \leq \frac{\omega}{2} \|x^{(k)} - x^*\|^2.$$

Da $\|x^{(0)} - x^*\| = \rho$, folgt daraus

$$\|x^{(1)} - x^*\| \leq \underbrace{\frac{\omega}{2} \|x^{(0)} - x^*\|}_{= \frac{\omega\rho}{2} =: \alpha < 1} \|x^{(0)} - x^*\| < \|x^{(0)} - x^*\|$$

und somit induktiv für $k > 0$

$$\|x^{(k)} - x^*\| \leq \underbrace{\frac{\omega}{2} \|x^{(k-1)} - x^*\|}_{\leq \frac{\omega\rho}{2} = \alpha < 1} \|x^{(k-1)} - x^*\| \leq \alpha^k \|x^{(0)} - x^*\| < \|x^{(0)} - x^*\| \quad (k > 0).$$

Daraus folgt $\|x^{(k)} - x^*\| < \rho$ für alle $k > 0$ und die Folge $(x^{(k)})$ konvergiert gegen x^* .

Zum Beweis der Eindeutigkeit in der Umgebung $\mathcal{U}_{2/\omega}(x^*)$ um x^* mit Radius $2/\omega$ benutzen wir nochmals (1.9).

Sei $x^{**} \in \mathcal{U}_{2/\omega}(x^*)$ eine weitere Lösung, also $f(x^{**}) = 0$ und $\|x^* - x^{**}\| < 2/\omega$.

Einsetzen in (1.9) liefert

$$\begin{aligned} \|x^{**} - x^*\| &= \left\| J^{-1}(x^*) \left(0 - 0 - J^{-1}(x^*)(x^{**} - x^*) \right) \right\| \\ &\leq \underbrace{\frac{\omega}{2} \|x^{**} - x^*\|}_{< 1} \|x^{**} - x^*\|, \end{aligned}$$

dies ist aber nur möglich, falls $x^{**} = x^*$. □

Bemerkung 1.4.7 (Merkregel) Das Newton-Verfahren konvergiert lokal quadratisch.

Bemerkung 1.4.8 (Konvergenztest) Als Näherung an den Fehler $\|x - x^{(k)}\|$ verwenden wir den Term $\|J^{-1}(x^{(k)})f(x^{(k)})\|$ ($= \|J^{-1}(x^{(k)})(f(x) - f(x^{(k)}))\|$). Da man erwartet, dass der Fehler monoton fällt, d.h. $\|x - x^{(k+1)}\| \leq \|x - x^{(k)}\|$ gilt, testet man dieses für jedes k durch den natürlichen Monotonietest, d.h.

$$\|J^{-1}(x^{(k)})f(x^{(k+1)})\| \leq \bar{\theta} \|J^{-1}(x^{(k)})f(x^{(k)})\| \quad \text{sei erfüllt für ein } \bar{\theta} < 1. \quad (1.10)$$

Im Newton-Verfahren berechnen wir zu $x^{(k)}$,

$$J(x^{(k)})s^{(k)} = -f(x^{(k)}) \quad \text{und} \quad x^{(k+1)} = x^{(k)} + s^{(k)}.$$

Somit ist der Ausdruck $J^{-1}(x^{(k)})f(x^{(k)})$ auf der rechten Seite in (1.10) gleich dem Negativen der Newton-Korrektur $s^{(k)}$, die sowieso berechnet werden muss. Zusätzlich muss nur $\bar{s}^{(k)}$, definiert durch

$$J(x^{(k)})\bar{s}^{(k)} = f(x^{(k+1)}),$$

bestimmt werden. Theoretische Untersuchungen und numerische Experimente liefern $\bar{\theta} = 1/2$ als eine gute Wahl. Falls der natürliche Monotonietest, d.h.

$$\|\bar{s}^{(k)}\| \leq \frac{1}{2} \|s^{(k)}\|$$

für ein k verletzt ist, so ist das Newton-Verfahren abzubrechen und es bleibt nichts Anderes übrig, als einen (hoffentlich) besseren Startwert zu finden.

Bemerkung 1.4.9 (Dämpfung des Newtonverfahrens) Eine Möglichkeit die Konvergenz des Newton-Verfahrens zu retten, ist häufig eine Dämpfung in der Form

$$x^{(k+1)} = x^{(k)} - \lambda_k s^{(k)}, \quad \text{für } \lambda_k \in (0, 1].$$

Für eine einfache Dämpfungsstrategie können wir den Dämpfungsparameter λ_k derart wählen, so dass der natürliche Monotonietest für $\bar{\theta} = 1 - \lambda_k/2$ erfüllt ist, d.h.

$$\|F'(x^{(k)})^{-1}F(x^{(k)} + \lambda_k s^{(k)})\| \leq \left(1 - \frac{\lambda_k}{2}\right) \|F'(x^{(k)})^{-1}F(x^{(k)})\|.$$

Dabei wählen wir λ_k aus einer endlichen Folge $\{1, \frac{1}{2}, \frac{1}{4}, \dots, \lambda_{\min}\}$ und brechen ggf. das Verfahren ab, falls $\lambda_k < \lambda_{\min}$ notwendig wäre. War λ_k erfolgreich, so zeigt die Praxis, dass es effizienter ist, mit $\lambda_{k+1} = \min\{1, 2\lambda_k\}$ fortzufahren anstatt wieder mit $\lambda_{k+1} = 1$ anzufangen. War der Monotonietest mit λ_k verletzt, so testet man erneut mit $\lambda_k/2$.

MATLAB-Funktion: Newton.m

```

1 function [u,nit] = newton(u,F,DF,tol,maxit,param)
2 Fu = F(u,param);
3 DFu = DF(u,param);
4 s = -DFu\Fu;
5 lam = 1;
6 tmp = max(tol,tol*norm(s));
7 nit = 0;
8 while norm(s) > tmp && nit <= maxit

```

```

9   nit = nit + 1;
10  u_old = u;
11  lam = min(1,2*lam);
12  for k=1:30
13      u = u_old + lam * s; % Daempfung mit Parameter lam
14      Fu = F(u,param);
15      if norm(DFu\Fu) <= (1-lam/2) * norm(s)
16          break % Abbruch der for-Schleife, falls
17          end % Konvergenztest erfuehlt
18          lam = lam/2; % lam noch zu groe --> halbieren
19  end
20  DFu = DF(u,param);
21  s = -DFu\Fu;
22  end

```

Beispiel 1.4.10 (Extremalstellen der Rosenbrock-Funktion) Es gilt für

$$f(x) = \sum_{i=1}^{n-1} [(1-x_i)^2 + 100(x_{i+1} - x_i^2)^2] \quad (x \in \mathbb{R}^n)$$

aus Beispiel 1.0.3, dass die Jacobi-Matrix $A := A(x) := J_f(x)$ folgende Einträge enthält

$$\begin{aligned}
 a_{11} &= 2 + 1200x_1^2 - 400x_2 \\
 a_{jj} &= 202 + 1200x_j^2 - 400x_{j+1} \quad (j = 2, \dots, n-1) \\
 a_{nn} &= 200 \\
 a_{j,j+1} &= a_{j+1,j} = -400x_j \quad (j = 1, \dots, n-1) \\
 a_{jk} &= 0 \quad (|j-k| \geq 2).
 \end{aligned}$$

MATLAB-Funktion: rosenbrock.m

```

1  function value = rosenbrock(x,param)
2  n = size(x,1);
3  value(2:n,1) = 200 * (x(2:end)-x(1:end-1).^2);
4  value(1:n-1,1) = value(1:n-1,1) - 2 * (1-x(1:end-1)) ...
5      - 400*x(1:end-1).*(x(2:end)-x(1:end-1).^2);

```

MATLAB-Funktion: D_rosenbrock.m

```

1  function D = D_rosenbrock(x,param)
2  n = size(x,1);
3  d0(2:n,1) = 200;
4  d0(1:n-1) = d0(1:n-1) + 2 + 1200*x(1:n-1).^2-400*x(2:n);
5  dm1 = -400*x;
6  dp1 = -400*x([1,1:n-1]);
7  D = spdiags([dm1,d0,dp1],[-1 0 1],n,n);

```

MATLAB-Beispiel:

Man kann z.B. neben dem globalen Minimum $(x_1, \dots, x_n) = (1, \dots, 1)$ mit dem Newton-Verfahren ein weiteres lokales Minimum in der Umgebung von $(x_1, x_2, \dots, x_n) = (-1, 1, \dots, 1)$ von f in Bsp. 1.0.3 finden.

```
>> n = 6;
>> x0 = [-1;ones(n-1,1)];
>> [x,nit] = NewtonSimple(x0,@rosenbrock,
    @D_rosenbrock,1e-10,100,[],)
x =
-0.98657497957099
 0.98339822883618
 0.97210667005309
 0.94743743682644
 0.89865118485173
 0.80757395203542
nit =
 5
```

Beispiel 1.4.11 (p-Laplace) Im Folgenden betrachten wir das **p-Laplace Problem**.

Gegeben sei ein Gebiet $\Omega \subset \mathbb{R}^d$ ($d \in \mathbb{N}$) und $1 < p < \infty$. Finde $u \in W_0^{1,p}(\Omega) := \{u \in L^p \mid \nabla u \in (L^p(\Omega))^d, u|_{\partial\Omega} = 0\}$ mit

$$\begin{aligned} \operatorname{div}(|\nabla u|^{p-2} \nabla u) &= f && \text{im Gebiet } \Omega, \\ u &= 0 && \text{auf dem Rand } \Gamma := \partial\Omega, \end{aligned} \quad (1.11)$$

wobei div den Divergenzoperator⁴ bezeichne.

Die Lösung des p -Laplace-Problems (1.11) ist der Minimierer des Energiefunktional

$$J(u) := \int_{\Omega} \frac{1}{p} |\nabla u|^p - f u \, dx$$

über alle Funktionen aus dem Sobolev⁵-Raum $W_0^{1,p}(\Omega)$. Im Mehrdimensionalen ist dieses Problem im Allgemeinen nicht analytisch zu lösen. Für den Spezialfall $p = 2$, welches auf ein lineares Problem führt, spricht man einfach vom Laplace-Problem. Das p -Laplace-Problem ist ein typisches Beispiel für eine große Klasse von nichtlinearen Problemen.

Ohne jetzt auf die funktionalanalytischen Grundlagen einzugehen, beschränken wir uns kurzer Hand auf die Approximation von u durch eine stückweise lineare Funktion u_N und den eindimensionalen Fall ($d = 1$), um die Darstellung übersichtlich zu halten.

Gegeben sei $N \in \mathbb{N}$. Es sei $x_i = -1 + \frac{2i}{N+1}$ ($i = 1, \dots, N$), $h_i := x_i - x_{i-1}$ und

$$\varphi_i(x) := \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} & , \text{ falls } x \in [x_{i-1}, x_i], \\ \frac{x_{i+1}-x}{x_{i+1}-x_i} & , \text{ falls } x \in (x_i, x_{i+1}], \\ 0 & , \text{ sonst} \end{cases} \quad (1.12)$$

definiert sogenannte stückweise lineare **Hutfunktionen**. Man beachte, dass $\varphi_i(x_j) = \delta_{ij}$ gilt, wobei δ_{ij} das Kronecker-Symbol sei.

Wir suchen nun für $1 < p < \infty$ eine stückweise lineare Funktion $u_N(x) := \sum_{i=1}^N \alpha_i \varphi_i(x)$ (diese erfüllt per Definition die Randbedingungen $u_N(-1) = u_N(1) = 0$), welche zu gegebenem $f : [-1, 1] \rightarrow \mathbb{R}$ das Funktional

$$\widehat{J}(u_N) = J(\alpha) := \int_{-1}^1 \frac{1}{p} |u'_N(x)|^p - f(x) u_N(x) \, dx$$

⁴Für $u = (u_1, \dots, u_n) \in C^1(\Omega; \mathbb{R}^n)$ sei $\operatorname{div}(u) := \sum_{i=1}^n \frac{\partial u_i}{\partial x_i}$

⁵Sergei Lvovich Sobolev, 1908 - 1989

minimiert, mit $\alpha = (\alpha_1, \dots, \alpha_n)$. Man beachte $u'_N(x)|_{(x_{k-1}, x_k)} = h_k^{-1}(\alpha_{k-1} - \alpha_k)$ und

$$\begin{aligned} J(\alpha) &:= \sum_{i=1}^N \sum_{k=0}^N \int_{x_k}^{x_{k+1}} \frac{1}{p} |\alpha_i \varphi_i(x)'|^p - \alpha_i f(x) \varphi_i(x) dx \\ &= \sum_{k=0}^N \int_{x_k}^{x_{k+1}} \frac{1}{p} |\alpha_k h_{k+1}^{-1} - \alpha_{k+1} h_{k+1}^{-1}|^p - f(x) (\alpha_k h_{k+1}^{-1} - \alpha_{k+1} h_{k+1}^{-1}) dx. \end{aligned}$$

Ohne es zu beweisen, gehen wir davon aus, dass die gesuchte Lösung

$$\frac{\partial}{\partial \alpha_j} J(\alpha) = 0 \quad (j = 1, \dots, n)$$

erfüllt. Dies führt zu dem nichtlinearen Gleichungssystem

$$F(\alpha) = (F_1(\alpha), \dots, F_N(\alpha))^T = 0$$

mit

$$\begin{aligned} F_i(\alpha) &:= \frac{\partial}{\partial \alpha_i} J(\alpha) = \frac{\partial}{\partial \alpha_i} \sum_{k=i-1}^i \int_{x_k}^{x_{k+1}} |\alpha_k h_{k+1}^{-1} - \alpha_{k+1} h_{k+1}^{-1}|^p - f(x) (\alpha_k h_{k+1}^{-1} - \alpha_{k+1} h_{k+1}^{-1}) dx \\ &= -|\alpha_{i-1} h_i^{-1} - \alpha_i h_i^{-1}|^{p-2} (\alpha_{i-1} h_i^{-1} - \alpha_i h_i^{-1}) + h_i^{-1} \int_{x_{i-1}}^{x_i} f(x) dx \\ &\quad + |\alpha_i h_{i+1}^{-1} - \alpha_{i+1} h_{i+1}^{-1}|^{p-2} (\alpha_i h_{i+1}^{-1} - \alpha_{i+1} h_{i+1}^{-1}) - h_{i+1}^{-1} \int_{x_i}^{x_{i+1}} f(x) dx. \end{aligned}$$

Für $f \equiv 1$ erhalten wir

$$u(x) = \frac{p-1}{p} \left(1 - x^{\frac{p}{p-1}} \right) \quad 1 < p < \infty \quad (1.13)$$

bzw. für $p = 2$ gilt $u_N(x_i) = u(x_i)$ ($i = 0, \dots, N+1$) und für $p = 3/2$ lässt sich mit $N = 2M$

$$u(x) - u_N(x) = \frac{\frac{p-1}{p} - 2^{\frac{1}{1-p}}}{M^2} \left(1 - |x|^{(2-p)/(p-1)} \right) \quad (1.14)$$

zeigen.

Aufgabe 1.4.12 Man zeige (1.13) und (1.14).

Gilt $f \equiv 1$, so lässt sich $F(\alpha)$ exakt bestimmen, was in der Matlab-Routine `f2.m` realisiert ist. Die Berechnung der Funktionalmatrix $\frac{\partial^2}{\partial \alpha_i \partial \alpha_j} J(\alpha)$ ist in der Matlab-Routine `Df2.m` umgesetzt.

MATLAB-Funktionen: `f2.m` und `Df2.m`

```

1 function Fu = f2(u,param)
2 % Aufstellen des Vektors Fu
3 h = 1/param.M;
4 u = [0;u;0];
5 Fu=zeros(2*param.M+1,1);
6 for j = 1:2*param.M
7     stima = ([1 -1;-1 1]*u([j,j+1]))/h^2;
8     fac = (u([j,j+1])'*stima)^( (param.p-2)/2 );

```

```

9   Fu([j,j+1])=Fu([j,j+1])+h*fac*stima;
10  end
11  for j=1:2*param.M
12     Fu([j;j+1])=Fu([j,j+1])-h/2;
13  end
14  Fu([1,end])=[];

1  function DFu = Df2(u,param)
2  % Aufstellen der Jacobimatrix
3  h = 1/param.M; u = [0;u;0];
4  DFu=sparse(2*param.M+1,2*param.M+1);
5  for j = 1:2*param.M
6     stima = [1 -1;-1 1]/h^2;
7     fac = (param.p-1)*(u([j,j+1])'*stima*u([j,j+1]))^((param.p-2)/2);
8     DFu([j,j+1],[j,j+1])=DFu([j,j+1],[j,j+1])+h*fac*stima;
9  end
10 DFu = DFu(2:2*param.M,2:2*param.M);

```

MATLAB-Beispiel:

Man kann z.B. das p -Laplace-Problem näherungsweise mit dem folgenden Befehlen bestimmen.

```

>> M = 8;
>> p = 1.2;
>> u0 = (1-linspace(-1,1,2*M+1).^2)';
>> u0([1,end]) = [];
>> u = Newton(u0,@f2,@Df2,1e-12,...
             100,struct('M',M,'p',p));
>> xi = linspace(-1,1,2*M+1);
>> plot(xi,[0;u;0],'-*');

```

Für $p \rightarrow 1$ oder $p \rightarrow \infty$ nimmt die Nichtlinearität immer stärker zu, so liefert z.B. die Routine `NewtonSimple` für $p = 1.2$ und $M = 8$ schon keine Konvergenz mehr, jedoch die Routine `Newton` mit Schrittweitensteuerung und modifiziertem Abbruchkriterium.

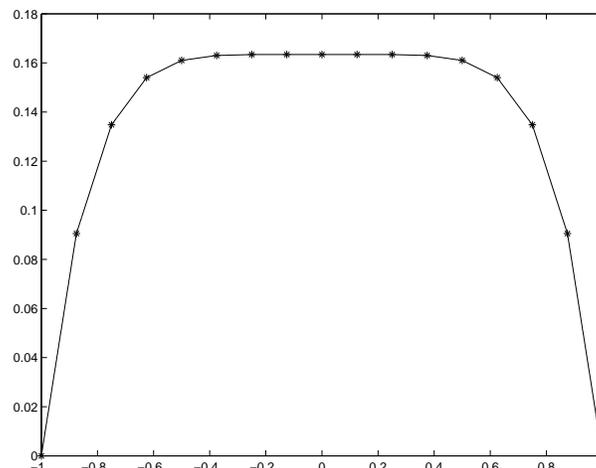


Abb. 1.7: Näherungsweise Lösung des p -Laplace-Problems mit $p = 1.2$ und $N = 15$.

1.5 DAS *Broyden*-VERFAHREN

Das *Newton*-Verfahren, wie wir es bisher diskutiert haben, hat aufgrund seiner quadratischen Konvergenz eine große Bedeutung, besitzt aber auch einige Nachteile. Einer davon ist, dass in jeder Iteration die *Jacobi*-Matrix benötigt wird. In vielen Beispielen sind die analytischen Ableitungen jedoch nicht bekannt.

Das auf *Broyden*⁶ zurückgehende Quasi-*Newton*-Verfahren ist ein Kompromiss zwischen Neuberechnung der *Jacobi*-Matrix in jedem Iterationsschritt (analog zum *Newton*-Verfahren) und der Verwendung einer festen Matrix im Lauf der gesamten Iteration. Ausgehend von einer Näherung an die *Jacobi*-Matrix werden die Matrizen in jedem Schritt so aktualisiert, dass sie möglichst ähnliche Abbildungseigenschaften aufweisen wie die exakte Funktionalmatrix. Gleichzeitig achtet man darauf, dass diese Aktualisierung möglichst wenig zusätzlichen Rechenaufwand erfordert. Sie erfolgt daher mittels Rang-1-Korrekturmatrizen, die sich aus Termen berechnen lassen, die ohnehin während der Iteration bestimmt werden müssen.

Die Näherungen an die *Jacobi*-Matrizen seien mit B_k bezeichnet. Dann wird im k -ten Schritt des *Broyden*-Verfahrens

$$B_k s^{(k)} = -f(x^{(k)}) \quad (1.15)$$

berechnet, wobei $x^{(k+1)} = x^{(k)} + s^{(k)}$ sei. Im Eindimensionalen liefert das Sekantenverfahren zu zwei gegebenen Werten $x^{(-1)}$ und $x^{(0)}$ iterativ die Steigung der Sekante durch

$$\beta_k = \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}} \quad (k \geq 0) \quad (1.16)$$

und daraus eine weitere Näherung $x^{(k+1)}$ an x mittels

$$x^{(k+1)} = x^{(k)} - \beta_k^{-1} f(x^{(k)}).$$

Die formale Verallgemeinerung der Sekanten-Bedingung (1.16) an B_k lautet

$$B_k s^{(k-1)} = f(x^{(k)}) - f(x^{(k-1)}) =: \delta f_k.$$

Die Bedingung genügt jedoch nicht, um $B_k \in \mathbb{R}^{n \times n}$ eindeutig zu bestimmen. Daher versucht man für $k \geq n$ die letzten gewonnenen Informationen zu nutzen, so dass B_k ($k \geq n$) eine Lösung der folgenden Menge von n Systemen

$$B_k (x^{(k)} - x^{(k-j)}) = f(x^{(k)}) - f(x^{(k-j)}) \quad (j = 1, \dots, n) \quad (1.17)$$

ist. Da im Allgemeinen die Vektoren $x^{(k-j)}, \dots, x^{(k)}$ nicht linear unabhängig sind, fordern wir zusätzlich, dass die Differenz zwischen den linearen Approximationen von $f(x^{(k-1)})$ und $f(x^{(k)})$, nämlich

$$d_k := f(x^{(k)}) + B_k(x - x^{(k)}) - \left(f(x^{(k-1)}) + B_{k-1}(x - x^{(k-1)}) \right), \quad (1.18)$$

in der euklidischen Norm minimiert wird. Setzen wir $j = 1$ in (1.17) so wird (1.18) zu

$$d_k = (B_k - B_{k-1})(x - x^{(k-1)}). \quad (1.19)$$

Zerlegt man den Vektor $x - x^{(k-1)}$ in der Form

$$x - x^{(k-1)} = \alpha s^{(k-1)} + r$$

⁶Broyden, Charles George (1933 -)

mit $\alpha \in \mathbb{R}$ und $r^T s^{(k-1)} = 0$, dann erhält man aus (1.19)

$$d_k = \alpha(B_k - B_{k-1})s^{(k-1)} + (B_k - B_{k-1})r. \quad (1.20)$$

Da $(B_k - B_{k-1})s^{(k-1)} = \delta f_k - B_{k-1}s^{(k-1)}$ gilt, ist somit der erste Term in (1.20) unabhängig von B_k und es bleibt nur der zweite Term in (1.20) zu minimieren. Die Matrix B_k , die $(B_k - B_{k-1})r$ minimiert für alle r , die orthogonal zu $s^{(k-1)}$ sind, unter der Restriktion, dass (1.17) gilt, kann rekursiv mittels des Rang-1-Updates von B_{k-1}

$$B_k = B_{k-1} + \frac{(\delta f_k - B_{k-1}s^{(k-1)}) (s^{(k-1)})^T}{(s^{(k-1)})^T s^{(k-1)}} \quad (1.21)$$

berechnet werden. Die Methode (1.15) mit der Wahl (1.21) wird als Broyden-Verfahren bezeichnet. Zur Initialisierung setzt man $B_0 = J_f(x_0)$ oder eine geeignete Approximation z.B. mittels Differenzenquotienten, d.h. $(B_0)_{ij} \approx (f_i(x_0 + he_j) - f_i(x_0))/h$.

Bemerkung 1.5.1 Ist eine QR-Zerlegung von $B_0 \in \mathbb{R}^{n \times n}$ gegeben, so läßt sich die QR-Zerlegung von B_k ($k > 0$) aus der QR-Zerlegung von B_{k-1} (wie wir in Numerik 1 gezeigt haben) mit $\mathcal{O}(n^2)$ bestimmen.

MATLAB-Funktion: Broyden.m

```

1 function [x,nit] = Broyden(x,f,B,tol,maxit,param)
2 fx = f(x,param);
3 fx1 = zeros(size(fx));
4 nit = 0; err = inf;
5 while nit < maxit && err > tol
6     s = - B\x;
7     x = x + s;
8     err = norm(s);
9     if err > tol
10        fx1 = f(x,param);
11        B = B + 1/(s'*s) * fx1 * s';
12    end
13    fx = fx1;
14    nit = nit + 1;
15 end

```

Unter Verwendung des Broyden-Verfahrens lösen wir das nichtlineare Problem aus Bsp. 1.0.3 für $n = 6$. Diese Methode konvergiert in 18 Iterationen verglichen mit den 5 Iterationen, die das Newton-Verfahren erforderte bei gleichem Startwert $x_0 = (-1, 1, \dots, 1)^T$. Die Matrix B_0 wurde gleich der Jacobi-Matrix im Punkt x_0 gesetzt. Abbildung 1.5 zeigt das Verhalten der Euklidischen Norm des Fehlers beider Methoden.

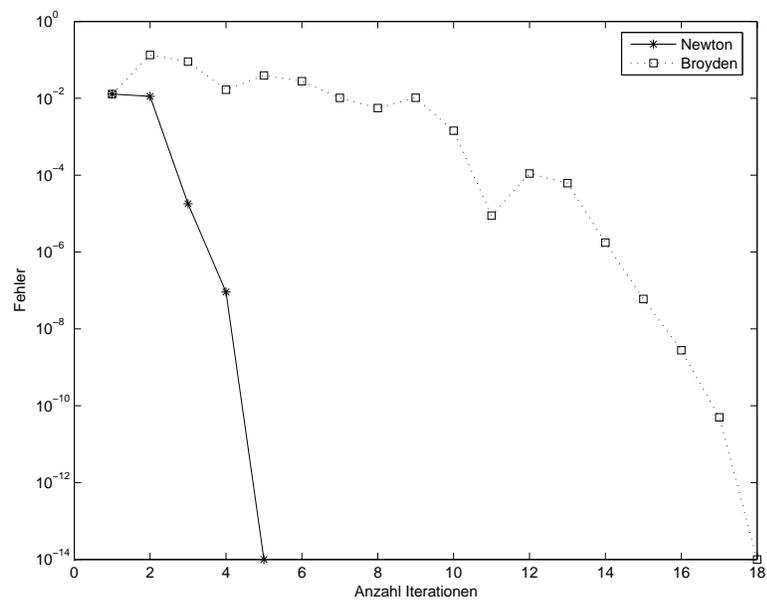


Abb. 1.8: Euklidische Norm des Fehlers für das Broyden- und Newton-Verfahren im Fall des nichtlinearen Problems aus Bsp. 1.0.3 für $n = 6$.

2 ORTHOGONALPOLYNOME UND GAUSS-QUADRATUR

Motivation Bei vielen Problemen der Mathematik lässt sich eine Lösung bzw. eine Lösungsfunktion bezüglich sog. speziellen Funktionen entwickeln, die sich dann durch ganz spezielle, dem Problem angemessene Eigenschaften auszeichnen. Beispiele hierfür sind etwa

- (i) *Tschebyscheff*¹-Polynome (siehe Kapitel über Interpolation),
- (ii) *Legendre*²-Polynome (siehe Kapitel über Quadratur).

Diese Funktionen zeichnen sich insbesondere durch ihre Orthogonalität aus.

2.1 ALLGEMEINE CHARAKTERISTIKA

Definition 2.1.1 (Skalarprodukt) Es sei V ein reeller Vektorraum. Ein **Skalarprodukt** (oder inneres Produkt) auf V ist eine symmetrische positiv definite Bilinearform $(\cdot, \cdot): V \times V \rightarrow \mathbb{R}$, d.h. für $x, y, z \in V$ und $\alpha, \beta \in \mathbb{R}$ gelten die folgenden Bedingungen:

i.) positive Definitheit

$$(x, x) \geq 0, \quad \text{und } (x, x) = 0 \text{ genau dann, wenn } x = 0,$$

ii.) Symmetrie

$$(x, y) = (y, x),$$

iii.) Bilinearität

$$(\alpha x + \beta y, z) = \alpha(x, z) + \beta(y, z).$$

Satz 2.1.2 (Cauchy-Schwarsche-Ungleichung) Es sei V ein reeller Vektorraum, $(\cdot, \cdot): V \times V \rightarrow \mathbb{R}$ ein Skalarprodukt auf V . Dann gilt

$$(x, y) \leq \sqrt{(x, x)} \sqrt{(y, y)}.$$

Beweis. Der Fall $y = 0$ ist trivial. Es bleibt also der Fall $y \neq 0$ und damit $(y, y) \neq 0$. Für jedes $\alpha \in \mathbb{R}$ gilt

$$0 \leq (x - \alpha y, x - \alpha y) = (x, x) - 2\alpha(x, y) + \alpha^2(y, y).$$

Wählt man nun speziell $\alpha := (x, y)/(y, y)$, so ergibt sich

$$0 \leq (x, x) - \frac{(x, y)^2}{(y, y)},$$

also

$$(x, y)^2 \leq (x, x)(y, y).$$

Nun liefert Ziehen der Quadratwurzel die Behauptung. □

¹Tschebyscheff, Pafnuti Lwowitsch (1821-1894), auch als Čebyšv, Chebyshev, Tschebyschow, Tschebyschew oder Tschebyschew transkribiert

²Legendre, Adrien-Marie (1752-1833)

Satz 2.1.3 Es sei $\omega \in C(a, b)$, $\omega(x) > 0$ für $x \in (a, b)$ eine positive Gewichtsfunktion. Dann ist

$$(f, g) := (f, g)_\omega := \int_a^b \omega(x) f(x) g(x) dx$$

für $f, g \in C[a, b]$ ein Skalarprodukt.

Beweis. Die Aussage ergibt sich unmittelbar aus der Verifizierung der bekannten Skalarprodukt-Eigenschaften aus der Linearen Algebra (Additivität, Homogenität, Symmetrie, positive Definitheit) und wird deshalb an dieser Stelle ausgespart. \square

Definition 2.1.4 (Orthogonalität) Wir bezeichnen zwei Funktionen $f, g \in C[a, b]$ bzgl. des Skalarprodukts $(\cdot, \cdot)_\omega$ als **orthogonal**, falls gilt:

$$(f, g)_\omega = 0.$$

Bemerkung 2.1.5 Im Folgenden setzen wir voraus, dass die durch das Skalarprodukt induzierte Norm

$$\|q\| := \|q\|_\omega := \sqrt{(q, q)_\omega}$$

für alle Polynome $q \in \mathbb{P}_k$ ($k \in \mathbb{N}$) wohldefiniert und endlich ist. Somit existieren auch die Momente

$$m_k := \int_a^b \omega(x) x^k dx,$$

da mit der **Cauchy-Schwarz-Ungleichung** folgt:

$$|m_k| = |(1, x^k)_\omega| \leq \|1\|_\omega \|x^k\|_\omega < \infty.$$

Beispiele 2.1.6 Einige Beispiele von Gewichtsfunktionen seien hier genannt, die ungewöhnlich genug sind, um numerische Techniken zu erfordern, aber in praktischen Anwendungen verwendet werden.

- $\omega(x) = x^\alpha \log(1/x)$ auf $[0, 1]$ mit $\alpha > 0$.
Die Momente $m_k = (r + \alpha + 1)^{-1}$ sind alle endlich und die zugehörigen Orthogonalpolynome werden verwendet, um Quadraturformeln zu konstruieren für Integrale über $[0, 1]$, deren Integranden zwei Singularitäten bei Null haben, eine logarithmische und eine algebraische (falls $\alpha \neq 0, 1, 2, \dots$).
- $\omega(x) = e^{-x}$ und $\omega(x) = e^{-x^2}$ auf $[0, c]$ ($0 < c < \infty$).
Dies sind Laguerre bzw. Hermite-Gewichte auf einem endlichen Intervall. Die Momente m_k lassen sich durch die unvollständige Gamma-Funktion $\gamma(\alpha, x) = \int_0^x t^{\alpha-1} e^{-t} dt$ ausdrücken, nämlich $m_k = \gamma(k+1, c)$ bzw. $m_k = \frac{1}{2} \gamma(\frac{1}{2}(k+1), c^2)$. Beide Varianten finden Anwendung bei der Gauss-Quadratur von Integralen in der molekularen Quantenmechanik.

Definition 2.1.7 (Orthogonalpolynome) Es sei (p_k) eine Folge paarweise orthogonaler Polynome $p_k \in \mathbb{P}_k$, d.h.

$$(p_i, p_j)_\omega = \delta_{ij} (p_i, p_i)_\omega \geq 0 \tag{2.1}$$

und exakt vom Grad k , dann heißen die p_k **Orthogonalpolynome** über $[a, b]$ bzgl. der Gewichtsfunktion ω .

Aufgabe 2.1.8 Man zeige, dass ein Polynom $p_n(x) = x^n + \dots \in \mathbb{P}_n$ genau dann ein orthogonales Polynom n -ten Grades ist, wenn

$$\int p_n^2(x) \omega(x) dx = \min \left\{ \int q_n^2(x) \omega(x) dx : q_n(x) = x^n + \dots \right\}$$

gilt.

Bemerkung 2.1.9 1. Da die p_0, \dots, p_k eine Basis des \mathbb{P}_k darstellen, folgt sofort

$$(p_k, q)_\omega = (p_k, \sum_{i=0}^{k-1} \alpha_i p_i)_\omega = 0 \quad \text{für alle } q \in \mathbb{P}_{k-1}. \quad (2.2)$$

2. Die Definition der Orthogonalpolynome über (2.1) ist keineswegs eindeutig. Eine mögliche Standardisierung ist z.B.

$$p_k(x) = x^k + a_{k-1}x^{k-1} + \dots,$$

d.h. der führende Koeffizient ist Eins.

Definition 2.1.10 (Monisches Polynom) Ist der führende Koeffizient eines Polynoms Eins, so bezeichnet man dieses als **monisch** (oder auch normiert).

Satz 2.1.11 Zu jedem positiv gewichteten Skalarprodukt $(\cdot, \cdot)_\omega$ gibt es eindeutig bestimmte monische Orthogonalpolynome $p_k \in \mathbb{P}_k$ (d.h. mit führendem Koeffizienten Eins). Gilt zusätzlich $(x p_k, p_j)_\omega = (p_k, x p_j)_\omega$ für $j, k \geq 0$, dann erfüllen die Orthogonalpolynome die folgende Drei-Term-Rekursion:

$$p_k(x) = (\alpha_k + x)p_{k-1}(x) + \gamma_k p_{k-2}(x), \quad k = 1, 2, \dots \quad (2.3)$$

mit $p_{-1} := 0, p_0 := 1$ und

$$\alpha_k = -\frac{(x p_{k-1}, p_{k-1})_\omega}{(p_{k-1}, p_{k-1})_\omega}, \quad \gamma_k = -\frac{(p_{k-1}, p_{k-1})_\omega}{(p_{k-2}, p_{k-2})_\omega}. \quad (2.4)$$

Bemerkung 2.1.12 Die Abbildung $(f, g) := -\int_{-1}^1 \int_{-1}^1 f(x)g(y) \log|x-y| dx dy$ ist zwar ein Skalarprodukt auf der Menge der stetigen Funktionen, aber die zugehörigen Orthogonalpolynome erfüllen keine 3-Term-Rekursion. Symmetrieüberlegungen zeigen sofort, dass für $0 < j + k$ ungerade

$$(x^j, x^k) = 0$$

gilt. Mit Hilfe von Maple verifiziert man leicht durch mehrfaches Anwenden der Anweisungen

```
j:=0;
k:=1;
-int(int(x^j*y^k*log((x-y)^2)/2, x=-1..1), y=-1..1);
```

zu verschiedenen $j, k \geq 0$, dass gilt

$$(x^j, x^k) = \begin{cases} 0 & , \text{ falls } 0 \leq k+j \text{ ungerade,} \\ \neq 0 & , \text{ falls } 0 \leq k+j \text{ gerade.} \end{cases}$$

Wir definieren $p_0 = 1, p_1 = x$ und normieren weitere p_k so, dass der führende Koeffizient 1 ist. Damit das Orthogonalpolynom $p_2 = x^2 + ax + b \in \mathbb{P}_2$ orthogonal zu x ist, muss $a = 0$ gelten und aus $(p_2, p_0) = 0$ folgt $b = -(x^2, 1)/(1, 1)$. Man beachte, dass hier $16/9 - 4/3 \log(2) = (x^2, 1) \neq (x, x) = 1$ gilt und somit die Formel der 3-Term-Rekursion (2.3) nicht gilt.

Beweis von Satz 2.1.11. Man sieht unmittelbar, dass $p_0 \equiv 1 \in \mathbb{P}_0$ gilt, den Rest der Behauptung zeigen wir induktiv. Seien also p_0, \dots, p_{k-1} paarweise orthogonale Polynome mit $p_j \in \mathbb{P}_j$ vom Grad j und führendem Koeffizienten gleich Eins. Ist $p_k \in \mathbb{P}_k$ ebenso normiert, dann folgt

$$p_k(x) - xp_{k-1}(x) \quad \text{ist vom Grade} \leq k-1.$$

Da jedoch p_0, \dots, p_{k-1} eine Orthogonalbasis von \mathbb{P}_{k-1} bzgl. $(\cdot, \cdot)_\omega$ bilden, gilt

$$p_k - xp_{k-1} = \sum_{j=0}^{k-1} \gamma_j p_j \quad \text{mit} \quad \gamma_j := \frac{(p_k - xp_{k-1}, p_j)_\omega}{(p_j, p_j)_\omega}.$$

(Man setze die linke Gleichung in $(\cdot, p_\ell)_\omega$ für $\ell = 0, \dots, k-1$ ein.) Außerdem folgt aus der Orthogonalität von p_k zu p_0, \dots, p_{k-1}

$$\gamma_j = -\frac{(xp_{k-1}, p_j)_\omega}{(p_j, p_j)_\omega} = -\frac{(p_{k-1}, xp_j)_\omega}{(p_j, p_j)_\omega}.$$

Da das Produkt xp_j als Linearkombination von $p_0, \dots, p_{j+1} \in \mathbb{P}_{j+1}$ geschrieben werden kann, gilt somit $\gamma_j = 0$ für $j+1 < k-1$, d.h.

$$\gamma_0 = \dots = \gamma_{k-3} = 0 \quad \text{bzw. es folgt} \quad p_k - xp_{k-1} = \gamma_{k-2}p_{k-2} + \gamma_{k-1}p_{k-1}.$$

Beachtet man $xp_{k-2} = p_{k-1} + \alpha_{k-2}p_{k-2} + \dots + \alpha_0p_0$, so folgt

$$\gamma_{k-2} = -\frac{(p_{k-1}, xp_{k-2})_\omega}{(p_{k-2}, p_{k-2})_\omega} = -\frac{(p_{k-1}, p_{k-1})_\omega}{(p_{k-2}, p_{k-2})_\omega}$$

und damit

$$\gamma_{k-1} = -\frac{(xp_{k-1}, p_{k-1})_\omega}{(p_{k-1}, p_{k-1})_\omega}.$$

□

Satz 2.1.13 Es seien \tilde{p}_k ($k = 0, 1, 2, \dots$) die Orthonormalpolynome bzgl. des Skalarprodukts $(\cdot, \cdot)_\omega$. Dann gilt

$$\sqrt{\beta_{k+1}}\tilde{p}_{k+1}(x) = (\alpha_k + x)\tilde{p}_k(x) - \sqrt{\beta_k}\tilde{p}_{k-1}(x) \quad k = 0, 1, 2, \dots \quad (2.5)$$

$$p_{-1}(x) = 0, \quad p_0(x) = 1/\sqrt{\beta_0}, \quad (2.6)$$

wobei die α 's und β 's durch (2.4) gegeben sind.

Definition 2.1.14 Wir definieren die unendlich-dimensionale, symmetrische und tridiagonale **Jacobi-Matrix**

$$J_\infty = J_\infty(\omega) := \begin{pmatrix} \alpha_0 & \sqrt{\beta_1} & & & 0 \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \alpha_2 & \sqrt{\beta_3} & \\ & & \ddots & \ddots & \ddots \\ 0 & & & & \end{pmatrix}.$$

Die führende $n \times n$ Untermatrix wird bezeichnet mit

$$J_n = J_n(\omega) = (J_\infty(\omega))_{i,j=1}^n$$

Falls man die ersten n Gleichungen in (2.5) in der Form

$$x \tilde{p}_k(x) = \sqrt{\beta_k} \tilde{p}_{k-1}(x) + \alpha_k \tilde{p}_{k+1}(x) + \sqrt{\beta_{k+1}} \tilde{p}_{k+1}(x) \quad k = 0, 1, 2, \dots, n-1 \quad (2.7)$$

schreibt und

$$\tilde{\mathbf{p}}(x) = (\tilde{p}_1(x), \tilde{p}_2(x), \dots, \tilde{p}_{n-1}(x))^T$$

definiert, dann lässt sich (2.7) wie folgt schreiben:

$$x \tilde{\mathbf{p}}(x) = J_n(\omega) \tilde{\mathbf{p}}(x) + \sqrt{\beta_n} \tilde{p}_n(x) \mathbf{e}_n \quad (2.8)$$

mit $\mathbf{e}_n = (0, \dots, 0, 1)^T$.

Satz 2.1.15 Die Nullstellen $x_i^{(n)}$ ($i = 1, \dots, n$) von p_n (oder \tilde{p}_n) sind die Nullstellen der $n \times n$ -Jacobi-Matrix $J_n(\omega)$ und $\tilde{p}_n(x_i^{(n)})$ sind die zugehörigen Eigenvektoren.

Bemerkung 2.1.16 Allgemeiner ist die Darstellung der Orthogonalpolynome in der Form

$$p_n(x) = k_n x^n + k'_n x^{n-1} + \dots \quad (n = 0, 1, 2, \dots). \quad (2.9)$$

Zusätzlich definieren wir

$$h_n(p_n) := h_n := \int_a^b \omega(x) p_n^2(x) dx = \|p_n\|_\omega^2. \quad (2.10)$$

Übertragen wir nun die Drei-Term-Rekursion aus Satz 2.1.11 für Orthogonalpolynome mit führendem Koeffizienten Eins auf die allgemeinere Form (2.9), so erhalten wir folgendes Resultat:

Satz 2.1.17 Zu jedem Skalarprodukt $(\cdot, \cdot)_\omega$ gibt es eindeutig bestimmte Orthogonalpolynome $p_n \in \mathbb{P}_n$ mit führendem Koeffizienten k_n . Diese Polynome erfüllen die folgende Drei-Term-Rekursion:

$$p_n(x) = (a_n + b_n x) p_{n-1} + c_n p_{n-2} \quad (n = 1, 2, \dots), \quad (2.11)$$

mit $p_{-1} = 0$, $p_0 = k_0$. Die Koeffizienten ergeben sich wie folgt:

$$b_n = \frac{k_n}{k_{n-1}}, \quad a_n = b_n \left(\frac{k'_n}{k_n} - \frac{k'_{n-1}}{k_{n-1}} \right), \quad c_n = -\frac{k_n k_{n-2} h_{n-1}}{k_{n-1}^2 h_{n-2}} \quad (n = 2, 3, \dots). \quad (2.12)$$

Beweis. Die Behauptung folgt aus Satz 2.1.11 mit der Darstellung aus Bemerkung 2.1.15. Sei $p_n(x) = k_n x^n + k'_n x^{n-1} + \dots$ ($n = 0, 1, 2, \dots$), dann hat $\bar{p}_n(x) := p_n(x)/k_n$ führenden Koeffizienten Eins und Satz 2.1.11 liefert

$$p_n(x) = k_n \bar{p}_n(x) = k_n (\alpha_n + x) \bar{p}_{n-1}(x) + k_n \gamma_n \bar{p}_{n-2}(x), \quad n = 1, 2, \dots \quad (2.13)$$

mit $\bar{p}_{-1} := 0$, $\bar{p}_0 := p_0/k_0 = 1$ und

$$\alpha_n = -\frac{(x \bar{p}_{n-1}, \bar{p}_{n-1})_\omega}{(\bar{p}_{n-1}, \bar{p}_{n-1})_\omega}$$

sowie

$$\gamma_n = -\frac{(\bar{p}_{n-1}, \bar{p}_{n-1})_\omega}{(\bar{p}_{n-2}, \bar{p}_{n-2})_\omega} = -\frac{1/k_{n-1}^2 (p_{n-1}, p_{n-1})_\omega}{1/k_{n-2}^2 (p_{n-2}, p_{n-2})_\omega} = -\frac{k_{n-2}^2 h_{n-1}}{k_{n-1}^2 h_{n-2}}. \quad (2.14)$$

Man beachte

$$x \bar{p}_{n-1} = x \left(x^{n-1} + \frac{k'_{n-1}}{k_{n-1}} x^{n-2} + \dots \right) = \bar{p}_n + \left(\frac{k'_{n-1}}{k_{n-1}} - \frac{k'_n}{k_n} \right) \bar{p}_{n-1} + q(x) \quad \text{mit } q \in \mathbb{P}_{n-2},$$

d.h.

$$\alpha_n = -\frac{(x\bar{p}_{n-1}, \bar{p}_{n-1})_\omega}{(\bar{p}_{n-1}, \bar{p}_{n-1})_\omega} = \frac{k'_n}{k_n} - \frac{k'_{n-1}}{k_{n-1}}. \quad (2.15)$$

Somit ergibt sich mit (2.14)

$$k_n \gamma_n \bar{p}_{n-2}(x) = -k_n \frac{k_{n-2}^2 h_{n-1}}{k_{n-1}^2 h_{n-2}} \frac{1}{k_{n-2}} p_{n-2}(x) = -\frac{k_n k_{n-2} h_{n-1}}{k_{n-1}^2 h_{n-2}} p_{n-2}(x)$$

bzw.

$$k_n x \bar{p}_{n-1}(x) = \frac{k_n}{k_{n-1}} x p_{n-1}(x)$$

und mit (2.15)

$$k_n \alpha_n \bar{p}_{n-1}(x) = \frac{k_n}{k_{n-1}} \left(\frac{k'_n}{k_n} - \frac{k'_{n-1}}{k_{n-1}} \right) p_{n-1}(x).$$

Die letzten drei Gleichungen mit (2.1) liefern dann die Behauptung. \square

Bemerkung 2.1.18 Für Orthogonalpolynome gilt zur Angabe der p_n die sog. Rodrigues³-Formel; d.h. alle p_n erfüllen die folgende explizite Formel

$$p_n(x) = \frac{1}{e_n \omega(x)} \frac{d^n}{dx^n} \left\{ \omega(x) (g(x))^n \right\}, \quad n \in \mathbb{N}_0 \quad (2.16)$$

wobei $g(x)$ ein von n unabhängiges Polynom sei. Die Koeffizienten e_n ergeben sich in einigen wichtigen Fällen aus der Tabelle des nachfolgenden Beispiels.

Beispiel 2.1.19 In der nachfolgenden Tabelle werden einige, v.a. historisch wichtige Polynomterme p_n , ihre Gewichte und Standardisierungen sowie weitere zentrale Eigenschaften aufgelistet.

Wichtige Beispiele von Orthogonalpolynomen								
$p_n(x)$	Name	a	b	$\omega(x)$	Standard.	h_n	e_n	$g(x)$
$P_n(x)$	Legendre	-1	1	1	$P_n(1) = 1$	$\frac{2}{2n+1}$	$(-2)^n n!$	$1 - x^2$
$T_n(x)$	Tscheby.	-1	1	$1/\sqrt{1-x^2}$	$T_n(1) = 1$	$\begin{pmatrix} \pi/2, n \neq 0 \\ \pi, n=0 \end{pmatrix}$	$(-2)^n n!$	$1 - x^2$
$L_n(x)$	Laguerre ⁴	0	∞	e^{-x}	$k_n = (-1)^n / n!$	1	$1/n!$	x
$H_n(x)$	Hermite	$-\infty$	∞	e^{-x^2}	$k_n = 2^n$	$\sqrt{\pi} 2^n n!$	$(-1)^n$	1
$P_n^{(\alpha, \beta)}(x)$	Jacobi $\alpha, \beta > -1$	-1	1	$(1-x)^\alpha \cdot (1+x)^\beta$	$P_n^{(\alpha, \beta)}(1) = \binom{n+\alpha}{n}$		$(-2)^n n!$	$1 - x^2$

Für die Jacobi-Polynome gilt

$$h_n(P_n^{(\alpha, \beta)}) = \frac{2^{\alpha-\beta+1}}{2n+\alpha+\beta+1} \frac{\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{n!\Gamma(n+\alpha+\beta+1)}.$$

Aufgabe 2.1.20 Mit Hilfe der Rodrigues-Formel zeige man, dass für die Legendre-Polynome gilt

$$\int_{-1}^1 P_m(x) P_n(x) dx = \begin{cases} 0 & \text{falls } m \neq n \\ \frac{2}{2n+1} & \text{falls } m = n. \end{cases} \quad (m, n \in \mathbb{N}_0)$$

Bemerkung 2.1.21 Weiterhin gelten für die Jacobi-Polynome folgende Eigenschaften:

(i) $P_n^{(0,0)}(x) = P_n(x)$

³Rodrigues, Olinde (1794-1851)

⁴Laguerre, Edmond Nicolas (1834-1886)

$$(ii) P_n^{(-1/2, -1/2)}(x) = \binom{n-1/2}{n} T_n(x) = \frac{1}{4^n} \binom{2n}{n} T_n(x)$$

$$(iii) (1-x^2) \left(P_n^{(\alpha, \beta)} \right)'' + (\beta - \alpha - (\alpha + \beta + 2)x) \left(P_n^{(\alpha, \beta)} \right)' + n(n + \alpha + \beta + 1) P_n^{(\alpha, \beta)} = 0$$

Satz 2.1.22 (Christoffel-Darboux) *Mit der für Orthogonalpolynome zuvor eingeführten Notation (2.9), (2.10) gilt*

$$\sum_{i=0}^n \frac{p_i(x)p_i(y)}{h_i} = \frac{k_n}{k_{n+1} \cdot h_n} \cdot \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{x-y}. \quad (2.17)$$

Beweis. Für $n = 0$ erhält man die Identität (2.17) durch Einsetzen von $p_0(x) = k_0$ und $p_1(x) = k_1x + k'_1$ und für $n \geq 1$ aus der Rekursionsformel (2.11). Aus dieser folgt nämlich für $n \geq 1$ durch jeweiliges Einsetzen für p_{n+1}

$$\begin{aligned} & p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y) \\ &= ((a_{n+1} + b_{n+1}x)p_n(x) + c_{n+1}p_{n-1}(x))p_n(y) \\ &\quad - p_n(x)((a_{n+1} + b_{n+1}y)p_n(y) + c_{n+1}p_{n-1}(y)) \\ &= b_{n+1}(x-y)p_n(x)p_n(y) + c_{n+1}(p_{n-1}(x)p_n(y) - p_n(x)p_{n-1}(y)) \end{aligned}$$

Mit (2.12) erhalten wir für $x \neq y$ ($b_{n+1} = k_{n+1}/k_n \neq 0$ nach Voraussetzung)

$$\begin{aligned} & \frac{1}{b_{n+1}} \cdot \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{x-y} \\ &= p_n(x)p_n(y) + \frac{c_{n+1}}{b_{n+1}} \cdot \frac{p_{n-1}(x)p_n(y) - p_n(x)p_{n-1}(y)}{x-y}, \end{aligned}$$

d.h.

$$\begin{aligned} & \frac{k_n}{k_{n+1}} \cdot \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{x-y} \\ &= p_n(x)p_n(y) + \frac{k_{n+1}k_{n-1}h_nk_n}{k_n^2 h_{n-1} k_{n+1}} \cdot \frac{p_n(x)p_{n-1}(y) - p_{n-1}(x)p_n(y)}{x-y} \\ &= p_n(x)p_n(y) + \frac{k_{n-1}h_n}{k_n h_{n-1}} \cdot \frac{p_n(x)p_{n-1}(y) - p_{n-1}(x)p_n(y)}{x-y}. \end{aligned} \quad (2.18)$$

Gelte nun (2.17) für ein $n-1 \geq 0$ dann schließen wir mit (2.18) auf n wie folgt

$$\begin{aligned} & \frac{k_n}{k_{n+1}} \cdot \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{x-y} \\ &= p_n(x)p_n(y) + \frac{k_{n-1}h_n}{k_n h_{n-1}} \cdot \frac{p_n(x)p_{n-1}(y) - p_{n-1}(x)p_n(y)}{x-y} \\ &= p_n(x)p_n(y) + h_n \sum_{i=0}^{n-1} \frac{p_i(x)p_i(y)}{h_i} \end{aligned}$$

Division durch h_n liefert dann die Behauptung. □

Bemerkung 2.1.23 *Man beachte den Spezialfall $y \rightarrow x$ in Satz 2.1.21. Es gilt*

$$\begin{aligned} \sum_{i=0}^n \frac{p_i^2(x)}{h_i} &= \frac{k_n}{k_{n+1}h_n} \cdot \lim_{y \rightarrow x} \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{x-y} \\ &= \frac{k_n}{k_{n+1}h_n} \cdot \lim_{y \rightarrow x} \frac{(p_{n+1}(x) - p_{n+1}(y))p_n(y) - p_{n+1}(y)(p_n(x) - p_n(y))}{x-y} \\ &= \frac{k_n}{k_{n+1}h_n} \left(p'_{n+1}(x)p_n(x) - p'_n(x)p_{n+1}(x) \right). \end{aligned}$$

Damit gilt insbesondere für Nullstellen x^* von p_{n+1}

$$p'_{n+1}(x^*) p_n(x^*) = \frac{k_{n+1} h_n}{k_n} \sum_{i=0}^n \frac{p_i^2(x^*)}{h_i}. \quad (2.19)$$

Satz 2.1.24 Es sei $\omega \in C(a, b)$, $\omega(x) > 0$ ($a < x < b$) eine positive Gewichtsfunktion, $(\cdot, \cdot)_\omega$ das zugehörige Skalarprodukt. Dann hat ein Orthogonalpolynom $p_k(x)$ von echtem Grad k genau k einfache Nullstellen in (a, b) .

Beweis. Es seien x_1, \dots, x_m die $m \leq k$ verschiedenen Nullstellen $a < x_i < b$, an denen p_k sein Vorzeichen wechselt. Das Polynom

$$q(x) := (x - x_1) \cdot \dots \cdot (x - x_m)$$

wechselt dann an den gleichen Stellen sein Vorzeichen, sodass die Funktion $\omega(x)p_k(x)q(x)$ ihr Vorzeichen in (a, b) nicht ändert ($\omega(x)$ ist eine positive Gewichtsfunktion) und daher gilt

$$(q, p_k)_\omega = \int_a^b \omega(x)q(x)p_k(x) dx \neq 0.$$

Da jedoch p_k auf allen Polynomen aus \mathbb{P}_{k-1} senkrecht steht, folgt unmittelbar: $\text{grad } q = m \geq k$ und damit die Behauptung. \square

Satz 2.1.25 Es sei $\omega \in C(a, b)$ eine positive Gewichtsfunktion auf (a, b) und (p_k) eine Folge von Orthogonalpolynomen bzgl. des Skalarprodukts $(\cdot, \cdot)_\omega$. Des Weiteren sei $n \in \mathbb{N}$ und $x_1 < \dots < x_n$ die Nullstellen von $p_n(x)$. Dann existieren (von n abhängige) eindeutig bestimmte Gewichte $\omega_1, \dots, \omega_n \in \mathbb{R}_+$, sodass gilt:

$$\int_a^b \omega(x)q(x) dx = \omega_1 q(x_1) + \dots + \omega_n q(x_n) \quad \text{für alle } q \in \mathbb{P}_{2n-1}. \quad (2.20)$$

Bemerkung 2.1.26 Diese Art der Integralberechnung wird als Gauß-Quadratur bezeichnet.

Beweis zu Satz 2.1.24. Es sei p_n das bis auf Vielfachheit eindeutig bestimmte Orthogonalpolynom bzgl. $(\cdot, \cdot)_\omega$ vom echten Grad n . Dann existiert zu jedem $q \in \mathbb{P}_{2n-1}$ die eindeutige Darstellung $q = \varphi p_n + \psi$ mit $\varphi, \psi \in \mathbb{P}_{n-1}$. Das Ausnutzen der Orthogonalität von $(p_n, x^k)_\omega = 0$ für $k = 0, \dots, n-1$ liefert

$$\int_a^b \omega(x)q(x) dx = \int_a^b \omega(x)\varphi(x)p_n(x) dx + \int_a^b \omega(x)\psi(x) dx = \int_a^b \omega(x)\psi(x) dx. \quad (2.21)$$

Seien $L_1(x), \dots, L_n(x) \in \mathbb{P}_{n-1}$ die Lagrange-Polynome

$$L_k(x) = \prod_{\substack{j=1 \\ j \neq k}}^n \frac{x - x_j}{x_k - x_j} \in \mathbb{P}_{n-1} \quad k \in \{1, \dots, n\}$$

(vgl. hierzu Kapitel 3, Interpolation) zu x_1, \dots, x_n . Man beachte die Eigenschaft $L_k(x_j) = \delta_{jk}$, wobei δ_{jk} das Kronecker-Symbol ist. Es gilt dann folgende Darstellung

$$\psi(x) = \sum_{i=1}^n \psi(x_i) L_i(x)$$

und somit für (2.21)

$$\int_a^b \omega(x)q(x) dx = \int_a^b \omega(x) \left(\sum_{i=1}^n \psi(x_i) L_i(x) \right) dx = \sum_{i=1}^n \psi(x_i) \int_a^b \omega(x) L_i(x) dx.$$

Dies ist aber genau die Darstellung (2.20) mit $\omega_i = \int_a^b \omega(x) L_i(x) dx$, da an den Nullstellen von p_n gilt $q(x_i) = \varphi(x_i) p_n(x_i) + \psi(x_i) = \psi(x_i)$.

Beweisen wir nun die Eindeutigkeit der ω_i . Dazu seien $\omega_1, \dots, \omega_n$ und $\widetilde{\omega}_1, \dots, \widetilde{\omega}_n$ zwei verschiedene Wahlen der Koeffizienten in (2.20). Durch Differenzbildung erhalten wir also

$$0 = \sum_{i=1}^n \psi(x_i) (\omega_i - \widetilde{\omega}_i),$$

setzen wir nacheinander $\psi(x) = L_k(x)$ ($k = 1, \dots, n$), so folgt sofort $\omega_k = \widetilde{\omega}_k$ für alle $k \in \{1, \dots, n\}$.

Zuletzt müssen wir noch die Positivität der ω_i nachweisen. Setzen wir dazu $q(x) = L_k^2(x)$, dann folgt mit $\omega(x) > 0$ ($x \in (a, b)$) und (2.20)

$$0 < \int_a^b \omega(x) L_k^2(x) dx = \sum_{i=1}^n \omega_i L_k^2(x_i) = \omega_k$$

und damit die Behauptung. □

Satz 2.1.27 Für die Gewichte ω_k ($k = 1, \dots, n$) in Satz 2.1.24 gelten die Darstellungen

$$\omega_k = \frac{k_n h_{n-1}}{p'_n(x_k) p_{n-1}(x_k) k_{n-1}} = \left(\sum_{i=0}^{n-1} \frac{p_i^2(x_k)}{h_i} \right)^{-1} \quad (k = 1, \dots, n),$$

wobei x_k die Nullstellen von p_n seien.

Beweis. Es sei $x_k, k \in \{1, \dots, n\}$, eine Nullstelle von p_n . Dann gilt

$$\lim_{x \rightarrow x_k} \frac{p_n(x)}{x - x_k} = \lim_{x \rightarrow x_k} \frac{p_n(x) - p_n(x_k)}{x - x_k} = p'_n(x_k).$$

Wir definieren

$$q(x) = \begin{cases} p_n(x) p_{n-1}(x) / (x - x_k) & \text{falls } x \in \mathbb{R} \setminus \{x_k\} \\ p'_n(x_k) p_{n-1}(x_k) & \text{falls } x = x_k \end{cases}.$$

Da $q \in \mathbb{P}_{2n-2}$, gilt

$$\int_a^b \omega(x) q(x) dx = \sum_{j=1}^n \omega_j q(x_j) = \omega_k q(x_k) = \omega_k p'_n(x_k) p_{n-1}(x_k). \quad (2.22)$$

Nutzen wir die Darstellung $p_n(x) = \frac{k_n}{k_{n-1}}(x - x_k)p_{n-1}(x) + (x - x_k)r(x)$ mit einem $r \in \mathbb{P}_{n-2}$, so ergibt sich

$$\begin{aligned} \int_a^b \omega(x) q(x) dx &= \int_a^b \omega(x) \left(\frac{k_n}{k_{n-1}} p_{n-1}(x) + r(x) \right) p_{n-1}(x) dx \\ &= \frac{k_n}{k_{n-1}} \int_a^b \omega(x) p_{n-1}^2(x) dx = \frac{k_n h_{n-1}}{k_{n-1}}. \end{aligned} \quad (2.23)$$

Aus (2.22) und (2.23) erhalten wir die erste und daraus mit (2.19) die zweite Darstellung. □

Satz 2.1.28 Zwischen zwei Nullstellen von $p_n(x)$ liegt mindestens eine Nullstelle von $p_m(x)$ mit $m > n$.

Beweis. Seien x_1, \dots, x_m die Nullstellen von $p_m(x)$. Nach Satz 2.1.24 (vgl. (2.20)) gilt die Darstellung

$$\sum_{k=1}^m \omega_k p_n(x_k) q(x_k) = \int_a^b \omega(x) p_n(x) q(x) dx = 0, \quad (2.24)$$

wobei ω_k die positiven Gewichte bzgl. $x_1 < \dots < x_m$ darstellen und $q \in \mathbb{P}_{n-1}$ ist. Wählt man $q = 1$, so sieht man, dass wegen $\omega_k > 0$ ($k = 1, \dots, m$) die Folge $\{p_n(x_1), \dots, p_n(x_m)\}$ wenigstens einmal das Vorzeichen wechselt. Dies geschehe im Folgenden zwischen x_i und x_{i+1} . Sei nun $\xi_{\mu_1} \in (x_i, x_{i+1})$. Dann gilt mit $q = x_k - \xi_{\mu_1}$ und (2.24)

$$\sum_{k=1}^m \omega_k p_n(x_k) (x_k - \xi_{\mu_1}) = 0,$$

d.h. $p_n(x_k)(x_k - \xi_{\mu_1})$ wechselt sein Vorzeichen, jedoch nicht zwischen x_i und x_{i+1} . Dies lässt sich $(n-1)$ -mal fortführen und somit zeigen, dass die Folge $\{p_n(x_1), \dots, p_n(x_m)\}$ mindestens n -mal das Vorzeichen wechselt. Nach Satz 2.1.23 hat p_n jetzt genau n einfache Nullstellen in (a, b) , d.h. die Folge $\{p_n(x_1), \dots, p_n(x_m)\}$ wechselt genau n -mal das Vorzeichen. Somit gibt es folglich n verschiedene Intervalle

$$x_{\mu_k} < x < x_{\mu_{k+1}} \quad k = 1, \dots, n, \quad 1 \leq \mu_1 \leq \dots \leq \mu_{n+1} \leq m,$$

in denen p_n genau eine Nullstelle besitzt. Folglich liegt zwischen zwei Nullstellen von p_n mindestens eine Nullstelle von p_m . \square

Zum Ende des Abschnitts beschäftigen wir uns mit der Frage, wie sich der Fehler zwischen exakter Integration und Quadratur für eine Klasse von Funktionen bestimmen lässt. Vorbereitend benötigen wir jedoch noch den Mittelwertsatz der Integralrechnung.

Satz 2.1.29 (Mittelwertsatz der Integralrechnung) *Es sei $f : [a, b] \rightarrow \mathbb{R}$ stetig auf $[a, b]$, g sei Riemann-integrierbar mit $g(x) \geq 0$ (≤ 0) für alle $x \in [a, b]$. Dann gilt*

$$\int_a^b f(x)g(x) dx = f(\xi) \int_a^b g(x) dx \quad \text{für mindestens ein } \xi \in [a, b].$$

Beweis. Nach Voraussetzung gilt $g(x) \geq 0$ für alle $x \in [a, b]$. Gilt $\int_a^b g(x) dx = 0$, so ist $g(x) = 0$ bis auf höchstens endlich viele Ausnahmen, da g Riemann-integrierbar und $g(x) \geq 0$ ($x \in [a, b]$). Hieraus ergibt sich $\int_a^b f(x)g(x) dx = 0$ und somit die Behauptung.

Gelte nun $\int_a^b g(x) dx \neq 0$. Mit der Notation

$$m := \min_{\xi \in [a, b]} f(\xi) \leq f(x) \leq \max_{\xi \in [a, b]} f(\xi) =: M$$

folgt

$$m \int_a^b g(x) dx \leq \int_a^b f(x)g(x) dx \leq M \int_a^b g(x) dx.$$

Mit dem Zwischenwertsatz folgt aufgrund der Stetigkeit von f , dass ein $\xi \in [a, b]$ existiert mit

$$f(\xi) = \frac{\int_a^b f(x)g(x) dx}{\int_a^b g(x) dx},$$

womit die Aussage bewiesen wäre. \square

Satz 2.1.30 (A. Markoff) Für jede Funktion $f \in C^{2n}[a, b]$ lässt sich der Approximationsfehler der Gauß-Quadratur $Q_n[f] = \sum_{i=1}^n w_i f(x_i)$ ausdrücken durch

$$\int_a^b \omega(x) f(x) dx - \sum_{i=1}^n w_i f(x_i) = \frac{f^{(2n)}(\xi)}{(2n)!} \cdot \frac{(p_n, p_n)_\omega}{k_n^2},$$

für ein $\xi \in [a, b]$, wobei k_n der führende Koeffizient des n -ten Orthogonalpolynoms p_n ist.

Beweis. Wir betrachten das Restglied für die Hermite-Interpolierende $h(x)$, welche die Werte und Ableitungen $f(x_1), f'(x_1), \dots, f(x_n), f'(x_n)$ von f interpoliert (vgl. hierzu Satz 3.2.15 aus Kapitel 3). Der Interpolationsfehler ergibt sich nach Satz 3.2.11 aus selbigem Kapitel zu

$$f(x) - h(x) = \prod_{j=1}^n (x - x_j)^2 \cdot \frac{f^{(2n)}(\rho(x))}{(2n)!}, \quad x \in [a, b], \rho(x) \in (a, b).$$

Nun sind x_i die Nullstellen von $p_n(x)$ und $p_n \in \mathbb{P}_n$. Da nach Voraussetzung k_n der führende Koeffizient von p_n ist, folgt $(x - x_1) \cdots (x - x_n) = p_n(x)/k_n$ und somit auch

$$f(x) - h(x) = \frac{p_n^2(x)}{k_n^2} \frac{f^{(2n)}(\rho(x))}{(2n)!}.$$

Nach Definition von h hat $f - h$ an den Stellen x_i doppelte Nullstellen, so dass

$$(2n)! k_n^2 \frac{f(x) - h(x)}{p_n^2(x)} = f^{(2n)}(\rho(x))$$

auf ganz $[a, b]$ definiert und stetig ist. Da weiter $\omega(x)p_n^2(x) \geq 0$ für alle $x \in [a, b]$ gilt, lässt sich der Mittelwertsatz der Integralrechnung (Satz 2.1.28) anwenden, d.h.

$$\int_a^b \omega(x)(f(x) - h(x)) dx \int_a^b \omega(x) \frac{p_n^2(x)}{k_n^2} \frac{f^{(2n)}(\rho(x))}{(2n)!} dx = \frac{f^{(2n)}(\xi)}{(2n)!} \frac{(p_n, p_n)_\omega}{k_n^2} \quad \text{mit } \xi \in [a, b].$$

Betrachtet man nun $h \in \mathbb{P}_{2n-1}$ mit $h(x_i) = f(x_i)$ und $h'(x_i) = f'(x_i)$ ($i = 1, \dots, n$), so folgt aus der Integrationseigenschaft der Gauß-Quadratur

$$\begin{aligned} \int_a^b \omega(x) f(x) dx - \sum_{i=1}^n w_i f(x_i) &= \int_a^b \omega(x) f(x) dx - \sum_{i=1}^n w_i h(x_i) \\ &= \int_a^b \omega(x) (f(x) - h(x)) dx = \frac{f^{(2n)}(\xi)}{(2n)!} \frac{(p_n, p_n)_\omega}{k_n^2}, \end{aligned}$$

womit die Behauptung bewiesen wäre. □

Es folgen nun für spezielle Orthogonalpolynome, nämlich Tschebyscheff-, Legendre- und Jacobi-Polynome, Zusammenfassungen einiger ihrer Eigenschaften.

2.2 Tschebyscheff-POLYNOME

Die Tschebyscheff-Polynome T_n sind orthogonal bezüglich des Skalarprodukts

$$(f, g)_\omega := \int_{-1}^1 \frac{f(x)g(x)}{\sqrt{1-x^2}} dx$$

und werden standardisiert durch $T_n(1) = 1$. Durch Einsetzen dieser Eigenschaft in die Formeln von Satz 2.1.11 gewinnt man für $x \in \mathbb{R}$ die folgende Drei-Term-Rekursion

$$\boxed{T_0(x) = 1, T_1(x) = x, T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), k \geq 2}$$

Die *Tschebyscheff*-Polynome besitzen weiterhin die folgenden Eigenschaften:

- (i) Sie haben stets ganzzahlige Koeffizienten
- (ii) Der höchste Koeffizient von T_n ist $a_n = 2^{n-1}$
- (iii) T_n ist stets eine gerade Funktion, falls n gerade, und eine ungerade, falls n ungerade ist
- (iv) $T_n(1) = 1, T_n(-1) = (-1)^n$
- (v) $|T_n(x)| \leq 1$ für $x \in [-1, 1]$
- (vi) Die Nullstellen von $T_n(x)$ sind

$$x_k := \cos\left(\frac{2k-1}{2n}\pi\right), \quad (k = 1, \dots, n)$$

(vii)

$$T_k(x) = \begin{cases} \cos(k \cdot \arccos(x)), & |x| \leq 1, \\ \cosh(k \cdot \operatorname{Arccosh}(x)), & x > 1, \\ (-1)^k \cosh(k \cdot \operatorname{Arccosh}(-x)), & x < -1. \end{cases}$$

(viii) Die *Tschebyscheff*-Polynome besitzen die globale Darstellung

$$T_k(x) = \frac{1}{2}((x + \sqrt{x^2 - 1})^k + (x - \sqrt{x^2 - 1})^k), \quad \text{wobei } x \in \mathbb{R}.$$

(ix) $|T_n(x)|$ nimmt seinen maximalen Wert im Intervall $[-1, 1]$ an den sogenannten *Tschebyscheff*-Abszissen $\bar{x}_k = \cos(\frac{k\pi}{n})$ für $k = 0, \dots, n$ an, d.h.

$$|T_n(x)| = 1 \Leftrightarrow x = \bar{x}_k = \cos\left(\frac{k\pi}{n}\right) \quad \text{mit } k = 0, \dots, n.$$

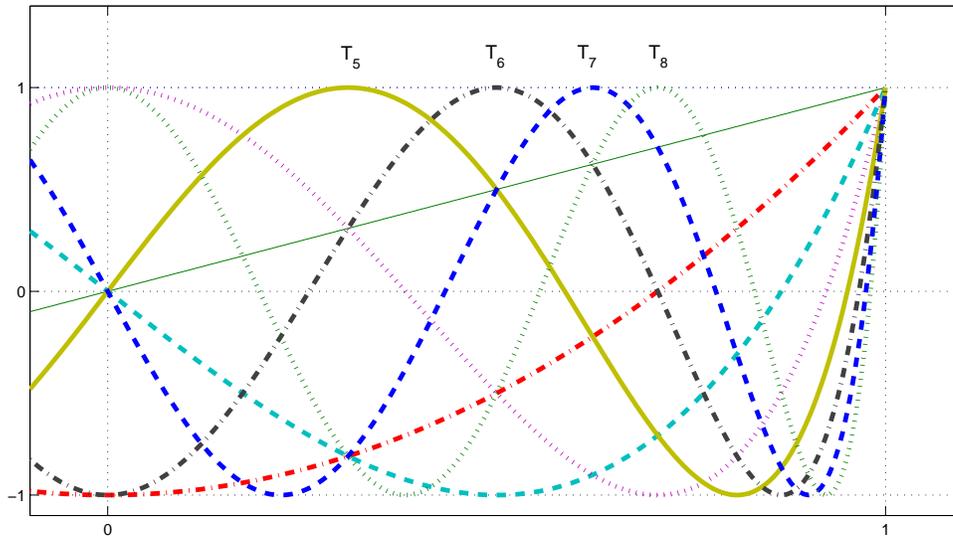
Abschließend möchten wir für $n = 0, \dots, 5$ die T_n explizit angeben und diese auch anhand der nachfolgenden Grafik veranschaulichen:

$$\begin{array}{ll} T_0 = 1 & T_3 = 4x^3 - 3x \\ T_1 = x & T_4 = 8x^4 - 8x^2 + 1 \\ T_2 = 2x^2 - 1 & T_5 = 16x^5 - 20x^3 + 5x \end{array}$$

2.3 Legendre-POLYNOME

Die *Legendre*-Polynome $P_n \in \mathbb{P}_n$ ($n = 0, 1, 2, \dots$) sind orthogonal bezüglich des L^2 -Skalarprodukts auf $[-1, 1]$, d.h.

$$(f, g)_\omega := \int_{-1}^1 f(x)g(x)dx,$$

Abb. 2.1: Tschebyscheff-Polynome T_n ($n = 0, \dots, 8$).

und sind durch $P_n(1) = 1$ standardisiert.

Die Legendre-Polynome besitzen folgende Eigenschaften:

(i) Sie erfüllen die Drei-Term-Rekursion

$$P_0(x) = 1, P_1(x) = x, (n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \quad n \in \mathbb{N}$$

(ii) $P_n(1) = 1, P_n(-1) = (-1)^n$ ($n = 0, 1, 2, \dots$)

(iii)

$$\int_{-1}^1 P_n^2(x) dx = \frac{2}{2n+1} \quad (n = 0, 1, 2, \dots) \quad (2.25)$$

(iv) Für Ableitung und Stammfunktion gilt

$$P'_n(x) = \frac{n(n+1)}{2n+1} \frac{P_{n+1}(x) - P_{n-1}(x)}{x^2 - 1} \quad (n \geq 1) \quad (2.26)$$

bzw.

$$\int_{-1}^x P_n(\xi) d\xi = \frac{1}{2n+1} \left(P_{n+1}(x) - P_{n-1}(x) \right) \quad (n \geq 1). \quad (2.27)$$

Im Folgenden sind P_0, \dots, P_5 explizit angegeben und in der nachfolgenden Grafik aufgezeichnet:

$$\begin{aligned} P_0 &= 1 & P_3 &= \frac{1}{2}(5x^3 - 3x) \\ P_1 &= x & P_4 &= \frac{1}{8}(35x^4 - 30x^2 + 3) \\ P_2 &= \frac{1}{2}(3x^2 - 1) & P_5 &= \frac{1}{8}(63x^5 - 70x^3 + 15x) \end{aligned}$$

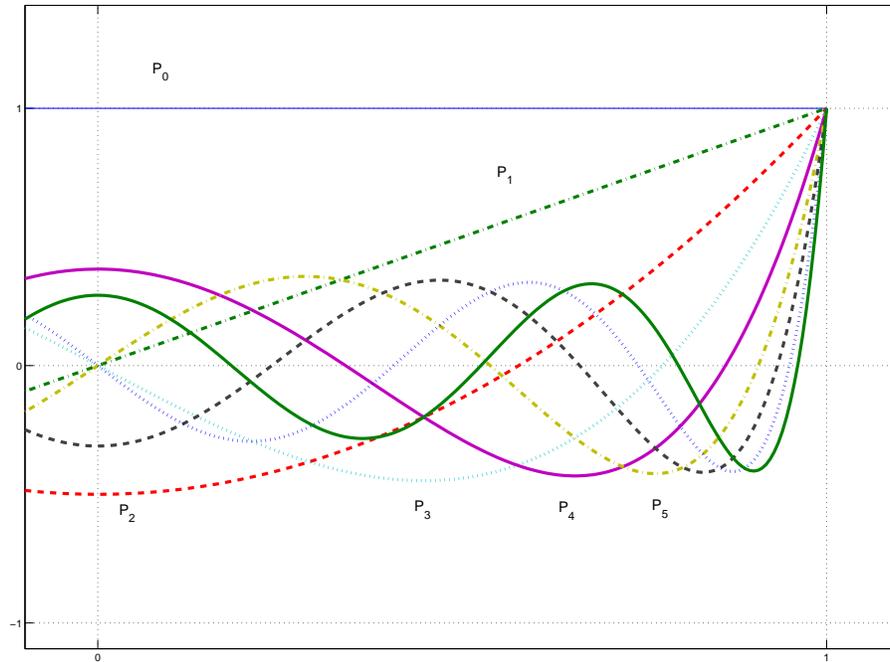


Abb. 2.2: Legendre-Polynome P_n ($n = 0, \dots, 8$).

Aufgabe 2.3.1 (i) Man zeige für $m \geq 1$

$$\int_{-1}^1 (1-x^2)P'_m(x)P_j(x) dx = \begin{cases} \frac{2m(m+1)}{(2m-1)(2m+1)} & \text{falls } j = m-1, \\ -\frac{2m(m+1)}{(2m+1)(2m+3)} & \text{falls } j = m+1, \\ 0 & \text{sonst.} \end{cases}$$

(ii) Man beweise (2.26) mit Hilfe von i).

(iii) Man zeige für $i, j \in \mathbb{N}_0$

$$\int_{-1}^1 P_i(x)P'_j(x) dx = \begin{cases} 2 & \text{falls } i < j \text{ und } j+i \text{ ungerade,} \\ 0 & \text{sonst.} \end{cases}$$

(iv) Man beweise (2.27) mit Hilfe von iii).

(v) Man zeige für $n \in \mathbb{N}$

$$\int_{-1}^x P_n(\xi) d\xi = \frac{1}{n(n+1)} (x^2-1) P'_n(x).$$

2.4 Jacobi-POLYNOME

Die *Jacobi*-Polynome $P_n^{(\alpha, \beta)}$ sind orthogonal bezüglich des durch die positive Gewichtsfunktion $(1-x)^\alpha(1+x)^\beta$ ($\alpha > -1, \beta > -1$) auf dem Intervall $(-1, 1)$ induzierten Skalarprodukts

$$(f, g)_\omega := \int_{-1}^1 (1-x)^\alpha(1+x)^\beta f(x)g(x) dx.$$

Somit sind *Legendre*- und *Tschebyscheff*-Polynome spezielle *Jacobi*-Polynome (vgl. hierzu auch Bemerkung 2.1.20), für die wir im Folgenden eine Aussage über die Verteilung der Nullstellen wiedergeben:

Satz 2.4.1 ([Sz]) Seien $|\alpha|, |\beta| \leq 1/2$ und $x_k = \cos \theta_k$ die Nullstellen von $P_n^{(\alpha, \beta)}(x)$ in absteigender Folge, d.h.

$$1 > x_1 > x_2 > \dots > x_n > -1; \quad 0 < \theta_1 < \theta_2 < \dots < \theta_n < \pi.$$

Dann gilt

$$\frac{k + (\alpha + \beta - 1)/2}{n + (\alpha + \beta + 1)/2} \pi < \theta_k < \frac{k}{n + (\alpha + \beta + 1)/2} \pi, \quad k = 1, 2, \dots$$

und im Falle $\alpha = \beta$ gilt

$$\theta_k \geq \frac{k + \alpha/2 - 1/4}{n + \alpha + 1/2} \pi, \quad k = 1, 2, \dots, [n/2],$$

wobei die Gleichheit für $\alpha = \beta = -1/2$ oder $\alpha = \beta = 1/2$ gilt.

Beweis. Wir verzichten an dieser Stelle auf die Wiedergabe eines Beweises dieser Aussage und verweisen auf [Sz]. \square

2.5 NULLSTELLEN VON ORTHOGONALPOLYNOMEN

Wie in Satz 2.1.16 bewiesen wurde, erfüllen die Orthogonalpolynome folgende Drei-Term-Rekursion mit den Startwerten $p_{-1} = 0, p_0 \in \mathbb{P}_0$:

$$p_n(x) = (a_n + b_n x)p_{n-1} + c_n p_{n-2} \quad (n = 1, 2, \dots).$$

Schematisch dargestellt bedeutet dies:

$$\begin{aligned} p_1 &= (a_1 + b_1 x)p_0 && \Leftrightarrow && -\frac{a_1}{b_1} \cdot p_0 + \frac{1}{b_1} \cdot p_1 = x p_0 \\ p_2 &= (a_2 + b_2 x)p_1 + c_2 p_0 && \Leftrightarrow && -\frac{c_2}{b_2} \cdot p_0 - \frac{a_2}{b_2} \cdot p_1 + \frac{1}{b_2} \cdot p_2 = x p_1 \\ &\vdots && && \vdots \\ p_{n+1} &= (a_{n+1} + b_{n+1} x)p_n + c_{n+1} p_{n-1} && \Leftrightarrow && -\frac{c_{n+1}}{b_{n+1}} \cdot p_{n-1} - \frac{a_{n+1}}{b_{n+1}} \cdot p_n = x p_n - \frac{1}{b_{n+1}} \cdot p_{n+1} \end{aligned}$$

Aus dem Schema ist aber ersichtlich, dass wir obige Rekursion auch als folgendes lineares Gleichungssystem interpretieren können, d.h.

$$\underbrace{\begin{pmatrix} -\frac{a_1}{b_1} & \frac{1}{b_1} & 0 & \dots & \dots & 0 \\ -\frac{c_2}{b_2} & -\frac{a_2}{b_2} & \frac{1}{b_2} & & & \vdots \\ 0 & -\frac{c_3}{b_3} & -\frac{a_3}{b_3} & \frac{1}{b_3} & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & -\frac{1}{b_{n+1}} \\ 0 & \dots & 0 & -\frac{c_{n+1}}{b_{n+1}} & -\frac{a_{n+1}}{b_{n+1}} & \end{pmatrix}}_{=: A} \cdot \begin{pmatrix} p_0(x) \\ \vdots \\ p_n(x) \end{pmatrix} = x \cdot \underbrace{\begin{pmatrix} p_0(x) \\ \vdots \\ p_n(x) \end{pmatrix}}_{=: p} + \underbrace{\begin{pmatrix} 0 \\ \vdots \\ 0 \\ -\frac{p_{n+1}(x)}{b_{n+1}} \end{pmatrix}}_{=: r}$$

In Matrixschreibweise also

$$Ap = xp + r. \quad (2.28)$$

Aus dieser Darstellung ergibt sich das folgende bemerkenswerte Resultat:

Satz 2.5.1 (Nullstellen von Orthogonalpolynomen) Die Nullstellen von p_{n+1} sind genau die Eigenwerte von A , d.h. die Lösungen der Gleichung $Ap = xp$ für $p \neq 0$. Die Eigenvektoren zu Eigenwerten x_k ($k = 1, \dots, n+1$) lauten bis auf ein Vielfaches $(p_0(x_k), \dots, p_n(x_k))^T$.

Beweis. Der Beweis ist nicht sehr schwer und bleibt dem Leser als Aufgabe überlassen. \square

Da p_n ausschließlich reelle Nullstellen besitzt, liegt die Idee nahe, dass sich A in eine symmetrische Matrix mittels einer Ähnlichkeitstransformation transformieren lässt. Die einfachste Möglichkeit wäre eine Skalierung mit einer Diagonalmatrix.

Da die Orthogonalität die Orthogonalpolynome nur bis auf ein Vielfaches eindeutig bestimmt, steht es uns frei, $b_k > 0$ ($k = 1, 2, 3, \dots$) zu wählen, zumal die Nullstellen der Orthogonalpolynome davon nicht betroffen sind. Wegen $b_k > 0$ folgt somit aus $h_k > 0$ und (2.12) $c_k < 0$ ($k = 2, 3, \dots$).

Satz 2.5.2 Es seien $a_j \in \mathbb{R}$, $b_j > 0$ ($j = 1, \dots, n$), $c_j < 0$ ($j = 2, \dots, n$) und

$$A := \begin{pmatrix} -\frac{a_1}{b_1} & \frac{1}{b_1} & 0 & \cdots & \cdots & 0 \\ -\frac{c_2}{b_2} & -\frac{a_2}{b_2} & \frac{1}{b_2} & & & \vdots \\ 0 & -\frac{c_3}{b_3} & -\frac{a_3}{b_3} & \frac{1}{b_3} & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & \frac{1}{b_{n-1}} \\ 0 & \cdots & & 0 & -\frac{c_n}{b_n} & -\frac{a_n}{b_n} \end{pmatrix} \in \mathbb{R}^{n \times n}. \quad (2.29)$$

Dann existiert eine Diagonalmatrix $D \in \mathbb{R}^{n \times n}$, so dass $\hat{A} := (\hat{a}_{ij})_{i,j=1}^n = DAD^{-1}$ symmetrisch ist, und es gilt

$$\hat{a}_{ii} = -\frac{a_i}{b_i}, \quad \hat{a}_{i,i-1} = \hat{a}_{i-1,i} = \frac{|c_i|^{1/2}}{b_i^{1/2} b_{i-1}^{1/2}} \quad (2.30)$$

und für $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ ergibt sich

$$Dx = \left(x_1 \sqrt{b_1}, x_2 \frac{\sqrt{b_2}}{\sqrt{|c_2|}}, \dots, x_n \frac{\sqrt{b_n}}{\sqrt{|c_2 \cdots c_n|}} \right).$$

Beweis. Allgemein gilt für eine Matrix $M = (m_{ij}) \in \mathbb{R}^{n \times n}$ und eine Diagonalmatrix $D = \text{diag}(d_1, \dots, d_n) \in \mathbb{R}^{n \times n}$, dass die Einträge von

$$\hat{M} = (\hat{m}_{ij}) := DMD^{-1} \quad \text{die Form} \quad \hat{m}_{ij} = \frac{d_i}{d_j} m_{ij} \quad \text{haben.}$$

Damit nun \hat{A} symmetrisch ist, muss $\hat{a}_{i,i-1} = \hat{a}_{i-1,i}$ gelten und somit

$$-\frac{c_i}{b_i} \frac{d_i}{d_{i-1}} = \frac{1}{b_{i-1}} \frac{d_{i-1}}{d_i}, \quad \text{d.h.} \quad d_i^2 = d_{i-1}^2 \cdot \frac{b_i}{b_{i-1} |c_i|}. \quad (i = 2, 3, \dots, n+1) \quad (2.31)$$

Dies erreichen wir nun z.B., indem wir $d_1 = \sqrt{b_1}$ setzen. Dann folgt mit (2.31)

$$d_2^2 = d_1^2 \frac{b_2}{b_1 |c_2|} = \frac{b_2}{|c_2|}, \quad d_3^2 = d_2^2 \frac{b_3}{b_2 |c_3|} = \frac{b_2}{|c_2|} \frac{b_3}{b_2 |c_3|} = \frac{b_3}{|c_2 c_3|}, \quad d_4^2 = d_3^2 \frac{b_4}{b_3 |c_4|} = \frac{b_4}{|c_2 c_3 c_4|}.$$

Allgemein folgt aus

$$d_{k-1}^2 = \frac{b_{k-1}}{|c_2 \cdot c_3 \cdot \dots \cdot c_{k-1}|}$$

die Darstellung

$$d_k^2 = d_{k-1}^2 \frac{b_k}{b_{k-1}|c_k|} = \frac{b_k}{|c_2 \cdot \dots \cdot c_k|}.$$

Bei dieser Wahl von D gilt für die Diagonaleinträge \hat{a}_{ii} ($i = 1, \dots, n$) der symmetrischen Tridiagonalmatrix \hat{A}

$$\hat{a}_{ii} = \frac{d_i}{d_i} a_{ii} = -\frac{a_i}{b_i},$$

und für die Nebendiagonaleinträge

$$\hat{a}_{i,i-1} = d_i \cdot a_{i,i-1} \cdot d_{i-1}^{-1} = \frac{b_i^{1/2}}{|c_2 \cdot \dots \cdot c_i|^{1/2}} \cdot \frac{|c_i|}{b_i} \cdot \frac{|c_2 \cdot \dots \cdot c_{i-1}|^{1/2}}{b_{i-1}^{1/2}} = \frac{|c_i|^{1/2}}{b_i^{1/2} b_{i-1}^{1/2}} = \hat{a}_{i-1,i}.$$

Für das Matrix-Vektorprodukt von D mit einem Vektor $x \in \mathbb{R}^n$ gilt

$$\begin{aligned} Dx &= \text{diag}(d_1, \dots, d_n)x = (d_1 x_1, d_2 x_2, \dots, d_n x_n) \\ &= \left(x_1 \sqrt{b_1}, x_2 \frac{\sqrt{b_2}}{\sqrt{|c_2|}}, \dots, x_n \frac{\sqrt{b_n}}{\sqrt{|c_2 \cdot \dots \cdot c_n|}} \right). \end{aligned}$$

□

Wie wir später sehen werden, lassen sich symmetrische Eigenwertprobleme numerisch besser behandeln. Anstatt also $Au = \lambda u$ zu analysieren, betrachten wir

$$DAD^{-1}Du = \lambda Du$$

d.h.

$$\hat{A}\hat{u} = \lambda\hat{u}$$

mit $\hat{A} := DAD^{-1}$ und $\hat{u} := Du$. Unter dieser Ähnlichkeitstransformation bleiben also die Eigenwerte unverändert und für die zugehörigen Eigenvektoren gilt, dass Eigenvektoren u in Du übergehen. Unter der im Beweis von Satz 2.5.2 verwendeten Ähnlichkeitstransformation lauten somit die Eigenvektoren zu $\hat{A}\hat{u} = \lambda\hat{u}$ zum Eigenwert x_k ($k = 1, \dots, n$) bis auf eine nichtverschwindende Konstante $c \in \mathbb{R}$

$$\hat{u}_k = c \left(p_0 \sqrt{b_1}, \frac{p_1(x_k) \sqrt{b_2}}{\sqrt{|c_2|}}, \dots, \frac{p_{n-1}(x_k) \sqrt{b_n}}{\sqrt{|c_2 \cdot \dots \cdot c_n|}} \right).$$

Satz 2.5.3 *Es gelten die Voraussetzungen wie in Satz 2.5.2 und \hat{A} sei durch (2.29) definiert. Des Weiteren sei \hat{U} die Matrix, die spaltenweise die Eigenvektoren $\hat{u}_k = (\hat{u}_{k1}, \dots, \hat{u}_{kn})^T$ zu den Eigenwerten x_k ($k = 1, \dots, n$) enthält, d.h.*

$$\hat{A}\hat{U} = \text{diag}(x_1, \dots, x_n)\hat{U}.$$

Dann sind die Gewichte in Satz 2.1.24 gegeben durch

$$\omega_k = (1, 1)_\omega \frac{\hat{u}_{k1}^2}{\hat{u}_k^T \hat{u}_k} \quad (k = 1, \dots, n). \quad (2.32)$$

Beweis. Die Vektoren \hat{v}_k seien Vielfache der Eigenvektoren \hat{u}_k ($k = 1, \dots, n$) mit der Eigenschaft, dass für den ersten Eintrag jeweils $\hat{v}_{k1} = p_0 \sqrt{b_1}$ gilt. Sei $\mathcal{V} = (\hat{v}_1 | \hat{v}_2 | \dots | \hat{v}_n)$ die Matrix, die spaltenweise die Eigenvektoren \hat{v}_k enthält und $\vec{\omega} = (\omega_1, \omega_2, \dots, \omega_n)^T$ der Vektor der gesuchten Gewichte. Aufgrund der verwendeten Normierung steht in der ersten Zeile jeweils $p_0 \sqrt{b_1}$.

Nach Satz 2.1.24 integriert die gesuchte Quadraturformel Polynome bis zum Grad $2n - 1$ exakt, d.h.

$$\sum_{j=1}^n \omega_j q(x_j) = \int_a^b \omega(x) q(x) dx = \begin{cases} (1, p_0)_\omega & \text{für } q = p_0, \\ 0 & \text{für } q = p_k, 1 \leq k \leq n. \end{cases}$$

Damit folgt

$$\mathcal{V} \vec{\omega} = \begin{pmatrix} \sum_j \omega_j p_0 \sqrt{b_1} \\ \sum_j \omega_j p_1(x_j) \sqrt{b_2 / |c_2|} \\ \vdots \\ \sum_j \omega_j p_{n-1}(x_j) \sqrt{b_n / |c_2 \cdot \dots \cdot c_n|} \end{pmatrix} = p_0 \sqrt{b_1} \begin{pmatrix} (1, 1)_\omega \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (2.33)$$

Die Nullstellen von p_n sind einfach nach Satz 2.1.23. Damit folgt aus Satz 2.5.1, dass auch die Eigenwerte von \hat{A} einfach sind. Für Eigenvektoren \hat{v}_k, \hat{v}_m zu verschiedenen Eigenwerten x_k, x_m ($k \neq m$) gilt nun, da \hat{A} symmetrisch ist

$$x_m \hat{v}_m^T \hat{v}_k = x_m \hat{v}_k^T \hat{v}_m = \hat{v}_k^T \hat{A} \hat{v}_m = \hat{v}_m^T \hat{A}^T \hat{v}_k = \hat{v}_m^T \hat{A} \hat{v}_k = x_k \hat{v}_m^T \hat{v}_k.$$

Somit gilt $\hat{v}_m^T \hat{v}_k = 0$ für $k \neq m$, d.h. Eigenvektoren zu paarweise verschiedenen Eigenwerten stehen senkrecht aufeinander. Für einen beliebigen Eigenvektor \hat{v}_k ergibt sich unter Ausnutzung von (2.33) und der eben gezeigten Orthogonalität

$$p_0^2 b_1 (1, 1)_\omega = \begin{pmatrix} p_0 \sqrt{b_1} \\ \frac{p_1(x_k) \sqrt{b_2}}{\sqrt{|c_2|}} \\ \vdots \\ \frac{p_{n-1}(x_k) \sqrt{b_n}}{\sqrt{|c_2 \cdot \dots \cdot c_n|}} \end{pmatrix}^T \cdot \begin{pmatrix} p_0 \sqrt{b_1} (1, 1)_\omega \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \hat{v}_k^T \mathcal{V} \vec{\omega} = \hat{v}_k^T \sum_{i=1}^n \omega_i \hat{v}_i = \omega_k \hat{v}_k^T \hat{v}_k.$$

Berücksichtigung der vorangegangenen Normierung, d.h. $\hat{v}_k = p_0 \sqrt{b_1} \hat{u}_k / \hat{u}_{k1}$, liefert das Ergebnis. \square

Bemerkung 2.5.4 Sind die Momente $m_k := \int_a^b \omega(x) x^k dx$ ($k = 0, \dots, 2n - 1$) bekannt, so lassen sich mit Satz 2.1.11 die Koeffizienten α_j ($j = 1, \dots, n$) und γ_j ($j = 2, \dots, n$) in der Drei-Term-Rekursion

$$p_n(x) = (\alpha_n + x)p_{n-1}(x) + \gamma_n p_{n-2}(x)$$

mittels

$$\alpha_n = -\frac{(xp_{n-1}, p_{n-1})_\omega}{(p_{n-1}, p_{n-1})_\omega}, \quad \gamma_n = -\frac{(p_{n-1}, p_{n-1})_\omega}{(p_{n-2}, p_{n-2})_\omega}$$

bestimmen. Denn gilt $p_k(x) = \sum_{j=0}^k \lambda_j x^j$ ($k \geq 0$), so erhält man für

$$(p_k, p_k)_\omega = \sum_{i=0}^k \sum_{j=0}^k \lambda_i \lambda_j \int_a^b \omega(x) x^{i+j} dx = \sum_{i=0}^k \sum_{j=0}^k \lambda_i \lambda_j \cdot m_{i+j}$$

und

$$(xp_k, p_k)_\omega = \sum_{i=0}^k \sum_{j=0}^k \lambda_i \lambda_j \cdot m_{i+j+1}.$$

Setzt man nun die so berechneten Werte α_j, γ_j in die Tridiagonalmatrix \hat{A} mit den Einträgen gegeben durch (2.30) für $a_j = \alpha_j, b_j = 1$ und $c_j = \gamma_j$ ein, so sind die Quadraturstellen zur Gewichtsfunktion ω gerade die Eigenwerte von \hat{A} und die Gewichte lassen sich mittels (2.32) aus den zugehörigen Eigenvektoren bestimmen.

MATLAB-Funktion: GaussWeightsNodes.m

```

1  function [weights,nodes,kn] = GaussWeightsNodes(momente)
2  n = (length(momente)+1)/2;
3  % Orth.polynome und Koeff. mit 3-Term-Rekursion bestimmen
4  % p_k = ( x - aa_(k-1) ) * p_(k-1) - cc_(k-1) * p_(k-2)
5  h(1) = momente(1);           % h = ( p_k , p_k )
6  hx(1) = momente(2);         % hx = ( x * p_k , p_k )
7  aa(1) = hx(1)/h(1);
8  p{1} = 1;
9  p{2} = [-aa(1);1];
10 for k = 3:n                   % bestimme p_k mit 3-Term-Rekurs.
11     q = mult(p{k-1},p{k-1});   % berechne q = p_(k-1) * p_(k-1)
12     h(k-1) = integ(q,momente); % berechne int_a^b omega*q dx
13     hx(k-1) = integ([0;q],momente); % berechne int_a^b omega*x*q dx
14     aa(k-1,1) = hx(k-1)/h(k-1); % bestimme Koeff. vor p_(k-1)
15     cc(k-1,1) = h(k-1)/h(k-2); % bestimme Koeff. vor p_(k-2)
16     p{k} = [0;p{k-1}] - aa(k-1)*[p{k-1};0] - cc(k-1) * [p{k-2};0;0];
17 end
18 if length(aa)==1              % sym. Matrix A aufstellen
19     A = aa;
20 else
21     A = full(spdiags([[sqrt(cc(2:end));0],aa,sqrt(cc)],...
22         [-1 0 1],n-1,n-1));
23 end
24 [V,D] = eig(A);               % Eigenwerte bestimmen
25 nodes = diag(D);              % Quad.-stellen sind Eigenw. von A
26 weights = zeros(n-1,1);      % Quadraturgewichte sind skalierte
27 for k = 1:n-1                  % Skalarprodukte der Eigenvektoren
28     weights(k) = momente(1) * V(1,k)^2 / (V(:,k)' * V(:,k));
29 end
30 kn = integ(mult(p{n},p{n}),momente); % Koeffizient im Fehlerterm
31 % einige Unterfunktionen zur Behandlung der auftretenden Polynome
32 % Polynome miteinander multiplizieren
33 function p = mult(s,t)
34     p = zeros(length(s)+length(t)-1,1);
35     for j = 1:length(s);
36         p(j:length(t)+j-1) = p(j:length(t)+j-1) + s(j) * t(:);
37     end
38 end
39 % Polynome addieren
40 function r = add(p,q)
41     if length(q) >= length(p)
42         r = q(:);
43         r(1:size(p,1)) = r(1:size(p,1)) + p(:);
44     else
45         r = p(:);
46         r(1:size(q,1)) = r(1:size(q,1)) + q(:);
47     end
48 end
49 % Integral auswerten
50 function r = integ(p,momente)
51     r = (p(:))' * momente(1:length(p))';
52 end
53 end

```

MATLAB-Funktion: GaussQuadratur.m

```

1 function [weights,nodes,kn] = GaussQuadratur(anz_Gewichte,typ)
2   N = 2 * anz_Gewichte+1;
3   switch typ
4     case 'legendre'
5       momente = 2./(1:N); momente(2:2:end) = 0;
6     case 'tschebyscheff'
7       momente = [pi,0,pi/2];
8       for j= 2:anz_Gewichte
9         momente = [momente,0,momente(end)*(2*j-1)/(2*j)];
10      end
11    case 'laguerre'
12      momente = factorial(0:N-1);
13    case 'hermite'
14      momente = gamma((1:N)./2); momente(2:2:end) = 0;
15    case 'log'
16      momente = (1./(1:N)).^2;
17    otherwise
18      disp('Nur "legendre", "tschebyscheff", ', ...
19           ' "laguerre", "hermite", "log" möglich!')
20      return
21    end
22   [weights,nodes,kn] = GaussWeightsNodes(momente);
23 end

```

MATLAB-Beispiel:

Man kann z.B. die Quadraturstellen und Gewichte zur Gewichtsfunktion

$$\omega(x) = -\log(x)$$

und $n = 2$ berechnen lassen, sie erfüllen dann

$$\begin{aligned}
 & - \int_0^1 \log(x) f(x) dx \\
 &= \sum_{j=1}^n \omega_j f(x_j) + \frac{f^{(2n)}(\xi)}{(2n)!} K_n,
 \end{aligned}$$

mit $K_n = (p_n, p_n)_\omega / k_n^2$.

```

>> [x,w,Kn] = GaussQuadratur(2,'log')
x =
    0.7185
    0.2815
w =
    0.1120
    0.6023
Kn =
    0.0029

```

Bemerkung 2.5.5 Umfangreiche Tabellen von Stützstellen und Gewichten zu verschiedenen Gewichtsfunktionen findet man z.B. in [AS] und [St].

Tab. 2.1: Momente für einige Standardfälle von Orthogonalpolynomen

Name	a	b	$\omega(x)$	$m_k := \int_a^b \omega(x)x^k dx$
Legendre	-1	1	1	$m_k = \begin{cases} 2/(k+1) & \text{falls } k \text{ gerade} \\ 0 & \text{falls } k \text{ ungerade} \end{cases}$
Tscheby.	-1	1	$\frac{1}{\sqrt{1-x^2}}$	$m_k = \begin{cases} \pi \cdot \frac{1 \cdot 3 \cdot 5 \cdot \dots \cdot k-1}{2 \cdot 4 \cdot 6 \cdot \dots \cdot k} & \text{falls } k \text{ gerade} \\ 0 & \text{falls } k \text{ ungerade} \end{cases}$
Laguerre	0	∞	e^{-x}	$m_k = k!$
Hermite	$-\infty$	∞	e^{-x^2}	$m_k = \begin{cases} \Gamma(\frac{k+1}{2}) & \text{falls } k \text{ gerade} \\ 0 & \text{falls } k \text{ ungerade} \end{cases}$
Jacobi $\alpha = \beta > -1$	-1	1	$(1-x)^\alpha(1+x)^\beta$	$m_k = \begin{cases} \sqrt{\pi} \frac{1 \cdot 3 \cdot 5 \cdot \dots \cdot k-1}{2^{k/2}} \frac{\Gamma(\alpha+1)}{\Gamma(\alpha+(k+3)/2)} & \text{falls } k \text{ gerade} \\ 0 & \text{falls } k \text{ ungerade} \end{cases}$
„log“	0	1	$-\log(x)$	$m_k = 1/(k+1)^2$

Bemerkung 2.5.6 In höheren Dimensionen hat man eine solche Theorie der Orthogonalpolynome nicht, aus der man Quadraturformeln gewinnen könnte. Hier bleibt einem häufig nichts anderes übrig, als diese näherungsweise als Lösung nichtlinearer Gleichungen zu bestimmen.

2.6 RADAU- UND LOBATTO-FORMELN

Manchmal bietet es sich an, den Funktionswert an einem Rand oder beiden Endpunkten mit in die Quadraturformel einzubringen. Sei dies z.B. der linke Randnoten a . Wie ersetzen n durch $n+1$ in (??) und schreiben $\tilde{\omega}(t) = (t-a)\omega(t)$. Die optimale Formel erfüllt nun

gg

$$f(x) = (x-a)(b-x)\psi(x) + \frac{b-x}{b-a}f(a) + \frac{x-a}{b-a}f(b)$$

bzw. falls $a < x < b$

$$\psi(x) = \frac{1}{(x-a)(b-x)} \left(f(x) - \frac{b-x}{b-a}f(a) - \frac{x-a}{b-a}f(b) \right)$$

Es gelte

$$\int_a^b p(x)(x-a)(b-x)\omega(x) dx = \sum_{i=1}^n \omega_i p(x_i) \quad (p \in \mathbb{P}_{2n+1})$$

und

$$m_0 = \frac{1}{b-a} \int_a^b \omega(x) dx \quad m_1 = \frac{1}{b-a} \int_a^b x\omega(x) dx$$

$$\begin{aligned}
\int_a^b f(x) \omega(x) dx &= \int_a^b \psi(x)(x-a)(b-x) \omega(x) dx + f(a) \int_a^b \frac{b-x}{b-a} \omega(x) dx + f(b) \int_a^b \frac{x-a}{b-a} \omega(x) dx \\
&= \sum_{i=1}^n \omega_i \psi(x_i) + f(a)(bm_0 - m_1) + f(b)(m_1 - am_0) \\
&= \sum_{i=1}^n \omega_i \frac{1}{(x_i - a)(b - x_i)} \left(f(x_i) - \frac{b - x_i}{b - a} f(a) - \frac{x_i - a}{b - a} f(b) \right) \\
&\quad + f(a)(bm_0 - m_1) + f(b)(m_1 - am_0) \\
&= \sum_{i=1}^n \frac{\omega_i}{(x_i - a)(b - x_i)} f(x_i) \\
&\quad + f(a) \left(\sum_{i=1}^n \frac{\omega_i}{(x_i - a)(b - a)} + bm_0 - m_1 \right) \\
&\quad + f(b) \left(\sum_{i=1}^n \frac{\omega_i}{(b - x_i)(b - a)} - am_0 + m_1 \right) \\
&= \sum_{i=1}^n \omega_i^* f(x_i) + \omega_a^* f(a) + \omega_b^* f(b)
\end{aligned}$$

Diese Kronrod-Formel ist exakt für $f \in \mathbb{P}_{2n+3}$.

Für den Fehlerterm gilt...

2.7 KRONROD FORMELN

Bei der praktischen Lösung von Integrationsproblemen haben Gauß-Quadraturformeln den Nachteil, dass zwei Quadraturformeln Q_m und Q_n mit $m > n$ keine gemeinsamen Stützstellen haben (eventuell den Intervallmittelpunkt). Die übliche und einfache Methode zur näherungsweise Bestimmung des Quadraturfehlers, zwei Formeln mit verschiedener Anzahl von Quadraturpunkten auszuwerten und die Differenz als Fehlerabschätzung zu verwenden, wäre hier nicht sehr effizient. Zu viele Funktionsauswertungen müssten berechnet werden. Dieser Nachteil kann durch eine naheliegende Vorgehensweise überwunden werden (erstmalig 1965 publiziert von A. S. Kronrod). Man gibt ähnlich wie bei der Konstruktion der Herleitung der Radau- und Lobatto-Formeln die n Quadraturstellen von Q_n vor und konstruiert eine Quadraturformel mit $(2n + 1)$ Stellen, die den größtmöglichen Genauigkeitsgrad $3n + 1$ für gerades n oder $3n + 2$ für ungerades n besitzt. Die neuen Stützstellen liegen im Inneren des Intervalls und werden durch die n Quadraturstellen von Q_n getrennt. Laurie ([Calvetti]) zeigte 1997, dass sich die Berechnung der Quadraturstellen (und daraus dann auch die Quadraturgewichte) wiederum auf ein Eigenwertproblem zurückführen lässt. Häufig sind die Quadraturstellen der ersten Quadraturformel jedoch. Einen Algorithmus, der die nochmalige Berechnung der ersten Quadraturstellen vermeidet, wurde von Calvetti, Golub, Gragg und Reichel [Calvetti] vorgestellt.

2.8 LINEARKOMBINATION VON ORTHOGONALPOLYNOMEN

Gegeben sei eine Linearkombination von Orthogonalpolynomen der Form

$$S_N(x) = \sum_{k=0}^N \alpha_k p_k(x),$$

wobei alle $p_k(x)$ einer Drei-Term-Rekursion der Form (2.11) mit den Startwerten p_0, p_1 genügen. Hierfür ergeben sich folgende Möglichkeiten zur Auswertung einer Linearkombination von Orthogonalpolynomen, nämlich Vorwärts- und Rückwärtsrekursion.

2.8.1 Vorwärtsrekursion

Eine naheliegende Berechnung der obigen Summe besteht darin, in jedem Schritt mit Hilfe von (2.11) die Werte p_k zu berechnen, diese mit den α_k zu multiplizieren und schließlich aufzuaddieren. Wir fassen im Folgenden die Summe $a_k + b_k x$ zum Faktor \underline{a}_k zusammen, d.h.

$$p_k = \underline{a}_k p_{k-1} + \underline{b}_k p_{k-2}$$

Aus obigem Schema gewinnen wir folgenden Algorithmus für die Vorwärtsrekursion:

Seien p_0, p_1 und $s_1 := \alpha_0 p_0 + \alpha_1 p_1$ gegeben, dann gilt für $k = 2, \dots, N$:

$$\begin{aligned} p_k &= \underline{a}_k p_{k-1} + \underline{b}_k p_{k-2} \\ s_k &= s_{k-1} + \alpha_k p_k \end{aligned}$$

Die Auswertung der $p = (p_0, \dots, p_N)$ ist nun äquivalent der Auswertung des gestaffelten Gleichungssystems

$$\underbrace{\begin{pmatrix} 1 & & & \cdots & & 0 \\ 0 & 1 & & & & \vdots \\ -\underline{b}_2 & -\underline{a}_2 & 1 & & & \\ & -\underline{b}_3 & -\underline{a}_3 & 1 & & \\ \vdots & & & \ddots & \ddots & \\ 0 & \cdots & & -\underline{b}_N & -\underline{a}_N & 1 \end{pmatrix}}_{=: L} \cdot \underbrace{\begin{pmatrix} p_0 \\ \vdots \\ p_N \end{pmatrix}}_{=: p} = \underbrace{\begin{pmatrix} p_0 \\ p_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{=: r}.$$

Die gegebene Linearkombination S_N ist also gerade das Skalarprodukt

$$S_N = \sum_{k=0}^N \alpha_k p_k = \langle \alpha, p \rangle (:= \alpha^T p), \quad \text{mit } Lp = r.$$

2.8.2 Rückwärtsrekursion

Sei u Lösung von $L^T u = \alpha$, d.h.

$$S_N = \langle \alpha, L^{-1} r \rangle = \langle L^{-T} \alpha, r \rangle = \langle u, r \rangle,$$

bzw. in der Matrixschreibweise

$$\begin{pmatrix} 1 & 0 & -\underline{b}_2 & \cdots & 0 \\ & 1 & -\underline{a}_2 & -\underline{b}_3 & \vdots \\ & & 1 & -\underline{a}_3 & \\ & & & 1 & \ddots & -\underline{b}_N \\ \vdots & & & & \ddots & -\underline{a}_N \\ 0 & \cdots & & & & 1 \end{pmatrix} \cdot \begin{pmatrix} u_0 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_N \end{pmatrix}$$

Als resultierenden Algorithmus erhalten wir:

Es gelte $u_{N+1} = u_{N+2} = 0$ sowie $u_0 := b_2 u_2 + \alpha_0$, dann erhalten wir für $k = N, N-1, \dots, 1$:

$$u_k = \underline{a}_{k+1} u_{k+1} + \underline{b}_{k+1} u_{k+2} + \alpha_k$$

und $S_N := u_0 p_0 + u_1 p_1$.

Bemerkung 2.8.1 Wir wollen nun die beiden vorgestellten Algorithmen hinsichtlich ihrer Operationen betrachten:

- Vorwärtsrekursion: $k = 2, \dots, N$ jeweils 5 Operationen und 3 zu Beginn ergibt $5(N-1) + 3 = 5N - 2$ Operationen
- Rückwärtsrekursion: $k = N, \dots, 1$ jeweils 4 Operationen und 2 Operationen für u_0 bzw. 3 Operationen von S_N im letzten Schritt ergeben insgesamt $4N + 5$

Demzufolge spart man bei Verwendung der Rückwärtsrekursion $\approx N$ Operationen, d.h. man ist gegenüber der Vorwärtsrekursion um bis 20 Prozent schneller.

3 INTERPOLATION

Häufig sind in der Praxis z.B. durch Marktanalysen, technische Messungen von einer Funktion nur einzelne Punkte bekannt, aber keine analytische Beschreibung der Funktion, um sie an beliebigen Stellen auswerten zu können. Könnte man die diskreten Daten durch eine (eventuell glatte) Kurve verbinden, so wäre es möglich, die unbekannte Funktion an den dazwischenliegenden Stellen zu schätzen. In anderen Fällen will man eine schwierig berechenbare Funktion näherungsweise durch eine einfachere darstellen. Eine Interpolationsfunktion kann diese Anforderung der Einfachheit erfüllen.

Interpolationsaufgabe: Eine gegebene Funktion $f : I \rightarrow \mathbb{R}$ sei geeignet zu approximieren unter der Vorgabe, dass f an diskreten (d.h. endlich vielen) Stützstellen die gegebenen Funktionswerte annehmen soll.

Die Interpolation ist somit eine Art der Approximation. Die Approximationsgüte hängt vom Ansatz ab. Um sie zu schätzen, werden Zusatzinformationen (Co-observations) über die Funktion f benötigt. Diese ergeben sich auch bei Unkenntnis von f häufig in natürlicher Weise: Beschränktheit, Stetigkeit oder Differenzierbarkeit lassen sich häufig voraussetzen.

Bei anderen Approximationsverfahren wie z. B. der Ausgleichsrechnung wird nicht gefordert, dass die Daten exakt wiedergegeben werden; das unterscheidet diese Verfahren von der Interpolation.

Bemerkung 3.0.1 i) Ist man am gesamten Verlauf von f interessiert, so sollte man eine Interpolierende $I f$ konstruieren, die sich „möglichst wenig“ von f unterscheidet.

ii) Diese Interpolierende $I f$ sollte eine leicht berechenbare Funktion sein - hierfür eignen sich **Polynome, trigonometrische Funktionen, Exponentialfunktionen** sowie **rationale Funktionen**.

3.1 KLASSISCHE POLYNOM-INTERPOLATION

Gegeben seien $(n+1)$ diskrete, paarweise verschiedene **Stützstellen** x_0, \dots, x_n und dazugehörige beliebige **Stützwerte** f_0, \dots, f_n .

Gesucht ist nun ein Polynom $P \in \mathbb{P}_n$ vom Grad $\text{grad} P \leq n$, d.h.

$$P(x) = a_n x^n + \dots + a_1 x + a_0 \quad \text{mit } a_\nu \in \mathbb{R} \quad (\nu = 0, \dots, n),$$

welches die Interpolationsbedingungen

$$P(x_i) = f_i \quad (i = 0, \dots, n) \tag{3.1}$$

erfüllt.

Die Frage nach der Existenz eines solchen Polynoms $P(x)$ führt uns zu folgendem wichtigen Resultat:

Satz 3.1.1 (Existenz und Eindeutigkeit der Polynominterpolation) Zu beliebigen $(n+1)$ Stützstellen (x_i, f_i) ($i = 0, \dots, n$) mit paarweise verschiedenen Stützstellen x_0, \dots, x_n existiert genau ein Interpolationspolynom $P \in \mathbb{P}_n$, das (3.1) erfüllt und höchstens den Grad n besitzt.

Beweis. (Existenz:) Die Existenz des Interpolationspolynoms $P(x)$ zeigen wir auf konstruktive Art. Zu diesem Zweck betrachten wir zu den gegebenen Stützstellen die $(n + 1)$ **Lagrange-Polynome**

$$L_i(x) := \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)}. \quad (3.2)$$

Diese Polynome sind vom echten Grad n und besitzen offensichtlich die Eigenschaft

$$L_i(x_k) = \delta_{ik} = \begin{cases} 1 & \text{für } i = k, \\ 0 & \text{für } i \neq k. \end{cases} \quad (3.3)$$

Demzufolge besitzt das Polynom

$$P(x) := \sum_{i=0}^n f_i L_i(x)$$

die geforderten Interpolationseigenschaften. Wegen (3.3) gilt:

$$P(x_k) = \sum_{i=0}^n f_i L_i(x_k) = \sum_{i=0}^n f_i \delta_{ik} = f_k \quad (k = 0, 1, \dots, n).$$

Ferner ist als Linearkombination von Polynomen vom Grad n der Grad von P kleiner oder gleich n .

(Eindeutigkeit:) Die Eindeutigkeit des Interpolationspolynoms ergibt sich wie folgt: Es seien $P(x)$ und $Q(x)$ zwei Polynome jeweils vom Grad höchstens gleich n mit

$$P(x_k) = Q(x_k) = f_k \quad (k = 0, 1, \dots, n). \quad (3.4)$$

Aus dieser Eigenschaft (3.4) folgt, dass $D(x) := P(x) - Q(x)$ ein Polynom vom Grad kleiner oder gleich n ist mit den $(n + 1)$ paarweise verschiedenen Nullstellen x_0, \dots, x_n . Nach dem Fundamentalsatz der Algebra muss nun aber $D(x) \equiv 0$ gelten, also $P(x) = Q(x)$ sein. \square

Bemerkung 3.1.2 Die **Lagrange-Darstellung** (3.2) ist für praktische Zwecke meist zu rechenaufwendig, sie eignet sich jedoch hervorragend für theoretische Fragestellungen.

Eine alternative Basis zur Darstellung des Interpolationspolynoms mit Lagrange-Polynomen bildet die sogenannte **monomiale Basis** $\{1, \dots, x^n\}$ von \mathbb{P} , d.h. wir schreiben P in folgender Koeffizientendarstellung

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

Bemerkung 3.1.3 (Vandermonde-Matrix) Die Interpolationsbedingungen $P(x_i) = f_i$ lassen sich nun als folgendes lineares Gleichungssystem auffassen:

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix}.$$

In Numerik I haben wir diese **Vandermonde-Matrix** schon kennengelernt und an dortiger Stelle auch eine zum **Gauß-Algorithmus** alternative Möglichkeit angegeben um das dazugehörige lineare Gleichungssystem zu lösen (Aufwand für dieses spezielle Verfahren $5n^2/2 + \mathcal{O}(n)$).

Bemerkung 3.1.4 Die Darstellung in monomialer Basis ist numerisch instabil und sollte daher für große n im Allgemeinen nicht verwendet werden.

3.2 Hermite-INTERPOLATION UND DIVIDIERTE DIFFERENZEN

Definition 3.2.1 Das nach Satz 3.1.1 eindeutig bestimmte Polynom P heißt **Interpolationspolynom** von f zu den paarweise verschiedenen Stützstellen x_0, \dots, x_n und wird mit

$$P = P(f|x_0, \dots, x_n)$$

bezeichnet.

Bemerkung 3.2.2 Ist man nur an der Auswertung des Interpolationspolynoms P an einer Stelle x interessiert, so muss man dazu nicht erst P bestimmen, sondern kann $P(x)$ durch rekursive Berechnung effektiver (d.h. vor allem effektiver bezüglich des Aufwands) bestimmen. Motiviert wird dies im Folgenden durch das Lemma von Aitken.

Lemma 3.2.3 (von Aitken) Für das Interpolationspolynom $P = P(f|x_0, \dots, x_n)$ gilt die Rekursionsformel

$$P(f|x_0, \dots, x_n)(x) = \frac{(x_0 - x)P(f|x_1, \dots, x_n)(x) - (x_n - x)P(f|x_0, \dots, x_{n-1})(x)}{x_0 - x_n}. \quad (3.5)$$

Hierbei gilt also insbesondere $P(f|x_k) = f(x_k)$.

Beweis. Sei $\phi(x)$ definiert als der Term auf der rechten Seite von (3.5). Dann ist $\phi \in \mathbb{P}_n$ und es gilt:

$$\phi(x_i) = \frac{(x_0 - x_i)f(x_i) - (x_n - x_i)f(x_i)}{x_0 - x_n} = f(x_i), \quad (i = 1, \dots, n-1).$$

Ebenso leicht folgt $\phi(x_0) = f(x_0)$ sowie $\phi(x_n) = f(x_n)$ und daher obige Behauptung. \square

Herleitung des Algorithmus von Aitken und Neville¹

Wir definieren $f_i := f(x_i)$ für $i = 0, \dots, n$. Man beachte hierbei

$$P(f|x_i) = f_i, \quad (i = 0, \dots, n).$$

Für festes x vereinfachen wir die Notation weiterhin durch

$$P_{ik} := P(f|x_{i-k}, \dots, x_i)(x), \quad (i \geq k).$$

Die Rekursion (3.5) schreibt sich nun

$$P_{nn} = \frac{(x_0 - x)P_{n,n-1} - (x_n - x)P_{n-1,n-1}}{x_0 - x_n},$$

oder allgemeiner für P_{ik} ($i \geq k$):

$$\begin{aligned} P_{ik} &= \frac{\overbrace{(x_{i-k} - x_i + x_i - x)}^{(x_{i-k} - x)} P_{i,k-1} - (x_i - x)P_{i-1,k-1}}{x_{i-k} - x_i} \\ &= P_{i,k-1} + \frac{x_i - x}{x_{i-k} - x_i} (P_{i,k-1} - P_{i-1,k-1}). \end{aligned}$$

¹Neville, Alexander Craig (1895-1967)

Nach dem Schema von *Neville* lässt sich P_{nn} ausgehend von den Daten (f_0, \dots, f_n) wie folgt berechnen:

$$\begin{array}{ccccccc}
 f_0 & = & P_{00} & & & & \\
 & & & \searrow & & & \\
 f_1 & = & P_{10} & \rightarrow & P_{11} & & \\
 \vdots & & \vdots & & \vdots & & \\
 \vdots & & \vdots & & \vdots & & \\
 & & & & & \searrow & \\
 & & \vdots & & \vdots & \rightarrow & P_{n-2,n-2} \\
 & & & & & \searrow & \\
 f_{n-1} & = & P_{n-1,0} & \rightarrow & \dots & \rightarrow & P_{n-1,n-2} \rightarrow P_{n-1,n-1} \\
 & & & \searrow & & \searrow & \searrow \\
 f_n & = & P_{n,0} & \rightarrow & P_{n,1} & \dots & \rightarrow P_{n,n-2} \rightarrow P_{n,n-1} \rightarrow P_{nn}
 \end{array}$$

Daraus gewinnen wir die folgende Rechenvorschrift:

Algorithmus 3.2.1: Aitken-Neville

$$P(j, 0) = f(j)$$

$$P(j, k) = P(j, k-1) + \frac{x-x(j)}{x(j)-x(j-k)}(P(j, k-1) - P(j-1, k-1)), \quad j \geq k$$

oder als Matlab-Routine (man beachte die Indexverschiebung) um $P(f|x_0, \dots, x_n)(x)$ an einer Stelle x auszuwerten.

MATLAB-Funktion: AitkenNeville.m

```

1 function value = AitkenNeville(x,fx,x0)
2 % evaluate the Interpolation polynomial given
3 % by (x,fx) at the point x0
4 for k = 2:length(fx)
5     for j = length(fx):-1:k
6         fx(j) = fx(j) + (x0-x(j))/(x(j)-x(j-k+1)) * (fx(j)-fx(j-1));
7     end
8 end
9 value = fx(end);

```

MATLAB-Beispiel:

Testen wir das Aitken-Neville-Verfahren anhand zweier Beispiele. Zum einen werten wir das Interpolationspolynom zu $f(x) = x^2$ und 3 Stützstellen an der Stelle 2 aus.

```
>> x = linspace(0,1,5);
>> AitkenNeville(x,x.^2,2)
ans =
     4
```

Zum anderen werten das Interpolationspolynom zu $f(x) = \sin(x)$ und 5 äquidistanten Stützstellen in $[0,1]$ an der Stelle $\pi/3$ aus. Man beachte, dass $\sin(x) = \sqrt{3}/2$ gilt.

```
>> x = linspace(0,1,5);
>> sqrt(3)/2 - AitkenNeville(x,sin(x),pi/3)
ans =
 4.387286117690792e-005
```

Bemerkung 3.2.4 Die rekursive Struktur des Algorithmus lässt sich auch dazu nutzen, das gesamte Polynom $P(f|x_0, \dots, x_n)$ zu bestimmen.

Dies gilt auch für die verallgemeinerte Interpolationsaufgabe, bei der neben den Funktionswerten $f(x_i)$ auch die Ableitungen gegeben sind, der sogenannten **Hermite-Interpolation**. Um diese einzuführen, benötigen wir noch einige Notationen.

Notation Wir definieren die Folge $\Delta := \{x_j\}_{j=0, \dots, n}$ mit

$$a = x_0 \leq x_1 \leq \dots \leq x_n = b,$$

wobei Stützstellen auch mehrfach auftreten können. Sind an einer Stelle x_i einerseits der Funktionswert $f(x_i)$ und andererseits die Ableitungen $f'(x_i), \dots, f^{(k)}(x_i)$ gegeben, so soll x_i in obiger Folge $(k+1)$ -mal auftreten.

Gleiche Knoten nummerieren wir hierbei mit

$$d_i := \max\{j \mid x_i = x_{i-j}\}$$

von links nach rechts durch; z.B.

$$\begin{array}{c|cccccc} x_i & x_0 = x_1 & < x_2 = x_3 = x_4 & < x_5 & < x_6 \\ d_i & 0 & 1 & 0 & 1 & 2 & 0 & 0 \end{array}$$

Führen wir nun mit diesen Abkürzungen nachfolgende lineare Abbildung

$$\mu_i : C^n[a, b] \rightarrow \mathbb{R}, \quad \mu_i(f) := f^{(d_i)}(x_i), \quad (i = 0, \dots, n)$$

ein, so lautet die Aufgabe der **Hermite-Interpolation**: Gegeben μ_i ($i = 0, \dots, n$). Finde $P \in \mathbb{P}_n$ mit

$$\mu_i(P) = \mu_i(f) \quad (i = 0, \dots, n). \quad (3.6)$$

Die Lösung $P = P(f|x_0, \dots, x_n) \in \mathbb{P}_n$ von (3.6) heißt **Hermite-Interpolierende**.

Satz 3.2.5 (Existenz und Eindeutigkeit) Zu jeder Funktion $f \in C^n[a, b]$ und jeder monotonen Folge

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

von (*i.Allg. nicht paarweise verschiedenen*) Knoten gibt es genau ein Polynom $P \in \mathbb{P}_n$, sodass gilt:

$$\mu_i(P) = \mu_i(f), \quad (i = 0, \dots, n).$$

Beweis. Die Abbildung

$$\mu : \mathbb{P}_n \rightarrow \mathbb{R}^{n+1}, P \mapsto (\mu_0(P), \dots, \mu_n(P))$$

ist offensichtlich eine lineare Abbildung zwischen den $(n+1)$ -dimensionalen reellen Vektorräumen \mathbb{P}_n und \mathbb{R}^{n+1} , sodass aus der Injektivität der Abbildung bereits die Surjektivität folgen würde und damit der Satz bewiesen wäre. Somit reicht es die Injektivität der linearen Abbildung zu zeigen.

Da $\mu(P) = 0$ gilt, folgt, dass P mindestens $(n+1)$ -Nullstellen inklusive Vielfachheiten besitzt, somit aber das Nullpolynom ist. Da ferner $\dim \mathbb{P}_n = \dim \mathbb{R}^{n+1} = n+1$, folgt daraus auch wieder die Existenz. \square

Definition 3.2.6 (Newton-Basis) Es seien $x_0, \dots, x_{n-1} \in \mathbb{R}$ und

$$\omega_0 := 1, \quad \omega_i(x) := \prod_{j=0}^{i-1} (x - x_j) \quad (\omega_i \in \mathbb{P}_i)$$

Wir bezeichnen $\omega_0, \dots, \omega_n$ als **Newton-Basis** des Polynomraums \mathbb{P}_n mit Basiselementen ω_i .

Bemerkung 3.2.7 Man beachte, dass bei der Definition der Newton-Basis weder eine Ordnung der Punkte x_k noch „paarweise verschieden“ vorgeschrieben wurde. Je nach Nummerierung, erhält man somit eine andere Newton-Basis. Dies ist bei der Stabilität der folgenden Verfahren zu berücksichtigen.

Um nun das Verfahren der dividierten Differenzen herzuleiten, mit dem sich die Hermite-Interpolationsaufgabe effizient lösen lässt, verwenden wir die Darstellung des Interpolationspolynoms in der Newton-Basis, d.h.

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots + a_n \prod_{j=0}^{n-1} (x - x_j) \\ &= \sum_{i=0}^n a_i \omega_i(x), \end{aligned}$$

wobei sich die unbekanntenen Koeffizienten a_0, \dots, a_n prinzipiell aus den Interpolationsbedingungen

$$\begin{array}{rcl} P(x_0) & = & a_0 & = & f(x_0) \\ P(x_1) & = & a_0 + a_1(x_1 - x_0) & = & f(x_1) \\ P(x_2) & = & a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_1)(x_2 - x_0) & = & f(x_2) \\ \vdots & & \vdots & & \vdots \end{array}$$

sukzessive berechnen lassen (dieses LGS hat Linksdreiecksgestalt!). Die Koeffizienten a_k nennt man **dividierte Differenzen**.

Definition 3.2.8 Der führende Koeffizient a_n des Interpolationspolynoms

$$P(f|x_0, \dots, x_n)(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

von f zu den Knoten $x_0 \leq x_1 \leq \dots \leq x_n$ heißt **n-te dividierte Differenz** von f an x_0, \dots, x_n und wird mit

$$[x_0, \dots, x_n]f := a_n$$

bezeichnet.

Satz 3.2.9 Für jede Funktion $f \in C^n(\mathbb{R})$ und Knoten $x_0 \leq \dots \leq x_n \in \mathbb{R}$ ist

$$P(x) = \sum_{i=0}^n [x_0, \dots, x_i] f \cdot \omega_i(x)$$

das Interpolationspolynom $P(f|x_0, \dots, x_n)$ von f an x_0, \dots, x_n .

Gilt darüber hinaus $f \in C^{n+1}(\mathbb{R})$, so folgt:

$$f(x) = P(x) + [x_0, \dots, x_n, x] f \cdot \omega_{n+1}(x). \quad (3.7)$$

Beweis. Wir zeigen die erste Behauptung durch Induktion nach $n \in \mathbb{N}$. Für $n = 0$ ist die Aussage trivialerweise erfüllt. Sei also im Folgenden $n > 0$ und

$$P_{n-1} := P(f|x_0, \dots, x_{n-1}) = \sum_{i=0}^{n-1} [x_0, \dots, x_i] f \cdot \omega_i$$

das Interpolationspolynom von f an den Stützstellen x_0, \dots, x_{n-1} . Damit erhalten wir für $P_n = P(f|x_0, \dots, x_n)$, dass

$$\begin{aligned} P_n(x) &= [x_0, \dots, x_n] f \cdot x^n + a_{n-1} x^{n-1} + \dots + a_0 \\ &= [x_0, \dots, x_n] f \cdot \omega_n(x) + Q_{n-1}(x) \end{aligned}$$

mit einem Polynom $Q_{n-1} \in \mathbb{P}_{n-1}$ gilt. Nun erfüllt aber

$$Q_{n-1} = P_n - [x_0, \dots, x_n] f \cdot \omega_n$$

offensichtlich die Interpolationsaufgabe für x_0, \dots, x_{n-1} , sodass wir erhalten:

$$Q_{n-1} = P_{n-1} = \sum_{i=0}^{n-1} [x_0, \dots, x_i] f \cdot \omega_i.$$

Dies beweist aber gerade die Aussage des Satzes. Insbesondere folgt nun, dass

$$P_n + [x_0, \dots, x_n, x] f \cdot \omega_{n+1}$$

die Funktion f an den Knoten x_0, \dots, x_n und x interpoliert und damit (3.7). \square

Aus den Eigenschaften der *Hermite*-Interpolation lassen sich sofort folgende Aussagen über die dividierten Differenzen zu f ableiten:

Lemma 3.2.10 i) Für $x_i \neq x_k$ gilt die Rekursionsformel

$$[x_0, \dots, x_n] f = \frac{[x_0, \dots, \hat{x}_i, \dots, x_n] f - [x_0, \dots, \hat{x}_k, \dots, x_n] f}{x_k - x_i},$$

wobei $\hat{}$ anzeigt, dass die entsprechende Stützstelle weggelassen wird ('seinen Hut nehmen muss')

ii) Für zusammenfallende Knoten $x_0 = \dots = x_n$ gilt

$$[x_0, \dots, x_n] f = f^{(n)}(x_0)/n!$$

Beweis. Für das *Hermite*-Interpolationspolynom gilt mit $x_i \neq x_k$:

$$P(f|x_0, \dots, x_n) = \frac{\overbrace{(x_0, \dots, \widehat{x}_i, \dots, x_n) f}^{[x_0, \dots, \widehat{x}_i, \dots, x_n] f x^{n-1} + \mathbb{P}_{n-2}} - (x_k - x)P(f|x_0, \dots, \widehat{x}_k, \dots, x_n)}{x_i - x_k}, \quad (3.8)$$

was sich durch Überprüfen der Interpolationseigenschaft zeigen lässt mittels Einsetzen der Definitionen. Aus der Eindeutigkeit des führenden Koeffizienten folgt aus (3.8) unmittelbar Behauptung *i*). Stimmen dagegen alle Knoten x_0, \dots, x_n überein, so ist das Interpolationspolynom

$$P(f|x_0, \dots, x_n)(x) = \sum_{j=0}^n \frac{(x - x_0)^j}{j!} f^{(j)}(x_0),$$

wie man durch Einsetzen in μ_i (vgl. oben) leicht einsieht. Sei nun $0 \leq k \leq n$, dann folgt:

$$\mu_k(P) = f^{(k)}(x_0) + \frac{k!(x - x_0)^1}{(k+1)!} f^{(k+1)}(x_0) + \dots + \frac{k!(x - x_0)^{n-k}}{n!} f^{(n)}(x_0).$$

Somit gilt auch *ii*). □

Satz 3.2.11 *Es sei $f \in C^n[a, b]$, $f^{(n+1)}(x)$ existiere für alle $x \in (a, b)$; weiterhin gelte $a \leq x_0 \leq x_1 \leq \dots \leq x_n \leq b$. Dann gilt:*

$$f(x) - P_n(f|x_0, \dots, x_n)(x) = \frac{(x - x_0) \cdot \dots \cdot (x - x_n)}{(n+1)!} f^{(n+1)}(\xi), \quad (3.9)$$

wobei $\min\{x, x_0, \dots, x_n\} < \xi < \max\{x, x_0, \dots, x_n\}$ ist.

Beweis. Nach Konstruktion von $P_n(f|x_0, \dots, x_n)$ gilt: $P_n(f|x_0, \dots, x_n) = f(x_k)$ ($k = 0, 1, \dots, n$). Es sei nun x fest und dabei ungleich x_0, x_1, \dots, x_n . Ferner sei

$$K(x) := \frac{f(x) - P_n(f|x_0, \dots, x_n)(x)}{(x - x_0) \cdot \dots \cdot (x - x_n)}. \quad (3.10)$$

Nun betrachten wir die Funktion

$$W(t) := f(t) - P_n(f|x_0, \dots, x_n)(t) - (t - x_0) \cdot \dots \cdot (t - x_n)K(x). \quad (3.11)$$

Die Funktion $W(t)$ verschwindet also an den Stellen $t = x_0, \dots, t = x_n$ und durch (3.10) auch an der Stelle $t = x$. Nach dem verallgemeinerten Satz von *Rolle*² verschwindet die Funktion $W^{(n+1)}(t)$ an einer Stelle ξ mit $\min\{x, x_0, \dots, x_n\} < \xi < \max\{x, x_0, \dots, x_n\}$. Das $(n+1)$ -fache Differenzieren von (3.11) liefert

$$W^{(n+1)}(t) = f^{(n+1)}(t) - (n+1)!K(x),$$

sodass gilt:

$$0 = W^{(n+1)}(\xi) = f^{(n+1)}(\xi) - (n+1)!K(x).$$

Damit erhalten wir aber unmittelbar

$$K(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi). \quad (3.12)$$

Nach Einsetzen von (3.12) in (3.11) erhalten wir die Behauptung für $t = x$, da x Nullstelle von W ist. □

²Rolle, Michel (1652-1719)

Bemerkung 3.2.12 Im Beweis zum Approximationsfehler (vgl. Satz 4, (3.9)) haben wir gezeigt

$$f(x) - P(f|x_0, \dots, x_n)(x) = \frac{\omega_{n+1}(x)}{n+1} f^{(n+1)}(\xi), \quad \min\{x_0, \dots, x_n, x\} < \xi < \max\{x_0, \dots, x_n, x\}.$$

Und mit dem Satz über die Newton-Darstellung gilt:

$$f(x) - P(f|x_0, \dots, x_n)(x) = [x_0, \dots, x_n, x]f \cdot \omega_{n+1}(x).$$

Somit folgern wir:

Für alle Knoten $x_0 \leq \dots \leq x_n$ existiert ein $\xi \in [x_0, x_n]$, sodass gilt:

$$[x_0, \dots, x_n]f = \frac{f^{(n)}(\xi)}{n!} \tag{3.13}$$

Die Auswertung der Rekursionsformel (3.8) erfolgt am zweckmäßigsten im Schema der dividierten Differenzen unter Verwendung der Startwerte $[x_i]f = f(x_i)$ für paarweise verschiedene Knoten.

x_0	$[x_0]f$				
x_1	$[x_1]f$	$[x_0, x_1]f$			
x_2	$[x_2]f$	$[x_1, x_2]f$	$[x_0, x_1, x_2]f$		
x_3	$[x_3]f$	$[x_2, x_3]f$	$[x_1, x_2, x_3]f$	$[x_0, x_1, x_2, x_3]f$	
x_4	$[x_4]f$	$[x_3, x_4]f$	$[x_2, x_3, x_4]f$	$[x_1, x_2, x_3, x_4]f$	$[x_0, \dots, x_4]f$

Die gesuchten Koeffizienten a_k des Newton-Interpolationspolynoms findet man im obigen Schema der dividierten Differenzen in der oberen Diagonalen.

Beispiel 3.2.13 Wenn wir das Schema auf gegebene Daten (x_i, f_i) ($i = 0, \dots, 3$) anwenden, so erhalten wir

$$\begin{array}{ccccccc}
 x_0 = 0 & : & f_0 = 1 & & & & \\
 & & & \searrow & & & \\
 x_1 = 3/2 & : & f_1 = 2 & \rightarrow & 2/3 & & \\
 & & & \searrow & & \searrow & \\
 x_2 = 5/2 & : & f_2 = 2 & \rightarrow & 0 & \rightarrow & -4/15 \\
 & & & \searrow & & \searrow & \searrow \\
 x_3 = 9/2 & : & f_3 = 1 & \rightarrow & -1/2 & \rightarrow & -1/6 & \rightarrow & 1/45
 \end{array}$$

und das Newtonsche Interpolationspolynom lautet demnach

$$P(x) = 1 + \frac{2}{3}(x - 0) - \frac{4}{15}(x - 0)(x - \frac{3}{2}) + \frac{1}{45}(x - 0)(x - \frac{3}{2})(x - \frac{5}{2})$$

MATLAB-Funktionen: NewtonInterpolation.m und EvalNewtonPoly.m

```

1 function fx = NewtonInterpolation(x,fx)
2 for k = 2:length(fx)
3     for j = length(fx):-1:k
4         fx(j) = (fx(j) - fx(j-1))/(x(j)-x(j-k+1));
5     end
6 end

1 function value = HornerNewton(a,x0,x)
2 % evaluate Newton polynom
3 % p(x0) = a(1) + a(2)*(x0-x(1)) + a(3)*(x0-x(1))*(x0-x(2)) + ...
4 %       = a(1) + (x0-x(1)) * ( a(2) + (x0-x(2)) * ( a(3) ....
5 value = a(end);
6 for k=length(a)-1:-1:1
7     value = a(k) + value.*(x0-x(k));
8 end

```

MATLAB-Beispiel:

Testen wir das Differenzen-Verfahren nochmals an den beiden Beispielen aus Bsp. 3.2. Zum einen werten wir das Interpolationspolynom zu $f(x) = x^2$ und 3 Stützstellen an der Stelle 2 aus.

```

>> x=linspace(0,2,3);
>> a = NewtonInterpolation(x,x.^2)
a =
     0     1     1
>> HornerNewton(a,2,x)
ans =
     4

```

Zum anderen werten das Interpolationspolynom zu $f(x) = \sin(x)$ und 5 äquidistanten Stützstellen in $[0, 1]$ an der Stelle $\pi/3$ aus.

```

>> x=linspace(0,1,5);
>> a = NewtonInterpolation(x,sin(x));
>> sqrt(3)/2-HornerNewton(a,pi/3,x)
ans =
 4.387286117690792e-005

```

Beispiel 3.2.14 Betrachten wir nun noch abschließend das Schema für die nichttriviale *Hermite*-Interpolationsaufgabe (d.h. auch Ableitungen werden interpoliert). Unter Beachtung von Lemma 3.2.10, insbesondere (3.8) ergibt sich:

$$\begin{array}{l}
 x_0 : \quad [x_0]f \\
 \quad \quad \searrow \\
 x_0 : \quad [x_0]f \rightarrow [x_0, x_0]f = f'(x_0) \\
 \quad \quad \searrow \\
 x_0 : \quad [x_0]f \rightarrow [x_0, x_0]f = f'(x_0) \rightarrow [x_0, x_0, x_0]f = \frac{f''}{2}(x_0) \\
 \quad \quad \searrow \quad \quad \quad \searrow \quad \quad \quad \searrow \\
 x_1 : \quad [x_1]f \rightarrow [x_0, x_1]f \rightarrow [x_0, x_0, x_1]f \rightarrow [x_0, x_0, x_0, x_1]f
 \end{array}$$

Das resultierende Polynom $P(x)$ mit

$$P(x) = f(x_0) + (x - x_0) \left(f'(x_0) + (x - x_0) \left(f''(x_0) + (x - x_0) [x_0, x_0, x_0, x_1]f \right) \right)$$

erfüllt dann die Interpolationsaufgaben

$$P(x_0) = f(x_0), P'(x_0) = f'(x_0), P''(x_0) = f''(x_0) \text{ und } P(x_1) = f(x_1).$$

Für den speziellen Fall, dass an allen Knoten x_0, \dots, x_n sowohl f als auch f' interpoliert werden sollen, erhält man die folgende Darstellung des Interpolationspolynoms $P(x)$

Satz 3.2.15 *Es sei $\omega(x) = (x - x_0) \cdot \dots \cdot (x - x_n)$ und $L_k(x)$ seien die Lagrange-Polynome zu den Knoten x_0, \dots, x_n . Dann hat*

$$P(x) = \sum_{k=1}^n f(x_k) \left(1 - \frac{\omega''(x_k)}{\omega'(x_k)} (x - x_k) \right) L_k^2(x) + \sum_{k=1}^n f'(x_k) (x - x_k) L_k^2(x)$$

die Interpolationseigenschaften

$$P(x_k) = f(x_k) \text{ und } P'(x_k) = f'(x_k) \quad (k = 0, \dots, n).$$

Beweis. Es sei x_ℓ einer der Knoten x_0, \dots, x_n , dann folgt sofort aus der Interpolationseigenschaft der Lagrange-Polynome

$$P(x_\ell) = f(x_\ell),$$

da

$$\omega'(x) = \sum_{i=0}^n \prod_{\substack{k=0 \\ k \neq i}}^n (x - x_k) \text{ und somit } \omega'(x_\ell) = \prod_{\substack{k=0 \\ k \neq \ell}}^n (x_\ell - x_k) \neq 0$$

gilt. Für die Ableitung von P ergibt sich

$$\begin{aligned} P'(x) &= \sum_{k=1}^n f(x_k) \left[\left(1 - \frac{\omega''(x_k)}{\omega'(x_k)} (x - x_k) \right) 2L_k'(x) - \frac{\omega''(x_k)}{\omega'(x_k)} L_k(x) \right] L_k(x) \\ &\quad + \sum_{k=1}^n f'(x_k) \left[(x - x_k) 2L_k'(x) + L_k(x) \right] L_k(x), \end{aligned}$$

sodass wir nun Folgendes erhalten:

$$P'(x_\ell) = f(x_\ell) \left(2L_\ell'(x_\ell) - \frac{\omega''(x_\ell)}{\omega'(x_\ell)} \right) + f'(x_\ell).$$

Nutzt man aus, dass $L_\ell(x) = \frac{\omega(x)}{(x - x_\ell)\omega'(x_\ell)}$ gilt (siehe Hausübung), so folgt aus

$$\omega(x) = L_\ell(x)(x - x_\ell)\omega'(x_\ell)$$

nach zweimaligem Differenzieren

$$\omega''(x) = L_\ell''(x)(x - x_\ell)\omega'(x_\ell) + 2L_\ell'(x)\omega'(x_\ell).$$

Damit gilt an der Stelle x_ℓ

$$\frac{\omega''(x_\ell)}{\omega'(x_\ell)} = 2L_\ell'(x_\ell)$$

Einsetzen in die Ableitung von P schließt den Beweis ab. □

3.3 *Tschebyscheff*-INTERPOLATION

Wie das folgende Beispiel zeigen wird, hat die Verteilung der Stützstellen x_0, \dots, x_n über das Interpolationsintervall entscheidenden Einfluss auf die Güte der Approximation. Ein klassisches Beispiel hierfür stammt von *Runge*³:

Die Interpolationspolynome $P(f|x_1, \dots, x_n)$ zur Funktion $f(x) = \frac{1}{1+x^2}$ im Intervall $I := [-5, 5]$ bei äquidistanten Stützstellen $x_k = -5 + \frac{10}{n}k$ zeigen bei wachsendem n einen zunehmenden Interpolationsfehler.

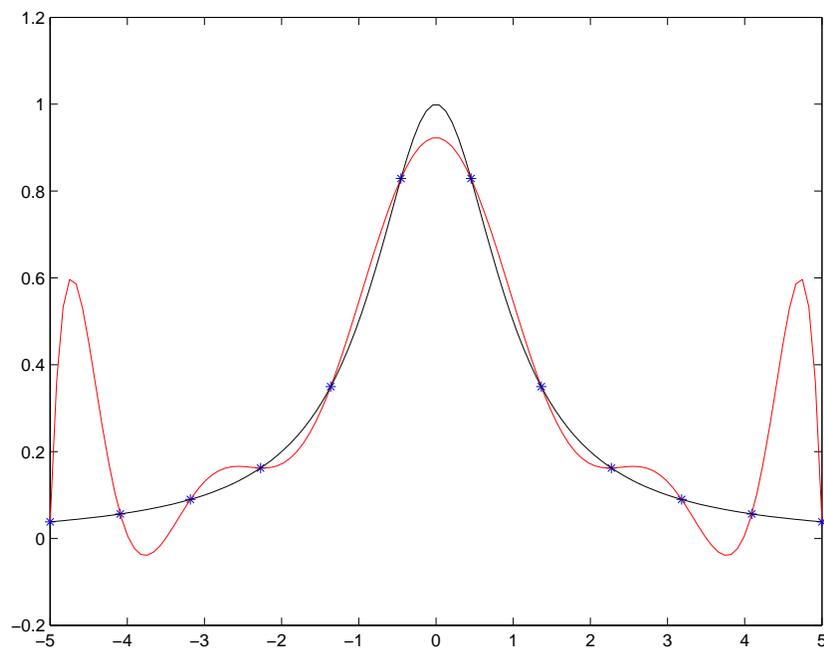
MATLAB-Beispiel:

Das Interpolationspolynom zu 12 äquidistanten Stützstellen und Stützwerten zur Funktion

$$f(x) = \frac{1}{1+x^2}$$

ist in Abb. 3.3 dargestellt.

```
n = 12;
f = @(x) 1./(x.^2+1);
x = linspace(-5,5,n);
fx = f(x);
s = linspace(x(1),x(end),10*n);
for j=1:length(s)
    ps(j) = AitkenNeville(x,fx,s(j));
end
plot(x,fx,'*','s,ps','r-',s,f(s),'k')
```



Wie wir im weiteren zeigen werden, kann man bei geschickter, nichtäquidistanter Wahl der Stützstellen dagegen eine Konvergenz erhalten. Genauer gesagt, wählen wir x_1, \dots, x_n als die Nullstellen der von $[-1, 1]$ auf I transformierten *Tschebyscheff*-Polynome und erhalten dadurch punktweise Konvergenz für $n \rightarrow \infty$. Man beachte das dieses Phänomen nicht von Rundungsfehlern abhängt.

³Runge, Carl (1856-1927)

Bei der Berechnung des Approximations- bzw. des Interpolationsfehlers haben wir gesehen, dass

$$f(x) - P(f|x_0, \dots, x_n)(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x) \quad (x \in [a, b])$$

für ein $\xi = \xi(x) \in (a, b)$ gilt. Wir suchen nun Knoten $x_0, \dots, x_n \in [a, b]$, die das *Minimax*-Problem

$$\max_{x \in [a, b]} |\omega_{n+1}(x)| = \max_{x \in [a, b]} |(x - x_0) \cdot \dots \cdot (x - x_n)| = \min$$

lösen. Anders formuliert, es gilt das normierte Polynom $\omega_{n+1} \in \mathbb{P}_{n+1}$ mit den reellen Nullstellen x_0, \dots, x_n zu bestimmen, für das

$$\max_{x \in [a, b]} |\omega_{n+1}(x)| = \min$$

gilt. Im Folgenden werden wir sehen, dass gerade die *Tschebyscheff*-Polynome T_n diese obige *Minimax*-Aufgabe lösen (sie lösen sie bis auf einen skalaren Faktor und eine affine Transformation). Somit sind die Nullstellen der *Tschebyscheff*-Polynome (bis auf eine affine Transformation) gerade die gesuchten Stützstellen x_0, \dots, x_n .

Zunächst reduzieren wir das Problem auf das Intervall $[-1, 1]$ mit Hilfe der Umkehrabbildung folgender Abbildung

$$x : [a, b] \rightarrow [-1, 1], \quad t \mapsto x = x(t) = \frac{2t - a - b}{b - a},$$

d.h. die Umkehrabbildung lautet:

$$t : [-1, 1] \rightarrow [a, b], \quad x \mapsto t = t(x) = \frac{b + a}{2} + \frac{b - a}{2}x.$$

Ist jetzt $P \in \mathbb{P}_n$ mit $\text{grad}P = n$ und führendem Koeffizienten 1 die Lösung des *Minimax*-problems

$$\max_{x \in [-1, 1]} |P(x)| = \min,$$

so stellt $\widehat{P}(t) := P(x(t))$ die Lösung des ursprünglichen Problems mit führendem Koeffizienten $2^n / (b - a)^n$ dar. Für $x \in [-1, 1]$ definieren wir die *Tschebyscheff*-Polynome durch (vgl. hierzu auch Kapitel 6.5):

$$T_n(x) = \cos(n \arccos(x)), \quad x \in [-1, 1] \quad (3.14)$$

und allgemein für $x \in \mathbb{R}$ durch die Drei-Term-Rekursion

$$T_0(x) = 1, T_1(x) = x, T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), \quad k \geq 2 \quad (3.15)$$

Wir benötigen im Folgenden die schon in Kapitel 2 diskutierten Eigenschaften der *Tschebyscheff* Polynome, die wir der Einfachheit halber hier nochmals wiedergeben:

Bemerkung 3.3.1 (i) Der führende Koeffizient von T_n ist $a_n = 2^{n-1}$ ($n \geq 1$)

(ii) $|T_n(x)| \leq 1$ für $x \in [-1, 1]$

(iii) Die Nullstellen von $T_n(x)$ sind

$$x_k := \cos\left(\frac{2k-1}{2n}\pi\right), \quad (k = 1, \dots, n)$$

(iv) $|T_n(x)|$ nimmt seinen maximalen Wert im Intervall $[-1, 1]$ an den Stellen $\bar{x}_k = \cos(\frac{k\pi}{n})$ für $k = 0, \dots, n$ an, d.h.

$$|T_n(x)| = 1 \Leftrightarrow x = \bar{x}_k = \cos(\frac{k\pi}{n}) \text{ mit } k = 0, \dots, n.$$

Satz 3.3.2 Jedes Polynom $P \in \mathbb{P}_n$ mit führendem Koeffizienten $a_n \neq 0$ nimmt im Intervall $[-1, 1]$ einen Wert vom Betrag $\geq |a_n|/2^{n-1}$ an. Insbesondere sind die Tschebyscheff-Polynome $T_n(x)$ minimal bezüglich der Maximumsnorm $\|f\|_\infty = \max_{x \in [-1, 1]} |f(x)|$ unter den Polynomen vom Grad n mit führendem Koeffizienten 2^{n-1} .

Beweis. (Annahme:) Sei $P \in \mathbb{P}_n$ ein Polynom mit führendem Koeffizienten $a_n = 2^{n-1}$ und $|P(x)| < 1$ für $x \in [-1, 1]$. Dann ist $T_n - P_n$ ein Polynom vom Grad kleiner oder gleich $(n-1)$ (beide besitzen a_n als führenden Koeffizienten). An den Tschebyscheff-Abszissen $\bar{x}_k := \cos(\frac{k\pi}{n})$ gilt:

$$T_n(\bar{x}_{2k}) = 1, \quad P_n(\bar{x}_{2k}) < 1 \Rightarrow P_n(\bar{x}_{2k}) - T_n(\bar{x}_{2k}) < 0$$

$$T_n(\bar{x}_{2k+1}) = -1, \quad P_n(\bar{x}_{2k+1}) > -1 \Rightarrow P_n(\bar{x}_{2k+1}) - T_n(\bar{x}_{2k+1}) > 0,$$

d.h. die Differenz $T_n - P_n$ ist an den $(n+1)$ -Tschebyscheff-Abszissen abwechselnd positiv und negativ, damit besitzt die Differenz mindestens n Nullstellen in $[-1, 1]$ im Widerspruch zu $0 \neq T_n - P_n \in \mathbb{P}_{n+1}$. Demnach muss es für jedes Polynom $P \in \mathbb{P}_n$ mit führendem Koeffizienten $a_n = 2^{n-1}$ ein $x \in [-1, 1]$ geben derart, dass $|P_n(x)| \geq 1$ erfüllt. Für ein beliebiges $P \in \mathbb{P}_n$ mit $a_n \neq 0$ folgt die Behauptung daraus, dass $\tilde{P}_n := \frac{2^{n-1}}{a_n} P_n$ ein Polynom mit $\tilde{a}_n = 2^{n-1}$ ist. \square

3.4 RATIONALE INTERPOLATION

Zur Interpolation einer Funktion, welche einen Pol besitzt oder deren Graph eine Asymptote aufweist sind im Allgemeinen Polynome nicht gut geeignet.

Zu diesem Zweck untersuchen wir im Folgenden die Interpolation mittels einer gebrochen rationalen Funktion

$$R(x) = \frac{p_0 + p_1x + \dots + p_\nu x^\nu}{q_0 + q_1x + \dots + q_\mu x^\mu} = \frac{P_\nu(x)}{Q_\mu(x)}. \quad (3.16)$$

Die Polynomgrade ν, μ seien hierbei vorgegeben, sodass $R(x)$ an $(n+1)$ paarweise verschiedenen Knoten x_0, \dots, x_n die vorgegebenen Funktionswerte $f_0 := f(x_0), \dots, f_n := f(x_n)$ annimmt, somit also gilt:

$$R(x_i) = f_i, \quad (i = 0, \dots, n).$$

Da Zähler und Nenner in (3.16) jeweils mit einer von Null verschiedenen Zahl multipliziert werden dürfen, enthält der Ansatz (3.16) $\nu + \mu + 1$ freie Unbekannte. Damit nun die Anzahl der Interpolationsbedingungen mit der Unbekanntenanzahl übereinstimmt, muss zwangsläufig $\nu + \mu = n$ erfüllt sein.

Beispiel 3.4.1 $x_0 = 1, x_1 = 1, x_2 = 2, y_0 = 2, y_1 = 3, y_2 = 3$

Wir setzen $\nu = 0, \mu = 2$, d.h. wir verwenden den Ansatz

$$R(x) = \frac{p_0}{q_0 + q_1x + q_2x^2}.$$

Die Interpolationsaufgabe führt uns zum Gleichungssystem

$$P_\nu(x_i) - Q_\mu(x_i) \cdot f(x_i) = 0,$$

d.h. im vorliegenden Fall

$$\begin{aligned} p_0 - 2(q_0 - q_1 + q_2) &= 0 \\ p_0 - 2(q_0 + q_1 + q_2) &= 0 \\ p_0 - 3(q_0 + 2q_1 + 4q_2) &= 0 \end{aligned}$$

Nach der *Cramerschen* Regel besitzt dieses lineare Gleichungssystem die einzige Lösung

$$R(x) = \frac{36}{14 - 3x + x^2}.$$

Beispiel 3.4.2 Wählen wir zu den Daten von Beispiel 1 den Ansatz $\mu = \nu = 1$, so setzen wir

$$R(x) = \frac{p_0 + p_1x}{q_0 + q_1x},$$

bzw.

$$p_0 + p_1x_i - f_i(q_0 + q_1x_i) = 0, \quad i = 0, 1, 2$$

Dies führt uns zu folgendem linearen Gleichungssystem:

$$\begin{aligned} p_0 - p_1 - 2(q_0 - q_1) &= 0 \\ p_0 + p_1 - 3(q_0 + q_1) &= 0 \\ p_0 + 2p_1 - 3(q_0 + 2q_1) &= 0, \end{aligned}$$

welches die (bis auf einen gemeinsamen Faktor) eindeutige Lösung

$$p_0 = 3, \quad p_1 = 3, \quad q_0 = 1, \quad q_1 = 1$$

besitzt, sodass wir folgendes Ergebnis erhalten:

$$R(x) = \frac{3 + 3x}{1 + x} = 3 \quad [\text{Widerspruch (vgl. } R(x_0) = 2)]$$

Bemerkung 3.4.3 Die nichttriviale Lösung des LGS zur Bestimmung von $R(x)$ braucht nicht in jedem Fall die Interpolationsaufgabe zu erfüllen. Da sowohl Zähler als auch Nenner einen gemeinsamen Linearfaktor $(x - x_j)$ besitzen können, wird sich dieser bei der Lösung wegekürzen. Die daraus resultierende gebrochen rationale Funktion $\tilde{R}(x)$ wird dann im Allgemeinen den Funktionswert f_j an der Stelle x_j nicht annehmen (vgl. hierzu Beispiel 3.4.2).

Bemerkung 3.4.4 Auch für die rationale Interpolation existieren Neville-artige Algorithmen zur Auswertung des Polynoms, ohne dies vorher bestimmt zu haben bzw. Algorithmen zur Bestimmung der Koeffizienten, ähnlich dem dividierten Differenzen Verfahren. Man schlage z.B. in [Stör/Bulirsch] unter den Stichworten **Thielscher Kettenbruch**, **reziproke Differenzen** nach.

MATLAB-Funktion: RationalAitkenNeville.m

```

1 function value = RationalAitkenNeville(x,fx,x0)
2 % evaluate the rational interpolation polynomial (n,n+1) resp. (n,n
  )
3 % given by (x,fx) at the point x0
4 fxml = fx;
5 for j = length(fx):-1:2
6     if x0~=x(j)
7         fx(j) = fx(j) + (x0-x(j))/(x(j)-x(j-1)) * (fx(j)-fx(j-1));
8     else
9         value = fx(j); return
10    end
11 end
12
13 for k = 3:length(fx)
14     temp = fx;
15     for j = length(fx):-1:k
16         d = fx(j)-fx(j-1);
17 %         if fx(j)==fxml(j-1) % abs(fx(j)-fxml(j-1))<1e-10*max(1,max(abs
  (fx(j)),abs(fxml(j-1))))
18 %             fx(j) = NaN;
19 %         else
20 %             %((x0-x(j-k+1))/(x0-x(j))*(1-d/(fx(j)-fxml(j-1)))-1)
21             fx(j) = fx(j)+d/((x0-x(j-k+1))/(x0-x(j))*(1-d/(fx(j)-fxml(j
  -1)))-1));
22 %         end
23     end
24     fxml = temp;
25 end
26 value = fx(end);

```

MATLAB-Beispiel:

Vergleichen wir die rationale Interpolation (Zähler- und Nennerpolynom quadratisch) mit der einfachen Polynominterpolation 4. Ordnung. Die Werte für $\cot(1^\circ)$, $\cot(2^\circ)$, ..., $\cot(5^\circ)$ liegen vor und der Wert für $\cot(2^\circ, 30')$ soll durch Interpolation angenähert werden. Man beachte die unterschiedliche Genauigkeit.

```

>> format long
>> f = @(x) cot(x*pi/180);
>> x = linspace(1,5,5);
>> AitkenNeville(x,f(x),2.5)
ans =
    22.74709740132512
>> RationalAitkenNeville(x,f(x),2.5)
ans =
    22.90376554340344
>> f(2.5)
ans =
    22.90376554843120

```

Das die deutlich bessere Approximation durch das rationale Interpolationspolynom kein einzelner Zufallsbefund ist wird durch die folgende Grafik bestätigt, die den relativen Fehler zwischen

Interpolationspolynom und exaktem Wert grafisch wiedergibt.

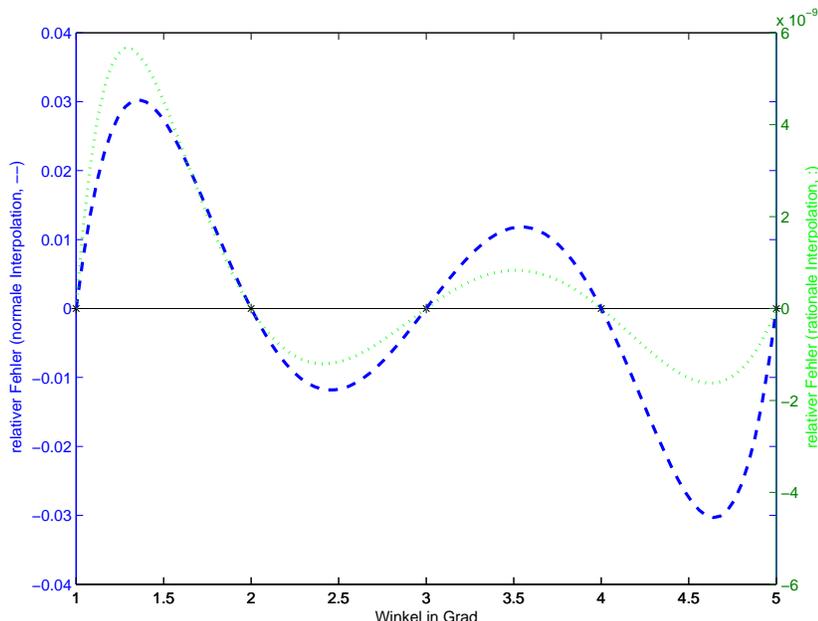


Abb. 3.1: Die Werte für $\cot(1^\circ)$, $\cot(2^\circ)$, \dots , $\cot(5^\circ)$ liegen vor. Dargestellt ist der relative Fehler zwischen Interpolationspolynom (zu den gegebenen Daten) und der exakten Funktion für das einfache Interpolationspolynom (gestichelte Linie, y-Achse links) mit einem Maximalwert ≈ 0.03 und für ein rationales Interpolationspolynom (gepunktete Linie, y-Achse rechts) mit einem maximalen Absolutwert $\approx 5.8 \cdot 10^{-9}$.

3.5 Padé-APPROXIMATION

Als *Padé-Approximation* wird eine rationale Funktion bezeichnet, deren Potenzreihenentwicklung mit einer gegebenen Potenzreihe bis zum höchsten Grad übereinstimmt. Falls die rationale Funktion die folgende Form

$$R(x) = \frac{\sum_{k=0}^m a_k x^k}{1 + \sum_{k=0}^n b_k x^k}$$

hat, dann wird $R(x)$ als *Padé-Approximation* zur Potenzreihe

$$f(x) = \sum_{k=0}^{\infty} c_k x^k$$

bezeichnet, falls

$$R(0) = f(0) \quad \text{und ebenso} \quad (3.17)$$

$$\left. \frac{d}{dx^k} R(x) \right|_{x=0} = \left. \frac{d}{dx^k} f(x) \right|_{x=0}, \quad k = 1, 2, \dots, m+n \quad (3.18)$$

gilt.

Die Gleichungen (3.17) und (3.43) führen zu $m+n+1$ Gleichungen mit den Unbekannten a_0, \dots, a_m und b_1, \dots, b_n . Der einfachste Weg zu diesen Gleichungen zu gelangen, besteht darin, $R(x)$ und $f(x)$ mit dem Nenner von $R(x)$ zu multiplizieren und einen Koeffizientenvergleich in den ersten $m+n+1$ Monomen $1, x, x^2, \dots, x^{m+n}$ durchzuführen.

Betrachten wir allerdings nur den Spezialfall $m = n$, so erhalten wir $a_0 = c_0$ und die Verbleibenden Koeffizienten a, b erfüllen die Gleichungen

$$\sum_{l=1}^n b_l c_{n-l+k} = -c_{n+k} \quad k = 1, \dots, n \quad (3.19)$$

$$\sum_{l=0}^k b_l c_{k-l} = a_k \quad k = 1, \dots, n \quad (3.20)$$

Man startet also zuerst mit (3.19), welches die Gleichungen zur Bestimmung von b_1, \dots, b_n darstellen. Obwohl diese Gleichungen eine *Töplitzmatrix* erzeugen, wendet man nicht das spezielle Verfahren für *Töplitzmatrizen* aus Numerik 1 an. Erfahrungen zeigen, dass diese Gleichungen häufig nahezu singular sind. Die *LR*-Zerlegung ist hier aus Stabilitätsgründen vorzuziehen. Nachdem die b 's nun bekannt sind, liefert (3.20) eine explizite Darstellung für die a 's.

Die *Padé*-Approximation wird häufig dann angewendet, wenn die zu approximierende Funktion $f(x)$ nicht bekannt ist. Wir gehen davon aus, dass es möglich ist, den Wert von $f(x)$ und einiger Ableitungen an der Stelle $x = 0$, z.B. $f(0), f'(0), f''(0)$ zu bestimmen. Dies sind natürlich bereits die ersten Koeffizienten der Potenzreihenentwicklung, aber es ist keinesfalls klar, ob diese betragsmäßig kleiner werden und ob die Reihe überhaupt konvergiert. Wir wollen die Möglichkeiten der *Padé*-Approximation an einem Beispiel illustrieren, für eine Analyse der Methode verweisen wir auf [Cuyt/Wuytack].

Beispiel 3.5.1 Man stelle sich vor, man könnte die ersten fünf Terme einer Potenzreihe einer unbekanntenen Funktion $f(x)$ generieren, und diese sind

$$f(x) \approx 2 + \frac{1}{9}x + \frac{1}{81}x^2 - \frac{49}{8748}x^3 + \frac{175}{78732}x^4 + \dots$$

Wir wählen $m = n = 2$ für unsere *Padé*-Approximation. In der nachfolgenden Abbildung sind die abgebrochene Potenzreihendarstellung, die *Padé*-Approximation und die exakte Funktion

$$f(x) = \left(7 + (1+x)^{4/3}\right)^{1/3}$$

Diese Funktion hat einen Verzweigungspunkt bei $x = -1$, daher konvergiert die Potenzreihe lediglich in $-1 < x < 1$. Im größten Teil der nachfolgenden Abbildung ist die Reihe divergent und der Wert der abgebrochenen Entwicklung nahezu sinnlos. Nichtsdestoweniger liefert die *Padé*-Approximation sehr gute Werte zumindest bis $x \sim 10$.

MATLAB-Funktionen: pade.m und horner.m

```

1 function [a,b] = pade(c)
2 c=c(:); n=(length(c)-1)/2;
3 b=[1;-toeplitz(c(n+1:end-1),c(n+1:-1:2))\c(n+2:end)];
4 a=[toeplitz(c(1:n+1),[c(1),zeros(1,n)])*b]

1 function value = horner(a,x0)
2 value = a(end);
3 for k = length(a)-1:-1:1
4     value = value.*x0 + a(k);
5 end

```

MATLAB-Beispiel:

Vergleich von abgebrochener
Reihenentwicklung, Padé-
Approximation und exakter
Darstellung der Funktion

$$f(x) = \left(7 + (1+x)^{4/3}\right)^{1/3}.$$

Das Ergebnis ist in Abb. 3.5 dargestellt.

```
>> c = [2,1/9,1/81,-49/8748,175/78732];
>> x = linspace(0,10,400);
>> [a,b] = pade(c)
>> plot(x,(7+(1+x).^(4/3)).^(1/3),'k-', ...
        x,horner(c,x),'b-.', ...
        x,horner(a,x)./ horner(b,x),'r:')
>> axis([0,10,1.5,6])
>> legend('exakt','abgebr. Entwicklung',
        ...
        'Pade',2)
```

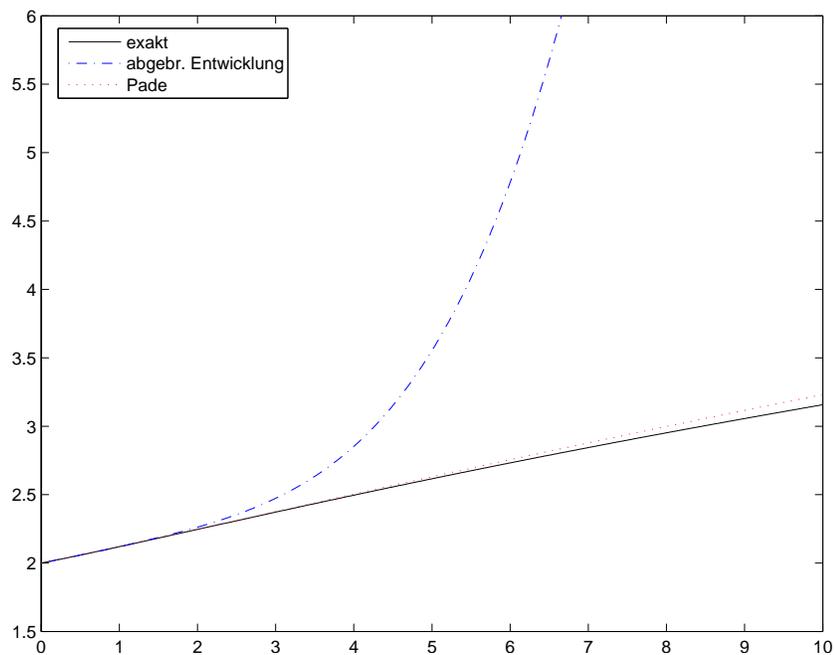


Abb. 3.2: Vergleich von abgebrochener Reihenentwicklung, Padé-Approximation und exakter Darstellung der Funktion $f(x) = (7 + (1+x)^{4/3})^{1/3}$..

4 SPLINES

In diesem Kapitel wird die Theorie polynomialer und rationaler Splinefunktionen dargestellt. Zuerst stellt sich jedoch die Frage, warum überhaupt die Spline-Approximation eingeführt wurde. Eine Antwort liefert der

Satz 4.0.1 (von Jackson) *Es gilt die Abschätzung*

$$\forall k \in \mathbb{N}_0 \exists c_k > 0 \forall (-\infty < a < b < \infty) \forall f \in C^k([a, b], \mathbb{R}) \forall n < k$$

$$E_n(f) \leq c_k \left(\frac{b-a}{n} \right)^k \omega \left(f^{(k)}, \frac{b-a}{2(n-k)} \right)$$

für die Bestapproximation E_n von f .

(Vgl. [Rivlin], Seite 23, in anderer Form [Schönhage], Seite 182.)

Dabei ist

$$E_n(f) := \text{dist}(f, \mathbb{P}_n) = \inf_{g \in \mathbb{P}_n} \|f - g\|_{\infty, [a, b]} \quad \text{und}$$

und

$$\omega(g, h) := \sup_{|\delta| < h} \|g(\cdot) - g(\cdot + \delta)\|_{\infty, [a, b]}$$

der Stetigkeitsmodul von g , ein Maß für die Glattheit von g .

Der Satz von Jackson besagt folgendes: Da der Stetigkeitsmodul einer wenig glatten Funktion groß ist, wird man keine „gute“ Näherung erwarten dürfen, da schon die Schranke für die Bestapproximation entsprechend groß wird. Ferner sind Interpolationspolynome

- abhängig von der Wahl der Knoten, wie schon das Beispiel von Runge (siehe Seite 58) zeigt,
- bei Änderung eines Koeffizienten eines Polynoms, tritt eine globale Änderung ein, und
- viele Basen sind schlecht konditioniert.

Die einfachste Form einer stetigen Funktion f , die die Bedingung $f(x_i) = y_i$ für gegebene geordnete Paare (x_i, y_i) ($i = 0, \dots, n$) erfüllt, ist sicherlich der Streckenzug, d.h.

$$f|_{(x_{i-1}, x_i)} = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}(\cdot - x_{i-1}) + y_{i-1} \quad (i = 1, \dots, n).$$

Auf jedem Intervall ist f also ein Polynom höchstens ersten Grades und insgesamt eine stetige Funktion.

Definition 4.0.2 (Spliner Raum $\mathcal{S}^k(\mathcal{T})$) *Es sei $\mathcal{T} = \{t_0, \dots, t_{n+1}\}$ eine Knotenfolge von $n + 2$ paarweise verschiedenen Knoten*

$$a = t_0 < \dots < t_{n+1} = b.$$

Ein **Spline** vom Grad k ($k \geq 0$) bezüglich \mathcal{T} ist eine Funktion $s \in C^{k-1}[a, b]$, für die auf jedem Intervall $[t_j, t_{j+1}]$, $j = 0, \dots, n$

$$s|_{[t_k, t_{k+1}]} \in \mathbb{P}_k$$

gilt. Den Raum aller Splines vom Grad k zur Knotenfolge \mathcal{T} bezeichnen wir mit $\mathcal{S}^k(\mathcal{T})$.

Unter $C^{-1}[a, b]$ ist der Raum der stückweise stetigen Funktionen zu verstehen, d.h. unstetig nur an den Knoten x_j ($j = 0, \dots, n + 1$).

4.1 KUBISCHE SPLINE-INTERPOLATION

Im Folgenden betrachten wir die Interpolation mit **kubischen Splines**.

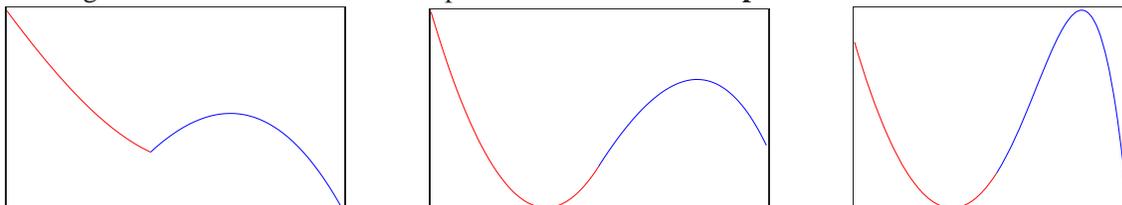


Abb. 4.1: Verschiedene Funktionen aus C^0 , C^1 und C^2

Welche dieser Funktionen in der Sequenz von Grafiken empfinden Sie als glatt?

Der Knick im linken Graphen ist offensichtlich, auch im mittleren Graphen erkennt man mit etwas Geduld und Erfahrung eine Unstetigkeit in der Krümmung der Funktion, welche jedoch in der rechten Grafik nicht mehr auszumachen ist.

In vielen grafischen Anwendungen genügen diese Glattheitsanforderungen an die Interpolationsfunktion, nämlich sie vom Auge als „glatt“ zu empfinden.

Dies ist einer der Hauptgründe, weswegen wir uns zuerst auf Funktionen aus $\mathcal{S}^k(\mathcal{T}) \subset C^2$ beschränken.

Untersuchen wir zuerst die Eigenschaften der kubischen Splines, bevor wir zu ihrer Konstruktion und Berechnung kommen.

Satz 4.1.1 Sei s ein interpolierender kubischer Spline zu der Funktion f an den Knoten $a = t_0 < \dots < t_{n+1} = b$ und y eine beliebige interpolierende Funktion von f , sodass

$$s''(t) \cdot (y'(t) - s'(t)) = 0 \quad (t \in [a, b]). \quad (4.1)$$

Dann gilt

$$\|s''\|_2 := \left(\int_a^b (s''(t))^2 dt \right)^{1/2} \leq \|y''\|_2. \quad (4.2)$$

Beweis. Aus (4.2) folgt mit $y'' = s'' + (y'' - s'')$

$$\begin{aligned} \int_a^b (y''(x))^2 dx &= \int_a^b (s''(x))^2 + 2s''(x)(y''(x) - s''(x)) + (y''(x) - s''(x))^2 dx \\ &= \int_a^b (s''(x))^2 + (y''(x) - s''(x))^2 dx \geq \int_a^b (s''(x))^2 dx, \end{aligned}$$

falls $\int_a^b s''(y'' - s'') dx$ verschwindet. Dies läßt sich aber mit (4.1) und partieller Integration unter Berücksichtigung von $s(x)|_{[x_{i-1}, x_i]} \in \mathbb{P}_3$ (und somit $s'''(x)|_{[x_{i-1}, x_i]} \equiv c_i \in \mathbb{R}$) wie folgt zeigen:

$$\begin{aligned} \int_a^b s''(x)(y''(x) - s''(x)) dx &= \sum_{i=1}^n \int_{x_{i-1}}^{x_i} s''(x)(y''(x) - s''(x)) dx \\ &= \sum_{i=1}^n \left(\underbrace{s''(x)(y'(x) - s'(x))}_{=0} \Big|_{x=x_{i-1}}^{x_i} - \int_{x_{i-1}}^{x_i} s'''(x)(y'(x) - s'(x)) dx \right) \\ &= - \sum_{i=1}^n c_i \int_{x_{i-1}}^{x_i} y'(x) - s'(x) dx = - \sum_{i=1}^n c_i \left[(y(x_i) - s(x_i)) - (y(x_{i-1}) - s(x_{i-1})) \right] = 0 \end{aligned}$$

□

Die Aussage dieses Satzes benötigen wir insbesondere zum Beweis des folgenden wichtigen Resultats über die Minimaleigenschaften der kubischen Splines.

Satz 4.1.2 (Minimaleigenschaft der kubischen Splines) *Es sei $\mathcal{T} = \{x_i\}$ eine Knotenfolge mit $a = x_0 \dots < x_{n+1} = b$ und $s \in \mathcal{S}^3(\mathcal{T})$ ein kubischer Spline, der neben den Interpolationsbedingungen $s(x_i) = f(x_i)$ eine der folgenden Randbedingungen erfülle:*

$$(i) \quad s'(a) = f'(a) \text{ und } s'(b) = f'(b) \quad (\text{vollständige Randbedingung})$$

$$(ii) \quad s''(a) = s''(b) = 0 \quad (\text{natürliche Randbedingung})$$

$$(iii) \quad s'(a) = s'(b) \text{ und } s''(a) = s''(b) \quad (\text{periodische Randbedingung})$$

(falls f periodisch mit Periode $b - a$)

Ein solches $s \in \mathcal{S}^3(\mathcal{T})$ existiert und ist eindeutig bestimmt. Für jede interpolierende Funktion $y \in C^2[a, b]$, die dieselben Interpolations- und Randbedingungen erfüllt, gilt ferner

$$\int_a^b (s''(x))^2 dx \leq \int_a^b (y''(x))^2 dx$$

Bevor wir diesen Satz beweisen noch eine Anmerkung zur Dimension von $\mathcal{S}^3(\mathcal{T})$ mit $\mathcal{T} = \{x_0, \dots, x_{n+1}\}$. Für das Teilintervall $[x_i, x_{i+1}]$ der Länge $h_i := x_{i+1} - x_i$ wählen wir folgenden Ansatz:

$$s_i(x) := s(x)|_{[x_i, x_{i+1}]} = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

Für seinen Wert und die Ableitungen s', s'' an den Endpunkten erhalten wir

$$s_i(x_i) = \quad \quad \quad d_i = f(x_i) \quad (4.3)$$

$$s_i(x_{i+1}) = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i = f(x_{i+1}) \quad (4.4)$$

$$s'_i(x_i) = \quad \quad \quad c_i \quad (4.5)$$

$$s'_i(x_{i+1}) = 3a_i h_i^2 + 2b_i h_i + c_i \quad (4.6)$$

$$s''_i(x_i) = \quad \quad \quad 2b_i \quad (4.7)$$

$$s''_i(x_{i+1}) = 6a_i h_i + 2b_i \quad (4.8)$$

Gehen wir nun davon aus, dass auf dem ersten Intervall $[x_0, x_1]$ die Koeffizienten a_0, b_0, c_0, d_0 gegeben seien, sodass die Interpolationsbedingungen auf $[x_0, x_1]$ erfüllt sind. Durch die Interpolationsbedingungen und die Stetigkeit von s' und s'' an den Knoten folgt:

$$s_1(x_1) = \quad \quad \quad d_1 = f(x_1) \quad (4.9)$$

$$s_1(x_2) = a_1 h_1^3 + b_1 h_1^2 + c_1 h_1 + d_1 = f(x_2) \quad (4.10)$$

$$s'_1(x_1) = \quad \quad \quad c_1 = 3a_0 h_0^2 + 2b_0 h_0 + c_0 \quad (4.11)$$

$$s''_1(x_1) = \quad \quad \quad 2b_1 = 6a_0 h_0 + 2b_0 \quad (4.12)$$

Aus den Gleichungen (3.24) – (3.27) folgt die Eindeutigkeit der a_1, b_1, c_1, d_1 . Per Induktion folgt dann auch die eindeutige Bestimmung der weiteren a_k, b_k, c_k, d_k , d.h. der Raum $\mathcal{S}^3(\mathcal{T})$ mit $\mathcal{T} = \{x_0, \dots, x_{n+1}\}$ besitzt $(n + 2) + 2 = n + 4$ Freiheitsgrade. Allgemein läßt sich zeigen:

$$\dim \mathcal{S}^k(\mathcal{T}) = n + k + 1.$$

Beweis von Satz 4.1.2. Die Interpolations- und Randbedingungen sind linear in s , und ihre Anzahl stimmt mit der Dimension von $\mathcal{S}_k(\mathcal{T})$ überein. Somit genügt es zu zeigen, dass für die Splinefunktion $f \equiv 0$ der triviale Spline $s \equiv 0$ einzige Lösung ist.

Da $y \equiv 0$ alle Bedingungen erfüllt, folgt mit Satz 6, dass auch $\|s''\| = 0$ gilt. Da s'' stetig, folgt somit auch $s'' = 0$, somit $s' = c_0$ und $s(x) = c_0 x + c_1$. Aus den Interpolationsbedingungen $s(x_i) = 0$ folgt sofort $s = 0$. \square

Bemerkung 4.1.3 Der Satz 4.1.2 gilt unter Verallgemeinerung der Randbedingungen für beliebige Splineräume \mathcal{S}^k ($k \geq 3$). Siehe z.B. [Hämmerlin/Hoffmann], Seite 252.

Berechnung kubischer Splines

Kommen wir nun zur Berechnung kubischer Splines. Zusätzlich zu den Daten $y_i := f(x_i)$ ($i = 0, \dots, n+1$) ist es zweckmäßig Variablen $y_i'' := s_i''(x_i) = s_{i-1}''(x_i)$ einzuführen und die Variablen a_i, \dots, d_i durch diese zu ersetzen. Aus den Gleichungen (3.18), (3.19) sowie (3.22) und (3.23) gewinnt man das Gleichungssystem

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ h_i^3 & h_i^2 & h_i & 1 \\ 0 & 2 & 0 & 0 \\ 6h_i & 2 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_i \\ b_i \\ c_i \\ d_i \end{pmatrix} = \begin{pmatrix} s_i(x_i) \\ s_i(x_{i+1}) \\ s_i''(x_i) \\ s_i''(x_{i+1}) \end{pmatrix} = \begin{pmatrix} y_i \\ y_{i+1} \\ y_i'' \\ y_{i+1}'' \end{pmatrix}. \quad (4.13)$$

Die Berechnung der Determinanten der Koeffizientenmatrix liefert

$$\det \begin{pmatrix} 0 & 0 & 0 & 1 \\ h_i^3 & h_i^2 & h_i & 1 \\ 0 & 2 & 0 & 0 \\ 6h_i & 2 & 0 & 0 \end{pmatrix} = -\det \begin{pmatrix} h_i^3 & h_i^2 & h_i \\ 0 & 2 & 0 \\ 6h_i & 2 & 0 \end{pmatrix} = -2 \det \begin{pmatrix} h_i^3 & h_i \\ 6h_i & 0 \end{pmatrix} = 12h_i^2$$

und damit die Eindeutigkeit der a_i, \dots, d_i und somit von s , falls s_i und s_i'' für alle $i = 0, \dots, n+1$ bekannt sein sollten. Das Lösen von (4.13) liefert nun

$$d_i = y_i \quad b_i = \frac{1}{2}y_i'' \quad (4.14)$$

$$a_i = \frac{1}{6h_i}(y_{i+1}'' - y_i'') \quad c_i = \frac{1}{h_i}(y_{i+1} - y_i) - \frac{1}{6}h_i(y_{i+1}'' + 2y_i''). \quad (4.15)$$

Es lassen somit sich die kubischen Polynome $s_i(x)$ in jedem Teilintervall eindeutig bestimmen, wenn neben den Stützwerten y_k auch die Größen y_k'' bekannt sind. Damit wäre neben der Interpolationseigenschaft auch gleich die Stetigkeit der zweiten Ableitung von $s(x)$ gesichert.

Was nun zu zeigen wäre, ist, dass aus den y_i, y_i'' auch die Stetigkeit von $s(x)$ folgt. Setzen wir die Darstellung der a_i, b_i, c_i und d_i in (3.21) ein, so erhalten wir

$$\begin{aligned} s_i'(x_{i+1}) &= 3h_i^2a_i + 2h_ib_i + c_i \\ &= 3h_i^2\left(\frac{1}{6h_i}(y_{i+1}'' - y_i'')\right) + 2h_i\left(\frac{1}{2}y_i'' + \frac{1}{h_i}(y_{i+1} - y_i) - \frac{1}{h_i}(y_{i+1}'' + 2y_i'')\right) \\ &= h_i\left(\frac{1}{2}y_{i+1}'' - \frac{1}{2}y_i'' + y_i'' - \frac{1}{6}y_{i+1}'' - \frac{1}{3}y_i''\right) + \frac{1}{h_i}(y_{i+1} - y_i) \\ &= \frac{h_i}{6}(2y_{i+1}'' + y_i'') + \frac{1}{h_i}(y_{i+1} - y_i) \end{aligned}$$

bzw.

$$s_{i-1}'(x_i) = \frac{1}{h_{i-1}}(y_i - y_{i-1}) + \frac{h_{i-1}}{6}(2y_i'' + y_{i-1}'') \quad (4.16)$$

Die Stetigkeit von $s'(x)$ an den inneren Knoten liefert mit der letzten Gleichung (4.16) wegen der Darstellung von c_i aus (3.20) und (3.29) die Gleichung

$$\begin{aligned} &\frac{1}{h_{i-1}}(y_i - y_{i-1}) + \frac{1}{6}h_{i-1}(2y_i'' + y_{i-1}'') \\ &= s_{i-1}'(x_i) \stackrel{!}{=} s_i'(x_i) = c_i = \frac{1}{h_i}(y_{i+1} - y_i) - \frac{1}{6}h_i(y_{i+1}'' + 2y_i'') \end{aligned}$$

Sortieren von y_k'' nach links, y_k nach rechts und anschließende Multiplikation mit 6 liefert

$$h_{i-1}y_{i-1}'' + 2(h_{i-1} + h_i)y_i'' + h_i y_{i+1}'' = \frac{6}{h_i}(y_{i+1} - y_i) - \frac{6}{h_{i-1}}(y_i - y_{i-1}). \quad (4.17)$$

Diese Bedingung muss für alle inneren Knoten x_1, \dots, x_n erfüllt sein und liefert somit n Gleichungen für die Unbekannten y_0'', \dots, y_{n+1}'' .

Allerdings reichen die n Gleichungen für $n + 2$ Unbekannte y_0'', \dots, y_{n+1}'' nicht aus um diese eindeutig zu bestimmen. Untersuchen wir nun die unterschiedliche Behandlung der Randbedingungen.

Berücksichtigung natürlicher und vollständiger Randbedingungen

Die vollständige Randbedingung $s'(a) = f'(a)$, bzw. $s'(b) = f'(b)$ liefert aufgrund von (3.20), (3.21) und (4.14) die weitere Gleichung

$$s'(a) = s'_0(x_0) = c_0 = \frac{1}{h_0}(y_1 - y_0) - \frac{1}{6}(y_1'' + 2y_0'') \stackrel{!}{=} f'(a).$$

Somit folgt

$$2h_0y_0'' + h_0y_1'' = \frac{6}{h_0}(y_1 - y_0) - 6f'(a), \quad (4.18)$$

bzw. aus

$$\begin{aligned} s'(b) &= s'_n(x_{n+1}) = 3a_n h_n^2 - 2b_n h_n + c_n \\ &= \frac{h_n}{2}(y_{n+1}'' - y_n'') + h_n y_n'' + \frac{1}{h_n}(y_{n+1} - y_n) - \frac{1}{6}h_n(y_{n+1}'' + 2y_n'') \\ &= h_n \left(\frac{1}{2}y_{n+1}'' - \frac{1}{2}y_n'' + y_n'' - \frac{1}{6}y_{n+1}'' - \frac{1}{3}y_n'' \right) + \frac{1}{h_n}(y_{n+1} - y_n) \\ &= h_n \left(\frac{1}{3}y_{n+1}'' + \frac{1}{6}y_n'' \right) + \frac{1}{h_n}(y_{n+1} - y_n) \stackrel{!}{=} f'(b), \end{aligned}$$

für den rechten Rand

$$h_n y_n'' + 2h_n y_{n+1}'' = -\frac{6}{h_n}(y_{n+1} - y_n) + 6f'(b) \quad (4.19)$$

Für die natürlichen Randbedingung $y_0'' = y_{n+1}'' = 0$ ergeben sich folgende Gleichungen

$$y_0'' = 0 \quad (4.20)$$

$$y_{n+1}'' = 0 \quad (4.21)$$

Natürliche und vollständige Randbedingungen können auch gemischt auftreten, d.h. an der Stelle x_0 ist neben $f(x_0)$ auch die Ableitung $f'(x_0)$ vorgegeben und an der Stelle x_{n+1} gilt $y_{n+1} = 0$.

Im Falle $n = 5$ erhalten wir daraus folgendes lineares Gleichungssystem:

$$\begin{aligned}
& \begin{pmatrix} \star & \star & \star & \star & \star & \star \\ h_0 & 2(h_0 + h_1) & h_1 & & & \\ & h_1 & 2(h_1 + h_2) & h_2 & & \\ & & h_2 & 2(h_2 + h_3) & h_3 & \\ & & & h_3 & 2(h_3 + h_4) & h_4 \\ \star & \star & \star & \star & \star & \star \end{pmatrix} \cdot \begin{pmatrix} y_0'' \\ \vdots \\ \vdots \\ y_5'' \end{pmatrix} \\
&= \begin{pmatrix} \star \\ \frac{6}{h_1}(y_2 - y_1) - \frac{6}{h_0}(y_1 - y_0) \\ \frac{6}{h_2}(y_3 - y_2) - \frac{6}{h_1}(y_2 - y_1) \\ \frac{6}{h_3}(y_4 - y_3) - \frac{6}{h_2}(y_3 - y_2) \\ \frac{6}{h_4}(y_5 - y_4) - \frac{6}{h_3}(y_4 - y_3) \\ \star \end{pmatrix} \quad (4.22)
\end{aligned}$$

Die in diesem Schema mittels \star gekennzeichnete erste und letzte Zeile ist dabei durch die Wahl der Randbedingungen (4.18), (3.22) bzw. (3.35), (3.36) gegeben.

Für vollständige Randbedingungen sind in (4.22) erste und letzte Zeile durch die Gleichungen (4.18) und (4.20) zu ersetzen, sodass wir erhalten:

$$\begin{pmatrix} 2h_0 & h_0 & & & & \\ \star & \star & \star & & & \\ & & \ddots & & & \\ & & & \star & \star & \star \\ & & & & h_n & 2h_n \end{pmatrix} \begin{pmatrix} y_0'' \\ \star \\ \vdots \\ \star \\ y_{n+1}'' \end{pmatrix} = \begin{pmatrix} \frac{6}{h_0}(y_1 - y_0) - 6f'(a) \\ \star \\ \vdots \\ \star \\ -\frac{6}{h_n}(y_{n+1} - y_n) + 6f'(b) \end{pmatrix}$$

Hierbei sind die mit \star gekennzeichneten Zeilen 1 bis n dem System (4.22) zu entnehmen. Analog erhalten wir für die Randbedingungen $y_0'' = f''(a)$, $y_{n+1}'' = f''(b)$

$$\begin{pmatrix} 1 & & & & \\ \star & \star & \star & & \\ & & \ddots & & \\ & & & \star & \star & \star \\ & & & & & 1 \end{pmatrix} \begin{pmatrix} y_0'' \\ \star \\ \vdots \\ \star \\ y_{n+1}'' \end{pmatrix} = \begin{pmatrix} f''(a) \\ \star \\ \vdots \\ \star \\ f''(b) \end{pmatrix}$$

Da im letzten Gleichungssystem y_0'' und y_{n+1}'' explizit bekannt sind, können wir es wie folgt reduzieren:

$$\begin{pmatrix} 2(h_0 + h_1) & h_1 & & & \\ h_1 & 2(h_1 + h_2) & h_2 & & \\ & & \ddots & & \\ & & & h_{n-1} & 2(h_{n-1} + h_n) \end{pmatrix} \cdot \begin{pmatrix} y_1'' \\ \vdots \\ \vdots \\ y_n'' \end{pmatrix} = \begin{pmatrix} \star & -h_0 y_0'' \\ \star & \\ \star & \\ \star & -h_n y_{n+1}'' \end{pmatrix}$$

Bemerkung 4.1.4 Man beachte, dass $y_0'' = c_0$ und $y_{n+1}'' = c_1$ mit $c_0, c_1 \in \mathbb{R}$ nicht die Matrix, sondern nur die rechte Seite modifiziert, damit also auch eine Verallgemeinerung der natürlichen Randbedingungen möglich ist.

Berücksichtigung periodischer Randbedingungen

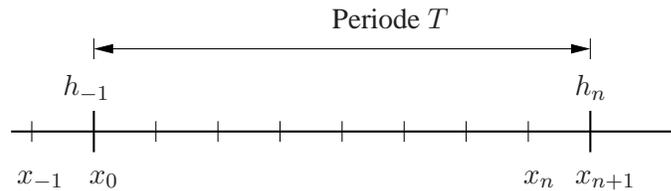
Die Stützstellen $x_0 < \dots < x_{n+1}$ seien nun so festgelegt, dass $x_{n+1} = x_0 + T$, wobei T die Periode der gesuchten Funktion darstelle.

Die periodischen Randbedingungen sind $f(x_0) = f(x_{n+1})$, $f'(x_0) = f'(x_{n+1})$ sowie $f''(x_0) = f''(x_{n+1})$. Mit den Größen $y_0 = y_{n+1}$, y_1, \dots, y_n und den Unbekannten $y''_0 = y''_{n+1}$, y''_1, \dots, y''_n ist die Stetigkeit an den $n + 1$ Stützstellen x_0, \dots, x_n zu erfüllen (beachte, dass die Stetigkeit bei x_0 der bei x_{n+1} aufgrund der Periodizität entspricht).

Für die inneren Knoten x_1, \dots, x_n sind die Gleichungen gegeben durch (4.17). Aber auch zum Knoten x_0 sind die Gleichungen durch (4.17) gegeben, wenn man bei der Formulierung

$$\begin{aligned} h_{-1} &= x_0 - x_{-1} = x_{n+1} - x_n = h_n \\ y''_{-1} &= y''_n \text{ und } y_{-1} = y_n \end{aligned}$$

beachtet, da zu jedem Knoten x_i nur die Informationen y_{i-1}, y_i, y_{i+1} und $y''_{i-1}, y''_i, y''_{i+1}$ benötigt werden, d.h. die Daten vom linken, rechten Nachbarn sowie vom Knoten selbst



Das System lautet dann allgemein für $n = N$, $x_{N+1} = x_0 + T$

$$\begin{pmatrix} 2(h_N + h_0) & h_0 & & & & & h_N \\ & h_0 & 2(h_0 + h_1) & h_1 & & & \\ & & h_1 & 2(h_1 + h_2) & h_2 & & \\ & & & \ddots & \ddots & & \\ & & & & h_{N-2} & 2(h_{N-2} + h_{N-1}) & h_{N-1} \\ h_N & & & & h_{N-1} & 2(h_{N-1} + h_N) & \end{pmatrix} \cdot \begin{pmatrix} y''_0 \\ y''_1 \\ y''_2 \\ \vdots \\ \vdots \\ y''_{N-1} \\ y''_N \end{pmatrix} \\ = \begin{pmatrix} \frac{6}{h_0}(y_1 - y_0) - \frac{6}{h_N}(y_0 - y_N) \\ \frac{6}{h_1}(y_2 - y_1) - \frac{6}{h_0}(y_1 - y_0) \\ \frac{6}{h_2}(y_3 - y_2) - \frac{6}{h_1}(y_2 - y_1) \\ \vdots \\ \frac{6}{h_{N-1}}(y_N - y_{N-1}) - \frac{6}{h_{N-2}}(y_{N-1} - y_{N-2}) \\ \frac{6}{h_N}(y_0 - y_N) - \frac{6}{h_{N-1}}(y_N - y_{N-1}) \end{pmatrix}$$

Bemerkung 4.1.5 Der Name *Spline* läßt sich aus dem Englischen wie *Latte* oder *Zeichenlineal* übersetzen. Wieso die oben diskutierten Funktionen als *Splines* bezeichnet werden, erklärt sich durch eine physikalische Interpretation der Eigenschaft aus Satz 6:

Die 'Biegeenergie' E eines elastischen Körpers ist gegeben durch

$$E = \int_a^b \left(\frac{y''(x)}{(1 + y'(x)^2)^{3/2}} \right)^2 dx,$$

also dem Integral über das Quadrat der Krümmung

$$\kappa(x) = \frac{y''(x)}{(1 + y'(x)^2)^{3/2}}.$$

Für kleine $y'(x)$ liefert $y''(x)$ eine vernünftige Näherung an $\kappa(x)$, d.h. unter allen Interpolierenden 'minimiert' ein Spline das Integral über das Quadrat der Krümmung, also die 'glatteste' Interpolationsfunktion.

Die natürlichen Randbedingungen entsprechen also gerade der Situation, dass das Zeichenwerkzeug außerhalb des Interpolationsintervalls gerade ist. Da bei den vollständigen Randbedingungen mehr Informationen über die zu interpolierende Funktion eingehen, sind auch deren Interpolationseigenschaften besser.

Eine Matlab-Realisierung zur Berechnung der Koeffizienten a_i, b_i, c_i, d_i für alle Intervalls $[x_i, x_{i+1}]$ ($i = 0, \dots, n$) ist im Folgenden wieder gegeben.

MATLAB-Funktion: CoeffSpline.m

```

1 function coeff = CoeffSpline(t,y,kind,param)
2 % param(1,1)=f'(a) bzw. f''(a) (abhängig von der Wahl von kind)
3 % param(2,1)=f'(b) bzw. f''(b)
4 n=length(t);
5 dy = y(2:end)-y(1:end-1);
6 %% Berechnung des "Kerns" von A und b
7 h1 = t(2:n)-t(1:n-1);
8 h2 = t(3:end)-t(1:end-2);
9 A = sparse(n,n);
10 B = [h1,[2*h2;0],[h1(2:end);0]];
11 A(2:n-1,:)=spdiags(B,[0,1,2],n-2,n);
12 b = zeros(n,1);
13 b(2:n-1)=6*((y(3:n)-y(2:n-1))./h1(2:n-1)-...
14           (y(2:n-1)-y(1:n-2))./h1(1:n-2)));
15 %% Erweiterung von A und b für unterschiedliche Randbedingungen
16 switch kind
17     case 'nat'
18         A(1,1)=1; A(n,n)=1;
19         b(1)=param(1); b(end)=param(2);
20     case 'per'
21         if y(1)~=y(end)
22             error('This data is not applicable for periodic splines')
23         end
24         A(1,[1,2,n-1])=[2*(h1(1)+h1(end)),h1(1),h1(end)];
25         A(n,[1,n])=[1,-1];
26         b(1)=6*((y(2)-y(1))/h1(1)-(y(1)-y(end-1))/h1(end));
27     case 'compl'
28         A(1,[1,2])=[2*h1(1),h1(1)];
29         A(n,[n-1,n])=[h1(end),2*h1(end)];
30         b(1)=6*(y(2)-y(1)-param(1));
31         b(n)=-6*(y(end)-y(end-1)+param(2));
32     otherwise
33         error('This kind does not exist!')
34 end
35 %% Berechnung der y''
36 y2d=A\b;
37 %% Berechnung der Koeffizienten a,b,c,d
38 coeff=[y(1:end-1),...
39        (y(2:end)-y(1:end-1))./h1(1:end)-h1(1:end).*(y2d(2:end)+2*
40        y2d(1:end-1))/6,...
41        y2d(1:end-1)/2,(y2d(2:end)-y2d(1:end-1))./(6*h1(1:end))];

```

4.2 PUNKTAUSWERTUNG KUBISCHER SPLINES

Nachdem wir nun im vorletzten Kapitel analysiert haben, wie sich aus den Daten $a = x_0 < x_1 < \dots < x_{n+1} = b$ und y_0, \dots, y_{n+1} ein Kubischer Spline bestimmen läßt, stellt sich nun die Frage, wie wir diesen auswerten können. Im Gegensatz zu einem Polynom, sind die Splines nur jeweils stückweise definiert, d.h. wollen wir an einer Stelle $x \in [a, b]$ den Spline $s(x)$ auswerten, müssen wir erst k mit $x \in [x_k, x_{k+1}]$ bestimmen.

Bei der trivialen Realisierung, bei der man nacheinander $j = 0, 1, \dots, n$ durchgeht und testet ob x in $[x_j, x_{j+1}]$ liegt, benötigt man bei einer äquidistanten Unterteilung $|x_{j+1} - x_j| =: h$ ($j = 0, \dots, n$) in $n + 1$ Teilintervalle durchschnittlich $n/2$ Abfragen. Da x mit der gleichen Wahrscheinlichkeit $1/(n + 1)$ in einem der Intervalle $[x_j, x_{j+1}]$ ($j = 0, \dots, n$) liegt, wird bei einem x , welches im letzten ($(n + 1)$ -ten) Intervall $[x_n, x_{n+1}]$ oder vorletzten Intervall $[x_{n-1}, x_n]$ vorher n -mal ein Test auf „ x liegt im Intervall“ durchgeführt. Für ein x , welches im j -ten Intervall $[x_{j-1}, x_j]$ liegt, wird eine solche Überprüfung j -mal durchgeführt, d.h. durchschnittlich werden $\frac{1}{n+1} + \frac{2}{n+1} + \dots + \frac{n-1}{n+1} + \frac{n}{n+1} + \frac{n}{n+1} = \frac{(n+1)n}{2(n+1)} + \frac{n}{n+1} \approx \frac{n}{2} + 1$ Tests benötigt.

Wie man schnell sieht, ist man mit einer binären Suche deutlich schneller. Vereinfachen wir die Voraussetzungen insofern, dass wir von $n = 2^s$ Intervallen ausgehen und unser x liege mit der gleichen Wahrscheinlichkeit in einem der Teilintervalle. Mit einem Test ob x in den ersten 2^{s-1} oder den letzten 2^{s-1} Intervallen liegt, reduzieren wir die Problemgröße auf die Hälfte und erhalten nach s Tests das gesuchte Intervall.

Gilt nun $2^{s-1} < n \leq 2^s$, so wähle man $2^s - n$ virtuelle Intervalle $[b, b]$ und nach s Bisektionen hat man das gesuchte Intervall gefunden. Die Anzahl der Tests ist für $2^{s-1} < n \leq 2^s$ gerade $s = \lceil \log_2 n \rceil$. Für $n = 100(10000)$ benötigt man durchschnittlich bei der sequentiellen Suche dem 501(5001) Tests und bei der binären Suche nur 7(14) Tests.

Nachfolgend führen wir die entsprechenden *Matlab*-Zeilen zur Auswertung mittels sequentieller und binärer Suche an. In beiden Fällen wird das Vorgehen durch die entsprechende *Matlab*-Ausgabe verdeutlicht:

MATLAB-Funktion: sequentiellesuche.m

```

1 function k = sequentielle_suche(x,x0)
2 % x_1 < x_2 < x_3 < < x_n
3 % Finde kleinstes k, sodass x0 in [x_k,x_(k+1)]
4 % und setze k = 0, wenn es kein solches k gibt
5 for k = 1:length(x)-1
6     if x(k) <= x0 && x0 <= x(k+1)
7         return
8     end
9 end
10 k = 0;

```

MATLAB-Funktion: binaeresuche.m

```

1 function k_unten = binaeresuche(x,x0)
2 % x_1 < x_2 < x_3 < ... < x_n
3 % Finde kleinstes k so dass x0 in [x_k,x_(k+1)]
4 % und setze k = 0, wenn es kein solches k gibt
5 if x0 < x(1) || x(end) < x0
6     k_unten = 0;
7     return
8 end
9 k_unten = 1;
10 k_oben = length(x);
11 while k_oben - k_unten > 1
12     k_mitte = floor((k_unten + k_oben)/2); %rundet -> -inf
13     if x(k_unten) <= x0 && x0 <= x(k_mitte)
14         k_oben = k_mitte;
15     else
16         k_unten = k_mitte;
17     end
18 end

```

MATLAB-Beispiel:

Als Ausgabe erhalten wir:

```

>> N=10^6,x=[1:N];x0=N*rand(100,1);
>> tic, for k=1:100,
    sequentiellesuche(x,x0(k)); end, toc
N =
    1000000
Elapsed time is 1.844000 seconds.
>> N=10^6,x=[1:N];x0=N*rand(100,1);
>> tic, for k=1:100,
    binaeresuche(x,x0(k)); end, toc
N =
    1000000
Elapsed time is 0.016000 seconds.

```

Suche mit korrelierten Daten

Häufig kommt man bei der Auswertung des Splines noch zu einer speziellen Situation, nämlich die Auswertung von s an einer aufsteigenden Folge ($t_j \leq t_{j+1}$) von Punkten $t_j \in [a, b]$ ($j = 1, \dots, m$). Gilt $t_j \in [x_{k_j}, x_{k_j+1}]$ ($k_j \in \{0, \dots, n\}$), so ist klar, dass man t_{j+1} nur noch in der Teilmenge $[x_{k_j}, b]$ suchen muss. Wäre nur noch ein Intervall zu suchen, böte die Bisektionsmethode einen effizienten Suchalgorithmus, wenn aber noch mehrere Intervalle für t_{j+1}, \dots, t_m zu lokalisieren sind, wird das nächste k_{j+1} in der Nähe des zuletzt bestimmten k_j liegen. Dies muss aber keineswegs dasselbe oder auch das nächste Intervall sein. Die Idee des folgenden „Jagd“-Algorithmus ist es, das nächste k_{j+1} durch *größer* werdende Schritte einzuschachteln.

Gilt $t_{j+1} \notin [x_{k_j}, x_{k_{j+1}}]$ so teste man nacheinander

$$\begin{aligned} t_{j+1} &\in [x_{k_j+1}, x_{k_j+2}]? \\ t_{j+1} &\in [x_{k_j+2}, x_{k_j+4}]? \\ t_{j+1} &\in [x_{k_j+4}, x_{k_j+8}]? \\ &\vdots \end{aligned}$$

Hat man hierdurch ein Intervall identifiziert, wendet man bezüglich diesem Intervall die Bisektionsmethode an. Im „Worstcase“ benötigt man zweimal länger als mit der Bisektionssuche, aber im besten Fall ist man um den Faktor $\log_2 n$ schneller.

Nachfolgend der oben erwähnte „Jagd-Algorithmus“ in *MATLAB*-Implementierung:

MATLAB-Funktion: lokalisierejagd.m

```

1 function ks = lokalisierejagd(xs,x0)
2 % xs(1) < xs(2) < xs(3) < ... < xs(n)
3 % Finde für alle x0's die kleinsten k's, sodass x0 in [xs(k),xs(k
  +1)]
4 ks = zeros(length(x0),1);
5 n = length(xs);
6 k = 1;
7 for j = 1:length(x0)
8     inc = 1;
9     k_next = k + 1;
10    while k_next <= n && x0(j) > xs(k_next) % hunting
11        k = k_next;
12        k_next = k_next + inc;
13        inc = 2 * inc;
14    end
15    k_next = min(k_next,n);
16    if k_next > k+1 % bisection
17        k = k + binaeresuche(xs(k:k_next),x0(j)) - 1;
18    end
19    ks(j) = k;
20 end
21 end

```

MATLAB-Beispiel:

Besteht der Spline aus n Intervallen und gesucht ist die Auswertung an $m \ll n$ Stützstellen so benötigen `lokalisierejagd` und `binaeresuche` etwa gleich viel Zeit.

```

>> m = 20001; n = 80001;
>> nodes = linspace(0,1,n);
>> x0 = linspace(0,1,m);
>> tic, s = lokalisierejagd(nodes,x0);toc
Elapsed time is 0.448687 seconds.
>> t=zeros(m,1);
>> tic,for j = 1:m,t(j) = binaeresuche(
    nodes,x0(j)); end,toc
Elapsed time is 0.349985 seconds.

```

Ist jedoch die Anzahl der Auswertung deutlich größer als die Anzahl der Intervalle, auf dem der Spline stückweise definiert ist, so ist `lokalisierejagd` deutlich schneller als `binaeresuche`.

```
>> m = 80001; n = 20001;
>> nodes = linspace(0,1,n);
>> x0 = linspace(0,1,m);
Elapsed time is 0.023416 seconds.
>> tic, s = lokalisierejagd(nodes,x0); toc
>> t=zeros(m,1);
>> tic,for j = 1:m,t(j) = binaeresuche(
    nodes,x0(j)); end,toc
Elapsed time is 1.382369 seconds.
```

4.3 PARAMETRISIERTE KURVEN UND FLÄCHEN

Implizite und parametrisierte Darstellung

Die beiden häufigsten Methoden um Kurven oder Flächen mathematisch zu beschreiben sind die implizite Darstellung und die parametrisierte Form.

Die implizite Darstellung einer Kurve in der xy -Ebene hat die Form $f(x, y) = 0$. Zu einer gegebenen Kurve ist diese Darstellung eindeutig bis auf eine multiplikative Konstante. Ein Beispiel ist der Einheitskreis, definiert durch die Gleichung $f(x, y) = x^2 + y^2 - 1 = 0$.

In der Parameterdarstellung wird jede Koordinate eines Punktes auf der Kurve separat durch eine explizite Funktion eines unabhängigen Parameters dargestellt,

$$\mathbf{C}(t) = (x(t), y(t)) \quad a \leq t \leq b.$$

Somit ist $\mathbf{C}(t)$ eine vektorwertige Funktion des Parameters t . Obwohl das Intervall $[a, b]$ beliebig sein kann, wird es üblicherweise auf $[0, 1]$ normiert. Der erste Quadrant des Einheitskreises ist definiert durch die Parameterdarstellung

$$x(t) = \cos(t), \quad y(t) = \sin(t) \quad a \leq t \leq \pi/2.$$

Substituiert man $u = \tan(t/2)$ so erhält man die alternative Darstellung

$$x(u) = \frac{1 - u^2}{1 + u^2}, \quad y(u) = \frac{2u}{1 + u^2} \quad 0 \leq u \leq 1.$$

Die parametrische Darstellung ist folglich nicht eindeutig.

Beide Darstellungsformen haben Vor- und Nachteile, von denen einige hier genannt seien.

- Fügt man eine z -Koordinate hinzu, so lässt sich die gegebene Parameterdarstellung einer Kurve einfach in 3-dimensionalen Raum einbetten. Durch die implizite Form lassen sich nur Kurven in der xy (oder yz oder xz) Ebene darstellen.
- Parametrisierte Kurven haben eine natürliche Richtung (von $\mathbf{C}(a)$ zu $\mathbf{C}(b)$ für $a \leq t \leq b$). Somit lassen sich einfach geordnete Folgen von Punkten erzeugen. Implizit gegebene Kurven haben diese Eigenschaft nicht.
- In der Parameterdarstellung muss man manchmal mit „Anomalien kämpfen“, die nicht im Zusammenhang stehen mit der wirklichen Geometrie. Ein Beispiel ist die Einheitskugel. Verwendet man Kugelkoordinaten so sind die Pole algorithmisch schwierige Punkte, obwohl sie sich von den anderen Punkten nicht unterscheiden.

- Die Komplexität vieler geometrischer Operationen und Manipulationen hängt stark von der Darstellung ab. Die Berechnung eines Punktes auf einer Kurve ist schwierig in der impliziten Darstellung. Die Entscheidung, ob ein Punkt auf einer Kurve oder Fläche liegt ist jedoch in impliziten Darstellung einfacher.
- Unbeschränkte Geometrien lassen sich nur schwer mit einer Parameterdarstellung beschreiben.

Lässt man beliebige Koordinatenfunktionen $x(t)$, $y(t)$, $z(t)$ zur Beschreibung von Kurven zu, so erhält man eine riesige Auswahl an möglichen Kurven. Möchte man dies aber mit Hilfe eines Rechners umsetzen, so gibt es einige Restriktionen zu berücksichtigen. Am besten wäre es, man beschränkt sich auf eine Klasse von Funktionen, die

- die gewünschten Kurven präzise genug darstellt, wie sie für Berechnungen oder Darstellungen benötigt werden,
- einfach, effizient und stabil sind,
- wenig Speicherplatz benötigen,
- mathematisch einfach gut verstanden sind (d.h. keine Heuristiken).

Eine naheliegende Wahl von Funktionen wären die Polynome. Obwohl sie die letzten beiden Punkte in der Wunschliste erfüllen, gibt es mehrere wichtiger Kurven und Flächen, die sich nicht durch Polynome darstellen lassen, z.B. Kreise und Kugeln.

Die Darstellung einer Kurve in monomilaer Basis n -ten Grades ist gegeben durch

$$\mathbf{C}(t) = (x(t), y(t), z(t)) = \sum_{j=0}^n \mathbf{a}_j t^j \quad 0 \leq t \leq 1$$

mit $\mathbf{a}_j = (x_j, y_j, z_j)$. Zu einem gegebenem t_0 lässt sich der Punkt $\mathbf{C}(t_0)$ möglichst effizient mit dem Horner-Schema berechnen:

- für den Grad = 1: $\mathbf{C}(t_0) = \mathbf{a}_0 + t_0 \mathbf{a}_1$
- Grad = 2: $\mathbf{C}(t_0) = \mathbf{a}_0 + t_0(\mathbf{a}_1 + t_0 \mathbf{a}_2)$
- Grad = 3: $\mathbf{C}(t_0) = \mathbf{a}_0 + t_0(\mathbf{a}_1 + t_0(\mathbf{a}_2 + t_0 \mathbf{a}_3))$
- \vdots
- Grad = n : $\mathbf{C}(t_0) = \mathbf{a}_0 + t_0(\dots t_0(\mathbf{a}_{n-2} + \dots t_0(\mathbf{a}_{n-1} + t_0 \mathbf{a}_n)))$

In Matlab sieht der Algorithmus wie folgt aus.

MATLAB-Funktion: horner.m

```

1 function f = horner(a,t0)
2 % Compute point on a power basis curve
3 f = a(:,end);
4 for k = size(a,2)-1:-1:1
5     f = f.*t0+a(:,k);
6 end

```

4.4 BERNSTEIN-POLYNOME UND BÉZIER-KURVEN

Die monomiale Basis ist nicht die einzige, um Polynome darzustellen. In Rahmen der Interpolation wurden auch schon die Lagrange- und Newton-Basis diskutiert. Wir definieren nun zuerst eine weitere Basis, nämlich die Bernstein-Polynome. Obwohl die parametrisierten Funktionen, dargestellt in monomialer Basis oder mit Bernstein-Polynomen, mathematisch äquivalent sind, so ist die Darstellung mit Hilfe der Bernstein-Polynome für die Darstellung von Kurven und Flächen deutlich geeigneter. An entsprechender Stelle kommen wir auf diesen Punkt zurück.

Definition 4.4.1 (Bernstein-Polynom) Das i -te **Bernstein-Polynom** vom Grad n bezüglich des Intervalls $[0, 1]$ ist das Polynom $B_i^n \in \mathbb{P}_n$ mit

$$B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}. \quad (4.23)$$

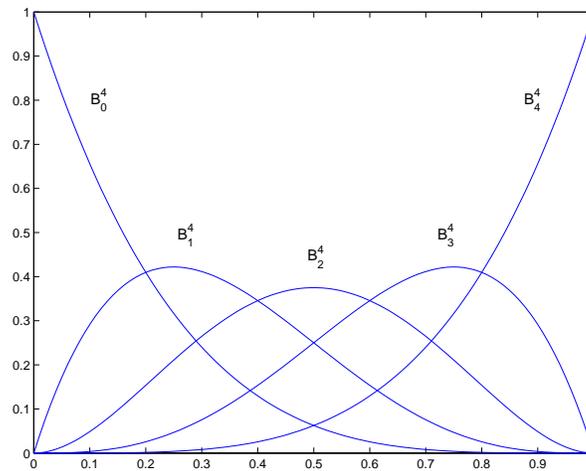


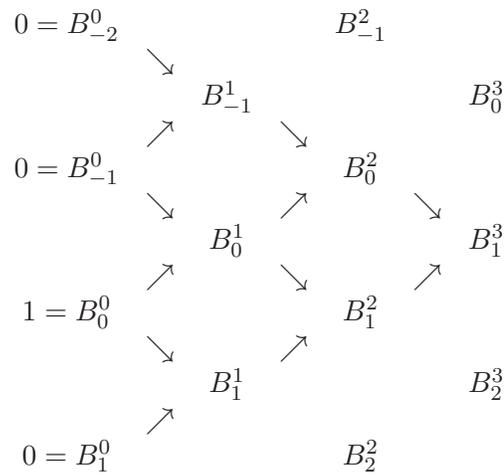
Abb. 4.2: Bernstein-Polynome B_0^4, \dots, B_4^4 auf dem Intervall $[0, 1]$.

Satz 4.4.2 (Eigenschaften der Bernstein-Polynome) Die Bernstein-Polynome haben folgende Eigenschaften:

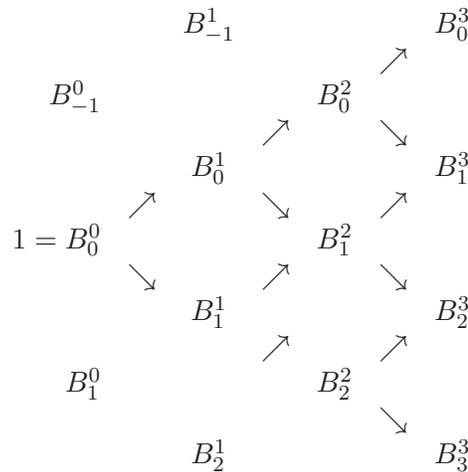
- (i) $B_i^n(t) \geq 0$ für alle i, n und $0 \leq t \leq 1$ (Positivität).
- (ii) $\sum_{i=0}^n B_i^n(t) = 1$ für alle $0 \leq t \leq 1$ (Zerlegung der Eins).
- (iii) $B_0^n(0) = B_n^n(1) = 1$.
- (iv) $B_i^n(t)$ hat genau ein Maximum im Intervall $[0, 1]$, und zwar bei $t = i/n$.
- (v) $B_i^n(t) = B_{n-i}^n(1-t)$ für $i = 0, \dots, n$ (Symmetrie).
- (vi) $B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t)$ (Rekursionsformel); wir definieren $B_i^n(t) \equiv 0$ für $i < 0$ oder $i > n$.
- (vii)

$$\frac{d}{dt} B_i^n(t) = n (B_{i-1}^{n-1}(t) - B_i^{n-1}(t))$$

$$\text{mit } B_{-1}^{n-1}(t) \equiv B_n^{n-1}(t) \equiv 0$$



Tab. 4.1: Berechnung von B_1^3 .



Tab. 4.2: Berechnung aller kubischen Bernstein-Polynome.

Die Gleichung (4.23) liefert $B_0^0(t) = 1$. Aus der Eigenschaft 4.4.2.vi gewinnen wir die linearen und quadratischen Bernstein-Polynome

$$\begin{aligned}
 B_0^1(t) &= (1-t)B_0^0(t) + tB_{-1}^0(t) = 1-t \\
 B_1^1(t) &= (1-t)B_1^0(t) + tB_0^0(t) = t \\
 B_0^2(t) &= (1-t)B_0^1(t) + tB_{-1}^1(t) = (1-t)^2 \\
 B_1^2(t) &= (1-t)B_1^1(t) + tB_0^1(t) = 2t(1-t) \\
 B_2^2(t) &= (1-t)B_2^1(t) + tB_1^1(t) = t^2
 \end{aligned}
 \tag{4.24}$$

Die Eigenschaft 4.4.2.vi liefert einen einfachen Algorithmus, um Werte der Bernstein-Polynome zu einem gegebenen t zu bestimmen. In Tabelle ?? ist dargestellt, welche B_k^0 benötigt werden, um B_1^3 zu berechnen. Berücksichtigt man die Nulleinträge ($B_{-2}^0 = B_{-1}^0 = B_1^0 = 0$), d.h. man vernachlässigt die Terme von denen man weiss das sie verschwinden, so lassen sich alle kubischen Bernstein-Polynome, wie in der folgenden Tabelle 4.4 dargestellt, effizient bestimmen.

Die Funktion `AllBernstein.m` kombiniert das in Tabelle 4.4 dargestellte Vorgehen mit Gleichung (4.23) um die Bernstein-Polynome n -ten Grades an einer gegebenen Stelle t zu bestimmen.

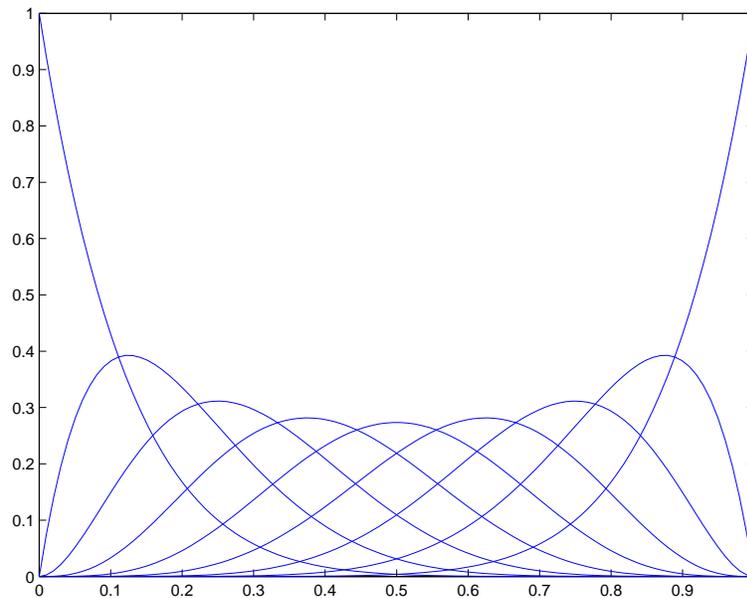


Abb. 4.3: Ergebnis der Matlab-Zeilen.

MATLAB-Funktion: AllBernstein.m

```

1 function B = AllBernstein(n,x)
2 % Compute all n-th Bernstein polynomials
3 B = zeros(n+1,1);
4 B(1) = 1;
5 for j=1:n
6     saved = 0;
7     for k=1:j
8         temp = B(k);
9         B(k) = saved + (1-x) * temp;
10        saved = x * temp;
11    end
12    B(j+1) = saved;
13 end

```

MATLAB-Beispiel:

Die folgenden Zeilen stellen die Bernstein-Polynome B_0^8, \dots, B_8^8 graphisch dar.

```

>> n = 8; no = 100;
>> t = linspace(0,1,no);
>> B = zeros(n+1,no);
>> for k=1:no,B(:,k)=AllBernstein(n,t(k));
>>     end
>> hold on, for k=1:n+1, plot(t,B(k,:)),
>>     end

```

Definition 4.4.3 (Bézierkurve) Gegeben seien $n + 1$ **Kontrollpunkte** $\mathbf{P}_j \in \mathbb{R}^d$ ($d \in \mathbb{N}$). Eine **Bézierkurve** n -ten Grades zu gegebenen $n + 1$ Punkten $\mathbf{P}_j \in \mathbb{R}^d$ ($i = 0, \dots, n, d \in \mathbb{N}$) ist für $t \in [0, 1]$ definiert als

$$\mathbf{C}(t) = \sum_{j=0}^n B_j^n(t) \mathbf{P}_j. \quad (4.25)$$

Bemerkung 4.4.4 Da die Bernstein-Polynome eine Zerlegung der Eins bilden, ist die Summe ausgewertet für ein festes t nichts anderes als die Linearkombination der gegebenen Punkte \mathbf{P}_j . Diese Punkte heißen **Kontrollpunkte** zur Splinekurve s .

Bemerkung 4.4.5 Ist $d = 2$, so heißt die geradlinige Verbindung der Punkte $\mathbf{P}_0, \dots, \mathbf{P}_n$ das **Kontrollpolygon**.

Für $n = 2$ und $\mathbf{C}(t) = \sum_{j=0}^2 B_j^2(t) \mathbf{P}_j$ gilt

$$\begin{aligned} \mathbf{C}(t) &= (1-t)^2 \mathbf{P}_0 + 2t(1-t) \mathbf{P}_1 + t^2 \mathbf{P}_2 \\ &= (1-t) \underbrace{\left((1-t) \mathbf{P}_0 + t \mathbf{P}_1 \right)}_{\text{linear}} + t \underbrace{\left((1-t) \mathbf{P}_1 + t \mathbf{P}_2 \right)}_{\text{linear}}. \end{aligned}$$

Somit kann $\mathbf{C}(t)$ als Linearkombination von zwei Bézierkurven 1-ten Grades bestimmt werden. Betrachten wir dies nun allgemeiner. Bezeichnen wir eine beliebige Bézierkurve n -ten Grades mit

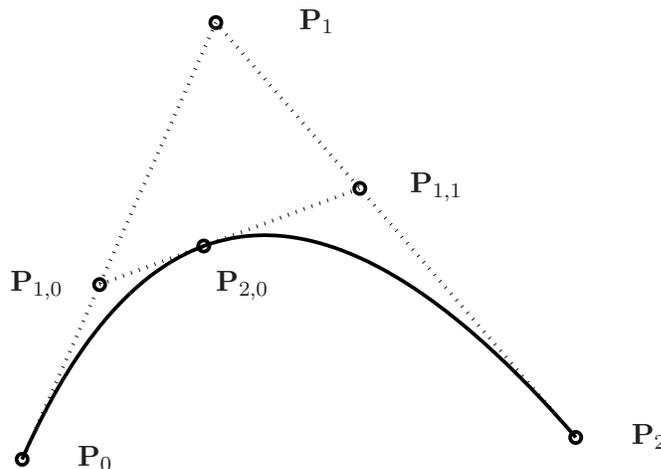


Abb. 4.4: Berechnung eines Punktes durch wiederholte lineare Interpolation für $t = 2/5$.

$\mathbf{C}_n(P_0, \dots, P_n)$, dann liefert uns die Rekursion 4.4.2.vi die Darstellung

$$\mathbf{C}_n(t : P_0, \dots, P_n) = (1-t) \mathbf{C}_{n-1}(t : P_0, \dots, P_{n-1}) + t \mathbf{C}_{n-1}(t : P_1, \dots, P_n).$$

Dies liefert ein rekursives Verfahren zur Bestimmung von $\mathbf{C}(t_0) = \mathbf{P}_{n,0}(t_0)$ auf einer Bézierkurve n -ten Grades, nämlich

$$\mathbf{P}_{k,j}(t_0) = (1-t_0) \mathbf{P}_{k-1,j}(t_0) + t_0 \mathbf{P}_{k-1,j+1}(t_0) \quad \text{für} \quad \begin{cases} k = 1, \dots, n \\ i = 0, \dots, n-k \end{cases} \quad (4.26)$$

Die Gleichung wird als **de Casteljau-Algorithmus** bezeichnet.

MATLAB-Funktion: deCasteljau1.m

```

1 function C = deCasteljau1(P,t)
2 % Compute point on a bezier curve using Casteljau
3 for k=1:size(P,2)-1
4     for i=1:size(P,2)-k
5         P(:,i) = (1-t)*P(:,i) + t*P(:,i+1);
6     end
7 end
8 C = P(:,1);

```

MATLAB-Beispiel:

Die folgenden Zeilen liefern das Kontrollpolygon und die Bézierkurve zu den Punkten $P_0 = (0,0)$, $P_1 = (1,-1)$, $P_2 = (3,4)$ und $P_3 = (3,3)$.

```

>> P=[0, 1, 2, 3;
>>     0 -1, 4, 3];
>> t = linspace(0,1,100);
>> for k=1:length(t)
>>     C(:,k)= deCasteljau1(P,t(k));
>> end
>> plot(C(1,:),C(2,:), 'k-',
>>       P(1,:),P(2,:), 'r*');

```

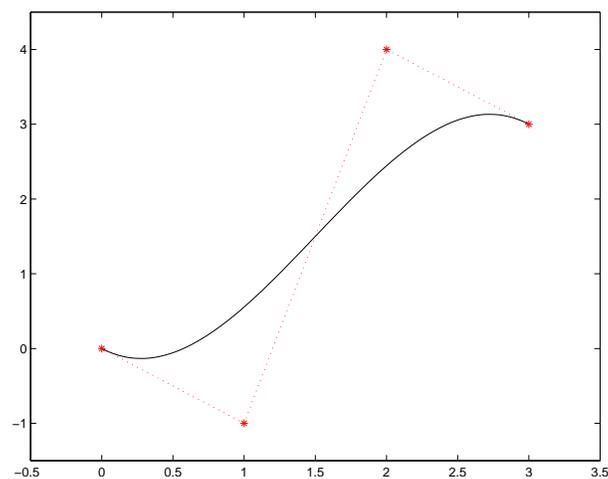


Abb. 4.5: Kontrollpolygon und Bézierkurve, Ergebnis des Matlab-Beispiels.

Kommen wir nochmals auf den Vergleich der Darstellungen zurück, d.h. die Koordinatenfunktionen $x(t)$, $y(t)$ und $z(t)$ sind Polynome in monomialer Basis oder in der Bernstein-Basis. Betrachten wir diesbezüglich einige Beispiele.

Beispiel 4.4.6 Es sei $n = 1$. Aus (4.24) erhalten wir $B_0^1(t) = 1 - t$ und $B_1^1(t) = t$. Die Darstellung (4.25) nimmt dann die Form $\mathbf{C}(t) = (1 - t)\mathbf{P}_0 + t\mathbf{P}_1$ an. Dies ist eine gerade Linie von \mathbf{P}_0 nach \mathbf{P}_1 .

Beispiel 4.4.7 Es sei $n = 2$. Aus (4.24) und (4.25) erhalten wir $C(t) = (1 - t)^2P_0 + 2t(1 - t)P_1 + t^2P_2$. Dies ist eine parabolische Kurve von P_0 nach P_2 (siehe Abb. 4.6 rechts). Man beachte, dass der Polygonzug mit den Punkten P_0, P_1, P_2 , sprich das Kontrollpolygon, die Form der Kurve näherungsweise gut approximiert. Für die Endpunkte gilt $P_0 = C(0)$ und $P_2 = C(1)$. Die tangentialen Richtungen an den Endpunkten sind parallel zu $P_1 - P_0$ und $P_2 - P_1$. Die Kurve liegt im Dreieck mit den Ecken P_0, P_1, P_2 .

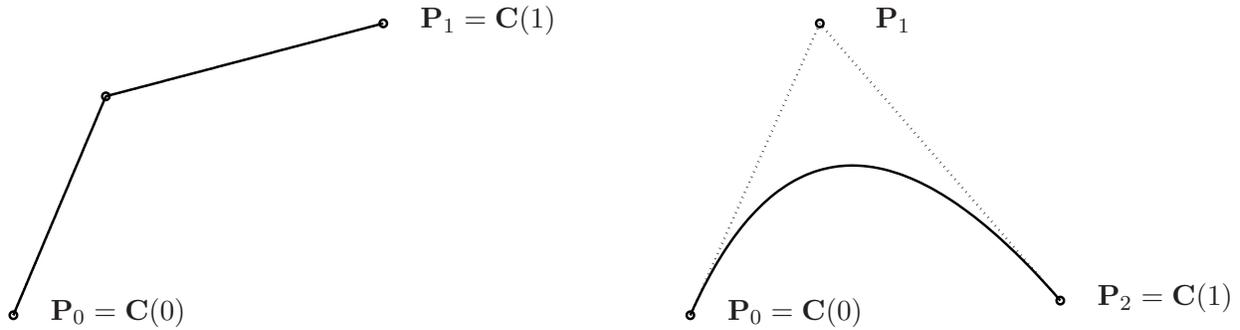


Abb. 4.6: Eine lineare (links) und quadratische (rechts) Bézierkurve.

Beispiel 4.4.8 Es sei $n = 3$. Wir erhalten $C(t) = (1 - t)^3P_0 + 3t(1 - t)^2P_1 + 3t^2(1 - t)P_2 + t^3P_3$. Beispiele kubischer Bézierkurven sind in Abb. 4.7 dargestellt. Man beachte, dass das Kontrollpolygon näherungsweise die Kurve beschreibt. Für die Endpunkte gilt $P_0 = C(0)$ und $P_3 = C(1)$. Die tangentialen Richtungen an den Endpunkten sind parallel zu $P_1 - P_0$ und $P_3 - P_2$. Die Kurve liegt in der konvexen Hülle der Punkte P_0, P_1, P_2, P_3 . Keine Gerade schneidet die Kurve häufiger als sie das Kontrollpolygon schneidet. Die Kurve krümmt sich bei $t = 0$ in die gleiche Richtung wie P_0, P_1, P_2 bzw. bei $t = 1$ wie P_1, P_2, P_3 .

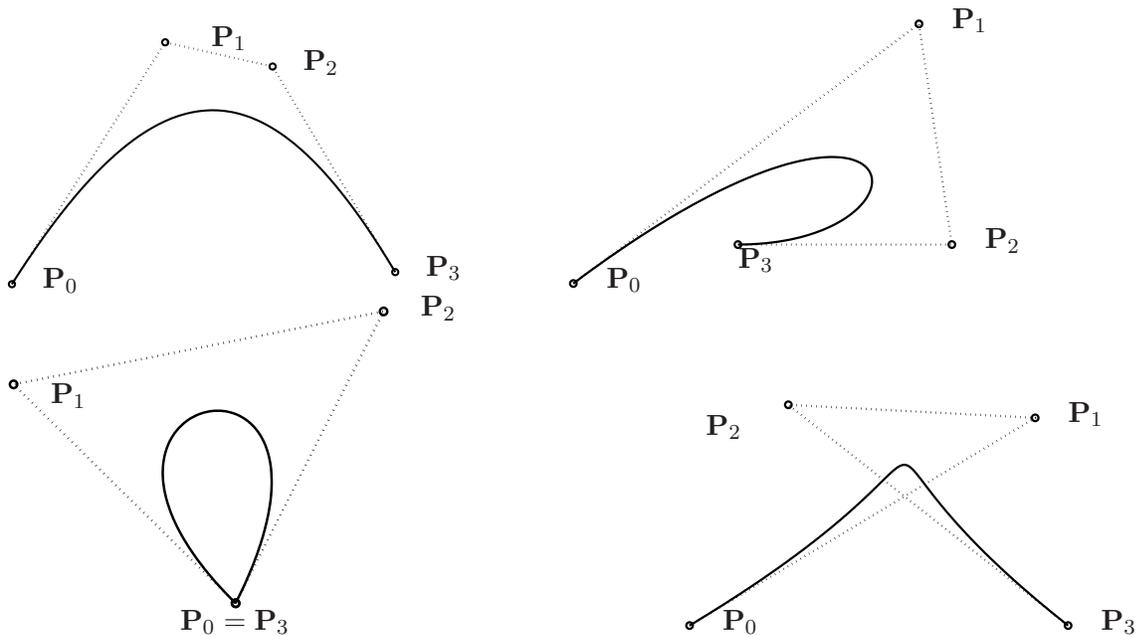


Abb. 4.7: Kubische Bézierkurve und zugehörige Kontrollpolygone.

Die Kurve $\mathbf{C}(t)$ ist eine vektorwertige Funktion in einer Variablen. Eine Fläche ist eine vektorwertige Funktion in zwei Parametern s und t und stellt die Abbildung eines Gebiets R von der st -Ebene in den 3-dimensionalen Raum dar, nämlich $\mathbf{S}(s, t) = (x(s, t), y(s, t), z(s, t))$, $(s, t) \in R$. Es gibt mehrere Möglichkeiten die Koordinatenfunktionen zu definieren. Der sicherlich einfachste und häufig verwendete Ansatz, ist der des Tensorprodukts, d.h.

$$\mathbf{S}(s, t) = (x(s, t), y(s, t), z(s, t)) = \sum_{i=0}^m \sum_{j=0}^n f_i(s) g_j(t) \mathbf{b}_{ij}$$

mit

$$\mathbf{b}_{ij} = (x_{ij}, y_{ij}, z_{ij}) \quad 0 \leq s, t \leq 1.$$

Man beachte, dass die Definitionsmenge dieser Abbildung das Quadrat $[0, 1]^2$ ist. Verwendet man als Basisfunktionen wieder die Bernstein-Polynome so erhält man

$$\mathbf{S}(s, t) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(s) B_j^n(t) \mathbf{P}_{ij} \quad 0 \leq s, t \leq 1. \quad (4.27)$$

Für ein festes s_0 gilt

$$\begin{aligned} \mathbf{C}_{s_0}(t) = \mathbf{S}(s_0, t) &= \sum_{i=0}^m \sum_{j=0}^n B_i^m(s_0) B_j^n(t) \mathbf{P}_{ij} \\ &= \sum_{j=0}^n B_j^n(t) \left(\sum_{i=0}^m B_i^m(s_0) \mathbf{P}_{ij} \right) \\ &= \sum_{j=0}^n B_j^n(t) \mathbf{Q}_j(s_0) \end{aligned} \quad (4.28)$$

wobei $\mathbf{Q}_j(s_0) = \sum_{i=0}^m B_i^m(s_0) \mathbf{P}_{ij}$, $j = 0, \dots, n$ eine Bézierkurve ist, die auf der Fläche liegt.

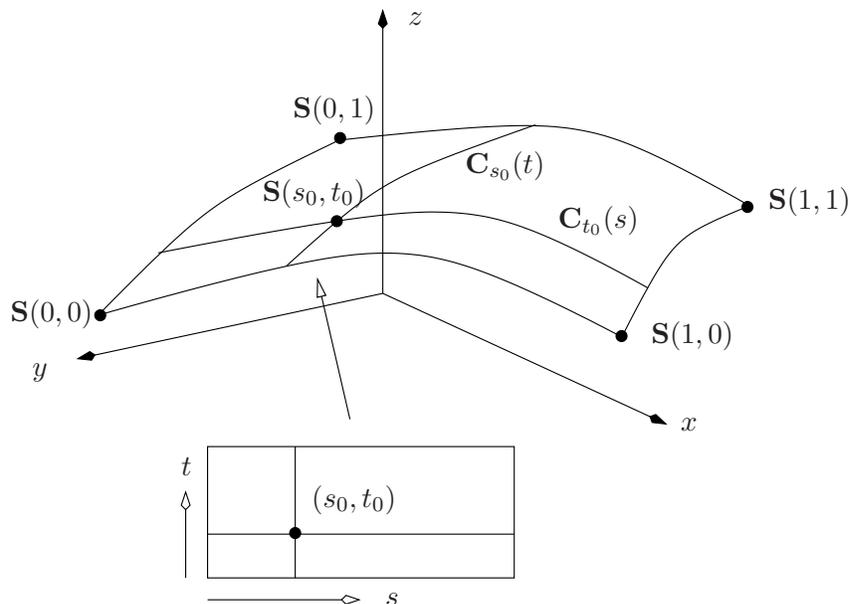
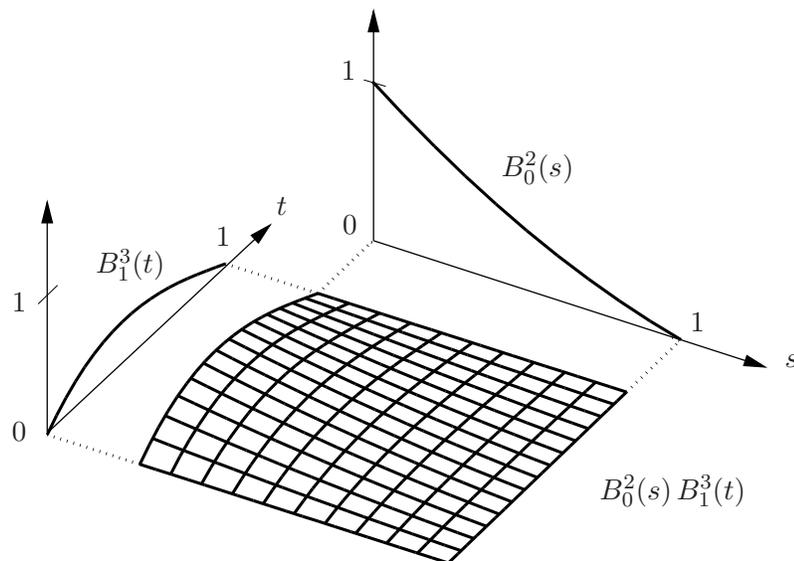


Abb. 4.8: Eine Tensorproduktfläche und isoparametrische Kurven.



Mittels (4.28) kann man also (4.27) zu gegebenen (s_0, t_0) durch mehrmaliges anwenden des eindimensionalen deCasteljau-Algorithmus bestimmen. Dieses Vorgehen ist in der Routine `deCasteljau2.m` realisiert.

MATLAB-Funktion: `deCasteljau2.m`

```

1 function S = deCasteljau2(P,s,t)
2 % Compute a point on a Bezier surface
3 n = size(P,2); m = size(P,3);
4 if n <= m
5     for j = 1:m
6         Q(:,j) = deCasteljau1(squeeze(P(:,:,j)),s);
7     end
8     S=deCasteljau1(Q,t);
9 else
10    for i = 1:n
11        Q(:,i)=deCasteljau1(squeeze(P(:,i,:)),t);
12    end
13    S=deCasteljau1(Q,s);
14 end

```

In Abb. 4.9 ist das Kontrollnetz und die Bézierfläche beispielhaft dargestellt.

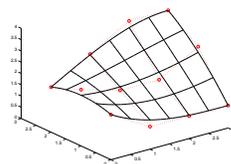


Abb. 4.9: Beispiel eines Kontrollnetz und Bézierfläche in \mathbb{R}^2 .

4.5 B-SPLINES

In Kapitel 4.1 haben wir uns mit den kubischen Splines beschäftigt und diese mit einem direkten Ansatz der Form

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i,$$

hergeleitet. Bis auf eine Verschiebung um x_i ist die ein monomialer Ansatz. Im Folgenden wollen wir uns mit einem allgemeineren¹ Ansatz beschäftigen, der durch die Anwendung in der Computergrafik motiviert ist. Wir werden eine weitere Basis einführen, deren Basisfunktionen einen Träger minimaler Länge haben (Monome haben ganz \mathbb{R} als Träger) und deren Elemente sich effektiv und numerisch stabil berechnen lassen. Wir erinnern daran, dass mit dem Träger einer Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ die Menge $\text{supp}(f) := \overline{\{x \in \mathbb{R}, f(x) \neq 0\}}$ bezeichnet wird, wobei der Strich den Abschluß der Menge bezeichnet. Splines, die in dieser Basis dargestellt werden, nennt man **B-Splines**.

Rekursive Definition der B-Splines-Basisfunktionen

Definition 4.5.1 (B-Splines-Basisfunktionen) Sei $\mathcal{T} = \{t_0, t_m\}$ eine nichtfallende Knotenfolge reeller Zahlen, d.h. $t_i \leq t_{i+1}$ ($i = 0, \dots, m-1$). Die t_i werden als **Knoten** und \mathcal{T} als **Knotenvektor** bezeichnet. Die i -te B-Spline Basisfunktion vom Grade p (Ordnung $p+1$) ist definiert für $p = 0$ als stückweise konstante Funktion der Form

$$N_j^0(t) := \begin{cases} 1 & , \text{ falls } t_j \leq t < t_{j+1} \\ 0 & , \text{ sonst} \end{cases} \quad (4.29)$$

und für $p > 0$ durch

$$N_j^p(t) := \frac{t - t_i}{t_{i+p} - t_i} N_i^{p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1}^{p-1}(t). \quad (4.30)$$

Bemerkung 4.5.2 (i) N_i^0 ist eine Treppenfunktion, die auf dem halboffenen Intervall $[t_j, t_{j+1})$ Eins ist und sonst verschwindet.

- (ii) Man beachte die rechtsseitige Stetigkeit in der Definition der N_j^0 , d.h. $\lim_{t \rightarrow t_{j+1}^+} N_j^0(t) = N_{j+1}^0(t_{j+1})$.
- (iii) Für $p > 0$ ist $N_i^p(t)$ eine Linearkombination von zwei Basisfunktionen vom Grade $(p-1)$.
- (iv) Die Berechnung einer Menge von Basisfunktionen erfordert einen Knotenvektor \mathcal{T} und einen Grad p .
- (v) In (4.30) kann der Nenner im Bruch Null werden; dieser Quotient sei per Definition Null.
- (vi) Die $N_i^p(t)$ sind stückweise polynomiale Funktionen auf der reellen Achse. Normalerweise ist nur das Intervall $[t_0, t_m]$ von Interesse.
- (vii) Das i -te Knotenintervall $[t_i, t_{i+1})$ kann die Länge Null haben, da aufeinanderfolgende Knoten nicht verschieden sein müssen.
- (viii) Die Berechnung der Basisfunktionen kann in dem bekannten Dreiecksschema erfolgen.
- (ix) Die N_j^0 liefern auf dem Intervall $[t_0, t_m)$ eine Zerlegung der Eins und sind positiv.

Die rekursive Definition der B-Splines (4.30) liefert einen einfachen Algorithmus, um Werte der B-Splines zu einem gegebenen $t \in [t_j, t_{j+1})$ zu bestimmen.

¹Allgemeiner bzgl. des Polynomgrads auf jedem Teilintervall, als auch der Glattheitsanforderung an den Knoten zwischen den Teilintervallen, bzw. an den beiden Endpunkten.

Beispiel 4.5.3 Es sei $\mathcal{T} = \{t_0 = 0, t_1 = 0, t_2 = 0, t_3 = 1, t_4 = 1, t_5 = 1\}$ und $p = 2$. Für B-Spline-Basisfunktionen vom Grade 0, 1 und 2 lauten dann

$$\begin{aligned}
 N_0^0 &= N_1^0 = 0 & -\infty < t < \infty \\
 N_2^0 &= \begin{cases} 1 & 0 \leq t < 1 \\ 0 & \text{sonst} \end{cases} \\
 N_3^0 &= N_4^0 = 0 & -\infty < t < \infty \\
 N_0^1 &= \frac{t-0}{0-0}N_0^0 + \frac{0-t}{0-0}N_1^0 = 0 & -\infty < t < \infty \\
 N_1^1 &= \frac{t-0}{0-0}N_1^0 + \frac{1-t}{1-0}N_2^0 = \begin{cases} 1-t & 0 \leq t < 1 \\ 0 & \text{sonst} \end{cases} \\
 N_2^1 &= \frac{t-0}{1-0}N_2^0 + \frac{1-t}{1-1}N_3^0 = \begin{cases} t & 0 \leq t < 1 \\ 0 & \text{sonst} \end{cases} \\
 N_3^1 &= \frac{t-1}{1-1}N_3^0 + \frac{1-t}{1-1}N_4^0 = 0 & -\infty < t < \infty \\
 N_0^2 &= \frac{t-0}{0-0}N_0^1 + \frac{1-t}{1-0}N_1^1 = \begin{cases} (1-t)^2 & 0 \leq t < 1 \\ 0 & \text{sonst} \end{cases} \\
 N_1^2 &= \frac{t-0}{1-0}N_1^1 + \frac{1-t}{1-0}N_2^1 = \begin{cases} 2t(1-t) & 0 \leq t < 1 \\ 0 & \text{sonst} \end{cases} \\
 N_2^2 &= \frac{t-0}{1-0}N_2^1 + \frac{1-t}{1-1}N_3^1 = \begin{cases} t^2 & 0 \leq t < 1 \\ 0 & \text{sonst} \end{cases}
 \end{aligned}$$

Man beachte, dass N_i^2 auf das Intervall $[0, 1)$ restringiert gerade die quadratischen Bernstein-Polynome sind. Aus diesem Grunde ist die B-Spline-Darstellung mit einem Knotenvektor der Form

$$U = \{ \underbrace{0, \dots, 0}_{(p+1)\text{-mal}}, \underbrace{1, \dots, 1}_{(p+1)\text{-mal}} \}$$

eine Verallgemeinerung der Bézier-Darstellung ist.

Beispiel 4.5.4 Es sei $\mathcal{T} = \{t_0 = t_1 = t_2 = 0, t_3 = 1, t_4 = 2, t_5 = 3, t_6 = t_7 = 4, t_8 = t_9 = t_{10} = 5\}$ und $p = 2$. Für stückweise konstanten, linearen und quadratischen B-Spline-Basisfunktionen lauten dann:

$$\begin{aligned}
 N_0^0 &= N_1^0 = 0 & -\infty < t < \infty \\
 N_2^0 &= \begin{cases} 1 & 0 \leq t < 1 \\ 0 & \text{sonst} \end{cases} \\
 N_3^0 &= \begin{cases} 1 & 1 \leq t < 2 \\ 0 & \text{sonst} \end{cases} \\
 N_4^0 &= \begin{cases} 1 & 2 \leq t < 3 \\ 0 & \text{sonst} \end{cases} \\
 N_5^0 &= \begin{cases} 1 & 3 \leq t < 4 \\ 0 & \text{sonst} \end{cases} \\
 N_6^0 &= 0 & -\infty < t < \infty \\
 N_7^0 &= \begin{cases} 1 & 4 \leq t < 5 \\ 0 & \text{sonst} \end{cases} \\
 N_8^0 &= N_9^0 = 0 & -\infty < t < \infty
 \end{aligned}$$

$$\begin{aligned}
N_0^1 &= \frac{t-0}{0-0}N_0^0 + \frac{0-t}{0-0}N_1^0 = 0 & -\infty < t < \infty \\
N_1^1 &= \frac{t-0}{0-0}N_1^0 + \frac{1-t}{1-0}N_2^0 = \begin{cases} 1-t & 0 \leq t < 1 \\ 0 & \text{sonst} \end{cases} \\
N_2^1 &= \frac{t-0}{1-0}N_2^0 + \frac{2-t}{2-1}N_3^0 = \begin{cases} t & 0 \leq t < 1 \\ 2-t & 1 \leq t < 2 \\ 0 & \text{sonst} \end{cases} \\
N_3^1 &= \frac{t-1}{2-1}N_3^0 + \frac{3-t}{3-2}N_4^0 = \begin{cases} t-1 & 1 \leq t < 2 \\ 3-t & 2 \leq t < 3 \\ 0 & \text{sonst} \end{cases} \\
N_4^1 &= \frac{t-2}{3-2}N_4^0 + \frac{4-t}{4-3}N_5^0 = \begin{cases} t-2 & 2 \leq t < 3 \\ 4-t & 3 \leq t < 4 \\ 0 & \text{sonst} \end{cases} \\
N_5^1 &= \frac{t-3}{4-3}N_5^0 + \frac{4-t}{4-4}N_6^0 = \begin{cases} t-3 & 3 \leq t < 4 \\ 0 & \text{sonst} \end{cases} \\
N_6^1 &= \frac{t-4}{4-4}N_6^0 + \frac{5-t}{5-4}N_7^0 = \begin{cases} 5-t & 4 \leq t < 5 \\ 0 & \text{sonst} \end{cases} \\
N_7^1 &= \frac{t-4}{5-4}N_7^0 + \frac{5-t}{5-5}N_8^0 = \begin{cases} t-4 & 4 \leq t < 5 \\ 0 & \text{sonst} \end{cases} \\
N_8^1 &= \frac{t-5}{5-5}N_8^0 + \frac{5-t}{5-5}N_9^0 = 0 & -\infty < t < \infty
\end{aligned}$$

Die folgenden N_i^2 sind bis auf die angegebenen Intervalle jeweils Null.

$$\begin{aligned}
N_0^2 &= \frac{t-0}{0-0}N_0^1 + \frac{1-t}{1-0}N_1^1 = (1-t)^2 & 0 \leq t < 1 \\
N_1^2 &= \frac{t-0}{1-0}N_1^1 + \frac{2-t}{2-0}N_2^1 = \begin{cases} 2t - \frac{3}{2}t^2 & 0 \leq t < 1 \\ \frac{1}{2}(2-t)^2 & 1 \leq t < 2 \end{cases} \\
N_2^2 &= \frac{t-0}{2-0}N_2^1 + \frac{3-t}{3-1}N_3^1 = \begin{cases} \frac{1}{2}t^2 & 0 \leq t < 1 \\ -\frac{3}{2} + 3t - t^2 & 1 \leq t < 2 \\ \frac{1}{2}(3-t)^2 & 2 \leq t < 3 \end{cases} \\
N_3^2 &= \frac{t-1}{3-1}N_3^1 + \frac{4-t}{4-2}N_4^1 = \begin{cases} \frac{1}{2}(t-1)^2 & 1 \leq t < 2 \\ -\frac{11}{2} + 5t - t^2 & 2 \leq t < 3 \\ \frac{1}{2}(4-t)^2 & 3 \leq t < 4 \end{cases} \\
N_4^2 &= \frac{t-2}{4-2}N_4^1 + \frac{4-t}{4-3}N_5^1 = \begin{cases} \frac{1}{2}(t-2)^2 & 2 \leq t < 3 \\ -16 + 10t - \frac{3}{2}t^2 & 3 \leq t < 4 \end{cases} \\
N_5^2 &= \frac{t-3}{4-3}N_5^1 + \frac{5-t}{5-4}N_6^1 = \begin{cases} (t-3)^2 & 3 \leq t < 4 \\ (5-t)^2 & 4 \leq t < 5 \end{cases} \\
N_6^1 &= \frac{t-4}{5-4}N_6^1 + \frac{5-t}{5-4}N_7^1 = 2(t-4)(5-t) & 4 \leq t < 5 \\
N_7^1 &= \frac{t-4}{5-4}N_7^1 + \frac{5-t}{5-5}N_8^1 = (t-4)^2 & 4 \leq t < 5
\end{aligned}$$

In Abbildung 4.10 ist die Zusammensetzung von N_3^2 aus den jeweiligen stückweise polynomialen Funktionen auf den einzelnen Teilintervallen grafisch dargestellt.

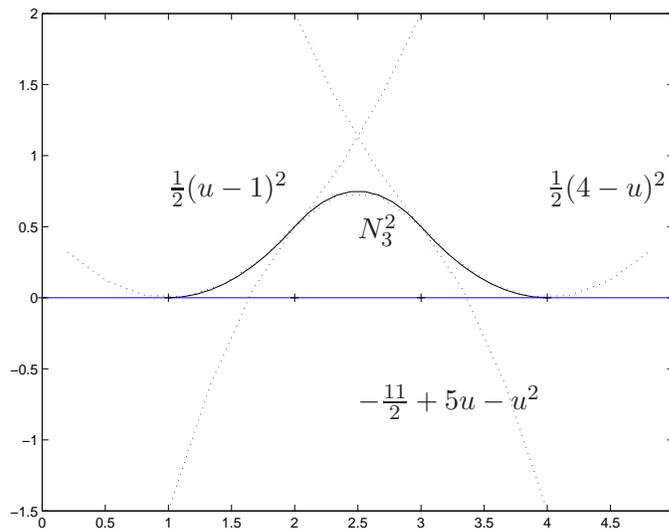


Abb. 4.10: Die Zerlegung von N_3^2 in seine stückweise polynomialen Teilfunktionen.

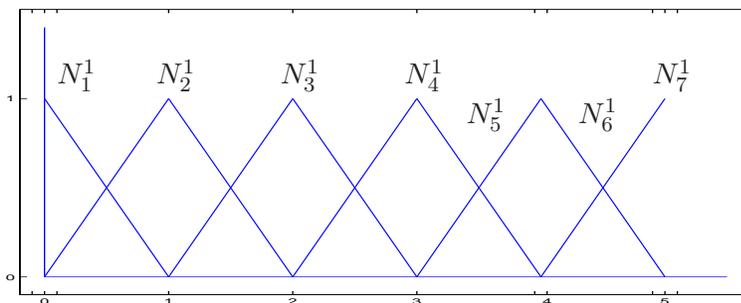


Abb. 4.11: Die stückweise linearen Basisfunktionen zu $\mathcal{T} = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$.

Es ist wichtig, den Effekt von mehrfachen Knoten zu verstehen. Man betrachte die Funktionen $N_0^2, N_1^2, N_2^2, N_5^2$ und N_6^2 in Abbildung 4.12. Beachtet man die rekursive Definition der Basisfunktionen (4.30), so stellt man fest, dass sie jeweils nur von 4 Knoten abhängen, nämlich:

$$\begin{aligned} N_0^2 &: \{0, 0, 0, 1\} \\ N_1^2 &: \{0, 0, 1, 2\} \\ N_2^2 &: \{0, 1, 2, 3\} \\ N_5^2 &: \{3, 4, 4, 5\} \\ N_6^2 &: \{4, 4, 5, 5\} \end{aligned}$$

Der Begriff Vielfachheit eines Knotens, kann man verstehen

- in Bezug auf eine Knoten im Knotenvektor oder
- in Bezug auf einen Knoten bezüglich einer Basisfunktion.

Zum Beispiel hat $t = 0$ die Vielfachheit 3 im o.g. Knotenvektor \mathcal{T} , aber in Bezug auf die Basisfunktion N_1^2 ist $t = 0$ ein Knoten mit der Vielfachheit 2. Die Basisfunktionen sind stückweise polynomiale Funktionen, d.h. im Inneren der Intervalle (t_j, t_{j+1}) sind sie beliebig glatt. Unstetigkeiten können also nur an den Knoten auftreten. Für $t = 0$ stellt man fest, dass N_0^2 unstetig ist,

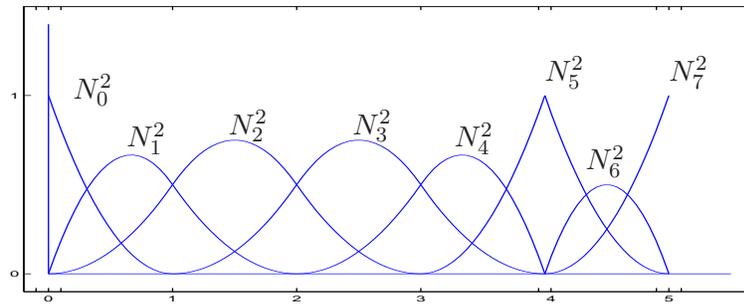


Abb. 4.12: Die stückweise quadratischen Basisfunktionen zu $\mathcal{T} = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$.

N_1^2 C^0 stetig ist, N_2^2 C^1 stetig ist und N_5^2 und all seine Ableitungen dort Null von beiden Seiten ist. N_1^2 sieht $t = 0$ als doppelten Knoten, N_2^2 sieht $t = 0$ als einfachen Knoten und N_5^2 enthält $t = 0$ gar nicht als Knoten.

Satz 4.5.5 Ist t_ℓ ein m -facher Knoten, d.h.

$$t_{\ell-1} < t_\ell = \dots = t_{\ell+m-1} < t_{\ell+m},$$

so ist N_j^k an der Stelle t_ℓ mindestens $(k - 1 - m)$ -mal stetig differenzierbar. Für die Ableitung von N_j^k gilt

$$\frac{d}{dt} N_j^k(t) = (k-1) \cdot \left(\frac{N_j^{k-1}(t)}{t_{j+k-1} - x_j} - \frac{N_{j+k}^{k-1}(t)}{t_{j+k} - t_{j+1}} \right)$$

Beweis. Der Beweis dieses Satzes wird an späterer Stelle erbracht □

Satz 4.5.6 (Marsdens² Identität) Bei gegebener Knotenfolge $\mathcal{T} := \{t_j\}, j \in \mathbb{Z}$, mit

$$\lim_{j \rightarrow \pm\infty} t_j = \pm\infty$$

sei für beliebiges $y \in \mathbb{R}$

$$\psi_{j1}(y) := 1, \quad \psi_{jk}(y) := (x_{j+1} - y)(x_{j+2} - y) \cdots (x_{j+k-1} - y), \quad k > 1.$$

Dann gilt

$$(x - y)^{k-1} = \sum_{j \in \mathbb{Z}} \psi_{jk}(y) B_{jk}(x)$$

Beweis. Sei $(a_j)_{j \in \mathbb{Z}}$ eine beliebige Folge reeller Zahlen. Dann folgt aus der Rekursion (??)

$$\sum_{j \in \mathbb{Z}} a_j B_{jk} = \sum_{j \in \mathbb{Z}} (a_{j-1}(1 - \omega_{jk} + a_j \omega_{jk}) B_{j,k-1})$$

Setzen wir $a_j := \psi_{jk}(x)$, so folgt

$$\begin{aligned} & a_{j-1}(1 - \omega_{jk}(x)) + a_k \omega_{jk}(x) \\ &= \left((x_j - y)(1 - \omega_{jk}(x)) + (x_{j+k-1} - y)\omega_{jk}(x) \right) \psi_{j,k-1}(y) \\ &= (x - y) \psi_{j,k-1}(y). \end{aligned}$$

⁵benannt nach M. J. Marsden, vlg. hierzu [Marsden]

Damit gewinnen wir die Gleichung

$$\sum_{j \in \mathbb{Z}} \psi_{jk}(y) B_{jk}(x) = (x - y) \sum_{j \in \mathbb{Z}} \psi_{j,k-1}(y) B_{j,k-1}(x)$$

und somit mithilfe der Definitionen der ψ_{jk} und B_{j1}

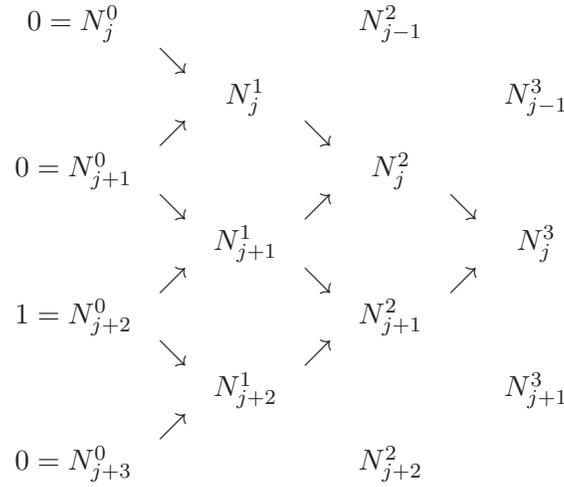
$$\sum_{j \in \mathbb{Z}} \psi_{jk} B_{jk}(x) = (x - y)^{k-1} \sum_{j \in \mathbb{Z}} \psi_{j,1}(y) B_{j,1}(x) = (x - y)^{k-1}$$

□

Bemerkung 4.5.7 Da y im obigen Satz beliebig war, haben wir also gezeigt, dass $\mathbb{P}_{k-1} \subset \mathcal{S}_{k,t}$.

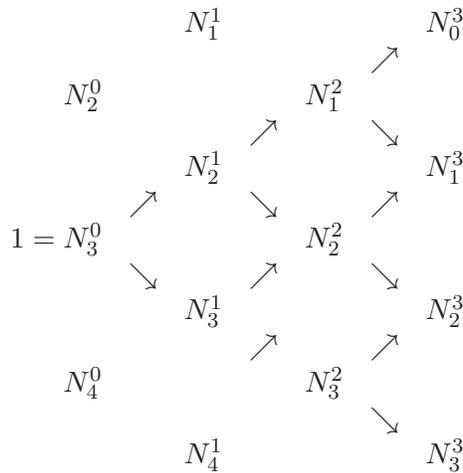
Effiziente Auswertung der B-Spline-Basisfunktionen

Die Funktion N_j^3 ist eine Linearkombination der Funktionen $N_j^0, N_{j+1}^0, N_{j+2}^0$ und N_{j+1}^1 . Somit ist N_j^3 nur von Null verschieden für $t \in [t_j, t_{j+4})$.



Tab. 4.3: N_j^3 ist nur auf dem Intervall $[t_j, t_{j+4})$ von Null verschieden.

In jedem Knotenintervall $[t_j, t_{j+1})$ sind maximal $p + 1$ der N_i^p von Null verschieden, nämlich N_{j-p}^p, \dots, N_j^p . Auf $[t_3, t_4)$ ist z.B. N_3^0 die einzige nichtverschwindende Basisfunktion vom Grad Null. Somit sind N_0^3, \dots, N_3^3 die einzigen von Null verschiedenen kubischen Funktionen auf $[t_3, t_4)$. Diese Eigenschaft ist in der folgenden Grafik 4.5 dargestellt.



Tab. 4.4: N_3^0 ist nur auf dem Intervall $[t_3, t_4)$ von Null verschieden, Somit sind auch N_0^3, \dots, N_3^3 die einzigen von Null verschiedenen kubischen Funktionen auf $[t_3, t_4)$.

MATLAB-Funktion: BasisFunc.m

```
1 function N = BasisFunc(i,p,U,t)
2 % compute the nonvanishing basis functions
3 N = zeros(p+1,1);
4 N(1) = 1;
5 for j=1:p
6     left(j) = t - U(i+1-j);
7     right(j) = U(i+j) - t;
8     saved = 0;
9     for r=1:j
10        temp = N(r)/(right(r)+left(j-r+1));
11        N(r) = saved + right(r) * temp;
12        saved = left(j-r+1) * temp;
13    end
14    N(j+1) = saved;
15 end
```

MATLAB-Funktion: FindSpan.m

```
1 function mid = FindSpan(p,U,t)
2 % returns the knot span index
3 n = length(U)-p;
4 if t==U(end) % special case
5     mid = n-1;
6     return
7 end
8 low = p;
9 high = n;
10 mid = floor((low+high)/2);
11 while t<U(mid) | t>= U(mid+1)
12     if t<U(mid)
13         high = mid;
14     else
15         low = mid;
16     end
17     mid = floor((low+high)/2);
18 end
```

MATLAB-Funktion: CurvePoint.m

```
1 function value = CurvePoint(p,U,P,t)
2 % compute point on B-spline curve
3 span = FindSpan(p,U,t);
4 B = BasisFunc(span,p,U,t);
```

```

5 value = 0*P(:,1);
6 for i=0:p
7     value = value + B(i+1) * P(:,span-p+i);
8 end

```

Ableitung der B-Splines

MATLAB-Funktion: AllBasisFunc.m

```

1 function N = AllBasisFunc(i,p,U,t)
2 % compute the nonvanishing basis functions
3 N = zeros(p+1,p+1);
4 N(1,1) = 1;
5 for j=1:p
6     left(j) = t - U(i+1-j);
7     right(j) = U(i+j) - t;
8     saved = 0;
9     for r=1:j
10        temp = N(r,j)/(right(r)+left(j-r+1));
11        N(r,j+1) = saved + right(r) * temp;
12        saved = left(j-r+1) * temp;
13    end
14    N(j+1,j+1) = saved;
15 end

```

MATLAB-Funktion: CurveDerivPts.m

```

1 function PK = CurveDerivPts(p,U,P,d,r1,r2)
2 % compute control points of curve derivatives
3 r = r2-r1;
4 for i=1:r+1
5     PK(:,1,i)=P(:,r1+i);
6 end
7 for k=2:d+1
8     tmp = p-k+2;
9     for i=1:r-k+2
10        PK(:,k,i)=tmp*(PK(:,k-1,i+1)-PK(:,k-1,i))/(U(r1+i+p+1)-U(r1+i+k-1));
11    end
12 end

```

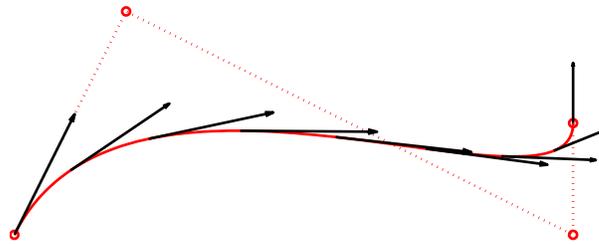


Abb. 4.13: Ergebnis der Matlab-Zeilen.

MATLAB-Funktion: CurveDerivs.m

```

1 function CK = CurveDerivs(p,U,P,t,d)
2 % Compute curve derivatives
3 du = min(d,p);
4 CK(1:size(P,1),[p+2:d+1]) = 0;
5 span = FindSpan(p,U,t);
6 N = AllBasisFunc(span,p,U,t);
7
8 PK = CurveDerivPts(p,U,P,du,span-p-1,span-1);
9
10 for k=1:du+1
11   CK(:,k) = 0;
12   for j=1:p-k+2
13     CK(:,k) = CK(:,k) + N(j,p-k+2)*PK(:,k,j);
14   end
15 end

```

MATLAB-Beispiel:

Die folgenden Zeilen stellen die Bernstein-Polynome B_0^8, \dots, B_8^8 graphisch dar.

```

>> P = [0,1,5,5;
        0,2,0,1];
>> U = [0,0,0,0,1,1,1,1]; p=3;
>> s = linspace(U(1),U(end),201);
>> for k = 1:length(s)
        C(:,k) = CurvePoint(p,U,P,s(k));
    end
>> plot(C(1,:),C(2,:),'-', ...
        P(1,:),P(2,:),'o:');
>> hold on
>> for j = 1 :25:length(s)
        V = CurveDerivs(p,U,P,s(j),1);
        quiver(V(1,1),V(2,1), ...
              0.2*V(1,2),0.2*V(2,2))
    end

```

4.6 RATIONALE B-SPLINES

MATLAB-Funktion: RatCurvePoint.m

```

1 function value = RatCurvePoint(p,U,Pw,t)
2 % compute point on B-spline curve
3 span = FindSpan(p,U,t);
4 B = BasisFunc(span,p,U,t);
5 value = 0*Pw(:,1);
6 for i=0:p
7     value = value + B(i+1) * Pw(:,span-p+i);
8 end
9 value = value(1:end-1)/value(end);

```

MATLAB-Funktion: RatCurveDerivs.m

```

1 function CK = RatCurveDerivs(p,U,Pw,t,d)
2 % compute derivatives on rational B-spline curve
3 du = min(d,p);
4
5 ders = CurveDerivs(p,U,Pw,t,d);
6 Aders = ders(1:end-1,:);
7 wders = ders(end,:);
8
9 CK(1:size(Aders,1),p+2:d+1) = 0;
10
11 for k=1:du+1
12     v = Aders(:,k);
13     for i=1:k-1
14         v = v - nchoosek(k-1,i)*wders(i+1)*CK(:,k-i);
15     end
16     CK(:,k) = v/wders(1);
17 end

```

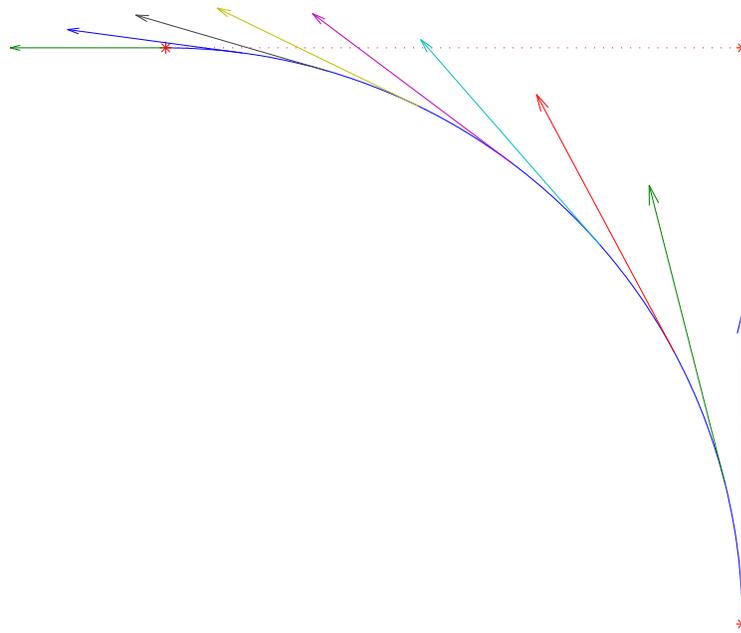


Abb. 4.14: Ergebnis der Matlab-Zeilen.

MATLAB-Beispiel:

Die folgenden Zeilen stellen die Bernstein-Polynome B_0^8, \dots, B_8^8 graphisch dar.

```
>> w = [1, 1, 2];
>> P = [1, 1, 0;
        0 1, 1];
>> U = [0,0,0,1,1,1]; p=2;
>> s = linspace(0,1,201);
>> Pw = [P(1,:).*w;P(2,:).*w/w];
>> for k = 1:length(s)
    Cw(:,k) = RatCurvePoint(p,U,Pw,s(k));
end
>> plot(Cw(1,:),Cw(2,:),'-', ...
        P(1,:),P(2,:),'*:');
>> hold on
>> for j = 1:25:length(s)
    CK = RatCurveDerivs(p,U,Pw,s(j),1);
    quiver(CK(1,1),CK(2,1),0.3*CK(1,2)
           ,0.3*CK(2,2))
end
```

4.7 GRUNDLEGENDE ALGORITHMEN

MATLAB-Funktion: CurveKnotIns.m

```

1 function [U,Q] = CurveKnotIns(p,U,P,t,k,s,r)
2 if p < s+r, Q=P; return, end
3 % compute new curve from knot insertion
4 np = length(U)-p-1;
5 % unaltered control points
6 Q = P(:,1:k-p);
7 Q(:,k-s+r:np+r)=P(:,k-s:np);
8 R = P(:,k-p:k-s);
9 for j=1:r % insert new knot r times
10     L=k-p+j-1;
11     for i=1:p-j-s+1
12         alpha = (t-U(L+i))/(U(i+k)-U(L+i));
13         R(:,i) = alpha*R(:,i+1)+(1-alpha)*R(:,i);
14     end
15     Q(:,L+1) = R(:,1);
16     Q(:,k+r-j-s)= R(:,p-j-s+1);
17 end
18 %copy remaining control points
19 Q(:,L+2:k-s)= R(:,2:k-s-L);
20 % new knot vevtor
21 U = [U(1:k),t*ones(1,r),U(k+1:end)];

```

MATLAB-Funktion: CurveSplit.m

```

1 function [U1,P1,U2,P2] = CurveSplit(p,U,P,t)
2 if t==U(1)
3     U1=[];P1=[];U2=U;P2=P;return
4 elseif t==U(end)
5     U1=U;P1=P;U2=[];P2=[];return
6 end
7 k = FindSpan(p,U,t);
8 s = sum((t==U));
9 if p>s
10     [U,P]=CurveKnotIns(p,U,P,t,k,s,p-s);
11 end
12 U1 = U([1:k+p-s,k+p-s])-U(1);
13 U1 = U1/max(U1); P1 = P(:,1:k);
14 U2 = U([k+1-s,k+1-s:end])-U(k+1-s);
15 U2 = U2/max(U2); P2 = P(:,k-s:end);

```

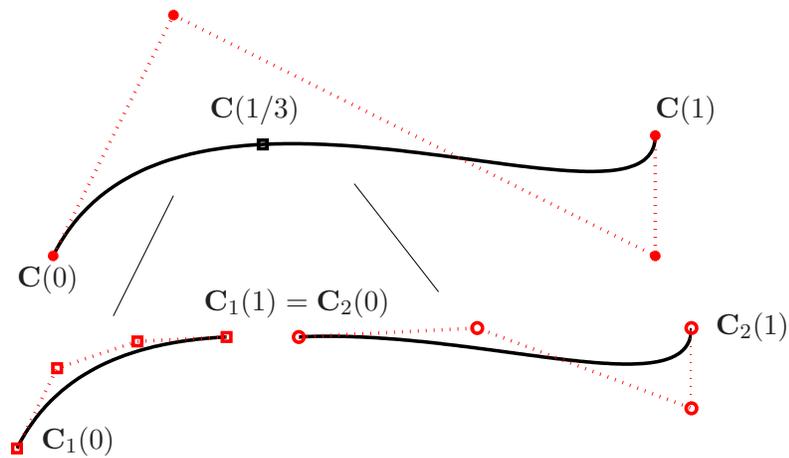


Abb. 4.15: Ergebnis der Matlab-Zeilen.

MATLAB-Beispiel:

Die folgenden Zeilen stellen die Bernstein-Polynome B_0^8, \dots, B_8^8 graphisch dar.

```
>> P = [0,1,5,5;
        0,2,0,1];
>> U = [0,0,0,0,1,1,1,1];
>> p = 3;
>> t = 1/3;
>> [U1,P1, U2, P2] = CurveSplit(p,U,P,t)
U1 =
    0    0    0    0    1    1    1    1
P1 =
         0  0.3333  1.0000  1.7407
         0  0.6667  0.8889  0.9259
U2 =
    0    0    0    0    1    1    1    1
P2 =
    1.7407  3.2222  5.0000  5.0000
    0.9259  1.0000  0.3333  1.0000
```

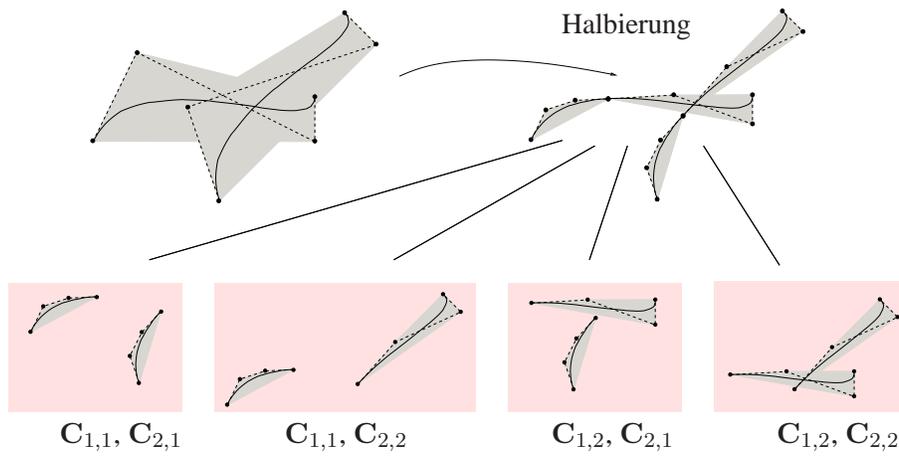


Abb. 4.16: Testen auf gemeinsamen Schnitt der einzelnen konvexen Hüllen nach Halbierung der einzelnen Bézierkurven $C_1 = C_{1,1} \cup C_{1,2}$ und $C_2 = C_{2,1} \cup C_{2,2}$.

MATLAB-Funktion: isLine.m

```

1 function [flag,P0,P1] = isLine(P,tol1,tol2)
2 S = mean(P,2);
3 PmS = P-S*ones(1,size(P,2));
4 [j,k] = max(sum(abs(PmS)));
5 rot = GivensRotMat(PmS(1,k),PmS(2,k));
6 Q = rot'*([PmS(1,:);PmS(2,:)]);
7 h = max(Q,[],2)-min(Q,[],2);
8 if h(2) <= max(tol1,tol2*h(1))
9     flag = 1;
10    P0 = S + rot*[h(1);0]/2;
11    P1 = S + rot*[-h(1);0]/2;
12 else
13     flag = 0; P0=[];P1=[];
14 end

```

MATLAB-Funktion: LineIntersect.m

```

1 function [flag,s] = LineIntersect(x,y)
2 % check for intersection of two line segments
3 % 0 no intersection, 1 one or more intersection points
4 dx = x(:,2)-x(:,1); mx = norm(dx);
5 dy = y(:,2)-y(:,1); my = norm(dy);
6 dw = (y(:,2)+y(:,1)-x(:,2)-x(:,1)); mw = norm(dw);
7 flag = 1;s=[];
8 if abs(det([dx,dy])) >= 1e-12 *mx*my % check for non parallel
9     if ~sum(abs([dx,dy]\dw)>1)
10        t = [dx,dy]\dw;
11        s = ((x(:,2)+x(:,1))+t(1)*dx)/2;
12        return
13    end
14 elseif det([dx,dw]) < max(1e+10*realmin,1e-12*mx*mw) % on common
    line
15    mm = [((x(:,2)+x(:,1))/2)*[1,1],((y(:,2)+y(:,1))/2)*[1,1]];
16    dd = [dx,dx,dy,dy];
17    xx = [dw+dy,dw-dy,-dw+dx,-dw+dx];
18    for j=1:4
19        t = xx(:,j)'*dd(:,j)/(dd(:,j)'*dd(:,j));
20        if abs(t)<=1
21            s = mm(:,j)+t/2*dd(:,j);
22            return
23        end
24    end
25 end
26 flag = 0;

```

MATLAB-Funktion: comConvHull.m

```

1 function flag = comConvHull(xy,st)
2 if comBoundingBoxes(min(xy,[],2),max(xy,[],2), ...
3                     min(st,[],2),max(st,[],2))
4     flag = 1;
5
6     if inConvHull(xy,mean(st(:,1:end-1),2)), return, end
7     if inConvHull(st,mean(xy(:,1:end-1),2)), return, end
8
9     for j = 1:size(xy,2)-1
10        for k = 1:size(st,2)-1
11            if LineIntersect(xy(:,j:j+1),st(:,k:k+1))
12                return
13            end
14        end
15    end
16 end
17 flag = 0;

```

MATLAB-Funktion: CurveBisection.m

```

1 function points = CurveBisection(p1,U1,Q1,p2,U2,Q2,tol)
2 [flag1,a1,b1] = isLine(Q1,tol(1),tol(2));
3 [flag2,a2,b2] = isLine(Q2,tol(1),tol(2));
4 if flag1 && flag2 % both segments are lines
5     [flag,points] = LineIntersect([a1,b1],[a2,b2]);
6 else
7     xy = myConvHull(Q1,flag1,a1,b1);
8     st = myConvHull(Q2,flag2,a2,b2);
9     points = [];
10    if comConvHull(xy,st) % intersection of convex hulls non empty
11        [U11,Q11,U21,Q21] = CurveSplit(p1,U1,Q1,(max(U1)-min(U1))/2);
12        [U12,Q12,U22,Q22] = CurveSplit(p2,U2,Q2,(max(U2)-min(U2))/2);
13        points = [points,CurveBisection(p1,U11,Q11,p2,U12,Q12,tol)];
14        points = [points,CurveBisection(p1,U11,Q11,p2,U22,Q22,tol)];
15        points = [points,CurveBisection(p1,U21,Q21,p2,U12,Q12,tol)];
16        points = [points,CurveBisection(p1,U21,Q21,p2,U22,Q22,tol)];
17    end
18 end
19
20 function h = myConvHull(Q,flag,a,b)
21 % Q
22 % [flag,a,b] = isLine(Q,1e-7,1e-7)
23
24 if flag
25     h = [a,b,a];
26 else
27     h = Q(:,convhull(Q(1,:),Q(2,:)));
28 end

```

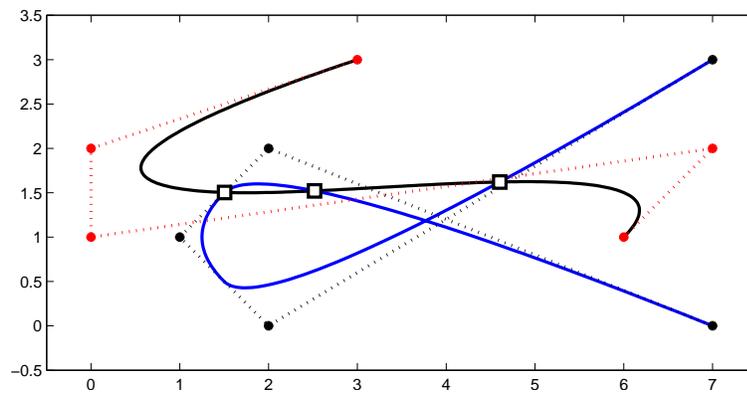


Abb. 4.17: Ergebnis der Matlab-Zeilen.

MATLAB-Beispiel:

Die folgenden Zeilen stellen die Bernstein-Polynome B_0^8, \dots, B_8^8 graphisch dar.

```
>> P1=[3 0,0,7,6;
      3,2,1,2,1]; p1=3;
>> U1 = [0,0,0,0,1/2,1,1,1,1];
>> P2 = [7,2,1,2,7;
      3,0,1,2,0]; p2=2;
>> U2 = [0,0,0,1/3,2/3,1,1,1];
>> CurveBisection(p1,U1,P1,p2,U2,P2, ...
      [1e-9,1e-7])
ans =
    1.5038    4.6028    2.5178
    1.5037    1.6214    1.5214
```

4.8 B-SPLINE INTERPOLATION

5 QUADRATURTHEORIE

Die Quadraturtheorie behandelt die Prinzipien und Algorithmen zur numerischen Berechnung von Integralen gegebener Funktionen. Hierbei beschränken wir uns vorerst auf die Berechnung des Riemann-Integrals¹

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

Dabei stellt f eine (von Computern ausführbare) Vorschrift dar, die es gestattet zu jedem $x \in [a, b]$ den entsprechenden Funktionswert f zu ermitteln (in den meisten Anwendungen ist f eine stückweise stetige oder stückweise glatte Funktion).

Es ist nun evident, dass im Allgemeinen die Berechnung der Funktionswerte allein zu keiner gesicherten Aussage bezüglich $I(f)$ führt; vielmehr bedarf es einer weiteren zusätzlichen Information über f , welche die Funktionswerte an denjenigen Stellen, an denen keine Werte errechnet worden sind, eingrenzt. Diese - globale - Information nennen wir die *Co-Observation* des gegebenen Problems, und interpretieren diese *Co-Observation* als eine Menge \mathcal{C} von Funktionen.

Es sei nun \mathcal{C} fixiert, Q ein Algorithmus, $Q[f]$ der durch diesen bei der Anwendung auf f erzeugte Schätzwert für $I(f)$. Dann stellt

$$\rho(Q; \mathcal{C}) := \sup_{f \in \mathcal{C}} \left| \int_a^b f(x) dx - Q[f] \right| \quad (5.1)$$

offenbar die durch Q in \mathcal{C} garantierte Genauigkeit dar. Für einige \mathcal{C} und Q wollen wir nun $\rho(Q; \mathcal{C})$ untersuchen.

5.1 MITTELPUNKT- UND TRAPEZREGEL

Beginnen wir mit der sicherlich einfachsten Quadraturformel.

Beispiel 5.1.1 (Mittelpunktsregel) Wir setzen

$$\mathcal{C}_M^{(r)} := \left\{ f \mid f^{(r)} \text{ stetig und } \sup_{a \leq x \leq b} |f^{(r)}(x)| \leq M \right\}$$

für $r \in \mathbb{N}$ und $M > 0$, und betrachten nachfolgende Grafik 5.1:

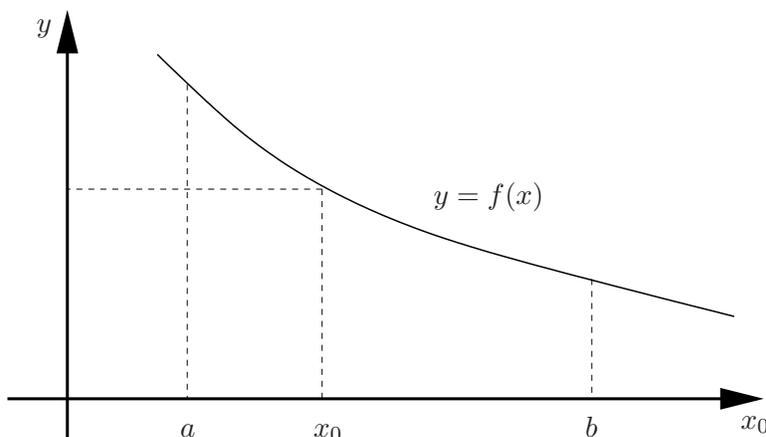
Sei nun $x_0 \in (a, b)$, dann gilt für $f \in C^1[a, b]$:

$$\int_a^b |f - P(f|x_0)| dx \leq \sup_{a \leq x \leq b} \frac{|f'(x)|}{1!} \int_a^b |x - x_0| dx. \quad (5.2)$$

Weiterhin ergibt sich mit $h := b - a$:

$$\begin{aligned} J(x_0) &:= \int_a^b |x - x_0| dx = - \int_a^{x_0} (x - x_0) dx + \int_{x_0}^b (x - x_0) dx \\ &= \frac{(x - x_0)^2}{2} \Big|_{x=x_0}^b - \frac{(x - x_0)^2}{2} \Big|_{x=a}^{x_0} \\ &= \frac{(b - x_0)^2}{2} + \frac{(a - x_0)^2}{2} \\ &= x_0^2 - x_0(a + b) + \frac{a^2 + b^2}{2}. \end{aligned}$$

¹Riemann, Georg Friedrich Bernhard (1826-1866)

Abb. 5.1: Graph einer Funktion f .

Wir wollen x_0 nun so wählen, dass $J(x_0)$ bzw. der rechte Teil der Ungleichung 5.2 minimal wird. Mit $J'(x_0) = 2x_0 - (a + b) \stackrel{!}{=} 0$ ergibt sich als einziger kritischer Punkt $x_0 = \frac{a+b}{2}$, welcher J minimiert, da $J'' = 2 > 0$.

Damit haben wir folgende Quadraturformel (**Mittelpunktsformel**) motiviert:

$$Q[f] := (b - a) \cdot f\left(\frac{a + b}{2}\right) \quad (5.3)$$

Mit obiger Herleitung haben wir auch gezeigt, dass

$$\begin{aligned} \rho(Q; \mathcal{C}_M^{(1)}) &\leq \frac{M}{1!} \cdot J\left(\frac{a + b}{2}\right) \\ &= M \cdot \left(\frac{(a + b)^2}{4} - \frac{(a + b)^2}{2} + \frac{a^2 + b^2}{2}\right) \\ &= M \cdot \frac{(b - a)^2}{4} \end{aligned}$$

als Maß für die Genauigkeit von (5.3) gilt.

Inwiefern läßt sich die Frage beantworten, ob eine Quadraturformel nun einen optimalen Algorithmus zur numerischen Berechnung beliebiger $I(f)$ darstellt.

Betrachten wir dazu die **summierte Trapezformel**:

$$Q_n^{Mi}[f] := \frac{b - a}{n} \sum_{\nu=1}^n f\left(a + \left(\nu - \frac{1}{2}\right) \frac{b - a}{n}\right) \quad (5.4)$$

Es sei ω eine auf $I := [0, b - a]$ definierte monoton-wachsende Funktion, die $w(0) = 0$ und $w(x + y) \leq w(x) + w(y)$ für $x, y \in I$ erfülle. Die Menge

$$H_w := \left\{ f \mid |f(x) - f(y)| \leq \omega(|x - y|) \forall x, y \in [a, b] \right\}$$

beinhaltet unter anderem die Menge aller auf $[a, b]$ stetigen Funktionen f (mit $w(t) := \sup \{|f(x) - f(y)| : |x - y| \leq t\}$).

Satz 5.1.2 ([Lebed]) Die beste Quadraturformel zu äquidistanten Stützstellen $x_i = a + i \cdot (b - a)/n$ ($i = 0, \dots, n$) und Funktionen aus H_ω hat Gewichte $\omega_0 = 1/(2n)$, $\omega_i = 1/n$ ($i = 1, \dots, n-1$) und $\omega_n = 1/(2n)$. Des Weiteren gilt für die Genauigkeit

$$\rho_n(H_\omega) = 2n \int_0^{\frac{b-a}{2n}} w(t) dt, \quad n \in \mathbb{N} \quad (5.5)$$

Das man mehr benötigt als summierte Mittelpunkts- oder Trapezformeln zeigt das folgende Beispiel.

MAPLE-Beispiel:

Möchten wir das Integral

$$\int_0^{e^{-2}} \frac{dx}{\log|x|}$$

mit einem möglichst kleinen Fehler, z.B. $< \frac{1}{2}10^{-4}$ und einer summierten Trapezformel berechnen, so liefert der Satz 5.1.2, dass dazu die Unterteilungseinheit von $n = 10^{500}$ nicht ausreicht! Man beachte, dass hier $w(t) = \frac{1}{\log|t|}$ gilt.

```
> restart:
> n:=10^500:
> 2*n*int(1/log(x), x=0..exp(-2)/(2*n)):
> evalf(%);
```

-0.0001171749067

Ein weiteres klassisches Ergebnis zur summierten Trapezformel liefert der folgende Satz.

Satz 5.1.3 ([Kiefer]) In der Klasse

$$\mathcal{C}_{A,B} := \{f \text{ wachsend, } f(a) = A, f(b) = B\}$$

ist die summierte Trapezformel zu äquidistanten Stützstellen optimal und es gilt darüber hinaus:

$$\rho_n(\mathcal{C}_{A,B}) = \frac{(B-A)(b-a)}{2(n+1)}, \quad n \in \mathbb{N}$$

Bemerkung 5.1.4 Die Fehlerabschätzungen für den Quadraturfehler, die wir im Folgenden betrachten werden, werden eher vom Typ 5.2 sein, d.h. bei dem die Approximationseigenschaft von Funktionen durch Polynome mittels höherer Ableitungen abgeschätzt wird.

5.2 Newton-Cotes-FORMELN

Die Kernidee der Mittelpunkts- bzw. der Trapezformel liegt darin, die Funktion f durch eine Interpolierende $P(f)$ zu ersetzen, sodass sich für diese die Quadratur einfach ausführen lässt. Wir verwenden dann $I(P(f))$ als Approximation zu $I(f) = \int_a^b f(x)dx$, setzen also:

$$Q[f] = I(P(f)).$$

Lemma 5.2.1 Zu $(n+1)$ paarweise verschiedenen Knoten x_0, \dots, x_n gibt es genau eine Quadraturformel

$$Q[f] = (b-a) \sum_{i=0}^n \omega_i f(x_i), \quad (\omega_i \hat{=} \text{weights}) \quad (5.6)$$

die für alle $P \in \mathbb{P}_n$ vom Grad kleiner oder gleich n exakt ist.

Beweis. Wir verwenden die *Lagrange*-Polynome

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

setzen diese in die Quadraturformel ein und erhalten:

$$I\left(\sum_{i=1}^n f(x_i)L_i(x)\right) = \sum_{i=1}^n f(x_i)I(L_i(x)) = (b-a) \sum_{i=1}^n \omega_i f(x_i). \quad (5.7)$$

Dadurch erhalten wir die Gewichte

$$\omega_i = \frac{1}{(b-a)} \int_a^b L_i(x) dx$$

auf eindeutige Weise zurück. □

Definition 5.2.2 (Newton-Cotes-Formeln) Bei äquidistanter Knotenwahl $a \leq x_0 < x_1 \dots < x_n \leq b$ heißen die resultierenden Integrationsformeln (5.7) **Newton-Cotes-Formeln**. Die Newton-Cotes-Formeln heißen **abgeschlossen**, wenn $x_0 = a$ und $x_n = b$, d.h.

$$x_i = a + ih, \quad h = \frac{b-a}{n} \quad (i = 0, \dots, n)$$

Die Newton-Cotes-Formeln heißen **offen**, wenn $x_0 < a$ und $x_n < b$, wobei meist

$$x_i = a + \left(i + \frac{1}{2}\right)h \quad \text{mit} \quad h = \frac{b-a}{n+1} \quad (i = 0, \dots, n) \quad (5.8)$$

gewählt wird.

Der Ausdruck für die abgeschlossenen Newton-Cotes-Gewichte ω_i vereinfacht sich durch die Substitution $s := \frac{(x-a)}{h}$ zu

$$\omega_i = \frac{1}{b-a} \int_a^b \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)} dx = \frac{1}{n} \int_0^n \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(s - j)}{(i - j)} ds,$$

und für die offenen Newton-Cotes-Formeln erhält man unter der Wahl (5.8) der x_i :

$$\omega_i = \frac{1}{b-a} \int_a^b \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)} dx = \frac{1}{n+1} \int_{-\frac{1}{2}}^{n+\frac{1}{2}} \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(s - j)}{(i - j)} ds$$

Die in Definition 5.2.2 definierten Gewichte ω_i sind von den Intervallgrenzen a , b unabhängig und müssen nur einmal zu gegebenem n bestimmt werden.

Für die abgeschlossenen Newton-Cotes-Formeln lauten die ersten Gewichte

n	Gewichte				Fehler	
1		1/2		1/2	$\frac{h^3}{12} \cdot f''(\xi)$	
2		1/6	4/6	1/6	$\frac{h^5}{90} \cdot f^{(4)}(\xi)$	
3	1/8		3/8	3/8	1/8	$\frac{3h^5}{80} \cdot f^{(4)}(\xi)$
4	7/90	32/90	12/90	32/90	7/90	$\frac{8h^7}{945} \cdot f^{(6)}(\xi)$

Im Fall $n = 1$ liegt die **Trapezregel** vor
 $n = 2$ ist die **Simpson-Keplersche²-Fassregel**
 $n = 3$ heißt **Newtonsche 3/8-Regel** und für
 $n = 4$ nennen wir die Quadraturformel **Milne-Regel**.

Mittels der folgenden Maple-Anweisungen lassen sich die Gewichte einfach bestimmen.

MAPLE-Beispiel:

```
> # geschlossene newton-Cotes-Formel
> n:=2:
> zaehler := (x, i) -> quo(product((x-k), k=0..n), (x-i), x):
> seq(int(zaehler(x, m)/subs(x=m, zaehler(x, m)), x=0..n)/n, m=0..n);
```

$$\frac{1}{6}, \frac{2}{3}, \frac{1}{6}$$

```
# offene newton-Cotes-Formel
> n:=2:
> zaehler := (x, i) -> quo(product((x-k), k=0..n), (x-i), x):
> seq(int(zaehler(x, m)/subs(x=m, zaehler(x, m)),
x=-1/2..n+1/2)/(n+1), m=0..n);
```

$$\frac{3}{8}, \frac{1}{4}, \frac{3}{8}$$

Definition 5.2.3 (Konvergenz einer Quadraturformel) Wir betrachten im weiteren Verlauf nun Folgen von Quadraturformeln und hierbei den Grenzwert

$$Q_n(f) \xrightarrow{n \rightarrow \infty} \int_a^b f(x) dx \quad \text{für jedes } f \in C[a, b]. \quad (5.9)$$

²Numere Kepler, Johannes (1571-1630)

Gilt (5.9), so spricht man von der **Konvergenz der Quadraturformeln** Q_n ($n \in \mathbb{N}$).

Satz 5.2.4 (Szegő) Sei $a \leq x_0^{(n)} < \dots < x_n^{(n)} \leq b$, und seien $\omega_k^{(n)} \in \mathbb{R}$ ($k = 0, \dots, n$).

Dann sind die für $n \in \mathbb{N}$ gemäß

$$Q_n : (C[a, b], \|\cdot\|_\infty) \rightarrow (\mathbb{R}, |\cdot|)$$

$$f \mapsto Q_n(f) := \sum_{k=0}^n \omega_k^{(n)} f(x_k^{(n)})$$

definierten Quadraturformeln Q_n genau dann konvergent, wenn die beiden folgenden Bedingungen erfüllt sind:

(i)

$$\sup_{n \in \mathbb{N}} \sum_{k=0}^n |\omega_k^{(n)}| < \infty$$

(ii)

$$I(p) = \lim_{n \rightarrow \infty} Q_n(p), \quad (p \in \mathbb{P} = \mathbb{P}(a, b))$$

Beweis. Der Beweis dieses Satzes erfordert grundlegende Kenntnisse der Funktionalanalysis und wird deshalb an dieser Stelle ausgespart. Man findet ihn nichtsdestoweniger beispielsweise in [Heuser][Seite 159]. \square

Beispiel 5.2.5 Wir wenden den Satz von Szegő auf die summierte Mittelpunktsregel an:

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

mit den Gewichten

$$\omega_k^{(n)} = \frac{b-a}{n+1}.$$

Es gilt: $\sum_k |\omega_k^{(n)}| = b-a$, demnach ist (i) in Satz 3 erfüllt. Weiterhin gilt (ii) aufgrund der Stetigkeit (damit der Riemann-Integrierbarkeit) der Polynome auf $[a, b]$ und der Fehlerabschätzung für die summierte Mittelpunktsregel.

Satz 5.2.6 (Steklov) Unter den Voraussetzungen des Satzes von Szegő gelte

$$\omega_k^{(n)} \geq 0 \quad (n \in \mathbb{N}, 0 \leq k \leq n) \tag{5.10}$$

für die Gewichte $\omega_k^{(n)}$ der Quadraturformel Q_n . Dann konvergieren die Quadraturformeln Q_n ($n \in \mathbb{N}$) genau dann, wenn sie für alle $p \in \mathbb{P}$ konvergieren.

Beweis. Nach dem Satz von Szegő ist der Beweis erbracht, wenn dort unter der Voraussetzung von (5.10) (i) aus (ii) folgt. Es gelte also (ii) im Satz von Szegő. Für $f \equiv 1$ gilt mit (ii)

$$b-a = I(f) = \lim_{n \rightarrow \infty} Q_n(f) = \lim_{n \rightarrow \infty} \sum_{k=0}^n \omega_k^{(n)},$$

was wegen (5.10) die Aussage (i) impliziert. \square

Bemerkung 5.2.7 Bei den Newton-Cotes-Formeln treten durchaus negative Gewichte auf, d.h. der Satz von Steklov kann dann nicht angewandt werden. Jedoch hilft auch der Satz von Szegó nicht weiter, da - wie G. Polya zeigte - eine Funktion existiert, für die die Newton-Cotes-Formeln nicht konvergieren. Für abgeschlossene Newton-Cotes-Formeln taucht bei $n = 8$, für offene Newton-Cotes-Formeln bei $n = 6$ das erste Mal ein negatives Gewicht auf.

Die folgenden Zeilen Maple-Code berechnen die Gewichte der abgeschlossenen Newton-Cotes-Formeln für $n = 8$ und die der offenen Newton-Cotes-Formeln für $n = 6$:

MAPLE-Beispiel:

```
> n:=8:
> zaehler := (x, i) -> quo(product((x-k), k=0..n), (x-i), x):
> seq(int(zaehler(x,m)/subs(x=m, zaehler(x,m)), x=0..n)/n, m=0..n);
```

$$\frac{989}{28350}, \frac{2944}{14175}, \frac{-464}{14175}, \frac{5248}{14175}, \frac{-454}{2835}, \frac{5248}{14175}, \frac{-464}{14175}, \frac{2944}{14175}, \frac{989}{28350}$$

```
> n:=6:
> zaehler := (x, i) -> quo(product((x-k), k=0..n), (x-i), x):
> seq(int(zaehler(x,m)/subs(x=m, zaehler(x,m)),
      x=-1/2..n+1/2)/(n+1), m=0..n);
```

$$\frac{4949}{27648}, \frac{49}{7680}, \frac{6223}{15360}, \frac{-6257}{34560}, \frac{6223}{15360}, \frac{49}{7680}, \frac{4949}{27648}$$

Wie man sieht, enthalten beide Quadratur-Formeln negative Gewichte.

Definition 3 Eine Quadraturformel Q_n (zu $n + 1$ Knoten) heißt von der Ordnung $k \in \mathbb{N}$, wenn

$$I(q) - Q_n[q] = 0 \quad (q \in \mathbb{P}_{k-1})$$

gilt und dabei k maximal gewählt ist.

Satz 5.2.8 Sind $a \leq x_0^{(n)} < \dots < x_n^{(n)} \leq b$ und ist Q_n eine Quadraturformel, so gilt für ihre Ordnung k :

$$k \leq 2n + 2.$$

Beweis. Die Anwendung der Quadraturformel in der Form

$$Q_n[f] = \sum_{k=0}^n \omega_k^{(n)} f(x_k^{(n)})$$

auf das Polynom

$$p(x) = \prod_{k=0}^n (x - x_k^{(n)})^2, \quad x \in [a, b]$$

vom Grad $2n + 2$ liefert die Aussage

$$\int_a^b \underbrace{p(x)}_{>0} dx - \sum_{k=0}^n \omega_k^{(n)} \underbrace{p(x_k^{(n)})}_{=0} > 0$$

und damit die Behauptung. □

Schon in Kapitel 2 haben wir folgenden Satz bewiesen (vgl. Satz 2.1.24).

Satz 5.2.9 *Es sei $\omega(x) > 0$, $(x \in (a, b))$ eine Gewichtsfunktion mit zugehörigen Orthogonalpolynomen $p_{n+1}(x)$, $n \geq 0$. Ferner seien $a < x_0 < \dots < x_n < b$ die Nullstellen von $p_{n+1}(x)$. Dann gibt es positive Gewichte $\omega_0, \dots, \omega_n$, sodass gilt:*

$$\int_a^b \omega(x)q(x)dx = \sum_{k=0}^n \omega_k q(x_k) \quad (p \in \Pi_{2n+1})$$

5.3 SCHWIERIGKEITEN BEI DER QUADRATUR

5.3.1 Unstetige Integranden

Häufig ist der Integrand nur durch Punktauswertungen bekannt, und als Komposition mehrerer Teilfunktionen definiert, die an den Nullstellen unstetig sind. Zum Beispiel ist eine Funktion auf unterschiedlichen Teilintervallen, durch unterschiedliche Approximationen definiert, wie z.B. die *Besselfunktion*². Sind diese Nahtstellen bekannt, so sollte man das Integrationsgebiet hier unterteilen, da kommerzielle Programme als auch die später noch diskutierte adaptive Quadratur hier versagen oder ineffizient sind.

5.3.2 Singuläre Integrale

Definition 5.3.1 (Singuläres Integral) *Unter einer singulären Funktion wollen wir eine Funktion verstehen, die mit Ausnahme endlich vieler Stellen auf einem Intervall $[a, b]$ definiert und stetig ist, sodass f in jeder Umgebung einer Unstetigkeitsstelle unbeschränkt ist.*

Beispiel 5.3.2 Zum Beispiel ist die Funktion $1/\sqrt{x}$ auf $[0, 1]$ unbeschränkt und trotzdem besitzt sie endliches Integral, nämlich

$$I(f) = \int_0^1 \frac{1}{\sqrt{x}} dx = 2$$

Betrachten wir nun mehrere Methoden, wie sich solche Integrale doch noch in „angenehmere“ Integrale umschreiben lassen:

a) Substitution

Für ein $x_0 \in [a, b]$ mögen die einseitigen Grenzwerte der Ableitungen nicht existieren, z.B. $f(x) = \sqrt{x} \sin(x)$ auf $[0, 1]$ und $x_0 = 0$. Hier läßt sich schon die zweite Ableitung $f''(0)$ nicht definieren. Die Substitution $t := \sqrt{x}$ führt hier zum Ziel

$$\int_0^1 \sqrt{x} \sin(x) dx = \int_0^1 2t^2 \sin(t^2) dt.$$

Der Integrand ist nun sogar beliebig oft differenzierbar.

²Bessel, Friedrich Wilhelm (1784-1846)

b) Regularisierung

Eine Idee, die unter dem Stichwort Abziehen der Singularität oder Regularisierung bekannt ist, besteht aus der Anwendung der Formel

$$I(f) = I(f - s) + I(s)$$

Im obigen Beispiel leistet das die Funktion $s = \sqrt{x} \cdot x$:

$$\begin{aligned} \int_0^1 \sqrt{x} \sin(x) dx &= \int_0^1 \sqrt{x} \sin(x) - \sqrt{x} \cdot x dx + \int_0^1 \sqrt{x} \cdot x dx \\ &= \int_0^1 \sqrt{x} (\sin(x) - x) dx + \frac{2}{5} \end{aligned}$$

Der neue Integrand ist jetzt dreimal stetig differenzierbar. Man berücksichtige jedoch, dass nun Nahe der Stelle $x_0 = 0$ für kleine x Auslöschungen auftreten können.

c) Aufspaltung des Integrals

Eine weitere Möglichkeit besteht manchmal in der Aufspaltung des Integrals, im Fall a) wäre dies

$$\int_0^1 \sqrt{x} \sin(x) dx = \int_0^\varepsilon \sqrt{x} \sin(x) dx + \int_\varepsilon^1 \sqrt{x} \sin(x) dx \quad 0 < \varepsilon 1$$

Der zweite Integrand ist nun glatt, wobei die Konstante der n -ten Ableitung von ε abhängt. Für das erste Integral erhält man, wenn man $\sin(x)$ in eine Reihe entwickelt

$$\begin{aligned} \int_0^\varepsilon \sqrt{x} \sin(x) dx &= \int_0^\varepsilon \sqrt{x} \sum_{k=0}^{\infty} \frac{x^{2k+1}}{(2k+1)!} (-1)^k \\ &= \sum_{k=0}^{\infty} (-1)^k \frac{\varepsilon^{2k+5/2}}{(2k+1)!(2k+5/2)}. \end{aligned}$$

Dieses Vorgehen ist gerechtfertigt, da man wegen der gleichmäßigen Konvergenz der Reihe diese gliedweise ausrechnen dar, bzw. Summation und Integration vertauschen darf. Für ein kleines ε kann man diese Entwicklung nach wenigen Schritten abbrechen. Das Problem hierbei ist die geeignete Wahl des ε .

d) Anwendung der Gaußquadratur

Ein weiterer Kunstgriff besteht darin, dass man das Integral in die Form

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

bringt, wobei $\omega(x)$ die Singularität enthält. Zur Integration zieht man dann eine Gauß-Formel zum Gewicht $\omega(x)$ heran. Erneut berechnen wir das oben verwendete Integral

$$\int_0^1 \sqrt{x} \sin(x) dx$$

mit Hilfe einer Gauß-Formel zum Gewicht \sqrt{x} für $x \in (0, 1]$. die Jacobi-Polynome $F_n^{0,1/2}$ sind gerade die bezüglich $\omega(x) = (1+x)^{1/2}$ auf $(-1, 1)$ orthogonalen Polynome. Mit den Überlegungen in Kapitel (Gauß-Gewichte) lassen sich

$$m_k = \int_0^1 \sqrt{x} x^k dx = \frac{2}{2k+3}$$

Quadraturpunkte und Gewichte bestimmen.

5.4 ADAPTIVE QUADRATUR

Motivation Für die Berechnung eines Integrals der Funktion

$$f(x) = \sqrt{x}$$

scheint die bisher vorgenommene äquidistante Unterteilung von $[a, b]$ nicht optimal zu sein.

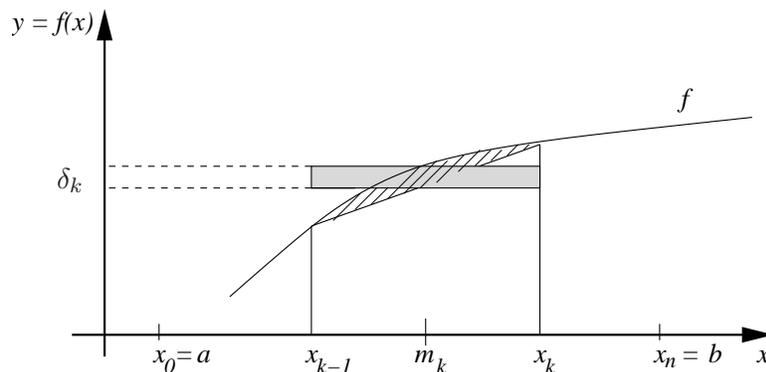
Wir betrachten deshalb die sog. adaptive Quadratur:

Herleitung des Algorithmus Es sei $\sum^n := \{x_j\}_{j=0, \dots, n}$ mit $a < x_0 < x_1 < \dots < x_n = b$. Für $f \in \mathcal{C}_{A,B}$ gilt

$$R_{k-1}^k(f) := \int_{x_{k-1}}^{x_k} f(x) dx - \frac{x_k - x_{k-1}}{2} (f(x_{k-1}) + f(x_k)) \leq (x_k - x_{k-1}) |\delta_k|,$$

mit $\delta_k := f(m_k) - \frac{f(x_{k-1}) + f(x_k)}{2}$.

Die geometrische Veranschaulichung des Faktors δ_k :



Vorgehensweise Falls die Fehler gleichverteilt sind, d.h. falls gilt:

$$R_{k-1}^k[f] \leq \varepsilon \frac{(x_{k-1} - x_k)}{(b - a)},$$

dann folgt

$$\left| \int_a^b f(x) dx - Q_{\sum^n}^{adap.}[f] \right| \leq \sum_{k=1}^n R_{k-1}^k[f] \leq \varepsilon.$$

Somit reicht es für jedes Teilintervall

$$|\delta_k| \leq \frac{\varepsilon}{b - a}$$

zu fordern.

In Pseudocodeschreibweise ergibt sich also folgender Algorithmus für die adaptive Quadratur:

Algorithmus 5.4.1: Adaptive Quadratur

Initialisierung: Gegeben $a = x_0 < x_1 < \dots < x_n = b$

1. Berechne $Q_{k-1}^k[f]$ ($k = 1, \dots, n$)

2. Falls $\sum(x_k - x_{k-1})|\delta_k| < \varepsilon$ ($\Rightarrow R(f) < \varepsilon$)

$$Q[f] = \sum Q_{k-1}^k[f]$$

sonst

verfeinere $\sum^n \rightarrow \sum^{\tilde{n}}$, d.h. für $k = 1, \dots, n$

füge \tilde{x}_k ein, mit $x_{k-1} < \tilde{x}_k < x_k$ (z.B. durch $\tilde{x} = \frac{x_{k-1} + x_k}{2}$),

falls $|\delta_k| > \frac{\varepsilon}{b-a}$.

Setze $\tilde{n} = n$ und gehe zu 1.

Ergebnis:

Die wesentlichen Bestandteile sind also:

- (i) Das Verfeinerungskriterium $|\delta_k| > \frac{\varepsilon}{b-a}$
- (ii) Die Verfeinerungsstrategie, i.e. das Einfügen von $\tilde{x} = \frac{x_{k-1} + x_k}{2}$
- (iii) Das Abbruchkriterium

5.5 EXTRAPOLATION

Motivation Wir betrachten ein Quadraturformel $Q_h[f]$ mit $h = \frac{b-a}{n}$, die eine Entwicklung des Quadraturfehlers in der Form

$$\int_a^b f(x)dx - Q_h[f] = \alpha h^s + \beta h^t + \mathcal{O}(h^{t+1}), \quad s < t$$

gestattet, wobei α, β von f abhängen aber nicht von h . Dann gilt mit $0 < q < 1$

$$\int_a^b f(x)dx - Q_{qh}[f] = \alpha(qh)^s + \beta(qh)^t + \mathcal{O}(h^{t+1})$$

also

$$\int_a^b f(x)dx - \frac{Q_{qh}[f] - q^s Q_h[f]}{1 - q^s} = -\frac{\beta q^s (1 - q^{t-s})}{1 - q^s} h^t + \mathcal{O}(h^{t+1}).$$

In der Linearkombination

$$Q_h^1[f] := \frac{Q_{qh}[f] - q^s Q_h[f]}{1 - q^s}$$

haben wir damit eine Quadraturformel, die sich bezüglich des Quadraturfehlers in h wesentlich günstiger verhält als $Q_h[f]$.

Beispiel 4 Für die summierte Trapezformel gilt die *Euler-Maclaurin*³-Summenformel in der Form

$$Q_h^{Trap}[f] = \int_a^b f(x)dx + \sum_{k=1}^N c_k h^{2k} + R_{N+1}(h)$$

mit bestimmten von h unabhängigen Koeffizienten c_k und einem Restglied $R_{N+1}(h) = \mathcal{O}(h^{2N+2})$ für jedes feste N und $h \rightarrow 0$

³Euler, Leonhard (1707-1783)

5.6 NUMERISCHE QUADRATUR VON STARK OSZILLIERENDEN INTEGRANDEN

Motivation Die Integration von stark oszillierenden Integranden ist ein numerisches Problem von weitreichender Bedeutung mit einer Vielzahl von Anwendungen, z.B. in der Quantenchemie, Bildanalyse, Elektrodynamik, Computertomographie und Strömungsmechanik. Allgemein wird dieses Problem als „schwierig“ eingestuft, bei dem man am besten die Oszillationen eliminiert, z.B. durch die Wahl überaus vieler kleiner Teilintervalle. Da das folgende Kapitel nur einen Einblick in diese Problematik und Techniken liefern soll, beschränken wir uns auf ein Integral der Form

$$I[f] = \int_0^1 f(x)e^{i\omega g(x)} dx \quad (5.11)$$

Bemerkung 4 Man beachte, dass sich ein beliebiges Integral $\int_a^b \tilde{f}(y)e^{i\omega g(y)} dy$ mit der Transformation $y = a + (b - a)x$ auf die Form von (5.11) überführen läßt, dabei ist dann $f(x) = \tilde{f}(a + (b - a)x)/(b - a)$ und $g(x) = \tilde{g}(a + (b - a)x)$.

Bemerkung 5 Das übliche numerische Verfahren ein Integral der Form (5.11) zu bestimmen, wobei f und g als glatt vorausgesetzt werden, wäre die *Gauß-Quadratur*: Wir interpolieren den Integranden an paarweise verschiedenen Knoten $0 \leq x_0 < \dots < x_n \leq 1$ durch ein Polynom $p \in \mathbb{P}_n$ und approximieren

$$I[f] \approx \int_0^1 p(x) dx.$$

Unglücklicherweise liefert diese Vorgehensweise für $w \gg 1$ völlig sinnlose Ergebnisse.

Beispiel 5.6.1

Ein alternativer Ansatz stammt in seiner ersten Idee von *Louis N. G. Filon* (1928). Anstatt den ganzen Interpolanden $f(x)e^{i\omega g(x)}$ zu interpolieren, interpolieren wir an den Stellen x_0, \dots, x_n die Funktion $f(x)$ durch ein Polynom $P[f](x)$ und setzen

$$Q^{Filon}[f] = \int_0^1 P[f](x)e^{i\omega g(x)} dx = \sum_{k=0}^n \lambda_k(\omega) f(x_k),$$

wobei $\lambda_k(\omega) := \int_0^1 L_k(x)e^{i\omega g(x)} dx$ und $L_k(x)$ sei das *Lagrange*-Polynom zu den Knoten x_0, \dots, x_n . Man beachte dabei, dass die Konstruktion von Q^{Filon} voraussetzt, dass sich die ersten Momente $\int_0^1 x^m e^{i\omega g(x)} dx$ berechnen lassen. Dies setzen wir stillschweigend für den Rest dieses Abschnitts voraus. Falls $g' \neq 0$ in $[0, 1]$, läßt sich zeigen (vgl. *Iserles*, 2003a), dass eine größer werdende Frequenz ω zu kleineren Fehlern führt, genauer ausgedrückt

$$Q^{Filon}[f] - I[f] \sim \begin{cases} \mathcal{O}(\omega^{-1}), & , \quad x_0 > 0 \text{ oder } x_n < 1 \\ \mathcal{O}(\omega^{-2}), & , \quad x_0 = 0 \text{ und } x_n = 1 \end{cases} \quad \omega \rightarrow \infty$$

Diese Methode läßt sich noch dadurch verbessern, dass man das Interpolationspolynom durch ein allgemeines *Hermite*-Polynom ersetzt, also ein Polynom wählt, welches an den Stellen x_0, \dots, x_n nicht nur die Funktionswerte $f(x_0), \dots, f(x_n)$ sondern auch die Ableitungen $f^{(m)}(x)$ an den Stützstellen exakt repräsentiert. Durch diese Verallgemeinerung läßt sich auch die Einschränkung $g' \neq 0$ in $[0, 1]$ umgehen, falls g' an endlich vielen Stellen ξ_k mit $g'(\xi_k) = \dots = g^{(r_k)}(\xi_k)$ und $g^{(r_k+1)}(\xi_k) \neq 0$ gilt.

Beispiel 5 Wie oben sei $f(x) = \cos(x)$, $g(x) = x$ und $\omega > 0$. Das Polynom

$$p(x) = \sum_{k=1}^n f(x_k) \cdot \left(1 - \frac{\omega''(x_n)}{\omega'(x_k)}(x - x_k)\right) L_k^2(x) = \sum_{k=1}^n f'(x_k) \cdot (x - x_k) L_k^2(x),$$

wobei L_k die *Lagrange*-Polynome sind und $\omega(x) = (x - x_1) \cdot \dots \cdot (x - x_n)$ ist, erfüllt das *Hermite'sche* Interpolationsproblem

$$p(x_k) = f(x_k), \quad p'(x_k) = f'(x_k) \quad (k = 1, 2, \dots, n);$$

für $n = 2$ lautet

$$\begin{aligned} p(x) &= \frac{(3x_1 - x_2 - 2x)(x - x_2)^2}{(x_1 - x_2)^2} f(x_1) + \frac{(3x_2 - x_1 - 2x)(x - x_1)^2}{(x_2 - x_1)^2} f(x_2) \\ &+ \frac{(x - x_1)(x - x_2)^2}{(x_1 - x_2)^2} f'(x_1) + \frac{(x - x_1)^2(x - x_2)}{(x_2 - x_1)^2} f'(x_2). \end{aligned}$$

Setzen wir $x_1 = 0$ und $x_2 = 1$, so erhalten wir

$$p(x) = (1 + 2x)(x - 1)^2 f(0) + (3 - 2x)x^2 f(1) + x(x - 1)^2 f'(0) + x^2(x - 1) f'(1).$$

Mit Maple lassen sich leicht die Integrale

$$\int_0^1 (1 + 2x)(x - 1)^2 e^{i\omega x} dx, \dots, \int_0^1 x^2(x - 1) e^{i\omega x} dx$$

berechnen, wobei die entsprechenden Zeilen in Maple lauten

```
> collect(int((1+2*x)*(x-1)^2*exp(I*omega*x) ,x=0..1) , omega);
collect(int((3-2*x)*x^2 *exp(I*omega*x) ,x=0..1) , omega);
collect(int(x*(x-1)^2 *exp(I*omega*x) ,x=0..1) , omega);
collect(int(x^2*(x-1) *exp(I*omega*x) ,x=0..1) , omega);
```

Die entsprechende Quadraturformel lautet dann

$$\begin{aligned}
 Q^{Filon}[f] &= \left(\frac{i}{\omega} + \frac{6i(1+e^{i\omega})}{\omega^3} + \frac{12(1-e^{i\omega})}{\omega^4} \right) f(0) \\
 &+ \left(\frac{-ie^{i\omega}}{\omega} - \frac{6(1+e^{i\omega})}{\omega^3} - \frac{12(1-e^{i\omega})}{\omega^4} \right) f(1) \\
 &+ \left(-\frac{1}{\omega^2} + \frac{2i(2+e^{i\omega})}{\omega^3} + \frac{6(1-e^{i\omega})}{\omega^4} \right) f'(0) \\
 &+ \left(\frac{e^{i\omega}}{\omega^2} + \frac{2i(1+2e^{i\omega})}{\omega^3} + \frac{6(1-e^{i\omega})}{\omega^4} \right) f'(1)
 \end{aligned}$$

Für $f(x) = \cos(x)$, $g(x) = x$ und $10 < \omega < 100$ ist der Fehler für Q^{Filon} skaliert mit ω^3 in der nachfolgenden Abbildung dargestellt.

Betrachten wir nun die weitere Möglichkeit das Integral (5.11) mittels einer asymptotischen Entwicklung zu berechnen. Aus

$$\frac{d}{dx} e^{i\omega g(x)} = i\omega g'(x) e^{i\omega g(x)}$$

erhalten wir nun unter der Voraussetzung $g' \neq 0$ in $[0, 1]$, $\omega > 0$ für eine beliebige Funktion h mittels partieller Integration

$$\begin{aligned}
 \int_0^1 h(x) e^{i\omega g(x)} dx &= \int_0^1 \frac{h(x)}{i\omega g'(x)} i\omega g'(x) e^{i\omega g(x)} dx \\
 &= \frac{h(x)}{i\omega g'(x)} \cdot e^{i\omega g(x)} \Big|_{x=0}^1 - \int_0^1 \frac{1}{i\omega} \frac{d}{dx} \left[\frac{h(x)}{g'(x)} \right] e^{i\omega g(x)} dx \quad (5.12)
 \end{aligned}$$

Mit der Notation

$$\sigma_0[f](x) = f(x), \quad \sigma_{k+1}[f](x) = \frac{d}{dx} \frac{\sigma_k[f](x)}{g'(x)}, \quad k = 0, 1, 2, \dots$$

und (5.12) erhalten wir somit für $h(x) = \sigma_k[f](x)$:

$$\begin{aligned}
 \int_0^1 \sigma_k[f](x) e^{i\omega g(x)} dx &= \frac{1}{i\omega} \left(\frac{\sigma_k[f](1)}{g'(1)} e^{i\omega g(1)} - \frac{\sigma_k[f](0)}{g'(0)} e^{i\omega g(0)} \right) \\
 &- \frac{1}{i\omega} \int_0^1 \sigma_{k+1}[f](x) e^{i\omega g(x)} dx.
 \end{aligned}$$

Per Induktion folgt nun für $n \geq 0$

$$\begin{aligned}
 \int_0^1 f(x) e^{i\omega g(x)} dx &= - \sum_{l=1}^n \left(\frac{i}{\omega} \right)^l \left(\frac{e^{i\omega g(1)}}{g'(1)} \cdot \sigma_{l-1}[f](1) - \frac{e^{i\omega g(0)}}{g'(0)} \cdot \sigma_{l-1}[f](0) \right) \\
 &+ \left(\frac{i}{\omega} \right)^n \int_0^1 \sigma_n[f](x) e^{i\omega g(x)} dx
 \end{aligned}$$

Die Idee der asymptotischen Reihenentwicklung ist nun, das Integral $\left(\frac{i}{\omega} \right)^n \int_0^1 \sigma_n[f](x) e^{i\omega g(x)} dx$ wegzulassen, da $\frac{1}{\omega^n}$ für $\omega \gg$ „hoffentlich schneller“ gegen Null strebt, als das entsprechende Integral. Somit definieren wir die Quadratur

$$Q^{asympt.}[f] = - \sum_{l=1}^n \left(\frac{i}{\omega} \right)^l \left(\frac{e^{i\omega g(1)}}{g'(1)} \cdot \sigma_{l-1}[f](1) - \frac{e^{i\omega g(0)}}{g'(0)} \cdot \sigma_{l-1}[f](0) \right)$$

Für streng monotone g erhalten wir durch direktes Ausrechnen

$$\begin{aligned}\sigma_0[f] &= f \\ \sigma_1[f] &= -\frac{g''}{g'^2}f + \frac{1}{g'}f' \\ \sigma_2[f] &= \frac{3g''^3 - gg'''}{g'^4}f - \frac{3g''}{g'^3}f' + \frac{1}{g'^2}f'' \\ \sigma_3[f] &= \frac{-15g''^3 + 10g'g''g''' - g^2g^{(4)}}{g'^6}f + \frac{15g''^2 - 4gg'''}{g'^5}f' - \frac{6g''}{g'^4}f'' + \frac{1}{g'^3}f'''\end{aligned}$$

und man sieht sofort, dass für jedes $\ell \geq 0$ und $j = 0, 1, \dots, \ell$ Funktionen $\sigma_{\ell,j}$ existieren, welche von g und entsprechenden Ableitungen abhängen, sodass

$$\sigma_\ell[f](x) = \sum_{j=0}^{\ell} \sigma_{\ell,j} f^{(j)}(x) \quad \text{und} \quad \sigma_{\ell,\ell} = \frac{1}{(g'(x))^\ell} \neq 0$$

Für unser Beispiel lautet die Quadraturformel Q^{app} mit 2 Entwicklungstermen

$$Q^{asympt.}[f] = \frac{e^{i\omega} f(1) - f(0)}{i\omega} + \frac{e^{i\omega} f'(1) - f'(0)}{\omega^2}.$$

Für $f(x) = \cos(x)$, $g(x) = x$ und $10 < \omega < 100$ ist der Fehler für $Q^{asympt.}$ skaliert mit ω^3 in nachfolgender Grafik dargestellt (oberer Graph)

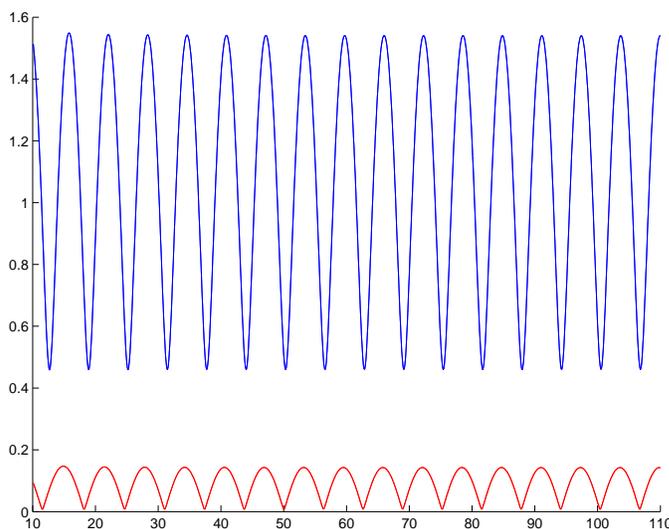


Abb. 5.2: Der Fehler von Q^{Filon} (oben) und $Q^{asympt.}$ (unten), skaliert mit ω^3 für $f(x) = \cos(x)$, $g(x) = x$ und $10 < \omega < 100$.

Widmen wir uns noch der Einschränkung, dass $g' \neq 0$ in $[0, 1]$ gelten soll, was wir in unseren Betrachtungen immer vorausgesetzt haben. Man beachte nun, dass für jedes $\xi \in [0, 1]$ und glatte Funktionen f und g mit $g'(\xi) = 0$, $g''(\xi) \neq 0$ und $g'(x) \neq 0$ für $x \in [0, 1] \setminus \{\xi\}$ die folgende Gleichheit gilt

$$\begin{aligned}\int_0^1 f(x) e^{i\omega g(x)} dx &= f(\xi) \int_0^1 e^{i\omega g(x)} dx + \int_0^1 (f(x) - f(\xi)) e^{i\omega g(x)} dx \\ &= f(\xi) \int_0^1 e^{i\omega g(x)} dx + \frac{1}{i\omega} \int_0^1 \frac{f(x) - f(\xi)}{g'(x)} \cdot \frac{d}{dx} e^{i\omega g(x)} dx\end{aligned}$$

Mit dem Satz von *de l'Hospital*⁴ folgt für genügend glatte f und g nun, da $g'' \neq 0$ in $[0, 1]$, dass der Grenzwert

$$\lim_{x \rightarrow \xi} \frac{f(x) - f(\xi)}{g'(x)} = \lim_{x \rightarrow \xi} \frac{f'(x)}{g''(x)}$$

existiert. Mit der Notation

$$\begin{aligned} \rho_0[f](x) &= f(x) \\ \rho_{k+1}[f](x) &= \frac{d}{dx} \frac{\rho_k[f](x) - \rho_k[f](\xi)}{g'(x)}, \quad k \geq 0 \end{aligned}$$

folgt dann wieder mittels partieller Integration und Induktion

$$\begin{aligned} \int_0^1 f(x) e^{i\omega g(x)} dx &= \int_0^1 e^{i\omega g(x)} dx \sum_{\ell=0}^{n-1} \left(\frac{i}{\omega}\right)^\ell \rho_\ell[f](\xi) \\ &- \sum_{\ell=1}^n \left(\frac{i}{\omega}\right)^\ell \left(\frac{e^{i\omega g(1)}}{g'(1)} \{ \rho_{\ell-1}[f](1) - \rho_{\ell-1}[f](\xi) \} - \frac{e^{i\omega g(0)}}{g'(0)} \{ \rho_{\ell-1}[f](0) - \rho_{\ell-1}[f](\xi) \} \right) \\ &+ \left(\frac{i}{\omega}\right)^n \int_0^1 \rho_n[f](x) e^{i\omega g(x)} dx \end{aligned}$$

Beispiel 5.6.2

$$\int_{-1}^1 \cos(x * \exp(4 * x^2)) dx = 2\Re \left(\int_0^1 \exp((2x - 1) * \exp(4 * (2x - 1)^2)) dx \right)$$

MATLAB-Funktion: asymp_bsp.txt

```

1 > restart:
2 > f := x -> 1:
3 > g := x -> (2*x-1) * exp(4*(2*x-1)^2):
4 > rho := proc(k)
5 >   local i,tmp;
6 >   tmp := f(x);
7 >   for i from 1 to k do
8 >     tmp:=diff(tmp/diff(g(x),x),x);
9 >   end do;
10 >   simplify(tmp)
11 > end proc:
12 >
13 > simplify(subs(x=(y+1)/2,rho(4)))

```

$$\begin{aligned} \rho_0[f](x) &= 1 \\ \rho_1[f](x) &= \frac{-8 \exp(-4y^2) y (8y^2 + 3)}{(8y^2 + 1)^2} \Big|_{y=2x-1} \\ \rho_2[f](x) &= \frac{8 \exp(-8y^2) (1024y^6 + 704y^4 + 144y^2 - 3)}{(8y^2 + 1)^4} \Big|_{y=2x-1} \\ \rho_3[f](x) &= \frac{-8 \exp(-12y^2) y (24576y^8 + 24064y^6 + 9024y^4 + 1160y^2 - 75)}{(8y^2 + 1)^6} \Big|_{y=2x-1} \end{aligned}$$

⁴de l'Hospital (1661-1704)

A LINEARE DIFFERENZENGLEICHUNG

Wir beschränken uns hier auf lineare Differenzgleichungen mit konstanten Koeffizienten.

Definition A.0.1 Sei $m \in \mathbb{N}$. Eine reelle Folge (a_n) ist durch a_0, a_1, \dots, a_{m-1} und $\mu_0, \mu_1, \dots, \mu_{m-1}$ ($\mu_0 \neq 0$) mit der Rekursionsvorschrift

$$a_{n+m} = \sum_{i=0}^{m-1} \mu_i a_{n+i} \quad \text{für alle } n \in \mathbb{N}_0 \quad (\text{A.1})$$

eindeutig bestimmt. Diese Rekursionsvorschrift heißt lineare Differenzgleichung m -ter Ordnung.

Definiert man

$$\hat{a}_k := \begin{pmatrix} a_k \\ \vdots \\ a_{k+m-1} \end{pmatrix} \quad \text{und} \quad A := \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & 1 \\ \mu_0 & \mu_1 & \dots & \dots & \mu_{m-1} \end{pmatrix} \quad (\text{A.2})$$

so folgt $\hat{a}_{k+1} = A\hat{a}_k$, also gilt $\hat{a}_n = A^n \hat{a}_0$.

Definition A.0.2 Die Eigenwerte $\lambda_1, \dots, \lambda_m$ von A aus (A.2) heißen auch Eigenwerte der Differenzgleichung aus (A.1).

Wir beschränken uns zunächst auf den Fall, dass die Eigenwerte $\lambda_1, \dots, \lambda_m$ von A paarweise verschieden sind, d. h. $A = B^{-1}DB$ ist diagonalisierbar mit der Diagonalmatrix

$$D = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{pmatrix}.$$

Damit gibt es Vektoren $b^{(1)}, \dots, b^{(m)} \in \mathbb{R}^m$ mit $\hat{a}_n = B^{-1}D^n B \hat{a}_0 = \lambda_1^n b^{(1)} + \dots + \lambda_m^n b^{(m)}$. Betrachtet man die erste Zeile, so gibt es Konstanten c_1, \dots, c_m mit

$$a_n = c_1 \lambda_1^n + \dots + c_m \lambda_m^n.$$

Mit Hilfe der Anfangswerte a_0, \dots, a_{m-1} berechnet man die Koeffizienten c_1, \dots, c_m .

Satz A.0.3 Seien die Nullstellen $x_1, \dots, x_m \in \mathbb{C}$ des Polynoms

$$p(x) = x^m - \sum_{k=0}^{m-1} \mu_k x^k$$

paarweise verschieden, so ist $(x_1^n), \dots, (x_m^n)$ eine Basis des Lösungsraums der linearen Differenzgleichung (A.1), d. h. a_n ist genau dann eine Lösung von (A.1), wenn es Konstanten $c_1, \dots, c_m \in \mathbb{R}$ gibt mit $a_n = c_1 x_1^n + \dots + c_m x_m^n$

Beweis. Wir zeigen mit vollständiger Induktion über m

$$\det(A - \lambda I) = (-1)^m + (-1)^{m-1} \sum_{k=0}^{m-1} \mu_k x^k.$$

für A aus (A.2). Für $m = 1$ gilt $\det(A - \lambda I) = -\lambda + \mu_0$. Für $m \geq 2$ gilt nach Auflösung der Determinante nach der ersten Spalte sowie nach Induktionsvoraussetzung

$$\begin{aligned} \det(A - \lambda I) &= -\lambda \left((-1)^{m-1} \lambda^{m-1} + (-1)^m \sum_{k=0}^{m-2} \mu_{k+1} \lambda^k \right) + (-1)^{m-1} \mu_0 \cdot 1 \\ &= (-1)^m \lambda^m + (-1)^{m-1} \sum_{k=0}^{m-1} \mu_k \lambda^k. \end{aligned}$$

Damit sind x_1, \dots, x_m die Eigenwerte von A . Da der Lösungsraum m -dimensional ist, ist $a_n = c_1 x_1^n + \dots + c_m x_m^n$ für alle $c_1, \dots, c_m \in \mathbb{R}$ Lösung von (A.1). \square

Beispiel A.0.4 Die Fibonacci-Zahlen $(a_n) = (0, 1, 1, 2, 3, 5, \dots)$ sind durch die Rekursionsformel $a_{n+2} = a_{n+1} + a_n$ und durch die Anfangswerte $a_0 = 0$ und $a_1 = 1$ definiert. Das Differenzgleichungssystem hat dann die Gestalt

$$\begin{pmatrix} a_{n+1} \\ a_{n+2} \end{pmatrix} = A \begin{pmatrix} a_n \\ a_{n+1} \end{pmatrix} \quad \text{mit} \quad A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Für das charakteristische Polynom gilt

$$-x(1-x) - 1 = x^2 - x - 1 = 0.$$

Man erhält zwei unterschiedliche Eigenwerte

$$x_1 = \frac{1 - \sqrt{5}}{2} \quad \text{und} \quad x_2 = \frac{1 + \sqrt{5}}{2},$$

woraus

$$a_n = c_1 \left(\frac{1 - \sqrt{5}}{2} \right)^n + c_2 \left(\frac{1 + \sqrt{5}}{2} \right)^n$$

folgt. Das Einsetzen der Anfangswerte bestimmt

$$c_1 = -\frac{1}{\sqrt{5}} \quad \text{und} \quad c_2 = \frac{1}{\sqrt{5}}$$

und führt zu

$$c_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1 - \sqrt{5}}{2} \right)^n - \left(\frac{1 + \sqrt{5}}{2} \right)^n \right).$$

Satz A.0.5 Seien $x_1, \dots, x_\ell \in \mathbb{C}$ die Nullstellen des Polynoms

$$p(x) = x^m - \sum_{k=0}^{m-1} \mu_k x^k$$

mit den algebraischen Vielfachheiten $k_1, \dots, k_\ell \in \mathbb{N}$, so ist a_n genau dann Lösung von (A.1), wenn es Polynome $p_i \in \mathbb{P}_{k_i-1}$ gibt mit $a_n = p_1(n)x_1^n + \dots + p_\ell(n)x_\ell^n$. Eine Basis des Lösungsraums wäre damit

$$x_1^n, nx_1^n, \dots, n^{k_1-1} x_1^n, x_2^n, nx_2^n, \dots, n^{k_2-1} x_2^n, \dots, x_\ell^n, nx_\ell^n, \dots, n^{k_\ell-1} x_\ell^n.$$

Beweis. Da mit A aus (A.2) der Rang von $A - x_i I$ gleich $n - 1$ ist, gibt es zu jedem Eigenwert x_i einen Jordanblock $J_i \in \mathbb{R}^{k_i \times k_i}$ mit der Gestalt

$$J_i = \begin{pmatrix} x_i & 1 & 0 & \dots & 0 \\ 0 & x_i & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & 1 \\ 0 & \dots & \dots & 0 & x_i \end{pmatrix},$$

sodass

$$A = B^{-1} \begin{pmatrix} J_1 & & 0 \\ & \ddots & \\ 0 & & J_n \end{pmatrix} B$$

gilt. Sei

$$E = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & 1 \\ 0 & \dots & \dots & \dots & 0 \end{pmatrix} \in \mathbb{R}^{k_i \times k_i} \quad \text{und} \quad \binom{y}{j} := \frac{(y-j+1) \cdots (y-1) \cdot y}{j!}.$$

Wegen

$$J_i^n = (x_i I + E)^n = \sum_{j=0}^n \binom{n}{j} x_i^{n-j} E^j = x_i^n \sum_{j=0}^{k_i-1} \underbrace{\binom{n}{j} x_i^{-j}}_{\in \mathbb{P}_j \text{ bzgl. } n} E^j$$

gibt es vektorwertige Polynome $q^{(i)} \in \mathbb{P}_{k_i-1}^n$ mit

$$\hat{a}_n = B^{-1} \begin{pmatrix} J_1^n & & 0 \\ & \ddots & \\ 0 & & J_\ell^n \end{pmatrix} B \hat{a}_0 = x_1^n q^{(1)}(n) + \dots + x_\ell^n q^{(\ell)}(n).$$

Da der Lösungsraum m -dimensional ist, ist $a_n = p_1(n)x_1^n + \dots + p_\ell(n)x_\ell^n$ für alle Polynome $p_i \in \mathbb{P}_{k_i-1}$ Lösung von (A.1). \square

Beispiel A.0.6 Ein Beispiel für gleiche Eigenwerte liefert die Rekursionsformel

$$b_{n+2} = 4b_{n+1} - 4b_n,$$

die zum charakteristischen Polynom $x^2 - 4x + 4 = 0$ führt. Mit $x_{1,2} = 2$ erhält man den Ansatz

$$a_n = (c_1 + c_2 n) \cdot 2^n.$$

Wählt man als Anfangswerte $b_0 = 1$ und $b_1 = 4$, so hat die explizite Darstellung die Gestalt $b_n = (1 + n) \cdot 2^n$.

Bemerkung A.0.7 Sollte eine der Eigenwerte x von (A.1) komplex sein, so ist auch der komplex konjugierte Eigenwert \bar{x} ebenfalls Eigenwert mit der gleichen algebraischen Vielfachheit. Sei

$$a_n = c_1 x^n + c_2 \bar{x}^n.$$

Dann folgt $c := c_1 = \bar{c}_2$, d. h. es gilt

$$a_n = 2\Re(c \cdot x^n) = 2|c||x|^n \cos \left(n \arctan \frac{\Im(x)}{\Re(x)} + \arctan \frac{\Im(c)}{\Re(c)} \right).$$

Bemerkung A.0.8 Eine weitere Möglichkeit Differenzgleichungen zu lösen besteht unter Verwendung der Potenzreihe

$$f(x) = \sum_{k=0}^{\infty} a_k x^k. \quad (\text{A.3})$$

(siehe auch [Meyberg/Vachenaer]). Sei nun $A := \max\{|\mu_i| : i = 0, \dots, m-1\} > 0$, dann zeigt man $|a_k| \leq C(nA)^{k+1-m}$ mit $C = \max\{|a_i| : i = 0, \dots, m-1\}$ per vollständiger Induktion. Damit existiert der Grenzwert der Potenzreihe $f(x)$ für $|x| < \frac{1}{mA}$. Geschicktes Aufsummieren

$$\left(-1 + \sum_{i=1}^m \mu_{m-i} x^i\right) f(x) = \sum_{i=0}^{\infty} b_i x^i = \sum_{i=0}^{m-1} b_i x^i$$

eliminiert wegen

$$b_{n+m} = -a_{m+n} + \sum_{i=0}^{m-1} \mu_i a_{n+i} = 0$$

bis auf die ersten $m-1$ alle weiteren Summanden. Die Koeffizienten der ersten $m-1$ Summanden haben die Gestalt

$$b_k = -a_k + \sum_{i=m-k}^{m-1} \mu_i a_{k-m+i}.$$

Umformen von (A.3) führt zu

$$f(x) = \frac{b_0 + b_1 x + \dots + b_{m-1} x^{m-1}}{-1 + \mu_{m-1} x + \mu_{m-2} x^2 + \dots + \mu_0 x^m}.$$

Zerlegt man die rationale Funktion in Partialbrüche der Gestalt

$$\frac{1}{(x-a)^k} = \left(-\frac{1}{a}\right)^k \sum_{i=0}^{\infty} \binom{k+i-1}{k-1} \left(\frac{x}{a}\right)^i,$$

so kann diese wiederum als Potenzreihen dargestellt und mit den Koeffizienten der ursprünglichen Potenzreihe verglichen werden.

Bemerkung A.0.9 Allgemeiner können die Koeffizienten μ_i von n abhängen, d. h. man erhält die lineare Differenzgleichung

$$a_{n+m} = \sum_{i=0}^{m-1} \mu_i(n) a_{n+i} \quad \text{für alle } n \in \mathbb{N}_0, \quad (\text{A.4})$$

bei der allgemeine Lösungsansätze nicht mehr existieren. Eine Behandlung dieses Themas zeigt [Elayadi]. Anwendungen und rechnergestützte Berechnungen zum Thema Differenzgleichungen findet man u. a. in [Cull].

A.1 INHOMOGENE LINEARE DIFFERENZENGLEICHUNGEN

Ersetzen wir (A.1) durch

$$a_{n+m} = \sum_{i=0}^{m-1} \mu_i a_{n+i} + c_n \quad \text{für alle } n \in \mathbb{N}_0, \quad (\text{A.5})$$

wobei (c_k) eine reelle Folge ist. Seien $(a_k^{(1)})$ und $(a_k^{(2)})$ zwei Lösungen von (A.5), so folgt

$$a_{n+m}^{(2)} - a_{n+m}^{(1)} = \sum_{i=0}^{m-1} \mu_i \left(a_{n+i}^{(2)} - a_{n+i}^{(1)} \right),$$

also ist $(a_k^{(2)} - a_k^{(1)})$ Lösung von (A.1). Ist eine sogenannte partikuläre Lösung (a_k) von (A.5) bekannt, so sind alle Lösungen von (A.5) durch Addition der homogenen Lösungen bekannt. Ein allgemeines Verfahren zur Bestimmung einer solchen partikulären Lösung gibt es nicht. Für einige Spezialfälle kann man wie folgt verfahren:

1. Fall: $c := c_n = \text{const.}$

Eine partikuläre Lösung wäre die konstante Folge

$$a_n = \frac{c}{1 - \sum_{i=0}^{m-1} \mu_i}.$$

2. Fall: $c_n = p(n)$, mit $p \in \mathbb{P}_r$

Verwenden Sie den Ansatz $a_k = \sum_{i=0}^r b_i k^i$, setzen Sie diese in (A.5) ein und vergleichen Sie die Koeffizienten vor den Monomen n^j .

3. Fall: $c_n = u^n$, mit $u \in \mathbb{R}$

Verwenden Sie den Ansatz $a_k = b \cdot u^k$, setzen Sie diese in (A.5) ein und ermitteln Sie b .

4. Fall: $c_n = A_0 \sin(\alpha n) + B_0 \cos(\alpha n)$ mit $A_0, B_0 \in \mathbb{R}$

Verwenden Sie den Ansatz $a_k = A \sin(\alpha k) + B \cos(\alpha k)$, setzen Sie diese in (A.5) ein und ermitteln Sie A und B .

Beispiel A.1.1 Die Fehlerfolge ε_n bei der Berechnung einer Nullstelle x^* einer Funktion $f \in \mathbb{C}^2([a, b])$ mit Hilfe des Sekantenverfahrens kann näherungsweise durch die Rekursionsformel

$$\varepsilon_{n+1} = u \cdot \varepsilon_n \cdot \varepsilon_{n-1}$$

mit $u > 0$ dargestellt werden. Setzt man nun $a_k = \log(\varepsilon_k)$, so ergibt sich die inhomogene lineare Differenzgleichung

$$a_{n+1} = \log(u) + a_n + a_{n-1}. \quad (\text{A.6})$$

Wie bei der Berechnung der Fibonacci-Zahlen sind

$$a_n^{(h)} = c_1 \left(\frac{1 - \sqrt{5}}{2} \right)^n + c_2 \left(\frac{1 + \sqrt{5}}{2} \right)^n$$

die Lösungen der homogenen Differenzgleichung und

$$a_n^{(p)} = -\log(u)$$

eine partikuläre Lösung von (A.6). Das führt zu

$$a_n = a_n^{(p)} + a_n^{(h)} = -\log(u) + c_1 \left(\frac{1 - \sqrt{5}}{2} \right)^n + c_2 \left(\frac{1 + \sqrt{5}}{2} \right)^n$$

und man erhält schließlich

$$\varepsilon_n = \frac{1}{u} \cdot e^{c_1 \left(\frac{1 - \sqrt{5}}{2} \right)^n + c_2 \left(\frac{1 + \sqrt{5}}{2} \right)^n}.$$

Mit $\lambda_1 := \frac{1-\sqrt{5}}{2}$ und $\lambda_2 := \frac{1+\sqrt{5}}{2}$ ist

$$\frac{\varepsilon_{n+1}}{\varepsilon_n^{\lambda_2}} = e^{c_1 \lambda_1^{n+1} + c_2 \lambda_2^{n+1} - c_1 \lambda_1^n \lambda_2 - c_2 \lambda_2^{n+1}} = e^{-\sqrt{5} c_1 \lambda_1^n} \rightarrow 1 \quad (n \rightarrow \infty)$$

in Abhängigkeit von n beschränkt. Also ist bei Konvergenz gegen 0 die maximale Konvergenzordnung gleich $\frac{1+\sqrt{5}}{2}$.

B AUFGABEN UND LÖSUNGEN

Aufgabe 1 (Nested intervals)

(3+3 Punkte)

- (i) Show first that the nonlinear equation $x = \cos x$ has a unique solution $x \in [0, \infty)$.
- (ii) Determine a numerical solution of the nonlinear equation $x = \cos x$ by using nested intervals with initial values $a = 0$ und $b = 1$ until the error is smaller than 0,05. You can use a calculator.

Lösung zu Aufgabe 1

- (i) Zu zeigen: Die Gleichung $x = \cos x$ hat eine eindeutige Lösung.
Sei $f : \mathbb{R} \rightarrow \mathbb{R}$ mit $f(x) = x - \cos x$ ist stetig.

$$f(0) = -1, f\left(\frac{\pi}{2}\right) = \frac{\pi}{2}$$

Damit gilt nach dem Zwischenwertsatz: es existiert ein $\xi \in (0, \frac{\pi}{2})$ mit $f(\xi) = 0$.

$$f'(x) = 1 + \sin x > 0 \text{ für } x \neq 2\pi k + \frac{3}{2}\pi, k \in \mathbb{Z}$$

Daraus folgt, dass f auf dem Intervall $[2\pi k + \frac{3}{2}\pi, 2\pi(k+1) + \frac{3}{2}\pi]$, $k \in \mathbb{Z}$, streng monoton steigend ist (Mittelwertsatz).

Damit ist dann f auf ganz \mathbb{R} streng monoton steigend und die die Nullstelle ξ ist eindeutig.

- (ii)

$$\begin{aligned} a_0 &= 0, b_0 = 1 \\ a_1 &= 0,5, b_1 = 1 \\ a_2 &= 0,5, b_2 = 0,75 \\ a_3 &= \frac{5}{8}, b_3 = \frac{3}{4} \\ a_4 &= \frac{11}{16}, b_4 = \frac{3}{4} \\ a_5 &= \frac{23}{32}, b_5 = \frac{3}{4} \end{aligned}$$

Die Länge des Intervalls ist nach der 5-ten Iteration $2^{-5} < 0,05$, damit ist auch der absolute Fehler für jedes $x \in [a_5, b_5]$ zur Nullstelle kleiner als 0,05.

Aufgabe 2 (Interpolation)

(3 Punkte)

Man finde das Polynom $p : \mathbb{R} \rightarrow \mathbb{R}$ von kleinstem Grad mit $p(-1) = 3$, $p(0) = -3$, $p(1) = -5$, $p(2) = 3$.

Lösung zu Aufgabe 2

$$\begin{aligned}
 p(x) &= ax^3 + bx^2 + cx + d \\
 p(-1) &= -a + b - c + d = 3 \\
 p(0) &= d = -3 \\
 p(1) &= a + b + c + d = -5 \\
 p(2) &= 8a + 4b + 2c + d = 3
 \end{aligned}$$

Nach Berechnung der Koeffizienten des linearen Gleichungssystems erhält man $p(x) = x^3 + 2x^2 - 5x - 3$.

Aufgabe 3 (Rolle, Taylor, Banach)**(3+3+4 Punkte)**

- (i) Zeigen Sie, dass für eine n -mal differenzierbare Funktion $f : [a, b] \rightarrow \mathbb{R}$ mit $n + 1$ verschiedenen Nullstellen ein $\xi \in (a, b)$ existiert mit $f^{(n)}(\xi) = 0$.
- (ii) Berechnen Sie näherungsweise das Integral

$$\int_0^1 e^{-x^2/2} dx,$$

indem Sie $f(x) = e^{-x^2/2}$ in ein Taylorpolynom 3. Grades mit Entwicklungspunkt 0 entwickeln. Schätzen sie den Fehler mit Hilfe des Restglieds von Lagrange ab.

- (iii) Zeigen Sie, dass durch

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{c}{x_n} \right)$$

ein iteratives Verfahren zur Berechnung der Quadratwurzel einer positiven Zahl c definiert ist. Welche Bedingungen müssen für den Startwert x_0 erfüllt sein? Berechnen Sie für den Startwert $x_0 = 1$ Näherungen x_1, \dots, x_4 für $\sqrt{2}$. Hier dürfen Sie einen Taschenrechner verwenden.

Lösung zu Aufgabe 3

- (i) Zu zeigen: Es existiert ein $\xi \in (a, b)$ mit $f^{(n)}(\xi) = 0$.
 Vollständige Induktion über n :
 IA $n = 1$: f ist differenzierbar und besitzt zwei verschiedene Nullstellen. Dann gilt nach dem Satz von Rolle:

$$\exists \xi \in [a, b] \text{ mit } f'(\xi) = 0$$

$A(n) \Rightarrow A(n+1)$: Sei f eine $n+1$ mal diff'bare Funktion mit $n+2$ verschiedene Nullstellen. Dann ist f' eine n mal diff'bare Funktion mit $n + 1$ verschiedenen Nullstellen (Satz von Rolle). Daraus folgt nach Induktionvoraussetzung: Es existiert ein $\xi \in [a, b]$ mit $f^{n+1}(\xi) = (f')^{(n)}(\xi) = 0$.

- (ii) Taylorpolynom dritten Grades mit Entwicklungspunkt 0 von $f(x) = e^{-x^2/2}$:

$$T(x) = 1 - \frac{1}{2}x^2$$

Lagrangerestglied: $R(x) = \frac{1}{4!}(3e^{-\xi^2/2} - 6\xi^2e^{-\xi^2/2} + \xi^4e^{-\xi^2/2})x^4$, $\xi \in (0, 1)$ Damit ist dann die Näherung des Integrals:

$$\int_0^1 f(x)dx \approx \int_0^1 (1 - \frac{1}{2}x^2)dx = 1 - \frac{1}{6} = \frac{5}{6}$$

Fehlerabschätzung:

$$|\int_0^1 (f(x) - T(x))dx| = |\int_0^1 R(x)dx| \leq \int_0^1 |R(x)|dx \leq \frac{1}{24} \int_0^1 x^4 dx \max_{y \in [0,1]} |3 - 6y^2 + y^4| = \frac{1}{40} = 0,025$$

da $g(y) = 3 - 6y^2 + y^4$ als Extremstellen $y_{1,2} = \pm\sqrt{3}$, $y_3 = 0$ hat und $g(0) = 3$ und $g(1) = -2$, also ist das $\max_{y \in [0,1]} |3 - 6y^2 + y^4| = 3$.

(iii) Mit $x_{n+1} = \frac{1}{2}(x_n - \frac{c}{x_n})$ kann man iterativ die Wurzel von c berechnen.

1.Schritt: Startwert x_0

Sei zunächst $f : \mathbb{R} \setminus \{0\} \rightarrow \mathbb{R}$ mit $f(x) = \frac{1}{2}(x - \frac{c}{x})$

$f(x) < 0$ für $x < 0 \Rightarrow$ Für die Konvergenz von (x_k) gegen \sqrt{c} muss x_0 positiv sein

2.Schritt: Minimum und Fixpunkt

$$f'(x) = \frac{1}{2} - \frac{1}{2} \frac{c}{x^2} = \frac{1}{2x^2}(x^2 - c)$$

Daraus folgt, dass f an der Stelle $x = \sqrt{c}$ ein Minimum hat.

$f(\sqrt{c}) = \frac{1}{2}(\sqrt{c} + \frac{c}{\sqrt{c}}) = \sqrt{c}$ damit ist \sqrt{c} ein Fixpunkt und es gilt: für $x_0 > 0$ folgt $x_1 \geq \sqrt{c}$.

3.Schritt: Banachscher Fixpunktsatz

Sei nun $f : [\sqrt{c}, \infty) \rightarrow [\sqrt{c}, \infty)$ ($[\sqrt{c}, \infty)$ vollständig)

$$\text{Kontraktion: } |f(x) - f(y)| = \frac{1}{2}|x - y + c(\frac{1}{x} - \frac{1}{y})| = \frac{1}{2}|1 - \underbrace{\frac{c}{xy}}_{<1}| |x - y| \leq \frac{1}{2}|x - y|$$

damit ist f eine Kontraktion mit $\frac{1}{2}$ als Konstante.

Nach dem Banachschen Fixpunktsatz konvergiert (x_k) gegen \sqrt{c} .

(Konvergenzgeschwindigkeit: $|x - x_n| \leq (\frac{1}{2})^{n-1}|x - x_1|$)

$$x_0 = 1$$

$$x_1 = 1,5$$

$$x_2 = 1,416667$$

$$x_3 = 1,414216$$

$$x_4 = 1,414214$$

Aufgabe 4 (Regula Falsi)

(4 Punkte)

Bei der Bestimmung einer Nullstelle mittels Regula Falsi wird zu einer stetigen Funktion $f : I \rightarrow \mathbb{R}$, $I \subset \mathbb{R}$ Intervall, und einem Intervall $[a, b] \subset I$, $a < b$, ein

$$\xi = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

berechnet und als nächstes Intervall eines der beiden Intervalle $[a, \xi]$ und $[\xi, b]$ verwendet, für das $f(a)f(\xi) < 0$ bzw. $f(\xi)f(b) < 0$ gilt. Durch mehrmalige Anwendung liegt wieder eine Intervallschachtelung vor. Führen Sie vier Schritte der Regula Falsi mit den Startwerten $a = 0$ und $b = 1$ zur näherungsweisen Bestimmung der Lösung der nichtlinearen Gleichung $x = \cos(x)$ durch. Sie dürfen einen Taschenrechner verwenden.

Lösung zu Aufgabe 4

$$\begin{aligned} a_0 &= 0, & b_0 &= 1 \\ a_1 &= 0,6851, & b_1 &= 1 \\ a_2 &= 0,7363, & b_2 &= 1 \\ a_3 &= 0,7389, & b_3 &= 1 \\ a_4 &= 0,7391 & b_4 &= 1 \end{aligned}$$

(Der exakte Wert der Nullstelle ist 0,739085...)

Aufgabe 5 (Newton-Iteration)**(2+4 Punkte)**

Gegeben Sei die Funktion

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad f(x_1, x_2) = \begin{pmatrix} 2x_1(1 + x_2^2) - 2 \\ 2x_1^2x_2 \end{pmatrix}$$

- (i) Berechnen Sie die Jacobi-Matrix von f .
- (ii) Führen Sie zwei Schritte des Newton-Verfahrens zur Bestimmung einer Nullstelle von f mit dem Startvektor $x^{(0)} = (\frac{3}{2}, \frac{1}{2})^T$ durch.

Lösung zu Aufgabe 5

- (i) Die Jacobi-Matrix von f lautet folgendermaßen:

$$J_f(x_1, x_2) = \begin{pmatrix} 2(1 + x_2^2) & 4x_1x_2 \\ 4x_1x_2 & 2x_1^2 \end{pmatrix}$$

- (ii) Die Iterationsvorschrift lautet:

$$x^{(k+1)} = x^{(k)} - (f'(x^{(k)}))^{-1} \cdot f(x^{(k)})$$

Mit dem Startvektor $x^{(0)} = (\frac{3}{2}, \frac{1}{2})^T$ kann man die Jacobi-Matrix für $k = 0$ berechnen:

$$J_f(x^{(0)}) = \begin{pmatrix} \frac{5}{2} & 3 \\ 3 & \frac{9}{2} \end{pmatrix}$$

Somit sieht die Inverse wie folgt aus:

$$(J_f(x^{(0)}))^{-1} = \begin{pmatrix} 2 & -\frac{4}{3} \\ -\frac{4}{3} & \frac{10}{9} \end{pmatrix}$$

Damit kann man $x^{(1)}$ berechnen:

$$x^{(1)} = \begin{pmatrix} \frac{3}{2} \\ \frac{1}{2} \end{pmatrix} - \begin{pmatrix} 2 & -\frac{4}{3} \\ -\frac{4}{3} & \frac{10}{9} \end{pmatrix} \cdot \begin{pmatrix} \frac{7}{4} \\ \frac{9}{4} \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{3} \end{pmatrix}$$

Damit sieht die Jacobi-Matrix für $k = 1$ folgendermaßen aus:

$$J_f(x^{(1)}) = \begin{pmatrix} \frac{20}{9} & \frac{4}{3} \\ \frac{4}{3} & 2 \end{pmatrix}$$

Somit sieht die Inverse wie folgt aus:

$$(J_f(x^{(1)}))^{-1} = \begin{pmatrix} \frac{3}{4} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{5}{6} \end{pmatrix}$$

Dadurch erhält man $x^{(2)}$:

$$x^{(2)} = \begin{pmatrix} 1 \\ \frac{1}{3} \end{pmatrix} - \begin{pmatrix} \frac{3}{4} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{5}{6} \end{pmatrix} \cdot \begin{pmatrix} \frac{2}{9} \\ \frac{2}{3} \end{pmatrix} = \begin{pmatrix} \frac{7}{6} \\ -\frac{1}{9} \end{pmatrix}$$

Aufgabe 6 (Newtonverfahren bei mehrfachen Nullstellen) (4+4 Punkte)

Es sei $f : [a, b] \rightarrow \mathbb{R}$ eine $(\ell + 1)$ -mal stetige differenzierbare Funktion und $x^* \in (a, b)$ sei eine ℓ -fache Nullstelle von f , d. h. es sei $f(x^*) = f'(x^*) = \dots = f^{(\ell-1)}(x^*) = 0$ und $f^{(\ell)}(x^*) \neq 0$.

(i) Zeigen Sie, dass für $\ell \geq 2$ das Newtonverfahren unter der Zusatzvoraussetzung $x_n \neq x^*$ in einer Umgebung von x^* die maximale Konvergenzordnung 1 hat.

(ii) Zeigen Sie, dass das modifizierte Newton-Verfahren

$$x_{n+1} = x_n - \ell \frac{f(x_n)}{f'(x_n)} \quad (n \in \mathbb{N}_0)$$

unter der Zusatzvoraussetzung $x_n \neq x^*$ in einer Umgebung von x^* mindestens die Konvergenzordnung 2 hat.

Lösung zu Aufgabe 6

(i) Für $f(x)$ gilt:

$$\begin{aligned} f(x) &= \sum_{k=0}^{\ell} \frac{1}{k!} f^{(k)}(x^*) (x - x^*)^k + \frac{1}{(\ell + 1)!} f^{(\ell+1)}(\xi(x)) (x - x^*)^{\ell+1} \\ &= \underbrace{\frac{1}{\ell!} f^{(\ell)}(x^*) (x - x^*)^{\ell}}_{=:a} + \underbrace{\frac{1}{(\ell + 1)!} f^{(\ell+1)}(\xi(x)) (x - x^*)^{\ell+1}}_{=:g(x)}, \quad \xi(x) \in (x^*, x) \end{aligned}$$

g ist beschränkt, da $f^{(\ell+1)}$ beschränkt ist.

Die erste Ableitung von f sieht dann folgendermaßen aus:

$$\begin{aligned} f'(x) &= \sum_{k=0}^{\ell-1} \frac{1}{k!} (f')^{(k)}(x^*) (x - x^*)^k + \frac{1}{\ell!} (f')^{(\ell)}(\xi_1(x)) (x - x^*)^{\ell} \\ &= \underbrace{\frac{1}{(\ell - 1)!} f^{(\ell)}(x^*) (x - x^*)^{\ell-1}}_{=:l \cdot a} + \underbrace{\frac{1}{\ell!} f^{(\ell+1)}(\xi_1(x)) (x - x^*)^{\ell}}_{=:h(x)} \end{aligned}$$

Wiederum gilt, dass $h(x)$ beschränkt ist, da $f^{(\ell+1)}$ beschränkt ist.

$$\begin{aligned} x_{n+1} - x^* &= x_n - x^* - \frac{f(x_n)}{f'(x_n)} = x_n - x^* - \frac{a(x_n - x^*)^{\ell} + g(x_n)(x_n - x^*)^{\ell+1}}{la(x_n - x^*)^{\ell-1} + h(x_n)(x_n - x^*)^{\ell}} \\ &= (x_n - x^*) \left(1 - \frac{a + g(x_n)(x_n - x^*)}{la + h(x_n)(x_n - x^*)} \right) \end{aligned}$$

Des Weiteren kann man den zweiten Faktor folgendermaßen abschätzen:

$$\begin{aligned} \left| 1 - \frac{a + g(x_n)(x_n - x^*)}{la + h(x_n)(x_n - x^*)} \right| &\leq 1 + (|a| + C_g|b - a|) \left| \frac{1}{|la| - |h(x_n)||x_n - x^*|} \right| \\ &\leq \underbrace{1 + (|a| + C_g|b - a|) \frac{2}{|la|}}_{\text{konstant}} \quad \text{falls } |x_n - x^*| \text{ genügend klein} \end{aligned}$$

Deshalb besitzt das Verfahren die Konvergenzordnung 1.

Angenommen das Verfahren besäße eine höhere Konvergenzordnung als 1, dann würde folgendes gelten:

Sei x_n eine Folge die gegen x^* für $n \rightarrow \infty$ konvergiert. Dann gilt:

$$0 \neq 1 - \frac{1}{l} = \lim_{x_n \rightarrow x^*} \left| 1 - \frac{a + g(x_n)(x_n - x^*)}{la + h(x_n)(x_n - x^*)} \right| = \lim_{x_n \rightarrow x^*} \frac{|x_{n+1} - x^*|}{|x_n - x^*|} \leq \lim_{x_n \rightarrow x^*} C|x_n - x^*|^{p-1} = 0$$

Dies ist aber ein Widerspruch.

(ii) Es gilt:

$$\begin{aligned} x_{n+1} - x^* &= x_n - x^* - l \frac{f(x_n)}{f'(x_n)} = x_n - x^* - \frac{la(x_n - x^*)^l + lg(x_n)(x_n - x^*)^{l+1}}{la(x_n - x^*)^{l-1} + h(x_n)(x_n - x^*)^l} \\ &= (x_n - x^*) \left(1 - \frac{la + lg(x_n)(x_n - x^*)}{la + h(x_n)(x_n - x^*)} \right) \\ &= (x_n - x^*) \frac{la + h(x_n)(x_n - x^*) - la - lg(x_n)(x_n - x^*)}{la + h(x_n)(x_n - x^*)} \\ &= (x_n - x^*)^2 \frac{h(x_n) - lg(x_n)}{la + h(x_n)(x_n - x^*)} \end{aligned}$$

$$\begin{aligned} \left| \frac{h(x_n) - lg(x_n)}{la + h(x_n)(x_n - x^*)} \right| &\leq (C_h + l \cdot C_g) \left| \frac{1}{|la| - |h(x_n)||x_n - x^*|} \right| \\ &\leq \underbrace{(C_h + l \cdot C_g) \frac{2}{|la|}}_{\text{konstant}} \quad \text{falls } |x_n - x^*| \text{ genügend klein} \end{aligned}$$

Deshalb hat das modifizierte Newtonverfahren die Konvergenzordnung 2.

Aufgabe 7 (Lineare Differenzgleichung)

(2+4+3 Punkte)

(i) Seien $\mu_0, \mu_1, \dots, \mu_{m-1}$ reelle Zahlen und (a_n) eine reelle Folge, die durch die Anfangswerte a_0, a_1, \dots, a_{m-1} und durch die lineare Differenzgleichungen

$$a_{n+m} = \sum_{i=0}^{m-1} \mu_i a_{n+i}$$

für $n \in \mathbb{N}_0$ eindeutig bestimmt ist. Nennen Sie die Matrix $A \in \mathbb{R}^{m \times m}$, die für alle $n \in \mathbb{N}_0$ den Vektor \hat{a}_n auf den Vektor \hat{a}_{n+1} abbildet, wobei

$$\hat{a}_k := \begin{pmatrix} a_k \\ \vdots \\ a_{k+m-1} \end{pmatrix}$$

für $k \in \mathbb{N}_0$ definiert ist. D. h. es gilt $\hat{a}_{n+1} = A \hat{a}_n$ und damit $\hat{a}_n = A^n \hat{a}_0$.

- (ii) Gegeben sei eine Folge (a_n) mit der Eigenschaft $a_{n+2} = a_{n+1} + a_n$ für alle $n \in \mathbb{N}_0$ und den Startwerten $a_0 = 1$ und $a_1 = 3$. Geben Sie eine explizite Darstellung von a_n an.
Hinweis: Geben Sie zunächst wie in Aufgabe 10 (i) für $m = 2$ die Matrix A an, für die $\hat{a}_{n+1} = A \hat{a}_n$ und damit auch $\hat{a}_n = A^n \hat{a}_0$ gilt. Diagonalisieren Sie A . Wie sieht dann A^n aus?
- (iii) Gegeben sei eine Folge (a_n) mit der Eigenschaft $a_{n+2} = a_{n+1} \cdot a_n$ für alle $n \in \mathbb{N}_0$ und den Startwerten $a_0 = 1$ und $a_1 = 2$. Wie kann man analog zu Aufgabe 10 (ii) die explizite Darstellung von a_n gewinnen? Geben Sie diese an.

Lösung zu Aufgabe 7

- (i) Man erhält für \hat{a}_{n+1} , \hat{a}_n und A folgende Vektoren bzw. Matrix:

$$\underbrace{\begin{pmatrix} a_{k+1} \\ a_{k+2} \\ a_{k+3} \\ \vdots \\ a_{k+m} \end{pmatrix}}_{\hat{a}_{k+1}} = \underbrace{\begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & 1 \\ \mu_0 & \mu_1 & \mu_2 & \dots & \mu_{m-1} \end{pmatrix}}_A \underbrace{\begin{pmatrix} a_k \\ a_{k+1} \\ a_{k+2} \\ \vdots \\ a_{k+m-1} \end{pmatrix}}_{\hat{a}_k}$$

- (ii) Dadurch das $a_{n+2} = a_n + a_{n+1}$ gilt, kann man folgern:

$$\hat{a}_{n+1} = \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}}_{=:A} \hat{a}_n \quad \text{mit} \quad \hat{a}_k = (a_k, a_{k+1})^T$$

Das charakteristische Polynom von A lautet dann folgendermaßen:

$$\det \begin{pmatrix} -\lambda & 1 \\ 1 & 1 - \lambda \end{pmatrix} = \lambda^2 - \lambda - 1 \stackrel{!}{=} 0$$

Daraus ergeben sich dann folgende Eigenwerte:

$$\lambda_1 = \frac{1 + \sqrt{5}}{2}, \quad \lambda_2 = \frac{1 - \sqrt{5}}{2}$$

Nun lässt sich \hat{a}_n darstellen:

$$\hat{a}_n = A^n \hat{a}_0 = \left(B^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} B \right)^n \hat{a}_0 = B^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}^n B \cdot \hat{a}_0 = B^{-1} \begin{pmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{pmatrix} B \cdot \hat{a}_0$$

Nun kann man a_n berechnen:

$$\begin{aligned} a_n &= c_1 \lambda_1^n + c_2 \lambda_2^n \\ a_0 &= c_1 + c_2 = 1 \quad (I) \\ a_1 &= c_1 \frac{1 + \sqrt{5}}{2} + c_2 \frac{1 - \sqrt{5}}{2} = 3 \quad \Leftrightarrow \underbrace{c_1 + c_2}_{=1} + (c_1 - c_2) \sqrt{5} = 6 \\ &\Rightarrow c_1 - c_2 = \sqrt{5} \quad (II) \end{aligned}$$

Durch Addition der beiden Gleichungen ergeben sich für c_1 und c_2 folgende Werte:

$$c_1 = \frac{1 + \sqrt{5}}{2} \quad c_2 = \frac{1 - \sqrt{5}}{2}$$

Daraus lässt sich dann a_n bestimmen:

$$a_n = \left(\frac{1 + \sqrt{5}}{2}\right)^{n+1} + \left(\frac{1 - \sqrt{5}}{2}\right)^{n+1}$$

(iii) Durch Logarithmieren der gegebenen Gleichung erhält man:

$$\begin{aligned} a_{n+2} &= a_n \cdot a_{n+1} \\ \underbrace{\log_2 a_{n+2}}_{=:b_{n+2}} &= \log_2(a_n \cdot a_{n+1}) = \underbrace{\log_2(a_n)}_{=:b_n} + \underbrace{\log_2(a_{n+1})}_{=:b_{n+1}} \end{aligned}$$

Nun kann man Aufgabe 10 (ii) anwenden und erhält analog dazu:

$$\begin{aligned} b_n &= c_1 \lambda_1^n + c_2 \lambda_2^n \\ b_0 &= c_1 + c_2 = 0 \quad (I) \\ b_1 &= c_1 \frac{1 + \sqrt{5}}{2} + c_2 \frac{1 - \sqrt{5}}{2} = 1 \quad \Leftrightarrow \underbrace{c_1 + c_2}_{=0} + (c_1 - c_2)\sqrt{5} = 2 \\ &\Rightarrow c_1 - c_2 = \frac{2}{\sqrt{5}} \quad (II) \end{aligned}$$

Durch Umformulierung der beiden Gleichungen ergeben sich für c_1 und c_2 folgende Werte:

$$c_1 = \frac{1}{\sqrt{5}} \quad c_2 = -\frac{1}{\sqrt{5}}$$

Dadurch ergibt sich für b_n :

$$b_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2}\right)^n - \left(\frac{1 - \sqrt{5}}{2}\right)^n \right]$$

Weshalb man dann für a_n erhält:

$$a_n = 2^{\frac{1}{\sqrt{5}}} \left[\left(\frac{1 + \sqrt{5}}{2}\right)^n - \left(\frac{1 - \sqrt{5}}{2}\right)^n \right]$$

Aufgabe 8 (Symmetry of orthogonal polynomials)

(4 Punkte)

Let (p_n) be a sequence of orthogonal polynomials with respect to the scalar product

$$(f, g)_\omega = \int_{-a}^a \omega(x) f(x) g(x) dx$$

provided by a symmetric, positive and integrable weight function $\omega : [-a, a] \rightarrow \mathbb{R}$. Show that $p_k(-x) = (-1)^k p_k(x)$ holds for all $x \in [-a, a]$ and $k \in \mathbb{N}_0$.

Lösung zu Aufgabe 8

Wir erhalten die Aussage $p_n(-x) = (-1)^n p_n(x)$ durch vollständige Induktion

n=0: $p_0(x) = \text{const.} \Rightarrow$ Aussage klar.

Induktionsschritt: Sei $f_n(x) = p_n(-x)$, dann gilt für $k = 0, \dots, n-1$

$$\begin{aligned} \int_{-a}^a \omega(x) f_n(x) p_k(x) dx &= \int_{-a}^a \omega(-x) p_n(-x) p_k(x) dx = \int_{-a}^a \omega(x) p_n(x) p_k(-x) dx \\ &\stackrel{IV}{=} (-1)^k \int_{-a}^a \omega(x) p_n(x) p_k(x) dx = 0. \end{aligned}$$

Also folgt $f_n \in \mathbb{P}_n \cap (\mathbb{P}_{n-1})^\perp$ und wir erhalten $f_n = \lambda p_n$ und somit $p_k(-x) = \lambda p_k(x)$

Sei

$$p_n(x) = \sum_{l=0}^n a_l x^l, \quad \text{also gilt} \quad \sum_{l=0}^n a_l (-1)^l x^l = \lambda a_l x^l.$$

Wir erhalten $a_n(-1)^n = \lambda a_n$ durch Vergleich des führenden Koeffizienten. Damit folgt $\lambda = (-1)^n$ und die Aussage gilt.

Aufgabe 9 (Tschebyscheff-Polynome)

(4+2+4+3 Punkte)

Gegeben sei für $k \geq 2$ und $x \in \mathbb{R}$ die Drei-Term-Rekursion der Tschebyscheff-Polynome:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), \quad k \geq 2, \quad x \in \mathbb{R}.$$

Zeigen Sie:

(i) Es gilt $T_k(x) = \cos(k \arccos(x))$ für $x \in [-1, 1]$.

(ii) Die Nullstellen von T_n sind

$$x_k := \cos\left(\frac{2k-1}{2n}\pi\right), \quad (k = 1, \dots, n).$$

(iii) Die Tschebyscheff-Polynome besitzen die globale Darstellung

$$T_k(x) = \frac{1}{2} \left((x + \sqrt{x^2 - 1})^k + (x - \sqrt{x^2 - 1})^k \right), \quad \text{wobei } x \in \mathbb{R}.$$

(iv) Zeigen Sie, dass die Quadratur

$$\int_{-1}^1 \frac{q(x)}{\sqrt{1-x^2}} = \omega_1 q(x_1) + \dots + \omega_n q(x_n) \quad \text{für alle } q \in \mathbb{P}_{2n-1}$$

zu den Nullstellen x_1, \dots, x_n des Tschebyscheff-Polynoms T_n die Gewichte $\omega_k = \frac{\pi}{n}$ besitzt.

Lösung zu Aufgabe 9

(i) Sei $f_k(x) = \cos(k \arccos(x))$.

Induktionsanfang:

$$f_0(x) = 1 = T_0(x)$$

$$f_1(x) = x = T_1(x)$$

Induktionsschritt:

$$\begin{aligned} T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x) = 2x \cos(k \arccos(x)) - \cos((k-1) \arccos(x)) \\ &= \cos(\arccos(x)) \cos(k \arccos(x)) + x \cos(k \arccos(x)) - \cos((k-1) \arccos(x)) \\ &= \cos((k+1) \arccos(x)) + \sin(\arccos(x)) \sin(k \arccos(x)) + x \cos(k \arccos(x)) \\ &\quad - \cos(k \arccos(x)) \cos(\arccos(x)) + \sin(-\arccos(x)) \sin(k \arccos(x)) \\ &= f_{k+1}(x) \end{aligned}$$

(ii) Berechnung der Nullstellen

$$\begin{aligned}
T_k(x) = 0 &\Leftrightarrow n \arccos(x) = \frac{1}{2}(2k-1)\pi, \quad k \in \mathbb{Z} \\
&\Leftrightarrow \arccos(x) = \frac{2k-1}{2n}\pi, \quad k \in \mathbb{Z} \\
&\Leftrightarrow x = \cos\left(\frac{2k-1}{2n}\pi\right), \quad k \in \mathbb{Z} \\
&\Leftrightarrow x = \cos\left(\frac{2k-1}{2n}\pi\right), \quad k \in \{1, 2, \dots, n\}
\end{aligned}$$

(iii) Sei $g_k(x) = \frac{1}{2}((x + \sqrt{x^2 - 1})^k + (x - \sqrt{x^2 - 1})^k)$ Vollständige Induktion über k :Induktionsanfang: $g_0(x) = 1 = T_0(x)$, $g_1(x) = x = T_1(x)$ Induktionsschritt: Wir definieren uns $a := x + \sqrt{x^2 - 1}$, $b := x - \sqrt{x^2 - 1}$ und wir erhalten $a + b = 2x$ sowie $a \cdot b = 1$.

Somit ergibt sich

$$\begin{aligned}
T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x) = (a+b)\frac{1}{2}(a^k + b^k) - \frac{1}{2}(a^{k-1} + b^{k-1}) \\
&= \frac{1}{2}(a^{k+1} + b^{k+1}) + \frac{1}{2}(a^k b + a b^k - a^{k-1} - b^{k-1}) \\
&= \frac{1}{2}(a^{k+1} + b^{k+1}) = g_{k+1}(x)
\end{aligned}$$

und die Behauptung ist gezeigt.

(iv) Laut Skript sind die Gewichte durch

$$\begin{aligned}
\omega_k &= \frac{k_n}{k_{n-1}} \frac{h_{n-1}}{T'_n(x_k) T_{n-1}(x_k)} \\
&= \frac{2}{n} \cdot \frac{\sqrt{1-x_k^2}}{\sin(n \arccos(x_k))} \cdot \frac{1}{\cos((n-1) \arccos(x_k))} \int_{-1}^1 \frac{\cos^2((n-1) \arccos(x))}{\sqrt{1-x^2}} dx \\
&= \frac{2}{n} \cdot \frac{\sin(\frac{2k-1}{2n}\pi)}{\sin(\frac{2k-1}{2}\pi)} \cdot \frac{1}{\cos(\frac{2k-1}{2}\pi - \frac{2k-1}{2n}\pi)} \cdot \int_0^\pi \cos^2((n-1)t) dt \\
&= \frac{2}{n} \cdot \underbrace{\frac{\sin(\frac{2k-1}{2n}\pi)}{\sin(\frac{2k-1}{2}\pi)}}_{(-1)^{k-1}} \cdot \underbrace{\frac{1}{\sin(\frac{2k-1}{2}\pi) \sin(\frac{2k-1}{2n}\pi)}}_{(-1)^{k-1}} \cdot \frac{\pi}{2} = \frac{\pi}{n}
\end{aligned}$$

gegeben.

Aufgabe 10 (Legendre-Polynome)**(5+4 Punkte)**(i) Zeigen Sie mit Hilfe der Rodrigues-Formel $P_n(x) = \frac{1}{(-2)^n n!} \frac{d^n}{dx^n} (1-x^2)^n$, dass für die Legendre-Polynome P_n

$$\int_{-1}^1 P_m(x) P_n(x) dx = \begin{cases} 0 & \text{falls } m \neq n \\ \frac{2}{2n+1} & \text{falls } m = n \end{cases} \quad (m, n \in \mathbb{N}_0)$$

gilt. Sie dürfen dabei die Formel

$$\int_{-1}^1 (1-x^2)^n dx = \frac{2^{n+1}n!}{3 \cdot 5 \cdot 7 \cdot \dots \cdot (2n+1)}$$

verwenden.

Hinweis: Integrieren Sie mehrmals partiell.

(ii) Zeigen Sie, dass für $n \in \mathbb{N}$ die Rekursionsformel

$$(1-x^2)P'_n(x) = \frac{n(n+1)}{2n+1}(P_{n-1}(x) - P_{n+1}(x))$$

für die Legendre-Polynome P_n gilt.

Hinweis: Zeigen Sie, dass die Differenz aus linker und rechter Seite orthogonal zu allen P_k mit $k = 0, \dots, n+1$ ist. Das Verhältnis von den führenden Koeffizienten der Legendre-Polynome P_n und P_{n+1} erhalten Sie aus der Rekursionsformel

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x),$$

die Sie nicht beweisen müssen.

Lösung zu Aufgabe 10

(i) 1. Schritt. Für $k \leq m$ existiert ein Polynom p abhängig von k und m mit

$$\frac{\partial^k}{\partial x^k}(1-x^2)^m = (1-x^2)^{m-k}p(x)$$

Vollständige Induktion über k :

Induktionsanfang für $k=0$ ist klar.

Induktionsschritt:

$$\begin{aligned} \frac{\partial^{k+1}}{\partial x^{k+1}}(1-x^2)^m &= \frac{\partial}{\partial x}((1-x^2)^{m-k}p(x)) \\ &= (m-k)(1-x^2)^{m-k-1}(-2x) \cdot p(x) + (1-x^2)^{m-k}p'(x) \\ &= (1-x^2)^{m-(k+1)} \underbrace{((m-k)p(x) + (1-x^2)p'(x))}_{\text{Polynom}} \end{aligned}$$

2. Schritt. Durch mehrmalige Anwendung der partiellen Integration erhält man

$$\begin{aligned} &\int_{-1}^1 \frac{\partial^m}{\partial x^m}(1-x^2)^m \frac{\partial^k}{\partial x^k}(1-x^2)^k dx \\ &= \underbrace{\frac{\partial^{m-1}}{\partial x^{m-1}}(1-x^2)^m \frac{\partial^k}{\partial x^k}(1-x^2)^k \Big|_{-1}^1}_{=0} - \int_{-1}^1 \frac{\partial^{m-1}}{\partial x^{m-1}}(1-x^2)^m \frac{\partial^{k+1}}{\partial x^{k+1}}(1-x^2)^k dx \\ &= \dots = (-1)^m \int_{-1}^1 (1-x^2)^m \frac{\partial^{n+m}}{\partial x^{n+m}}(1-x^2)^n dx =: a_{n,m}, \end{aligned}$$

wobei stets der erste Summand aufgrund des ersten Schrittes gleich Null ist.

3.Schritt 1.Fall $m > n$ (analog $m < n$)

$$\begin{aligned}(P_m, P_n) &= \frac{1}{(-2)^m m! (-2)^n n!} a_{n,m} \\ &= \frac{1}{(-2)^m m! (-2)^n n!} (-1)^m \int_{-1}^1 (1-x^2)^m \underbrace{\frac{\partial^{n+m}}{\partial x^{n+m}} (1-x^2)^n dx}_{=0} = 0\end{aligned}$$

2.Fall $m=n$

$$\begin{aligned}(P_n, P_n) &= \left(\frac{1}{(-2)^n n!} \right)^2 a_{n,n} = \left(\frac{1}{(-2)^n n!} \right)^2 (-1)^n \int_{-1}^1 (1-x^2)^n \frac{\partial^{2n}}{\partial x^{2n}} (1-x^2)^n dx \\ &= \left(\frac{1}{(-2)^n n!} \right)^2 (-1)^n (2n)! (-1)^n \int_{-1}^1 (1-x^2)^n dx \\ &= \frac{(2n)! \cdot n! \cdot 2^{n+1}}{2^{2n} \cdot (n!)^2 \cdot 3 \cdot 5 \cdot \dots \cdot (2n+1)} = \frac{2}{2n+1} \cdot \frac{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2n}{2^n \cdot n!} = \frac{2}{2n+1}\end{aligned}$$

$$(ii) \underbrace{(1-x^2)P_n'(x)}_{=LHS} = \underbrace{\frac{n(n+1)}{2n+1}(P_{n-1}(x) - P_{n+1}(x))}_{=RHS}$$

1. $LHS - RHS \in \mathbb{P}_{n+1}$

2. (a) Sei $j \leq n-2$
Wir erhalten

$$(RHS, P_j) = \frac{n(n+1)}{2n+1} \left[\int_{-1}^1 P_{n-1}(x) P_j(x) dx - \int_{-1}^1 P_{n+1}(x) P_j(x) dx \right] = 0$$

und

$$\begin{aligned}(LHS, P_j) &= \int_{-1}^1 (1-x^2) P_n'(x) P_j(x) dx \\ &= \underbrace{(1-x^2) P_n(x) P_j(x) \Big|_{-1}^1}_{=0} - \int_{-1}^1 P_n(x) \underbrace{(1-x^2) P_j'(x) dx}_{\in \mathbb{P}_{n-1}} \\ &\quad + \int_{-1}^1 P_n(x) \underbrace{2x P_j(x) dx}_{\in \mathbb{P}_{n-1}} \\ &= 0\end{aligned}$$

(b) Weiterhin gilt:

$$(RHS, P_n) = 0$$

$$\begin{aligned}
(LHS, P_n) &= \int_{-1}^1 (1-x^2)P_n'P_n dx = (1-x^2)P_n^2(x) \Big|_{-1}^1 - \underbrace{\int_{-1}^1 (1-x^2)P_nP_n' dx}_{=(LHS, P_n)} \\
&\quad + \int_{-1}^1 2xP_n^2 dx \\
&\Rightarrow 2(LHS, P_n) = 2 \int_{-1}^1 xP_n^2 dx = 0
\end{aligned}$$

da nach Aufgabe 11 P_n^2 eine gerade Funktion ist und xP_n^2 damit ungerade ist. Es folgt $(LHS, P_n) = 0$.

(c) Hinweis: vergleicht man die führenden Koeffizienten in der Rekursionsformel

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$

so folgt

$$(n+1)k_{n+1} = (2n+1)k_n \Rightarrow \frac{k_n}{k_{n+1}} = \frac{n+1}{2n+1}.$$

Es folgt nun mit dem Hinweis und der nachfolgenden Anmerkung

$$(RHS, P_{n-1}) = \frac{n(n+1)}{2n+1} (P_{n-1}, P_{n-1}) \stackrel{13(i)}{=} \frac{n(n+1)}{2n+1} \cdot \frac{2}{2n-1}$$

$$\begin{aligned}
(LHS, P_{n-1}) &= (1-x^2)P_nP_{n-1} \Big|_{-1}^1 - \int_{-1}^1 [(1-x^2)P_{n-1}]' P_n dx \\
&= - \int_{-1}^1 (1-x^2)P_nP_{n-1}' dx + 2 \int_{-1}^1 xP_nP_{n-1} dx \\
&= (n-1) \frac{k_{n-1}}{k_n} \int_{-1}^1 P_n^2 dx + 2 \frac{k_{n-1}}{k_n} \int_{-1}^1 P_n^2 dx \\
&= (n+1) \frac{k_{n-1}}{k_n} \cdot \frac{2}{2n+1} = (n+1) \frac{n}{2n-1} \cdot \frac{2}{2n+1}
\end{aligned}$$

wobei k_l der führende Koeffizient von P_l ist.

Anmerkung:

$$(1-x^2)P_{n-1}' = -(n-1) \frac{k_{n-1}}{k_n} P_n + q_1 \quad \text{mit } q_1 \in \mathbb{P}_{n-1}$$

$$xP_{n-1} = \frac{k_{n-1}}{k_n} P_n + q_2 \quad \text{mit } q_2 \in \mathbb{P}_{n-1}$$

(d)

$$(RHS, P_{n+1}) = -\frac{n(n+1)}{2n+1} \cdot \frac{2}{2n+3}$$

$$\begin{aligned} (LHS, P_{n+1}) &= \int_{-1}^1 (1-x^2)P_n'(x)P_{n+1}(x)dx = -n \cdot \frac{k_n}{k_{n+1}} \cdot \frac{2}{2n+3} \\ &= -n \cdot \frac{n+1}{2n+1} \cdot \frac{2}{2n+3} \end{aligned}$$

Anmerkung:

$$(1-x^2)P_n' = -n \frac{k_n}{k_{n+1}} P_{n+1} + q_3 \quad \text{mit } q_3 \in \mathbb{P}_n$$

insgesamt: LHS-RHS ist orthogonal zu einer Basis von \mathbb{P}_{n+1} . Es folgt $LHS - RHS = 0$.

Aufgabe 11 (Nullstellen von Orthogonalpolynome)

(3 Punkte)

Beweisen Sie Satz 2.5.1: Sei $Ap = xp + r$ das durch die Drei-Term-Rekursion für Orthogonalpolynome entstehende Gleichungssystem (siehe 2.16 im Skript, Kapitel 2). Dann sind die Nullstellen von p_{n+1} die Eigenwerte von A .

Lösung zu Aufgabe 11

Sei x_k eine Nullstelle von p_{k+1} . Dann ist x_k Eigenwert von A , da $Ap = xp$ mit $p \neq 0$ ($p_0(x_k) \neq 0$). Nach Satz 2.1.16 besitzt p_{n+1} genau $n+1$ verschiedene Nullstellen x_1, \dots, x_{n+1} . Also sind x_1, \dots, x_{n+1} Eigenwerte und es gilt $\text{Spektrum}(A) = \{x_1, \dots, x_{n+1}\}$ wegen $A \in \mathbb{R}^{(n+1) \times (n+1)}$.

Aufgabe 12 (Newton, Lagrange, Aitken-Neville, Hermite)

(6+3 Punkte)

- (i) Berechnen Sie das Interpolationspolynom $P \in \mathbb{P}_3$ in der Darstellung von Lagrange, mittels dividierter Differenzen in der Darstellung von Newton für die Punkte

$$\begin{array}{c|cccc} x_i & -1 & 1 & 2 & 3 \\ \hline y_i & 4 & 0 & -2 & 4 \end{array}$$

sowie den Wert $P(\frac{3}{2})$ mit Hilfe des Algorithmus von Aitken und Neville.

- (ii) Berechnen Sie mit Hilfe dividierter Differenzen das Polynom P bezüglich Newton-Basis vom kleinsten Grad mit den Eigenschaften $P(1) = -2$, $P'(1) = -2$, $P''(1) = 0$, $P(2) = -3$ und $P'(2) = 1$.

Lösung zu Aufgabe 12

- (i) Per Lagrange erhält man:

$$\begin{aligned} P(x) &= 4 \frac{(x-1)(x-2)(x-3)}{(-1-1)(-1-2)(-1-3)} + 0 \frac{(x+1)(x-2)(x-3)}{(1+1)(1-2)(1-3)} \\ &\quad - 2 \frac{(x+1)(x-1)(x-3)}{(2+1)(2-1)(2-3)} + 4 \frac{(x+1)(x-1)(x-2)}{(3+1)(3-1)(3-2)} \\ &= -\frac{1}{6}(x-1)(x-2)(x-3) + \frac{2}{3}(x+1)(x-1)(x-3) + \frac{1}{2}(x+1)(x-1)(x-2) \\ &= x^3 - 2x^2 - 3x + 4 \end{aligned}$$

Man erhält das Polynom

$$p(x) = 4 - 2(x+1) + 0 \cdot (x+1)(x-1) + 1 \cdot (x+1)(x-1)(x-2)$$

mittels dividierter Differenzen.

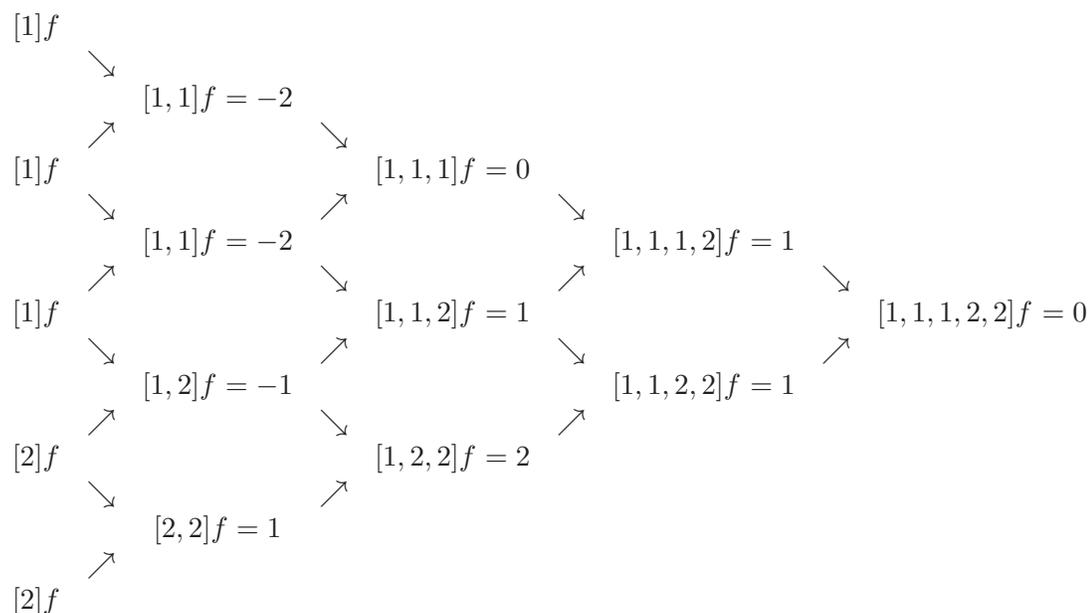
$$\begin{array}{r}
 [-1]f = 4 \\
 \searrow \\
 [-1, 1]f = \frac{0-4}{1-(-1)} = -2 \\
 \nearrow \\
 [1]f = 0 \\
 \searrow \\
 [1, 2]f = \frac{-2-0}{2-1} = -2 \\
 \nearrow \\
 [2]f = -2 \\
 \searrow \\
 [2, 3]f = \frac{4+2}{3-2} = 6 \\
 \nearrow \\
 [3]f = 4
 \end{array}
 \quad
 \begin{array}{r}
 \searrow \\
 [-1, 1, 2]f = 0 \\
 \nearrow \\
 \searrow \\
 [1, 2, 3]f = 4 \\
 \nearrow \\
 [-1, 1, 2, 3]f = 1
 \end{array}$$

Aitken-Neville liefert an der Stelle $\frac{3}{2}$ den Wert $P(\frac{3}{2}) = -\frac{13}{8}$.

$$\begin{array}{r}
 -1 \mid 4 \\
 \searrow \\
 0 + \frac{\frac{3}{2}-1}{+1-(-1)}(0-4) = -1 \\
 \nearrow \\
 1 \mid 0 \\
 \searrow \\
 -2 + \frac{\frac{3}{2}-2}{2-1}(-2-0) = -1 \\
 \nearrow \\
 2 \mid -2 \\
 \searrow \\
 4 + \frac{\frac{3}{2}-3}{3-2}(4-(-2)) = -5 \\
 \nearrow \\
 3 \mid 4
 \end{array}
 \quad
 \begin{array}{r}
 \searrow \\
 -1 + \frac{\frac{3}{2}-2}{2+1}(-1+1) = -1 \\
 \nearrow \\
 -5 + \frac{\frac{3}{2}-3}{3-1}(-5-(-1)) = -2 \\
 \nearrow \\
 -\frac{13}{8}
 \end{array}$$

(ii) Mit Hilfe der dividierten Differenzen erhält man

$$p(x) = -2 - 2(x-1) + (x-1)^3$$

**Aufgabe 13 (Interpolationsfehler und Konvergenzeigenschaften)****(3+3 Punkte)**

- (i) Es seien $f : [a, b] \rightarrow \mathbb{R}$ dreimal stetig differenzierbar und $P \in \mathbb{P}_2$ das zugehörige quadratische Interpolationspolynom zu den Stützstellen $x_i := a + hi$, $i = 0, 1, 2$, $h := (b - a)/2$. Zeigen Sie:

$$\|f - P\|_\infty \leq \frac{\sqrt{3}}{27} h^3 \|f'''\|_\infty.$$

- (ii) Die Funktion $f : [0, 1] \rightarrow \mathbb{R}$ sei definiert durch $f(x) := \log(x + 1)$. Zu einer beliebig vorgegebenen Folge von Stützstellenvektoren

$$\left(x_i^{(n)}\right)_{0 \leq i \leq n} \subset [0, 1], \quad n \in \mathbb{N}$$

sei P_n das zugehörige Interpolationspolynom aus \mathbb{P}_n .

Zeigen Sie, daß die Folge der P_n auf $[0, 1]$ gleichmäßig gegen f konvergiert.

Lösung zu Aufgabe 13

- (i) Nach dem Satz 3.2.11 gilt

$$f(x) - P(x) = \frac{1}{3!} (x - a) \left(x - \frac{a+b}{2}\right) (x - b) f'''(\xi)$$

Sei

$$\begin{aligned} g(t) &= (t - h)t(t + h) = t^3 - h^2t \\ g'(t) &= 3t^2 - h^2 \end{aligned}$$

und man erhält

$$x_E = \pm \frac{h}{\sqrt{3}}$$

als Extremstellen, welche man in g einsetzt

$$g(x_E) = \pm \frac{h^3}{3\sqrt{3}} - h^2 \left(\pm \frac{h}{\sqrt{3}}\right) = \mp \frac{2}{3\sqrt{3}} h^3$$

und man kann folgern:

$$\|\omega\|_{[a,b],\infty} = \|g\|_{[-h,h],\infty} = \frac{2}{3\sqrt{3}}h^3$$

Die Abschätzung

$$\|f - P_n\|_\infty \leq \frac{1}{6} \|(x-a)(x-\frac{a+b}{2})(x-b)\|_\infty \|f'''(\xi)\|_\infty = \frac{1}{6} \|g\|_\infty \|f'''(\xi)\|_\infty = \frac{\sqrt{3}}{27} h^3 \|f'''\|_\infty$$

liefert die Behauptung.

(ii) Es gilt

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

und man erhält für $f(x) = \log(x+1)$

$$f^{(n)}(x) = (-1)^{n-1} (n-1)! \frac{1}{(1+x)^n}.$$

Diese Aussage kann induktiv gezeigt werden. Und nun gilt

$$\|f - P_n\|_\infty \leq \underbrace{\left\| \prod_{j=0}^n \underbrace{(x - x_j^{(n)})}_{\leq 1} \right\|_\infty}_{< 1} \frac{n!}{(n+1)!} \leq \frac{1}{n+1} \rightarrow 0 \quad (n \rightarrow \infty)$$

Aufgabe 14 (Einheitskreis als parametrisierte Kurve)

(3+3+2+3 Punkte)

- (i) Zeigen Sie, dass das Kreissegment $\{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1, x \geq 0, y \geq 0\}$ nicht durch eine parametrisierte Kurve $s = (s_1, s_2) : [a, b] \rightarrow \mathbb{R}^2$ dargestellt werden kann, deren Komponenten s_1, s_2 Polynome sind.
- (ii) Gegeben sei $s : [0, 1] \rightarrow \mathbb{R}^2$ mit

$$s(t) = \left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2} \right).$$

Zeigen Sie $\text{Bild}(s) = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1, x \geq 0, y \geq 0\}$.

- (iii) Seien $a > 0$ und $b > 0$. Zeigen Sie, dass es zwei rationale Funktionen $s_1, s_2 : [0, 1] \rightarrow \mathbb{R}$ gibt mit

$$\text{Bild}(s) = \left\{ (x, y) \in \mathbb{R}^2 : \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1, x \geq 0, y \geq 0 \right\},$$

d. h. dass die Kurve $s = (s_1, s_2)$ ein Ellipsensegment beschreibt.

- (iv) Seien $a > 0$ und $b > 0$. Zeigen Sie, dass es zwei rationale Funktionen $s_1, s_2 : (-1, 1) \rightarrow \mathbb{R}$ gibt mit

$$\text{Bild}(s) = \left\{ (x, y) \in \mathbb{R}^2 : \left(\frac{x}{a}\right)^2 - \left(\frac{y}{b}\right)^2 = 1, x > 0 \right\},$$

d. h. dass die Kurve $s = (s_1, s_2)$ einen Hyperbelast beschreibt.

Lösung zu Aufgabe 14

(i) Seien s_1, s_2 Polynome mit $\text{Bild}(x) = \{(x, y) : x^2 + y^2 = 1, x \geq 0, y \geq 0\}$ dann gilt

$$0 = \text{grad}(1) = \text{grad}(s_1^2 + s_2^2) = \max(2\text{grad}(s_1), 2\text{grad}(s_2))$$

da sich der Grad bei der Addition aufgrund der positiven führenden Koeffizienten von s_1^2 und s_2^2 nicht verkleinern kann. Deshalb ist der

$$\text{grad}(s_1) = \text{grad}(s_2) = 0$$

weshalb $s_1 = \text{const}$ und $s_2 = \text{const}$.

Nun folgt, dass das Bild(s) nur einelementig ist. Dies stellt aber einen Widerspruch dar und die Behauptung folgt unmittelbar.

(ii) Es gelten folgende Eigenschaften:

- $s(0) = (1, 0)$, $s(1) = (0, 1)$
- $s_1(t) \geq 0$, $s_2(t) \geq 0$
- $s_1^2 + s_2^2 = \left(\frac{1-t^2}{t^2+1}\right)^2 + \left(\frac{2t}{t^2+1}\right)^2 = \left(\frac{t^2+1}{t^2+1}\right)^2 = 1$

Also gilt mit dem ZWS

$$\text{Bild}(s) = \{(x, y) \in \mathbb{R} : x^2 + y^2 = 1, x \geq 0, y \geq 0\}.$$

(iii) Analoge Rechnung wie in (ii) nur mit

$$s(t) = \left(a \frac{1-t^2}{1+t^2}, b \frac{2t}{1+t^2} \right).$$

(iv) Sei $\tilde{s}_1^2 + \tilde{s}_2^2 = 1$ dann folgt:

$$\begin{aligned} \tilde{s}_1^2 + \tilde{s}_2^2 = 1 &\iff \tilde{s}_1^2 = 1 - \tilde{s}_2^2 \\ &\iff 1 = \frac{1}{\tilde{s}_1^2} - \left(\frac{\tilde{s}_2}{\tilde{s}_1}\right)^2 \\ &\iff 1 = \frac{1}{a^2} \underbrace{\left(\frac{a}{\tilde{s}_1}\right)^2}_{:=s_1} - \frac{1}{b^2} \underbrace{\left(\frac{b\tilde{s}_2}{\tilde{s}_1}\right)^2}_{:=s_2} \end{aligned}$$

Für $\tilde{s}_1(t) = \frac{1-t^2}{1+t^2}$ und $\tilde{s}_2(t) = \frac{2t}{1+t^2}$ erhält man

$$s_1(t) = a \frac{1+t^2}{1-t^2} > 0, \quad s_2(t) = b \frac{2t}{1-t^2}$$

für $t \in (-1, 1)$. Betrachtet man das Verhalten von $s(t)$ für $t \rightarrow 1$ und $t \rightarrow -1$ folgt dass

$$\lim_{t \rightarrow 1} = (\infty, \infty), \quad \lim_{t \rightarrow -1} = (\infty, -\infty)$$

gilt. Und somit ergibt sich

$$\text{Bild}(s) = \{(x, y) \in \mathbb{R} : \left(\frac{x}{a}\right)^2 - \left(\frac{y}{b}\right)^2 = 1, x > 0\}$$

Aufgabe 15 (Thielscher Kettenbruch)**(6 Punkte)**

Definition. Zu $(n+1)$ paarweise verschiedenen Stützstellen x_0, \dots, x_n und zugehörigen Werten $y_0 = f(x_0), \dots, y_n = f(x_n)$ definieren wir:

1. die ersten **inversen dividierten Differenzen**:

$$\varphi_1(x_i, x_0) := \frac{x_i - x_0}{f(x_i) - f(x_0)}, \quad (i = 1, \dots, n),$$

2. die **k -ten inversen dividierten Differenzen** ($k = 2, \dots, n$):

$$\varphi_k(x_i, x_{k-1}, \dots, x_0) := \frac{x_i - x_{k-1}}{\varphi_{k-1}(x_i, x_{k-2}, \dots, x_0) - \varphi_{k-1}(x_{k-1}, x_{k-2}, \dots, x_0)},$$

für $(i = k, \dots, n)$.

Definition. Für $n \in \mathbb{N}$ definieren wir den abgebrochenen Thielschen Kettenbruch:

$$R_n(x) = c_0 + \frac{x - x_0}{c_1 + \frac{x - x_1}{c_2 + \frac{x - x_2}{\ddots + \frac{x - x_{n-2}}{c_{n-1} + \frac{x - x_{n-1}}{c_n}}}}$$

mit $c_0 := f(x_0)$, $c_k := \varphi_k(x_k, \dots, x_0)$, $k = 1, \dots, n$.

Zeigen Sie, dass der abgebrochene **Thielsche Kettenbruch** $R_n(x)$ die spezielle **rationale Interpolationsaufgabe**

(a)

$$R_n(x) = \frac{p_0 + p_1x + \dots + p_\zeta x^\zeta}{q_0 + q_1x + \dots + q_\nu x^\nu} = \frac{P_\zeta(x)}{Q_\nu(x)},$$

(b)

$$R_n(x_i) = y_i, \quad (i = 0, \dots, n),$$

(c)

$$\begin{cases} \zeta = \nu = \frac{n}{2}, & n \text{ gerade,} \\ \zeta = \frac{1}{2}(n+1), \nu = \frac{1}{2}(n-1), & n \text{ ungerade,} \end{cases}$$

löst.

Lösung zu Aufgabe 15

Sei

$$S_k(x) = c_k + \frac{x - x_k}{c_{k+1} + \frac{x - x_{k+1}}{c_{k+2} + \frac{x - x_{k+2}}{\ddots + \frac{x - x_{n-1}}{c_n}}}}$$

Es gilt:

- $S_0(x) = R_n(x), S_n(x) = c_n$
- $S_k(x) = c_k + \frac{x-x_k}{S_{k+1}(x)} (*)$
- $S_k(x_k) = c_k \Rightarrow R_n(x_k) = R_k(x_k)$

Für Teil (b) muss $R_n(x_n) = f(x_n), n \in \mathbb{N}_0$ gezeigt werden.

Wir beweisen die Behauptung $S_k(x_n) = \varphi_k(x_n, x_{k-1}, x_{k-2}, \dots, x_0)$ für $k = 1, 2, \dots, n$ mit vollständiger Induktion (rückwärts) über k

Induktionsanfang: $S_n(x_n) = c_n = \varphi_n(x_n, x_{n-1}, \dots, x_0)$

Induktionsvoraussetzung: $S_{k+1}(x_n) = \varphi_{k+1}(x_n, x_k, x_{k-1}, \dots, x_0)$

Induktionsschluss:

$$\begin{aligned} S_k(x_n) &= c_k + \frac{x_n - x_k}{S_{k+1}(x_n)} = \varphi_k(x_k, x_{k-1}, \dots, x_0) + \frac{x_n - x_k}{\varphi_{k+1}(x_n, x_k, \dots, x_0)} \\ &\stackrel{Def.}{=} \varphi_k(x_k, x_{k-1}, \dots, x_0) + \varphi_k(x_n, x_{k-1}, \dots, x_0) - \varphi_k(x_k, x_{k-1}, \dots, x_0) \\ &= \varphi_k(x_n, x_{k-1}, x_{k-2}, \dots, x_0) \end{aligned}$$

Damit gilt:

$$R_n(x_n) = S_0(x_n) = c_0 + \frac{x_n - x_0}{S_1(x_n)} = f(x_0) + \frac{x_n - x_0}{\varphi_1(x_n, x_0)} = f(x_0) + f(x_n) - f(x_0) = f(x_n)$$

Zu Teil (a) und (c):

Wir betrachten die Behauptung: \exists Polynome P_k, Q_k mit $S_k = \frac{P_k}{Q_k} (k = 0, \dots, n)$ und

$$\begin{aligned} \text{grad}(P_k) &\leq \frac{n-k}{2}, \quad \text{grad}(Q_k) \leq \frac{n-k}{2}, \quad \text{falls } (n-k) \text{ gerade} \\ \text{grad}(P_k) &\leq \frac{1}{2}(n-k+1), \quad \text{grad}(Q_k) \leq \frac{1}{2}(n-k-1), \quad \text{falls } (n-k) \text{ ungerade} \end{aligned}$$

und beweisen sie per vollständiger Induktion (rückwärts).

Induktionsanfang: $S_n(x) = \frac{P_n(x)}{Q_n(x)} = c_n, P_n = c_n, Q_n = 1 \Rightarrow \text{grad}(P_n) = \text{grad}(Q_n) = 0$

Induktionsschluss:

$$S_k(x) = c_k + \frac{x - x_k}{S_{k+1}(x)} = c_k + \frac{x - x_k}{\frac{P_{k+1}(x)}{Q_{k+1}(x)}} = \frac{\overbrace{c_k + P_{k+1}(x) + (x - x_k)Q_{k+1}(x)}^{=: P_k(x)}}{\underbrace{P_{k+1}(x)}_{=: Q_k(x)}} = \frac{P_k(x)}{Q_k(x)}$$

1.Fall $(n - (k + 1))$ gerade $\Leftrightarrow (n - k)$ ungerade

$$\begin{aligned} \Rightarrow \text{grad}(Q_k) &= \text{grad}(P_{k+1}) \leq \frac{n-k-1}{2} \\ \Rightarrow \text{grad}(P_k) &\leq \max(\underbrace{\text{grad}(P_{k+1})}_{\leq \frac{n-k-1}{2}}, 1 + \underbrace{\text{grad}(Q_{k+1})}_{\leq \frac{n-k-1}{2}}) \leq \frac{n-k+1}{2} \end{aligned}$$

2.Fall $(n - (k + 1))$ ungerade $\Leftrightarrow (n - k)$ gerade

$$\begin{aligned} \Rightarrow \text{grad}(Q_k) &= \text{grad}(P_{k+1}) \leq \frac{1}{2}(n - (k + 1) + 1) = \frac{1}{2}(n - k) \\ \Rightarrow \text{grad}(P_k) &\leq \max(\underbrace{\text{grad}(P_{k+1})}_{\leq \frac{1}{2}(n-k)}, 1 + \underbrace{\text{grad}(Q_{k+1})}_{\leq \frac{1}{2}(n-k-2)}) \leq \frac{1}{2}(n - k) \end{aligned}$$

Somit erhalten wir $R_n(x) = S_0(x) = \frac{P_0(x)}{Q_0(x)}$ mit der bewiesenen Behauptung für $k = 0$.

Aufgabe 16 (Fehlerabschätzung für numerische Intergration) (2 Punkte)

Let f be a sufficiently smooth function on $[a, b]$. The polynomial $p_n \in P_n$ is the corresponding interpolation polynomial of f for $n \in \mathbb{N}_0$ and $x_0, \dots, x_n \in [a, b]$. Proof, that the inequality

$$\int_a^b |f(x) - p_n(x)| dx \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty \int_a^b \left| \prod_{j=0}^n (x - x_j) \right| dx$$

holds.

Lösung zu Aufgabe 16

Durch die Gleichung

$$|f(x) - p(x)| = \frac{1}{(n+1)!} |(x - x_0)(x - x_1) \dots (x - x_n)| |f^{(n+1)}(\xi)|$$

erhält man die Abschätzung:

$$\begin{aligned} \int_a^b |f(x) - p(x)| dx &= \frac{1}{(n+1)!} \int_a^b \left| \prod_{j=0}^n (x - x_j) \right| |f^{(n+1)}(\xi(x))| dx \\ &\leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty \int_a^b \left| \prod_{j=0}^n (x - x_j) \right| dx \end{aligned}$$

Aufgabe 17 (Newton-Cotes-Formel) (3+4+3+4+3+4 Punkte)

Sei f eine hinreichend glatte Funktion auf $[a, b]$. Für $n \in \mathbb{N}$ betrachten wir die äquidistante Intervalleinteilung $[x_i, x_{i+1}]$ von $[a, b]$ mit $x_i = a + ih$, $i = 0, \dots, n$, $h = \frac{b-a}{n}$. Sei $p \in \mathbb{P}_n$ das Interpolationspolynom von f bzgl. der Stützstellen x_i .

(i) Zeigen Sie, dass

$$\int_a^b p(x) dx = (b-a) \sum_{i=0}^n \lambda_i f_i \quad \text{mit} \quad \lambda_i = \frac{1}{n} \int_0^n \prod_{\substack{k=0 \\ k \neq i}}^n \frac{z-k}{i-k} dz \quad (\text{B.1})$$

gilt.

(ii) Zeigen Sie, dass $\lambda_{n-i} = \lambda_i$ für $i = 1, \dots, n$ und $\sum_{i=0}^n \lambda_i = 1$ gilt.

(iii) Sei n gerade. Zeigen Sie, dass die Quadraturformel (C.1) sogar für $p \in \mathbb{P}_{n+1}$ gilt.

(iv) Berechnen Sie die Gewichte $\lambda_0, \dots, \lambda_n$ für $n = 1$ (Trapezregel), $n = 2$ (Simpsonregel) und $n = 3$ (3/8 Regel).

(v) Für die Trapezregel zeige man für den Approximationsfehler

$$\left| \int_a^b p(x) dx - \int_a^b f(x) dx \right| \leq \frac{1}{12} (b-a)^3 \|f''\|_\infty.$$

(vi) Die Unterteilung von $[a, b]$ in N Teilintervalle $[t_i, t_{i+1}]$, $t_i = a + ih$, $i = 0, \dots, N$, $h = \frac{b-a}{N}$, die Anwendung der Trapezformel in jedem Teilintervall und die anschließende Summation der Näherungswerte führt zu den summierten Quadraturformeln. Geben Sie die summierte Trapezregel an und schätzen Sie ihren Approximationsfehler ab.

Lösung zu Aufgabe 17

(i) Für das Interpolationspolynom gilt:

$$\begin{aligned} \int_a^b p(x) dx &= \int_a^b \sum_{i=0}^n f_i L_i(x) dx = \sum_{i=0}^n f_i \int_a^b \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j} dx \\ &\stackrel{y = n \frac{x-a}{b-a}}{=} \sum_{i=0}^n f_i \frac{b-a}{n} \int_0^n \prod_{j=0, j \neq i}^n \frac{hy + a - a - jh}{a + hi - a - hj} dy \\ &= (b-a) \sum_{i=0}^n f_i \int_0^n \prod_{j=0, j \neq i}^n \frac{y-j}{i-j} dy \end{aligned}$$

(ii) Zuerst betrachten wir $n \cdot \lambda_{n-i}$.

$$\begin{aligned} n \cdot \lambda_{n-i} &= \int_0^n \prod_{k=0, k \neq n-i}^n \frac{z-k}{n-i-k} dz \stackrel{y=-z+n}{=} \int_0^n \prod_{k=0, k \neq n-i}^n \frac{n-y-k}{n-i-k} dy \\ &\stackrel{k=n-j}{=} \int_0^n \prod_{j=0, j \neq i}^n \frac{n-y-n+j}{n-i-n+j} dy = \int_0^n \prod_{j=0, j \neq i}^n \frac{y-j}{i-j} dy = n \cdot \lambda_i \end{aligned}$$

Daraus folgt unmittelbar die erste Behauptung.

Um die zweite Behauptung zu zeigen, setzen wir $p(x) = 1$.

$$\begin{aligned} \Rightarrow b-a &= \int_a^b 1 dx = (b-a) \sum_{i=0}^n \lambda_i \\ \Rightarrow \sum_{i=0}^n \lambda_i &= 1. \end{aligned}$$

(iii) Sei n gerade und $p \in \mathbb{P}_n$. Des Weiteren sei $p_n \in \mathbb{P}_n$ der Interpolant von p zu den Stützstellen x_0, \dots, x_n und $p(x_i) = \bar{p}_i$. Die Behauptung folgt, da

$$\begin{aligned} \int_a^b p(x) dx - (b-a) \sum_{i=0}^n \lambda_i \bar{p}_i &= \int_a^b p(x) dx - \int_a^b p_n(x) dx = \int_a^b (p - p_n) dx \\ &= \frac{1}{(n+1)!} \int_a^b \underbrace{p^{(n+1)}(\xi)}_{=: c \text{ da } p \in \mathbb{P}_{n+1}} (x-x_0) \cdot \dots \cdot (x-x_n) dx \\ &= \frac{c}{(n+1)!} \underbrace{\int_a^b (x-x_0) \cdot \dots \cdot (x-x_n) dx}_{=0 \text{ da punktsym. zu } (\frac{a+b}{2}, 0)} = 0 \end{aligned}$$

(iv) Für die Trapezregel ($n = 1$) sind die Gewichte durch

$$\lambda_0 = \int_0^1 \frac{z-1}{0-1} dz = \frac{1}{2} = \lambda_1$$

gegeben. Für $n = 2$ kann man die Gewichte durch

$$\begin{aligned} \lambda_0 &= \frac{1}{2} \int_0^2 \frac{(z-1)(z-2)}{2} dz = \frac{1}{4} \int_0^2 (z^2 - 3z + 2) dz = \frac{1}{4} \left[\frac{1}{3} z^3 - \frac{3}{2} z^2 + 2z \right]_0^2 \\ &= \frac{1}{6} = \lambda_2 \\ \lambda_1 &= 1 - \lambda_0 - \lambda_2 = \frac{2}{3} \end{aligned}$$

berechnen. Für die 3/8-Regel ($n = 3$) erhält man die Gewichte

$$\begin{aligned}\lambda_0 &= \frac{1}{3} \int_0^3 \frac{(z-1)(z-2)(z-3)}{(0-1)(0-2)(0-3)} dz = -\frac{1}{18} \int_0^3 (z^3 - 6z^2 + 11z - 6) dz \\ &= -\frac{1}{18} \left[\frac{1}{4}z^4 - 2z^3 + \frac{11}{2}z^2 - 6z \right]_0^3 = \frac{1}{8} = \lambda_3 \\ 2\lambda_2 &= 1 - \lambda_0 - \lambda_3 = \frac{3}{4} \Rightarrow \lambda_2 = \lambda_1 = \frac{3}{8}\end{aligned}$$

(v) Die Abschätzung

$$\begin{aligned}\left| \int_a^b (f(x) - p(x)) dx \right| &= \left| \frac{1}{2} \int_a^b (x-a)(x-b) f''(\xi(x)) dx \right| \\ &\leq \frac{1}{2} \|f''\|_\infty \left| \int_a^b (x-a)(x-b) dx \right| \\ &= \frac{1}{2} \|f''\|_\infty \left| \int_{-\frac{b-a}{2}}^{\frac{b-a}{2}} \frac{b-a}{2} x^2 - \left(\frac{b-a}{2}\right)^2 dx \right| \\ &= \frac{1}{2} \|f''\|_\infty 2 \left| \left[\frac{1}{3} x^3 - \left(\frac{b-a}{2}\right)^2 x \right]_0^{\frac{b-a}{2}} \right| \\ &= \frac{1}{12} (b-a)^3 \|f''\|_\infty\end{aligned}$$

liefert die Behauptung.

(vi) Die summierte Trapezregel

$$\begin{aligned}\int_a^b p(x) dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} p(x) dx = \sum_{i=0}^{n-1} h \left(\frac{1}{2} p(x_i) + \frac{1}{2} p(x_{i+1}) \right) \\ &= \frac{b-a}{n} \left[\frac{1}{2} p(x_1) + \sum_{i=1}^{n-1} p(x_i) + \frac{1}{2} p(x_n) \right]\end{aligned}$$

liefert mit Hilfe folgender Abschätzung den Approximationsfehler

$$\begin{aligned}\left| \int_a^b (f(x) - p(x)) dx \right| &\leq \sum_{i=0}^{n-1} \left| \int_{x_i}^{x_{i+1}} (f(x) - p(x)) dx \right| \\ &\leq \sum_{i=0}^{n-1} \frac{1}{12} \left(\frac{b-a}{2} \right)^3 \|f''\|_\infty = \frac{(b-a)^3}{12n^2} \|f''\|_\infty.\end{aligned}$$

Aufgabe 18 (Gauß-Quadratur)

(4+4 Punkte)

(i) Bestimmen Sie x_1, x_2, x_3 und $\omega_1, \omega_2, \omega_3$, so dass die Quadraturformel

$$\int_{-1}^1 p(x) dx = \omega_1 p(x_1) + \omega_2 p(x_2) + \omega_3 p(x_3)$$

für alle Polynome $p \in \mathbb{P}_5$ gilt.

Hinweis: Verwenden Sie Satz 2.1.17 und die Rodriguesformel für die Legendre-Polynome P_n .

(ii) Bestimmen Sie x_1, x_2, x_3 und $\omega_1, \omega_2, \omega_3$, so dass

$$\int_{-\infty}^{\infty} e^{-x^2} p(x) dx = \omega_1 p(x_1) + \omega_2 p(x_2) + \omega_3 p(x_3)$$

für alle $p \in \mathbb{P}_5$ gilt.

Hinweis: Verwenden Sie Satz 2.1.17 und die Rodriguesformel

$$H_k(x) = (-1)^k e^{x^2} \frac{d^k}{dx^k} e^{-x^2}, \quad k \in \mathbb{N}_0.$$

für die Hermite-Polynome H_k .

Lösung zu Aufgabe 18

Aufgabe 19 (Mehrdimensionale Integration)

(4+4 Punkte)

(i) Seien $(0, 0), (1, 0), (0, 1) \in \mathbb{R}^2$ die Eckpunkte des Dreiecks D . Zeigen Sie, dass Gewichte $\omega_1, \dots, \omega_4$ existieren, sodass die Quadraturformel

$$\frac{1}{|D|} \int_D p(x, y) d(x, y) = \omega_1 p(0, 0) + \omega_2 p(1, 0) + \omega_3 p(0, 1) + \omega_4 p\left(\frac{1}{3}, \frac{1}{3}\right)$$

von allen Polynomen

$$p \in \{q : D \rightarrow \mathbb{R} : q(x, y) = \alpha_{0,0} + \alpha_{1,0}x + \alpha_{0,1}y + \alpha_{2,0}x^2 + \alpha_{1,1}xy + \alpha_{0,2}y^2 \text{ mit } \alpha_{i,j} \in \mathbb{R}\}$$

erfüllt wird. Dabei ist $|D|$ der Flächeninhalt von D . Wie lauten die Gewichte ω_i ?

(ii) Seien $a, b, c \in \mathbb{R}^2$ die Eckpunkte eines Dreiecks T . Zeigen Sie, dass Gewichte $\omega_1, \dots, \omega_4$ existieren, sodass die Quadraturformel

$$\frac{1}{|T|} \int_T p(x, y) d(x, y) = \omega_1 p(a) + \omega_2 p(b) + \omega_3 p(c) + \omega_4 p\left(\frac{a+b+c}{3}\right)$$

von allen Polynomen

$$p \in \{q : T \rightarrow \mathbb{R} : q(x, y) = \alpha_{0,0} + \alpha_{1,0}x + \alpha_{0,1}y + \alpha_{2,0}x^2 + \alpha_{1,1}xy + \alpha_{0,2}y^2 \text{ mit } \alpha_{i,j} \in \mathbb{R}\}$$

erfüllt wird. Dabei ist $|T|$ der Flächeninhalt von T . Wie lauten die Gewichte ω_i ?

Hinweis: Verwenden Sie das Ergebnis von 22 (i), indem Sie T affin auf D abbilden.

Lösung zu Aufgabe 19

C AUFGABEN UND LÖSUNGEN

Aufgabe 20 (Nested intervals)

(3+3 Punkte)

- (i) Show first that the nonlinear equation $x = \cos x$ has a unique solution $x \in [0, \infty)$.
- (ii) Determine a numerical solution of the nonlinear equation $x = \cos x$ by using nested intervals with initial values $a = 0$ und $b = 1$ until the error is smaller than 0,05. You can use a calculator.

Lösung zu Aufgabe 20

- (i) Zu zeigen: Die Gleichung $x = \cos x$ hat eine eindeutige Lösung.
Sei $f : \mathbb{R} \rightarrow \mathbb{R}$ mit $f(x) = x - \cos x$ ist stetig.

$$f(0) = -1, f\left(\frac{\pi}{2}\right) = \frac{\pi}{2}$$

Damit gilt nach dem Zwischenwertsatz: es existiert ein $\xi \in (0, \frac{\pi}{2})$ mit $f(\xi) = 0$.

$$f'(x) = 1 + \sin x > 0 \text{ für } x \neq 2\pi k + \frac{3}{2}\pi, k \in \mathbb{Z}$$

Daraus folgt, dass f auf dem Intervall $[2\pi k + \frac{3}{2}\pi, 2\pi(k+1) + \frac{3}{2}\pi]$, $k \in \mathbb{Z}$, streng monoton steigend ist (Mittelwertsatz).

Damit ist dann f auf ganz \mathbb{R} streng monoton steigend und die die Nullstelle ξ ist eindeutig.

- (ii)

$$\begin{aligned} a_0 &= 0, b_0 = 1 \\ a_1 &= 0,5, b_1 = 1 \\ a_2 &= 0,5, b_2 = 0,75 \\ a_3 &= \frac{5}{8}, b_3 = \frac{3}{4} \\ a_4 &= \frac{11}{16}, b_4 = \frac{3}{4} \\ a_5 &= \frac{23}{32}, b_5 = \frac{3}{4} \end{aligned}$$

Die Länge des Intervalls ist nach der 5-ten Iteration $2^{-5} < 0,05$, damit ist auch der absolute Fehler für jedes $x \in [a_5, b_5]$ zur Nullstelle kleiner als 0,05.

Aufgabe 21 (Interpolation)

(3 Punkte)

Man finde das Polynom $p : \mathbb{R} \rightarrow \mathbb{R}$ von kleinstem Grad mit $p(-1) = 3$, $p(0) = -3$, $p(1) = -5$, $p(2) = 3$.

Lösung zu Aufgabe 21

$$\begin{aligned}
 p(x) &= ax^3 + bx^2 + cx + d \\
 p(-1) &= -a + b - c + d = 3 \\
 p(0) &= d = -3 \\
 p(1) &= a + b + c + d = -5 \\
 p(2) &= 8a + 4b + 2c + d = 3
 \end{aligned}$$

Nach Berechnung der Koeffizienten des linearen Gleichungssystems erhält man $p(x) = x^3 + 2x^2 - 5x - 3$.

Aufgabe 22 (Rolle, Taylor, Banach)**(3+3+4 Punkte)**

- (i) Zeigen Sie, dass für eine n -mal differenzierbare Funktion $f : [a, b] \rightarrow \mathbb{R}$ mit $n + 1$ verschiedenen Nullstellen ein $\xi \in (a, b)$ existiert mit $f^{(n)}(\xi) = 0$.
- (ii) Berechnen Sie näherungsweise das Integral

$$\int_0^1 e^{-x^2/2} dx,$$

indem Sie $f(x) = e^{-x^2/2}$ in ein Taylorpolynom 3. Grades mit Entwicklungspunkt 0 entwickeln. Schätzen sie den Fehler mit Hilfe des Restglieds von Lagrange ab.

- (iii) Zeigen Sie, dass durch

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{c}{x_n} \right)$$

ein iteratives Verfahren zur Berechnung der Quadratwurzel einer positiven Zahl c definiert ist. Welche Bedingungen müssen für den Startwert x_0 erfüllt sein? Berechnen Sie für den Startwert $x_0 = 1$ Näherungen x_1, \dots, x_4 für $\sqrt{2}$. Hier dürfen Sie einen Taschenrechner verwenden.

Lösung zu Aufgabe 22

- (i) Zu zeigen: Es existiert ein $\xi \in (a, b)$ mit $f^{(n)}(\xi) = 0$.
Vollständige Induktion über n :
IA $n = 1$: f ist differenzierbar und besitzt zwei verschiedene Nullstellen. Dann gilt nach dem Satz von Rolle:

$$\exists \xi \in [a, b] \text{ mit } f'(\xi) = 0$$

$A(n) \Rightarrow A(n+1)$: Sei f eine $n+1$ mal diff'bare Funktion mit $n+2$ verschiedene Nullstellen. Dann ist f' eine n mal diff'bare Funktion mit $n + 1$ verschiedenen Nullstellen (Satz von Rolle). Daraus folgt nach Induktionvoraussetzung: Es existiert ein $\xi \in [a, b]$ mit $f^{n+1}(\xi) = (f')^{(n)}(\xi) = 0$.

- (ii) Taylorpolynom dritten Grades mit Entwicklungspunkt 0 von $f(x) = e^{-x^2/2}$:

$$T(x) = 1 - \frac{1}{2}x^2$$

Lagrangerestglied: $R(x) = \frac{1}{4!}(3e^{-\xi^2/2} - 6\xi^2e^{-\xi^2/2} + \xi^4e^{-\xi^2/2})x^4$, $\xi \in (0, 1)$ Damit ist dann die Näherung des Integrals:

$$\int_0^1 f(x)dx \approx \int_0^1 (1 - \frac{1}{2}x^2)dx = 1 - \frac{1}{6} = \frac{5}{6}$$

Fehlerabschätzung:

$$|\int_0^1 (f(x)-T(x))dx| = |\int_0^1 R(x)dx| \leq \int_0^1 |R(x)|dx \leq \frac{1}{24} \int_0^1 x^4 dx \max_{y \in [0,1]} |3-6y^2+y^4| = \frac{1}{40} = 0,025$$

da $g(y) = 3 - 6y^2 + y^4$ als Extremstellen $y_{1,2} = \pm\sqrt{3}$, $y_3 = 0$ hat und $g(0) = 3$ und $g(1) = -2$, also ist das $\max_{y \in [0,1]} |3 - 6y^2 + y^4| = 3$.

(iii) Mit $x_{n+1} = \frac{1}{2}(x_n - \frac{c}{x_n})$ kann man iterativ die Wurzel von c berechnen.

1.Schritt: Startwert x_0

Sei zunächst $f : \mathbb{R} \setminus \{0\} \rightarrow \mathbb{R}$ mit $f(x) = \frac{1}{2}(x - \frac{c}{x})$

$f(x) < 0$ für $x < 0 \Rightarrow$ Für die Konvergenz von (x_k) gegen \sqrt{c} muss x_0 positiv sein

2.Schritt: Minimum und Fixpunkt

$$f'(x) = \frac{1}{2} - \frac{1}{2} \frac{c}{x^2} = \frac{1}{2x^2}(x^2 - c)$$

Daraus folgt, dass f an der Stelle $x = \sqrt{c}$ ein Minimum hat.

$f(\sqrt{c}) = \frac{1}{2}(\sqrt{c} + \sqrt{c}) = \sqrt{c}$ damit ist \sqrt{c} ein Fixpunkt und es gilt: für $x_0 > 0$ folgt $x_1 \geq \sqrt{c}$.

3.Schritt: Banachscher Fixpunktsatz

Sei nun $f : [\sqrt{c}, \infty) \rightarrow [\sqrt{c}, \infty)$ ($[\sqrt{c}, \infty)$ vollständig)

$$\text{Kontraktion: } |f(x) - f(y)| = \frac{1}{2}|x - y + c(\frac{1}{x} - \frac{1}{y})| = \frac{1}{2} \underbrace{|1 - \frac{c}{xy}|}_{<1} |x - y| \leq \frac{1}{2}|x - y|$$

damit ist f eine Kontraktion mit $\frac{1}{2}$ als Konstante.

Nach dem Banachschen Fixpunktsatz konvergiert (x_k) gegen \sqrt{c} .

(Konvergenzgeschwindigkeit: $|x - x_n| \leq (\frac{1}{2})^{n-1}|x - x_1|$)

$$x_0 = 1$$

$$x_1 = 1,5$$

$$x_2 = 1,416667$$

$$x_3 = 1,414216$$

$$x_4 = 1,414214$$

Aufgabe 23 (Regula Falsi)

(4 Punkte)

Bei der Bestimmung einer Nullstelle mittels Regula Falsi wird zu einer stetigen Funktion $f : I \rightarrow \mathbb{R}$, $I \subset \mathbb{R}$ Intervall, und einem Intervall $[a, b] \subset I$, $a < b$, ein

$$\xi = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

berechnet und als nächstes Intervall eines der beiden Intervalle $[a, \xi]$ und $[\xi, b]$ verwendet, für das $f(a)f(\xi) < 0$ bzw. $f(\xi)f(b) < 0$ gilt. Durch mehrmalige Anwendung liegt wieder eine Intervallschachtelung vor. Führen Sie vier Schritte der Regula Falsi mit den Startwerten $a = 0$ und $b = 1$ zur näherungsweisen Bestimmung der Lösung der nichtlinearen Gleichung $x = \cos(x)$ durch. Sie dürfen einen Taschenrechner verwenden.

Lösung zu Aufgabe 23

$$\begin{aligned} a_0 &= 0, & b_0 &= 1 \\ a_1 &= 0,6851, & b_1 &= 1 \\ a_2 &= 0,7363, & b_2 &= 1 \\ a_3 &= 0,7389, & b_3 &= 1 \\ a_4 &= 0,7391 & b_4 &= 1 \end{aligned}$$

(Der exakte Wert der Nullstelle ist 0,739085...)

Aufgabe 24 (Newton-Iteration)**(2+4 Punkte)**

Gegeben Sei die Funktion

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad f(x_1, x_2) = \begin{pmatrix} 2x_1(1 + x_2^2) - 2 \\ 2x_1^2x_2 \end{pmatrix}$$

- (i) Berechnen Sie die Jacobi-Matrix von f .
- (ii) Führen Sie zwei Schritte des Newton-Verfahrens zur Bestimmung einer Nullstelle von f mit dem Startvektor $x^{(0)} = \left(\frac{3}{2}, \frac{1}{2}\right)^T$ durch.

Lösung zu Aufgabe 24

- (i) Die Jacobi-Matrix von f lautet folgendermaßen:

$$J_f(x_1, x_2) = \begin{pmatrix} 2(1 + x_2^2) & 4x_1x_2 \\ 4x_1x_2 & 2x_1^2 \end{pmatrix}$$

- (ii) Die Iterationsvorschrift lautet:

$$x^{(k+1)} = x^{(k)} - (f'(x^{(k)}))^{-1} \cdot f(x^{(k)})$$

Mit dem Startvektor $x^{(0)} = \left(\frac{3}{2}, \frac{1}{2}\right)^T$ kann man die Jacobi-Matrix für $k = 0$ berechnen:

$$J_f(x^{(0)}) = \begin{pmatrix} \frac{5}{2} & 3 \\ 3 & \frac{9}{2} \end{pmatrix}$$

Somit sieht die Inverse wie folgt aus:

$$(J_f(x^{(0)}))^{-1} = \begin{pmatrix} 2 & -\frac{4}{3} \\ -\frac{4}{3} & \frac{10}{9} \end{pmatrix}$$

Damit kann man $x^{(1)}$ berechnen:

$$x^{(1)} = \begin{pmatrix} \frac{3}{2} \\ \frac{1}{2} \end{pmatrix} - \begin{pmatrix} 2 & -\frac{4}{3} \\ -\frac{4}{3} & \frac{10}{9} \end{pmatrix} \cdot \begin{pmatrix} \frac{7}{4} \\ \frac{9}{4} \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{3} \end{pmatrix}$$

Damit sieht die Jacobi-Matrix für $k = 1$ folgendermaßen aus:

$$J_f(x^{(1)}) = \begin{pmatrix} \frac{20}{9} & \frac{4}{3} \\ \frac{4}{3} & 2 \end{pmatrix}$$

Somit sieht die Inverse wie folgt aus:

$$(J_f(x^{(1)}))^{-1} = \begin{pmatrix} \frac{3}{4} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{5}{6} \end{pmatrix}$$

Dadurch erhält man $x^{(2)}$:

$$x^{(2)} = \begin{pmatrix} 1 \\ \frac{1}{3} \end{pmatrix} - \begin{pmatrix} \frac{3}{4} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{5}{6} \end{pmatrix} \cdot \begin{pmatrix} \frac{2}{9} \\ \frac{2}{3} \end{pmatrix} = \begin{pmatrix} \frac{7}{6} \\ -\frac{1}{9} \end{pmatrix}$$

Aufgabe 25 (Newtonverfahren bei mehrfachen Nullstellen) (4+4 Punkte)

Es sei $f : [a, b] \rightarrow \mathbb{R}$ eine $(\ell + 1)$ -mal stetige differenzierbare Funktion und $x^* \in (a, b)$ sei eine ℓ -fache Nullstelle von f , d. h. es sei $f(x^*) = f'(x^*) = \dots = f^{(\ell-1)}(x^*) = 0$ und $f^{(\ell)}(x^*) \neq 0$.

(i) Zeigen Sie, dass für $\ell \geq 2$ das Newtonverfahren unter der Zusatzvoraussetzung $x_n \neq x^*$ in einer Umgebung von x^* die maximale Konvergenzordnung 1 hat.

(ii) Zeigen Sie, dass das modifizierte Newton-Verfahren

$$x_{n+1} = x_n - \ell \frac{f(x_n)}{f'(x_n)} \quad (n \in \mathbb{N}_0)$$

unter der Zusatzvoraussetzung $x_n \neq x^*$ in einer Umgebung von x^* mindestens die Konvergenzordnung 2 hat.

Lösung zu Aufgabe 25

(i) Für $f(x)$ gilt:

$$\begin{aligned} f(x) &= \sum_{k=0}^{\ell} \frac{1}{k!} f^{(k)}(x^*) (x - x^*)^k + \frac{1}{(\ell + 1)!} f^{(\ell+1)}(\xi(x)) (x - x^*)^{\ell+1} \\ &= \underbrace{\frac{1}{\ell!} f^{(\ell)}(x^*) (x - x^*)^{\ell}}_{=:a} + \underbrace{\frac{1}{(\ell + 1)!} f^{(\ell+1)}(\xi(x)) (x - x^*)^{\ell+1}}_{=:g(x)}, \quad \xi(x) \in (x^*, x) \end{aligned}$$

g ist beschränkt, da $f^{(\ell+1)}$ beschränkt ist.

Die erste Ableitung von f sieht dann folgendermaßen aus:

$$\begin{aligned} f'(x) &= \sum_{k=0}^{\ell-1} \frac{1}{k!} (f')^{(k)}(x^*) (x - x^*)^k + \frac{1}{\ell!} (f')^{(\ell)}(\xi_1(x)) (x - x^*)^{\ell} \\ &= \underbrace{\frac{1}{(\ell - 1)!} f^{(\ell)}(x^*) (x - x^*)^{\ell-1}}_{=:l \cdot a} + \underbrace{\frac{1}{\ell!} f^{(\ell+1)}(\xi_1(x)) (x - x^*)^{\ell}}_{=:h(x)} \end{aligned}$$

Wiederum gilt, dass $h(x)$ beschränkt ist, da $f^{(\ell+1)}$ beschränkt ist.

$$\begin{aligned} x_{n+1} - x^* &= x_n - x^* - \frac{f(x_n)}{f'(x_n)} = x_n - x^* - \frac{a(x_n - x^*)^{\ell} + g(x_n)(x_n - x^*)^{\ell+1}}{la(x_n - x^*)^{\ell-1} + h(x_n)(x_n - x^*)^{\ell}} \\ &= (x_n - x^*) \left(1 - \frac{a + g(x_n)(x_n - x^*)}{la + h(x_n)(x_n - x^*)} \right) \end{aligned}$$

Des Weiteren kann man den zweiten Faktor folgendermaßen abschätzen:

$$\begin{aligned} \left| 1 - \frac{a + g(x_n)(x_n - x^*)}{la + h(x_n)(x_n - x^*)} \right| &\leq 1 + (|a| + C_g|b - a|) \left| \frac{1}{|la| - |h(x_n)||x_n - x^*|} \right| \\ &\leq \underbrace{1 + (|a| + C_g|b - a|) \frac{2}{|la|}}_{\text{konstant}} \quad \text{falls } |x_n - x^*| \text{ genügend klein} \end{aligned}$$

Deshalb besitzt das Verfahren die Konvergenzordnung 1.

Angenommen das Verfahren besäße eine höhere Konvergenzordnung als 1, dann würde folgendes gelten:

Sei x_n eine Folge die gegen x^* für $n \rightarrow \infty$ konvergiert. Dann gilt:

$$0 \neq 1 - \frac{1}{l} = \lim_{x_n \rightarrow x^*} \left| 1 - \frac{a + g(x_n)(x_n - x^*)}{la + h(x_n)(x_n - x^*)} \right| = \lim_{x_n \rightarrow x^*} \frac{|x_{n+1} - x^*|}{|x_n - x^*|} \leq \lim_{x_n \rightarrow x^*} C|x_n - x^*|^{p-1} = 0$$

Dies ist aber ein Widerspruch.

(ii) Es gilt:

$$\begin{aligned} x_{n+1} - x^* &= x_n - x^* - l \frac{f(x_n)}{f'(x_n)} = x_n - x^* - \frac{la(x_n - x^*)^l + lg(x_n)(x_n - x^*)^{l+1}}{la(x_n - x^*)^{l-1} + h(x_n)(x_n - x^*)^l} \\ &= (x_n - x^*) \left(1 - \frac{la + lg(x_n)(x_n - x^*)}{la + h(x_n)(x_n - x^*)} \right) \\ &= (x_n - x^*) \frac{la + h(x_n)(x_n - x^*) - la - lg(x_n)(x_n - x^*)}{la + h(x_n)(x_n - x^*)} \\ &= (x_n - x^*)^2 \frac{h(x_n) - lg(x_n)}{la + h(x_n)(x_n - x^*)} \end{aligned}$$

$$\begin{aligned} \left| \frac{h(x_n) - lg(x_n)}{la + h(x_n)(x_n - x^*)} \right| &\leq (C_h + l \cdot C_g) \left| \frac{1}{|la| - |h(x_n)||x_n - x^*|} \right| \\ &\leq \underbrace{(C_h + l \cdot C_g) \frac{2}{|la|}}_{\text{konstant}} \quad \text{falls } |x_n - x^*| \text{ genügend klein} \end{aligned}$$

Deshalb hat das modifizierte Newtonverfahren die Konvergenzordnung 2.

Aufgabe 26 (Lineare Differenzgleichung)

(2+4+3 Punkte)

(i) Seien $\mu_0, \mu_1, \dots, \mu_{m-1}$ reelle Zahlen und (a_n) eine reelle Folge, die durch die Anfangswerte a_0, a_1, \dots, a_{m-1} und durch die lineare Differenzgleichungen

$$a_{n+m} = \sum_{i=0}^{m-1} \mu_i a_{n+i}$$

für $n \in \mathbb{N}_0$ eindeutig bestimmt ist. Nennen Sie die Matrix $A \in \mathbb{R}^{m \times m}$, die für alle $n \in \mathbb{N}_0$ den Vektor \hat{a}_n auf den Vektor \hat{a}_{n+1} abbildet, wobei

$$\hat{a}_k := \begin{pmatrix} a_k \\ \vdots \\ a_{k+m-1} \end{pmatrix}$$

für $k \in \mathbb{N}_0$ definiert ist. D.h. es gilt $\hat{a}_{n+1} = A \hat{a}_n$ und damit $\hat{a}_n = A^n \hat{a}_0$.

- (ii) Gegeben sei eine Folge (a_n) mit der Eigenschaft $a_{n+2} = a_{n+1} + a_n$ für alle $n \in \mathbb{N}_0$ und den Startwerten $a_0 = 1$ und $a_1 = 3$. Geben Sie eine explizite Darstellung von a_n an.
Hinweis: Geben Sie zunächst wie in Aufgabe 10 (i) für $m = 2$ die Matrix A an, für die $\hat{a}_{n+1} = A \hat{a}_n$ und damit auch $\hat{a}_n = A^n \hat{a}_0$ gilt. Diagonalisieren Sie A . Wie sieht dann A^n aus?
- (iii) Gegeben sei eine Folge (a_n) mit der Eigenschaft $a_{n+2} = a_{n+1} \cdot a_n$ für alle $n \in \mathbb{N}_0$ und den Startwerten $a_0 = 1$ und $a_1 = 2$. Wie kann man analog zu Aufgabe 10 (ii) die explizite Darstellung von a_n gewinnen? Geben Sie diese an.

Lösung zu Aufgabe 26

- (i) Man erhält für \hat{a}_{n+1} , \hat{a}_n und A folgende Vektoren bzw. Matrix:

$$\underbrace{\begin{pmatrix} a_{k+1} \\ a_{k+2} \\ a_{k+3} \\ \vdots \\ a_{k+m} \end{pmatrix}}_{\hat{a}_{k+1}} = \underbrace{\begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & 1 \\ \mu_0 & \mu_1 & \mu_2 & \dots & \mu_{m-1} \end{pmatrix}}_A \underbrace{\begin{pmatrix} a_k \\ a_{k+1} \\ a_{k+2} \\ \vdots \\ a_{k+m-1} \end{pmatrix}}_{\hat{a}_k}$$

- (ii) Dadurch das $a_{n+2} = a_n + a_{n+1}$ gilt, kann man folgern:

$$\hat{a}_{n+1} = \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}}_{=:A} \hat{a}_n \quad \text{mit} \quad \hat{a}_k = (a_k, a_{k+1})^T$$

Das charakteristische Polynom von A lautet dann folgendermaßen:

$$\det \begin{pmatrix} -\lambda & 1 \\ 1 & 1 - \lambda \end{pmatrix} = \lambda^2 - \lambda - 1 \stackrel{!}{=} 0$$

Daraus ergeben sich dann folgende Eigenwerte:

$$\lambda_1 = \frac{1 + \sqrt{5}}{2}, \quad \lambda_2 = \frac{1 - \sqrt{5}}{2}$$

Nun lässt sich \hat{a}_n darstellen:

$$\hat{a}_n = A^n \hat{a}_0 = \left(B^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} B \right)^n \hat{a}_0 = B^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}^n B \cdot \hat{a}_0 = B^{-1} \begin{pmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{pmatrix} B \cdot \hat{a}_0$$

Nun kann man a_n berechnen:

$$\begin{aligned} a_n &= c_1 \lambda_1^n + c_2 \lambda_2^n \\ a_0 &= c_1 + c_2 = 1 \quad (I) \\ a_1 &= c_1 \frac{1 + \sqrt{5}}{2} + c_2 \frac{1 - \sqrt{5}}{2} = 3 \quad \Leftrightarrow \underbrace{c_1 + c_2}_{=1} + (c_1 - c_2) \sqrt{5} = 6 \\ &\Rightarrow c_1 - c_2 = \sqrt{5} \quad (II) \end{aligned}$$

Durch Addition der beiden Gleichungen ergeben sich für c_1 und c_2 folgende Werte:

$$c_1 = \frac{1 + \sqrt{5}}{2} \quad c_2 = \frac{1 - \sqrt{5}}{2}$$

Daraus lässt sich dann a_n bestimmen:

$$a_n = \left(\frac{1 + \sqrt{5}}{2}\right)^{n+1} + \left(\frac{1 - \sqrt{5}}{2}\right)^{n+1}$$

(iii) Durch Logarithmieren der gegebenen Gleichung erhält man:

$$\begin{aligned} a_{n+2} &= a_n \cdot a_{n+1} \\ \underbrace{\log_2 a_{n+2}}_{=:b_{n+2}} &= \log_2(a_n \cdot a_{n+1}) = \underbrace{\log_2(a_n)}_{=:b_n} + \underbrace{\log_2(a_{n+1})}_{=:b_{n+1}} \end{aligned}$$

Nun kann man Aufgabe 10 (ii) anwenden und erhält analog dazu:

$$\begin{aligned} b_n &= c_1 \lambda_1^n + c_2 \lambda_2^n \\ b_0 &= c_1 + c_2 = 0 \quad (I) \\ b_1 &= c_1 \frac{1 + \sqrt{5}}{2} + c_2 \frac{1 - \sqrt{5}}{2} = 1 \quad \Leftrightarrow \underbrace{c_1 + c_2}_{=0} + (c_1 - c_2)\sqrt{5} = 2 \\ &\Rightarrow c_1 - c_2 = \frac{2}{\sqrt{5}} \quad (II) \end{aligned}$$

Durch Umformulierung der beiden Gleichungen ergeben sich für c_1 und c_2 folgende Werte:

$$c_1 = \frac{1}{\sqrt{5}} \quad c_2 = -\frac{1}{\sqrt{5}}$$

Dadurch ergibt sich für b_n :

$$b_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2}\right)^n - \left(\frac{1 - \sqrt{5}}{2}\right)^n \right]$$

Weshalb man dann für a_n erhält:

$$a_n = 2^{\frac{1}{\sqrt{5}}} \left[\left(\frac{1 + \sqrt{5}}{2}\right)^n - \left(\frac{1 - \sqrt{5}}{2}\right)^n \right]$$

Aufgabe 27 (Symmetry of orthogonal polynomials)

(4 Punkte)

Let (p_n) be a sequence of orthogonal polynomials with respect to the scalar product

$$(f, g)_\omega = \int_{-a}^a \omega(x) f(x) g(x) dx$$

provided by a symmetric, positive and integrable weight function $\omega : [-a, a] \rightarrow \mathbb{R}$. Show that $p_k(-x) = (-1)^k p_k(x)$ holds for all $x \in [-a, a]$ and $k \in \mathbb{N}_0$.

Lösung zu Aufgabe 27

Wir erhalten die Aussage $p_n(-x) = (-1)^n p_n(x)$ durch vollständige Induktion

n=0: $p_0(x) = \text{const.} \Rightarrow$ Aussage klar.

Induktionsschritt: Sei $f_n(x) = p_n(-x)$, dann gilt für $k = 0, \dots, n-1$

$$\begin{aligned} \int_{-a}^a \omega(x) f_n(x) p_k(x) dx &= \int_{-a}^a \omega(-x) p_n(-x) p_k(x) dx = \int_{-a}^a \omega(x) p_n(x) p_k(-x) dx \\ &\stackrel{IV}{=} (-1)^k \int_{-a}^a \omega(x) p_n(x) p_k(x) dx = 0. \end{aligned}$$

Also folgt $f_n \in \mathbb{P}_n \cap (\mathbb{P}_{n-1})^\perp$ und wir erhalten $f_n = \lambda p_n$ und somit $p_k(-x) = \lambda p_k(x)$

Sei

$$p_n(x) = \sum_{l=0}^n a_l x^l, \quad \text{also gilt} \quad \sum_{l=0}^n a_l (-1)^l x^l = \lambda a_l x^l.$$

Wir erhalten $a_n(-1)^n = \lambda a_n$ durch Vergleich des führenden Koeffizienten. Damit folgt $\lambda = (-1)^n$ und die Aussage gilt.

Aufgabe 28 (Tschebyscheff-Polynome)

(4+2+4+3 Punkte)

Gegeben sei für $k \geq 2$ und $x \in \mathbb{R}$ die Drei-Term-Rekursion der Tschebyscheff-Polynome:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), \quad k \geq 2, \quad x \in \mathbb{R}.$$

Zeigen Sie:

(i) Es gilt $T_k(x) = \cos(k \arccos(x))$ für $x \in [-1, 1]$.

(ii) Die Nullstellen von T_n sind

$$x_k := \cos\left(\frac{2k-1}{2n}\pi\right), \quad (k = 1, \dots, n).$$

(iii) Die Tschebyscheff-Polynome besitzen die globale Darstellung

$$T_k(x) = \frac{1}{2} \left(\left(x + \sqrt{x^2 - 1} \right)^k + \left(x - \sqrt{x^2 - 1} \right)^k \right), \quad \text{wobei } x \in \mathbb{R}.$$

(iv) Zeigen Sie, dass die Quadratur

$$\int_{-1}^1 \frac{q(x)}{\sqrt{1-x^2}} = \omega_1 q(x_1) + \dots + \omega_n q(x_n) \quad \text{für alle } q \in \mathbb{P}_{2n-1}$$

zu den Nullstellen x_1, \dots, x_n des Tschebyscheff-Polynoms T_n die Gewichte $\omega_k = \frac{\pi}{n}$ besitzt.

Lösung zu Aufgabe 28

(i) Sei $f_k(x) = \cos(k \arccos(x))$.

Induktionsanfang:

$$f_0(x) = 1 = T_0(x)$$

$$f_1(x) = x = T_1(x)$$

Induktionsschritt:

$$\begin{aligned} T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x) = 2x \cos(k \arccos(x)) - \cos((k-1) \arccos(x)) \\ &= \cos(\arccos(x)) \cos(k \arccos(x)) + x \cos(k \arccos(x)) - \cos((k-1) \arccos(x)) \\ &= \cos((k+1) \arccos(x)) + \sin(\arccos(x)) \sin(k \arccos(x)) + x \cos(k \arccos(x)) \\ &\quad - \cos(k \arccos(x)) \cos(\arccos(x)) + \sin(-\arccos(x)) \sin(k \arccos(x)) \\ &= f_{k+1}(x) \end{aligned}$$

(ii) Berechnung der Nullstellen

$$\begin{aligned}
T_k(x) = 0 &\Leftrightarrow n \arccos(x) = \frac{1}{2}(2k-1)\pi, \quad k \in \mathbb{Z} \\
&\Leftrightarrow \arccos(x) = \frac{2k-1}{2n}\pi, \quad k \in \mathbb{Z} \\
&\Leftrightarrow x = \cos\left(\frac{2k-1}{2n}\pi\right), \quad k \in \mathbb{Z} \\
&\Leftrightarrow x = \cos\left(\frac{2k-1}{2n}\pi\right), \quad k \in \{1, 2, \dots, n\}
\end{aligned}$$

(iii) Sei $g_k(x) = \frac{1}{2}((x + \sqrt{x^2 - 1})^k + (x - \sqrt{x^2 - 1})^k)$ Vollständige Induktion über k :Induktionsanfang: $g_0(x) = 1 = T_0(x)$, $g_1(x) = x = T_1(x)$ Induktionsschritt: Wir definieren uns $a := x + \sqrt{x^2 - 1}$, $b := x - \sqrt{x^2 - 1}$ und wir erhalten $a + b = 2x$ sowie $a \cdot b = 1$.

Somit ergibt sich

$$\begin{aligned}
T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x) = (a+b)\frac{1}{2}(a^k + b^k) - \frac{1}{2}(a^{k-1} + b^{k-1}) \\
&= \frac{1}{2}(a^{k+1} + b^{k+1}) + \frac{1}{2}(a^k b + a b^k - a^{k-1} - b^{k-1}) \\
&= \frac{1}{2}(a^{k+1} + b^{k+1}) = g_{k+1}(x)
\end{aligned}$$

und die Behauptung ist gezeigt.

(iv) Laut Skript sind die Gewichte durch

$$\begin{aligned}
\omega_k &= \frac{k_n}{k_{n-1}} \frac{h_{n-1}}{T'_n(x_k)T_{n-1}(x_k)} \\
&= \frac{2}{n} \cdot \frac{\sqrt{1-x_k^2}}{\sin(n \arccos(x_k))} \cdot \frac{1}{\cos((n-1) \arccos(x_k))} \int_{-1}^1 \frac{\cos^2((n-1) \arccos(x))}{\sqrt{1-x^2}} dx \\
&= \frac{2}{n} \cdot \frac{\sin(\frac{2k-1}{2n}\pi)}{\sin(\frac{2k-1}{2}\pi)} \cdot \frac{1}{\cos(\frac{2k-1}{2}\pi - \frac{2k-1}{2n}\pi)} \cdot \int_0^\pi \cos^2((n-1)t) dt \\
&= \frac{2}{n} \cdot \underbrace{\frac{\sin(\frac{2k-1}{2n}\pi)}{\sin(\frac{2k-1}{2}\pi)}}_{(-1)^{k-1}} \cdot \underbrace{\frac{1}{\sin(\frac{2k-1}{2}\pi) \sin(\frac{2k-1}{2n}\pi)}}_{(-1)^{k-1}} \cdot \frac{\pi}{2} = \frac{\pi}{n}
\end{aligned}$$

gegeben.

Aufgabe 29 (Legendre-Polynome)**(5+4 Punkte)**(i) Zeigen Sie mit Hilfe der Rodrigues-Formel $P_n(x) = \frac{1}{(-2)^n n!} \frac{d^n}{dx^n} (1-x^2)^n$, dass für die Legendre-Polynome P_n

$$\int_{-1}^1 P_m(x) P_n(x) dx = \begin{cases} 0 & \text{falls } m \neq n \\ \frac{2}{2n+1} & \text{falls } m = n \end{cases} \quad (m, n \in \mathbb{N}_0)$$

gilt. Sie dürfen dabei die Formel

$$\int_{-1}^1 (1-x^2)^n dx = \frac{2^{n+1}n!}{3 \cdot 5 \cdot 7 \cdot \dots \cdot (2n+1)}$$

verwenden.

Hinweis: Integrieren Sie mehrmals partiell.

(ii) Zeigen Sie, dass für $n \in \mathbb{N}$ die Rekursionsformel

$$(1-x^2)P'_n(x) = \frac{n(n+1)}{2n+1}(P_{n-1}(x) - P_{n+1}(x))$$

für die Legendre-Polynome P_n gilt.

Hinweis: Zeigen Sie, dass die Differenz aus linker und rechter Seite orthogonal zu allen P_k mit $k = 0, \dots, n+1$ ist. Das Verhältnis von den führenden Koeffizienten der Legendre-Polynome P_n und P_{n+1} erhalten Sie aus der Rekursionsformel

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x),$$

die Sie nicht beweisen müssen.

Lösung zu Aufgabe 29

(i) 1. Schritt. Für $k \leq m$ existiert ein Polynom p abhängig von k und m mit

$$\frac{\partial^k}{\partial x^k}(1-x^2)^m = (1-x^2)^{m-k}p(x)$$

Vollständige Induktion über k :

Induktionsanfang für $k=0$ ist klar.

Induktionsschritt:

$$\begin{aligned} \frac{\partial^{k+1}}{\partial x^{k+1}}(1-x^2)^m &= \frac{\partial}{\partial x}((1-x^2)^{m-k}p(x)) \\ &= (m-k)(1-x^2)^{m-k-1}(-2x) \cdot p(x) + (1-x^2)^{m-k}p'(x) \\ &= (1-x^2)^{m-(k+1)} \underbrace{((m-k)p(x) + (1-x^2)p'(x))}_{\text{Polynom}} \end{aligned}$$

2. Schritt. Durch mehrmalige Anwendung der partiellen Integration erhält man

$$\begin{aligned} &\int_{-1}^1 \frac{\partial^m}{\partial x^m}(1-x^2)^m \frac{\partial^k}{\partial x^k}(1-x^2)^k dx \\ &= \underbrace{\frac{\partial^{m-1}}{\partial x^{m-1}}(1-x^2)^m \frac{\partial^k}{\partial x^k}(1-x^2)^k \Big|_{-1}^1}_{=0} - \int_{-1}^1 \frac{\partial^{m-1}}{\partial x^{m-1}}(1-x^2)^m \frac{\partial^{k+1}}{\partial x^{k+1}}(1-x^2)^k dx \\ &= \dots = (-1)^m \int_{-1}^1 (1-x^2)^m \frac{\partial^{n+m}}{\partial x^{n+m}}(1-x^2)^n dx =: a_{n,m}, \end{aligned}$$

wobei stets der erste Summand aufgrund des ersten Schrittes gleich Null ist.

3.Schritt 1.Fall $m > n$ (analog $m < n$)

$$\begin{aligned}(P_m, P_n) &= \frac{1}{(-2)^m m! (-2)^n n!} a_{n,m} \\ &= \frac{1}{(-2)^m m! (-2)^n n!} (-1)^m \int_{-1}^1 (1-x^2)^m \underbrace{\frac{\partial^{n+m}}{\partial x^{n+m}} (1-x^2)^n dx}_{=0} = 0\end{aligned}$$

2.Fall $m=n$

$$\begin{aligned}(P_n, P_n) &= \left(\frac{1}{(-2)^n n!} \right)^2 a_{n,n} = \left(\frac{1}{(-2)^n n!} \right)^2 (-1)^n \int_{-1}^1 (1-x^2)^n \frac{\partial^{2n}}{\partial x^{2n}} (1-x^2)^n dx \\ &= \left(\frac{1}{(-2)^n n!} \right)^2 (-1)^n (2n)! (-1)^n \int_{-1}^1 (1-x^2)^n dx \\ &= \frac{(2n)! \cdot n! \cdot 2^{n+1}}{2^{2n} \cdot (n!)^2 \cdot 3 \cdot 5 \cdot \dots \cdot (2n+1)} = \frac{2}{2n+1} \cdot \frac{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2n}{2^n \cdot n!} = \frac{2}{2n+1}\end{aligned}$$

$$(ii) \underbrace{(1-x^2)P_n'(x)}_{=LHS} = \underbrace{\frac{n(n+1)}{2n+1}(P_{n-1}(x) - P_{n+1}(x))}_{=RHS}$$

1. $LHS - RHS \in \mathbb{P}_{n+1}$

2. (a) Sei $j \leq n-2$
Wir erhalten

$$(RHS, P_j) = \frac{n(n+1)}{2n+1} \left[\int_{-1}^1 P_{n-1}(x)P_j(x)dx - \int_{-1}^1 P_{n+1}(x)P_j(x)dx \right] = 0$$

und

$$\begin{aligned}(LHS, P_j) &= \int_{-1}^1 (1-x^2)P_n'(x)P_j(x)dx \\ &= \underbrace{(1-x^2)P_n(x)P_j(x)}_{=0} \Big|_{-1}^1 - \int_{-1}^1 P_n(x) \underbrace{(1-x^2)P_j'(x)}_{\in \mathbb{P}_{n-1}} dx \\ &\quad + \int_{-1}^1 P_n(x) \underbrace{2xP_j(x)}_{\in \mathbb{P}_{n-1}} dx \\ &= 0\end{aligned}$$

(b) Weiterhin gilt:

$$(RHS, P_n) = 0$$

$$\begin{aligned}
(LHS, P_n) &= \int_{-1}^1 (1-x^2)P_n'P_n dx = (1-x^2)P_n^2(x) \Big|_{-1}^1 - \underbrace{\int_{-1}^1 (1-x^2)P_nP_n' dx}_{=(LHS, P_n)} \\
&\quad + \int_{-1}^1 2xP_n^2 dx \\
&\Rightarrow 2(LHS, P_n) = 2 \int_{-1}^1 xP_n^2 dx = 0
\end{aligned}$$

da nach Aufgabe 11 P_n^2 eine gerade Funktion ist und xP_n^2 damit ungerade ist. Es folgt $(LHS, P_n) = 0$.

(c) Hinweis: vergleicht man die führenden Koeffizienten in der Rekursionsformel

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$

so folgt

$$(n+1)k_{n+1} = (2n+1)k_n \Rightarrow \frac{k_n}{k_{n+1}} = \frac{n+1}{2n+1}.$$

Es folgt nun mit dem Hinweis und der nachfolgenden Anmerkung

$$(RHS, P_{n-1}) = \frac{n(n+1)}{2n+1} (P_{n-1}, P_{n-1}) \stackrel{13(i)}{=} \frac{n(n+1)}{2n+1} \cdot \frac{2}{2n-1}$$

$$\begin{aligned}
(LHS, P_{n-1}) &= (1-x^2)P_nP_{n-1} \Big|_{-1}^1 - \int_{-1}^1 [(1-x^2)P_{n-1}]' P_n dx \\
&= - \int_{-1}^1 (1-x^2)P_nP_{n-1}' dx + 2 \int_{-1}^1 xP_nP_{n-1} dx \\
&= (n-1) \frac{k_{n-1}}{k_n} \int_{-1}^1 P_n^2 dx + 2 \frac{k_{n-1}}{k_n} \int_{-1}^1 P_n^2 dx \\
&= (n+1) \frac{k_{n-1}}{k_n} \cdot \frac{2}{2n+1} = (n+1) \frac{n}{2n-1} \cdot \frac{2}{2n+1}
\end{aligned}$$

wobei k_l der führende Koeffizient von P_l ist.

Anmerkung:

$$(1-x^2)P_{n-1}' = -(n-1) \frac{k_{n-1}}{k_n} P_n + q_1 \quad \text{mit } q_1 \in \mathbb{P}_{n-1}$$

$$xP_{n-1} = \frac{k_{n-1}}{k_n} P_n + q_2 \quad \text{mit } q_2 \in \mathbb{P}_{n-1}$$

(d)

$$(RHS, P_{n+1}) = -\frac{n(n+1)}{2n+1} \cdot \frac{2}{2n+3}$$

$$\begin{aligned} (LHS, P_{n+1}) &= \int_{-1}^1 (1-x^2)P_n'(x)P_{n+1}(x)dx = -n \cdot \frac{k_n}{k_{n+1}} \cdot \frac{2}{2n+3} \\ &= -n \cdot \frac{n+1}{2n+1} \cdot \frac{2}{2n+3} \end{aligned}$$

Anmerkung:

$$(1-x^2)P_n' = -n \frac{k_n}{k_{n+1}} P_{n+1} + q_3 \quad \text{mit } q_3 \in \mathbb{P}_n$$

insgesamt: LHS-RHS ist orthogonal zu einer Basis von \mathbb{P}_{n+1} . Es folgt $LHS - RHS = 0$.

Aufgabe 30 (Nullstellen von Orthogonalpolynome)

(3 Punkte)

Beweisen Sie Satz 2.5.1: Sei $Ap = xp + r$ das durch die Drei-Term-Rekursion für Orthogonalpolynome entstehende Gleichungssystem (siehe 2.16 im Skript, Kapitel 2). Dann sind die Nullstellen von p_{n+1} die Eigenwerte von A .

Lösung zu Aufgabe 30

Sei x_k eine Nullstelle von p_{k+1} . Dann ist x_k Eigenwert von A , da $Ap = xp$ mit $p \neq 0$ ($p_0(x_k) \neq 0$). Nach Satz 2.1.16 besitzt p_{n+1} genau $n+1$ verschiedene Nullstellen x_1, \dots, x_{n+1} . Also sind x_1, \dots, x_{n+1} Eigenwerte und es gilt $\text{Spektrum}(A) = \{x_1, \dots, x_{n+1}\}$ wegen $A \in \mathbb{R}^{(n+1) \times (n+1)}$.

Aufgabe 31 (Newton, Lagrange, Aitken-Neville, Hermite)

(6+3 Punkte)

- (i) Berechnen Sie das Interpolationspolynom $P \in \mathbb{P}_3$ in der Darstellung von Lagrange, mittels dividierter Differenzen in der Darstellung von Newton für die Punkte

$$\begin{array}{c|cccc} x_i & -1 & 1 & 2 & 3 \\ \hline y_i & 4 & 0 & -2 & 4 \end{array}$$

sowie den Wert $P(\frac{3}{2})$ mit Hilfe des Algorithmus von Aitken und Neville.

- (ii) Berechnen Sie mit Hilfe dividierter Differenzen das Polynom P bezüglich Newton-Basis vom kleinsten Grad mit den Eigenschaften $P(1) = -2$, $P'(1) = -2$, $P''(1) = 0$, $P(2) = -3$ und $P'(2) = 1$.

Lösung zu Aufgabe 31

- (i) Per Lagrange erhält man:

$$\begin{aligned} P(x) &= 4 \frac{(x-1)(x-2)(x-3)}{(-1-1)(-1-2)(-1-3)} + 0 \frac{(x+1)(x-2)(x-3)}{(1+1)(1-2)(1-3)} \\ &\quad - 2 \frac{(x+1)(x-1)(x-3)}{(2+1)(2-1)(2-3)} + 4 \frac{(x+1)(x-1)(x-2)}{(3+1)(3-1)(3-2)} \\ &= -\frac{1}{6}(x-1)(x-2)(x-3) + \frac{2}{3}(x+1)(x-1)(x-3) + \frac{1}{2}(x+1)(x-1)(x-2) \\ &= x^3 - 2x^2 - 3x + 4 \end{aligned}$$

Man erhält das Polynom

$$p(x) = 4 - 2(x+1) + 0 \cdot (x+1)(x-1) + 1 \cdot (x+1)(x-1)(x-2)$$

mittels dividierter Differenzen.

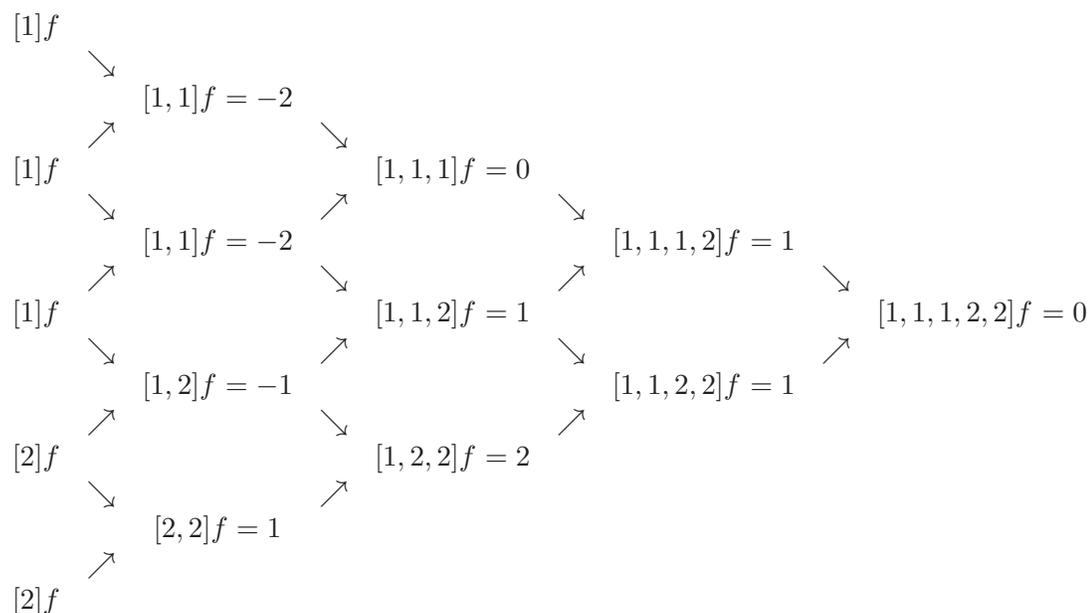
$$\begin{array}{r}
 [-1]f = 4 \\
 \searrow \\
 [-1, 1]f = \frac{0-4}{1-(-1)} = -2 \\
 \nearrow \\
 [1]f = 0 \\
 \searrow \\
 [1, 2]f = \frac{-2-0}{2-1} = -2 \\
 \nearrow \\
 [2]f = -2 \\
 \searrow \\
 [2, 3]f = \frac{4+2}{3-2} = 6 \\
 \nearrow \\
 [3]f = 4
 \end{array}
 \quad
 \begin{array}{r}
 \searrow \\
 [-1, 1, 2]f = 0 \\
 \nearrow \\
 \searrow \\
 [1, 2, 3]f = 4 \\
 \nearrow \\
 [-1, 1, 2, 3]f = 1
 \end{array}$$

Aitken-Neville liefert an der Stelle $\frac{3}{2}$ den Wert $P(\frac{3}{2}) = -\frac{13}{8}$.

$$\begin{array}{r}
 -1 \mid 4 \\
 \searrow \\
 0 + \frac{\frac{3}{2}-1}{+1-(-1)}(0-4) = -1 \\
 \nearrow \\
 1 \mid 0 \\
 \searrow \\
 -2 + \frac{\frac{3}{2}-2}{2-1}(-2-0) = -1 \\
 \nearrow \\
 2 \mid -2 \\
 \searrow \\
 4 + \frac{\frac{3}{2}-3}{3-2}(4-(-2)) = -5 \\
 \nearrow \\
 3 \mid 4
 \end{array}
 \quad
 \begin{array}{r}
 \searrow \\
 -1 + \frac{\frac{3}{2}-2}{2+1}(-1+1) = -1 \\
 \nearrow \\
 \searrow \\
 -5 + \frac{\frac{3}{2}-3}{3-1}(-5-(-1)) = -2 \\
 \nearrow \\
 -\frac{13}{8}
 \end{array}$$

(ii) Mit Hilfe der dividierten Differenzen erhält man

$$p(x) = -2 - 2(x-1) + (x-1)^3$$

**Aufgabe 32 (Interpolationsfehler und Konvergenzeigenschaften)****(3+3 Punkte)**

- (i) Es seien $f : [a, b] \rightarrow \mathbb{R}$ dreimal stetig differenzierbar und $P \in \mathbb{P}_2$ das zugehörige quadratische Interpolationspolynom zu den Stützstellen $x_i := a + hi$, $i = 0, 1, 2$, $h := (b - a)/2$. Zeigen Sie:

$$\|f - P\|_\infty \leq \frac{\sqrt{3}}{27} h^3 \|f'''\|_\infty.$$

- (ii) Die Funktion $f : [0, 1] \rightarrow \mathbb{R}$ sei definiert durch $f(x) := \log(x + 1)$. Zu einer beliebig vorgegebenen Folge von Stützstellenvektoren

$$\left(x_i^{(n)}\right)_{0 \leq i \leq n} \subset [0, 1], \quad n \in \mathbb{N}$$

sei P_n das zugehörige Interpolationspolynom aus \mathbb{P}_n .

Zeigen Sie, daß die Folge der P_n auf $[0, 1]$ gleichmäßig gegen f konvergiert.

Lösung zu Aufgabe 32

- (i) Nach dem Satz 3.2.11 gilt

$$f(x) - P(x) = \frac{1}{3!} (x - a) \left(x - \frac{a+b}{2}\right) (x - b) f'''(\xi)$$

Sei

$$\begin{aligned} g(t) &= (t - h)t(t + h) = t^3 - h^2t \\ g'(t) &= 3t^2 - h^2 \end{aligned}$$

und man erhält

$$x_E = \pm \frac{h}{\sqrt{3}}$$

als Extremstellen, welche man in g einsetzt

$$g(x_E) = \pm \frac{h^3}{3\sqrt{3}} - h^2 \left(\pm \frac{h}{\sqrt{3}}\right) = \mp \frac{2}{3\sqrt{3}} h^3$$

und man kann folgern:

$$\|\omega\|_{[a,b],\infty} = \|g\|_{[-h,h],\infty} = \frac{2}{3\sqrt{3}}h^3$$

Die Abschätzung

$$\|f - P_n\|_\infty \leq \frac{1}{6} \|(x-a)(x-\frac{a+b}{2})(x-b)\|_\infty \|f'''(\xi)\|_\infty = \frac{1}{6} \|g\|_\infty \|f'''(\xi)\|_\infty = \frac{\sqrt{3}}{27} h^3 \|f'''\|_\infty$$

liefert die Behauptung.

(ii) Es gilt

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

und man erhält für $f(x) = \log(x+1)$

$$f^{(n)}(x) = (-1)^{n-1} (n-1)! \frac{1}{(1+x)^n}.$$

Diese Aussage kann induktiv gezeigt werden. Und nun gilt

$$\|f - P_n\|_\infty \leq \underbrace{\left\| \prod_{j=0}^n \underbrace{(x - x_j^{(n)})}_{\leq 1} \right\|_\infty}_{< 1} \frac{n!}{(n+1)!} \leq \frac{1}{n+1} \rightarrow 0 \quad (n \rightarrow \infty)$$

Aufgabe 33 (Einheitskreis als parametrisierte Kurve)

(3+3+2+3 Punkte)

- (i) Zeigen Sie, dass das Kreissegment $\{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1, x \geq 0, y \geq 0\}$ nicht durch eine parametrisierte Kurve $s = (s_1, s_2) : [a, b] \rightarrow \mathbb{R}^2$ dargestellt werden kann, deren Komponenten s_1, s_2 Polynome sind.
- (ii) Gegeben sei $s : [0, 1] \rightarrow \mathbb{R}^2$ mit

$$s(t) = \left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2} \right).$$

Zeigen Sie $\text{Bild}(s) = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1, x \geq 0, y \geq 0\}$.

- (iii) Seien $a > 0$ und $b > 0$. Zeigen Sie, dass es zwei rationale Funktionen $s_1, s_2 : [0, 1] \rightarrow \mathbb{R}$ gibt mit

$$\text{Bild}(s) = \left\{ (x, y) \in \mathbb{R}^2 : \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1, x \geq 0, y \geq 0 \right\},$$

d. h. dass die Kurve $s = (s_1, s_2)$ ein Ellipsensegment beschreibt.

- (iv) Seien $a > 0$ und $b > 0$. Zeigen Sie, dass es zwei rationale Funktionen $s_1, s_2 : (-1, 1) \rightarrow \mathbb{R}$ gibt mit

$$\text{Bild}(s) = \left\{ (x, y) \in \mathbb{R}^2 : \left(\frac{x}{a}\right)^2 - \left(\frac{y}{b}\right)^2 = 1, x > 0 \right\},$$

d. h. dass die Kurve $s = (s_1, s_2)$ einen Hyperbelast beschreibt.

Lösung zu Aufgabe 33

(i) Seien s_1, s_2 Polynome mit $\text{Bild}(x) = \{(x, y) : x^2 + y^2 = 1, x \geq 0, y \geq 0\}$ dann gilt

$$0 = \text{grad}(1) = \text{grad}(s_1^2 + s_2^2) = \max(2\text{grad}(s_1), 2\text{grad}(s_2))$$

da sich der Grad bei der Addition aufgrund der positiven führenden Koeffizienten von s_1^2 und s_2^2 nicht verkleinern kann. Deshalb ist der

$$\text{grad}(s_1) = \text{grad}(s_2) = 0$$

weshalb $s_1 = \text{const}$ und $s_2 = \text{const}$.

Nun folgt, dass das Bild(s) nur einelementig ist. Dies stellt aber einen Widerspruch dar und die Behauptung folgt unmittelbar.

(ii) Es gelten folgende Eigenschaften:

- $s(0) = (1, 0)$, $s(1) = (0, 1)$
- $s_1(t) \geq 0$, $s_2(t) \geq 0$
- $s_1^2 + s_2^2 = \left(\frac{1-t^2}{t^2+1}\right)^2 + \left(\frac{2t}{t^2+1}\right)^2 = \left(\frac{t^2+1}{t^2+1}\right)^2 = 1$

Also gilt mit dem ZWS

$$\text{Bild}(s) = \{(x, y) \in \mathbb{R} : x^2 + y^2 = 1, x \geq 0, y \geq 0\}.$$

(iii) Analoge Rechnung wie in (ii) nur mit

$$s(t) = \left(a \frac{1-t^2}{1+t^2}, b \frac{2t}{1+t^2} \right).$$

(iv) Sei $\tilde{s}_1^2 + \tilde{s}_2^2 = 1$ dann folgt:

$$\begin{aligned} \tilde{s}_1^2 + \tilde{s}_2^2 = 1 &\iff \tilde{s}_1^2 = 1 - \tilde{s}_2^2 \\ &\iff 1 = \frac{1}{\tilde{s}_1^2} - \left(\frac{\tilde{s}_2}{\tilde{s}_1}\right)^2 \\ &\iff 1 = \frac{1}{a^2} \underbrace{\left(\frac{a}{\tilde{s}_1}\right)^2}_{:=s_1} - \frac{1}{b^2} \underbrace{\left(\frac{b\tilde{s}_2}{\tilde{s}_1}\right)^2}_{:=s_2} \end{aligned}$$

Für $\tilde{s}_1(t) = \frac{1-t^2}{1+t^2}$ und $\tilde{s}_2(t) = \frac{2t}{1+t^2}$ erhält man

$$s_1(t) = a \frac{1+t^2}{1-t^2} > 0, \quad s_2(t) = b \frac{2t}{1-t^2}$$

für $t \in (-1, 1)$. Betrachtet man das Verhalten von $s(t)$ für $t \rightarrow 1$ und $t \rightarrow -1$ folgt dass

$$\lim_{t \rightarrow 1} = (\infty, \infty), \quad \lim_{t \rightarrow -1} = (\infty, -\infty)$$

gilt. Und somit ergibt sich

$$\text{Bild}(s) = \{(x, y) \in \mathbb{R} : \left(\frac{x}{a}\right)^2 - \left(\frac{y}{b}\right)^2 = 1, x > 0\}$$

Aufgabe 34 (Thielscher Kettenbruch)**(6 Punkte)**

Definition. Zu $(n+1)$ paarweise verschiedenen Stützstellen x_0, \dots, x_n und zugehörigen Werten $y_0 = f(x_0), \dots, y_n = f(x_n)$ definieren wir:

1. die ersten **inversen dividierten Differenzen**:

$$\varphi_1(x_i, x_0) := \frac{x_i - x_0}{f(x_i) - f(x_0)}, \quad (i = 1, \dots, n),$$

2. die **k -ten inversen dividierten Differenzen** ($k = 2, \dots, n$):

$$\varphi_k(x_i, x_{k-1}, \dots, x_0) := \frac{x_i - x_{k-1}}{\varphi_{k-1}(x_i, x_{k-2}, \dots, x_0) - \varphi_{k-1}(x_{k-1}, x_{k-2}, \dots, x_0)},$$

für $(i = k, \dots, n)$.

Definition. Für $n \in \mathbb{N}$ definieren wir den abgebrochenen Thielschen Kettenbruch:

$$R_n(x) = c_0 + \frac{x - x_0}{c_1 + \frac{x - x_1}{c_2 + \frac{x - x_2}{\ddots + \frac{x - x_{n-2}}{c_{n-1} + \frac{x - x_{n-1}}{c_n}}}}$$

mit $c_0 := f(x_0)$, $c_k := \varphi_k(x_k, \dots, x_0)$, $k = 1, \dots, n$.

Zeigen Sie, dass der abgebrochene **Thielsche Kettenbruch** $R_n(x)$ die spezielle **rationale Interpolationsaufgabe**

(a)

$$R_n(x) = \frac{p_0 + p_1x + \dots + p_\zeta x^\zeta}{q_0 + q_1x + \dots + q_\nu x^\nu} = \frac{P_\zeta(x)}{Q_\nu(x)},$$

(b)

$$R_n(x_i) = y_i, \quad (i = 0, \dots, n),$$

(c)

$$\begin{cases} \zeta = \nu = \frac{n}{2}, & n \text{ gerade,} \\ \zeta = \frac{1}{2}(n+1), \nu = \frac{1}{2}(n-1), & n \text{ ungerade,} \end{cases}$$

löst.

Lösung zu Aufgabe 34

Sei

$$S_k(x) = c_k + \frac{x - x_k}{c_{k+1} + \frac{x - x_{k+1}}{c_{k+2} + \frac{x - x_{k+2}}{\ddots + \frac{x - x_{n-1}}{c_n}}}}$$

Es gilt:

- $S_0(x) = R_n(x), S_n(x) = c_n$
- $S_k(x) = c_k + \frac{x-x_k}{S_{k+1}(x)} (*)$
- $S_k(x_k) = c_k \Rightarrow R_n(x_k) = R_k(x_k)$

Für Teil (b) muss $R_n(x_n) = f(x_n), n \in \mathbb{N}_0$ gezeigt werden.

Wir beweisen die Behauptung $S_k(x_n) = \varphi_k(x_n, x_{k-1}, x_{k-2}, \dots, x_0)$ für $k = 1, 2, \dots, n$ mit vollständiger Induktion (rückwärts) über k

Induktionsanfang: $S_n(x_n) = c_n = \varphi_n(x_n, x_{n-1}, \dots, x_0)$

Induktionsvoraussetzung: $S_{k+1}(x_n) = \varphi_{k+1}(x_n, x_k, x_{k-1}, \dots, x_0)$

Induktionsschluss:

$$\begin{aligned} S_k(x_n) &= c_k + \frac{x_n - x_k}{S_{k+1}(x_n)} = \varphi_k(x_k, x_{k-1}, \dots, x_0) + \frac{x_n - x_k}{\varphi_{k+1}(x_n, x_k, \dots, x_0)} \\ &\stackrel{Def.}{=} \varphi_k(x_k, x_{k-1}, \dots, x_0) + \varphi_k(x_n, x_{k-1}, \dots, x_0) - \varphi_k(x_k, x_{k-1}, \dots, x_0) \\ &= \varphi_k(x_n, x_{k-1}, x_{k-2}, \dots, x_0) \end{aligned}$$

Damit gilt:

$$R_n(x_n) = S_0(x_n) = c_0 + \frac{x_n - x_0}{S_1(x_n)} = f(x_0) + \frac{x_n - x_0}{\varphi_1(x_n, x_0)} = f(x_0) + f(x_n) - f(x_0) = f(x_n)$$

Zu Teil (a) und (c):

Wir betrachten die Behauptung: \exists Polynome P_k, Q_k mit $S_k = \frac{P_k}{Q_k} (k = 0, \dots, n)$ und

$$\begin{aligned} \text{grad}(P_k) &\leq \frac{n-k}{2}, \quad \text{grad}(Q_k) \leq \frac{n-k}{2}, \quad \text{falls } (n-k) \text{ gerade} \\ \text{grad}(P_k) &\leq \frac{1}{2}(n-k+1), \quad \text{grad}(Q_k) \leq \frac{1}{2}(n-k-1), \quad \text{falls } (n-k) \text{ ungerade} \end{aligned}$$

und beweisen sie per vollständiger Induktion (rückwärts).

Induktionsanfang: $S_n(x) = \frac{P_n(x)}{Q_n(x)} = c_n, P_n = c_n, Q_n = 1 \Rightarrow \text{grad}(P_n) = \text{grad}(Q_n) = 0$

Induktionsschluss:

$$S_k(x) = c_k + \frac{x - x_k}{S_{k+1}(x)} = c_k + \frac{x - x_k}{\frac{P_{k+1}(x)}{Q_{k+1}(x)}} = \frac{\overbrace{c_k + P_{k+1}(x) + (x - x_k)Q_{k+1}(x)}^{=: P_k(x)}}{\underbrace{P_{k+1}(x)}_{=: Q_k(x)}} = \frac{P_k(x)}{Q_k(x)}$$

1.Fall $(n - (k + 1))$ gerade $\Leftrightarrow (n - k)$ ungerade

$$\begin{aligned} \Rightarrow \text{grad}(Q_k) &= \text{grad}(P_{k+1}) \leq \frac{n-k-1}{2} \\ \Rightarrow \text{grad}(P_k) &\leq \max(\underbrace{\text{grad}(P_{k+1})}_{\leq \frac{n-k-1}{2}}, 1 + \underbrace{\text{grad}(Q_{k+1})}_{\leq \frac{n-k-1}{2}}) \leq \frac{n-k+1}{2} \end{aligned}$$

2.Fall $(n - (k + 1))$ ungerade $\Leftrightarrow (n - k)$ gerade

$$\begin{aligned} \Rightarrow \text{grad}(Q_k) &= \text{grad}(P_{k+1}) \leq \frac{1}{2}(n - (k + 1) + 1) = \frac{1}{2}(n - k) \\ \Rightarrow \text{grad}(P_k) &\leq \max(\underbrace{\text{grad}(P_{k+1})}_{\leq \frac{1}{2}(n-k)}, 1 + \underbrace{\text{grad}(Q_{k+1})}_{\leq \frac{1}{2}(n-k-2)}) \leq \frac{1}{2}(n - k) \end{aligned}$$

Somit erhalten wir $R_n(x) = S_0(x) = \frac{P_0(x)}{Q_0(x)}$ mit der bewiesenen Behauptung für $k = 0$.

Aufgabe 35 (Fehlerabschätzung für numerische Intergration) (2 Punkte)

Let f be a sufficiently smooth function on $[a, b]$. The polynomial $p_n \in P_n$ is the corresponding interpolation polynomial of f for $n \in \mathbb{N}_0$ and $x_0, \dots, x_n \in [a, b]$. Proof, that the inequality

$$\int_a^b |f(x) - p_n(x)| dx \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty \int_a^b \left| \prod_{j=0}^n (x - x_j) \right| dx$$

holds.

Lösung zu Aufgabe 35

Durch die Gleichung

$$|f(x) - p(x)| = \frac{1}{(n+1)!} |(x - x_0)(x - x_1) \dots (x - x_n)| |f^{(n+1)}(\xi)|$$

erhält man die Abschätzung:

$$\begin{aligned} \int_a^b |f(x) - p(x)| dx &= \frac{1}{(n+1)!} \int_a^b \left| \prod_{j=0}^n (x - x_j) \right| |f^{(n+1)}(\xi(x))| dx \\ &\leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty \int_a^b \left| \prod_{j=0}^n (x - x_j) \right| dx \end{aligned}$$

Aufgabe 36 (Newton-Cotes-Formel) (3+4+3+4+3+4 Punkte)

Sei f eine hinreichend glatte Funktion auf $[a, b]$. Für $n \in \mathbb{N}$ betrachten wir die äquidistante Intervalleinteilung $[x_i, x_{i+1}]$ von $[a, b]$ mit $x_i = a + ih$, $i = 0, \dots, n$, $h = \frac{b-a}{n}$. Sei $p \in \mathbb{P}_n$ das Interpolationspolynom von f bzgl. der Stützstellen x_i .

(i) Zeigen Sie, dass

$$\int_a^b p(x) dx = (b-a) \sum_{i=0}^n \lambda_i f_i \quad \text{mit} \quad \lambda_i = \frac{1}{n} \int_0^n \prod_{\substack{k=0 \\ k \neq i}}^n \frac{z-k}{i-k} dz \quad (\text{C.1})$$

gilt.

(ii) Zeigen Sie, dass $\lambda_{n-i} = \lambda_i$ für $i = 1, \dots, n$ und $\sum_{i=0}^n \lambda_i = 1$ gilt.

(iii) Sei n gerade. Zeigen Sie, dass die Quadraturformel (C.1) sogar für $p \in \mathbb{P}_{n+1}$ gilt.

(iv) Berechnen Sie die Gewichte $\lambda_0, \dots, \lambda_n$ für $n = 1$ (Trapezregel), $n = 2$ (Simpsonregel) und $n = 3$ (3/8 Regel).

(v) Für die Trapezregel zeige man für den Approximationsfehler

$$\left| \int_a^b p(x) dx - \int_a^b f(x) dx \right| \leq \frac{1}{12} (b-a)^3 \|f''\|_\infty.$$

(vi) Die Unterteilung von $[a, b]$ in N Teilintervalle $[t_i, t_{i+1}]$, $t_i = a + ih$, $i = 0, \dots, N$, $h = \frac{b-a}{N}$, die Anwendung der Trapezformel in jedem Teilintervall und die anschließende Summation der Näherungswerte führt zu den summierten Quadraturformeln. Geben Sie die summierte Trapezregel an und schätzen Sie ihren Approximationsfehler ab.

Lösung zu Aufgabe 36

(i) Für das Interpolationspolynom gilt:

$$\begin{aligned} \int_a^b p(x) dx &= \int_a^b \sum_{i=0}^n f_i L_i(x) dx = \sum_{i=0}^n f_i \int_a^b \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j} dx \\ &\stackrel{y = n \frac{x-a}{b-a}}{=} \sum_{i=0}^n f_i \frac{b-a}{n} \int_0^n \prod_{j=0, j \neq i}^n \frac{hy + a - a - jh}{a + hi - a - hj} dy \\ &= (b-a) \sum_{i=0}^n f_i \int_0^n \prod_{j=0, j \neq i}^n \frac{y-j}{i-j} dy \end{aligned}$$

(ii) Zuerst betrachten wir $n \cdot \lambda_{n-i}$.

$$\begin{aligned} n \cdot \lambda_{n-i} &= \int_0^n \prod_{k=0, k \neq n-i}^n \frac{z-k}{n-i-k} dz \stackrel{y=-z+n}{=} \int_0^n \prod_{k=0, k \neq n-i}^n \frac{n-y-k}{n-i-k} dy \\ &\stackrel{k=n-j}{=} \int_0^n \prod_{j=0, j \neq i}^n \frac{n-y-n+j}{n-i-n+j} dy = \int_0^n \prod_{j=0, j \neq i}^n \frac{y-j}{i-j} dy = n \cdot \lambda_i \end{aligned}$$

Daraus folgt unmittelbar die erste Behauptung.

Um die zweite Behauptung zu zeigen, setzen wir $p(x) = 1$.

$$\begin{aligned} \Rightarrow b-a &= \int_a^b 1 dx = (b-a) \sum_{i=0}^n \lambda_i \\ \Rightarrow \sum_{i=0}^n \lambda_i &= 1. \end{aligned}$$

(iii) Sei n gerade und $p \in \mathbb{P}_n$. Des Weiteren sei $p_n \in \mathbb{P}_n$ der Interpolant von p zu den Stützstellen x_0, \dots, x_n und $p(x_i) = \bar{p}_i$. Die Behauptung folgt, da

$$\begin{aligned} \int_a^b p(x) dx - (b-a) \sum_{i=0}^n \lambda_i \bar{p}_i &= \int_a^b p(x) dx - \int_a^b p_n(x) dx = \int_a^b (p - p_n) dx \\ &= \frac{1}{(n+1)!} \int_a^b \underbrace{p^{(n+1)}(\xi)}_{=: c \text{ da } p \in \mathbb{P}_{n+1}} (x-x_0) \cdot \dots \cdot (x-x_n) dx \\ &= \frac{c}{(n+1)!} \underbrace{\int_a^b (x-x_0) \cdot \dots \cdot (x-x_n) dx}_{=0 \text{ da punktsym. zu } (\frac{a+b}{2}, 0)} = 0 \end{aligned}$$

(iv) Für die Trapezregel ($n = 1$) sind die Gewichte durch

$$\lambda_0 = \int_0^1 \frac{z-1}{0-1} dz = \frac{1}{2} = \lambda_1$$

gegeben. Für $n = 2$ kann man die Gewichte durch

$$\begin{aligned} \lambda_0 &= \frac{1}{2} \int_0^2 \frac{(z-1)(z-2)}{2} dz = \frac{1}{4} \int_0^2 (z^2 - 3z + 2) dz = \frac{1}{4} \left[\frac{1}{3} z^3 - \frac{3}{2} z^2 + 2z \right]_0^2 \\ &= \frac{1}{6} = \lambda_2 \\ \lambda_1 &= 1 - \lambda_0 - \lambda_2 = \frac{2}{3} \end{aligned}$$

berechnen. Für die 3/8-Regel ($n = 3$) erhält man die Gewichte

$$\begin{aligned}\lambda_0 &= \frac{1}{3} \int_0^3 \frac{(z-1)(z-2)(z-3)}{(0-1)(0-2)(0-3)} dz = -\frac{1}{18} \int_0^3 (z^3 - 6z^2 + 11z - 6) dz \\ &= -\frac{1}{18} \left[\frac{1}{4}z^4 - 2z^3 + \frac{11}{2}z^2 - 6z \right]_0^3 = \frac{1}{8} = \lambda_3 \\ 2\lambda_2 &= 1 - \lambda_0 - \lambda_3 = \frac{3}{4} \Rightarrow \lambda_2 = \lambda_1 = \frac{3}{8}\end{aligned}$$

(v) Die Abschätzung

$$\begin{aligned}\left| \int_a^b (f(x) - p(x)) dx \right| &= \left| \frac{1}{2} \int_a^b (x-a)(x-b) f''(\xi(x)) dx \right| \\ &\leq \frac{1}{2} \|f''\|_\infty \left| \int_a^b (x-a)(x-b) dx \right| \\ &= \frac{1}{2} \|f''\|_\infty \left| \int_{-\frac{b-a}{2}}^{\frac{b-a}{2}} \frac{b-a}{2} x^2 - \left(\frac{b-a}{2}\right)^2 dx \right| \\ &= \frac{1}{2} \|f''\|_\infty 2 \left| \left[\frac{1}{3} x^3 - \left(\frac{b-a}{2}\right)^2 x \right]_0^{\frac{b-a}{2}} \right| \\ &= \frac{1}{12} (b-a)^3 \|f''\|_\infty\end{aligned}$$

liefert die Behauptung.

(vi) Die summierte Trapezregel

$$\begin{aligned}\int_a^b p(x) dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} p(x) dx = \sum_{i=0}^{n-1} h \left(\frac{1}{2} p(x_i) + \frac{1}{2} p(x_{i+1}) \right) \\ &= \frac{b-a}{n} \left[\frac{1}{2} p(x_1) + \sum_{i=1}^{n-1} p(x_i) + \frac{1}{2} p(x_n) \right]\end{aligned}$$

liefert mit Hilfe folgender Abschätzung den Approximationsfehler

$$\begin{aligned}\left| \int_a^b (f(x) - p(x)) dx \right| &\leq \sum_{i=0}^{n-1} \left| \int_{x_i}^{x_{i+1}} (f(x) - p(x)) dx \right| \\ &\leq \sum_{i=0}^{n-1} \frac{1}{12} \left(\frac{b-a}{2} \right)^3 \|f''\|_\infty = \frac{(b-a)^3}{12n^2} \|f''\|_\infty.\end{aligned}$$

Aufgabe 37 (Gauß-Quadratur)

(4+4 Punkte)

(i) Bestimmen Sie x_1, x_2, x_3 und $\omega_1, \omega_2, \omega_3$, so dass die Quadraturformel

$$\int_{-1}^1 p(x) dx = \omega_1 p(x_1) + \omega_2 p(x_2) + \omega_3 p(x_3)$$

für alle Polynome $p \in \mathbb{P}_5$ gilt.

Hinweis: Verwenden Sie Satz 2.1.17 und die Rodriguesformel für die Legendre-Polynome P_n .

(ii) Bestimmen Sie x_1, x_2, x_3 und $\omega_1, \omega_2, \omega_3$, so dass

$$\int_{-\infty}^{\infty} e^{-x^2} p(x) dx = \omega_1 p(x_1) + \omega_2 p(x_2) + \omega_3 p(x_3)$$

für alle $p \in \mathbb{P}_5$ gilt.

Hinweis: Verwenden Sie Satz 2.1.17 und die Rodriguesformel

$$H_k(x) = (-1)^k e^{x^2} \frac{d^k}{dx^k} e^{-x^2}, \quad k \in \mathbb{N}_0.$$

für die Hermite-Polynome H_k .

Lösung zu Aufgabe 37

Aufgabe 38 (Mehrdimensionale Integration)

(4+4 Punkte)

(i) Seien $(0, 0), (1, 0), (0, 1) \in \mathbb{R}^2$ die Eckpunkte des Dreiecks D . Zeigen Sie, dass Gewichte $\omega_1, \dots, \omega_4$ existieren, sodass die Quadraturformel

$$\frac{1}{|D|} \int_D p(x, y) d(x, y) = \omega_1 p(0, 0) + \omega_2 p(1, 0) + \omega_3 p(0, 1) + \omega_4 p\left(\frac{1}{3}, \frac{1}{3}\right)$$

von allen Polynomen

$$p \in \{q : D \rightarrow \mathbb{R} : q(x, y) = \alpha_{0,0} + \alpha_{1,0}x + \alpha_{0,1}y + \alpha_{2,0}x^2 + \alpha_{1,1}xy + \alpha_{0,2}y^2 \text{ mit } \alpha_{i,j} \in \mathbb{R}\}$$

erfüllt wird. Dabei ist $|D|$ der Flächeninhalt von D . Wie lauten die Gewichte ω_i ?

(ii) Seien $a, b, c \in \mathbb{R}^2$ die Eckpunkte eines Dreiecks T . Zeigen Sie, dass Gewichte $\omega_1, \dots, \omega_4$ existieren, sodass die Quadraturformel

$$\frac{1}{|T|} \int_T p(x, y) d(x, y) = \omega_1 p(a) + \omega_2 p(b) + \omega_3 p(c) + \omega_4 p\left(\frac{a+b+c}{3}\right)$$

von allen Polynomen

$$p \in \{q : T \rightarrow \mathbb{R} : q(x, y) = \alpha_{0,0} + \alpha_{1,0}x + \alpha_{0,1}y + \alpha_{2,0}x^2 + \alpha_{1,1}xy + \alpha_{0,2}y^2 \text{ mit } \alpha_{i,j} \in \mathbb{R}\}$$

erfüllt wird. Dabei ist $|T|$ der Flächeninhalt von T . Wie lauten die Gewichte ω_i ?

Hinweis: Verwenden Sie das Ergebnis von 22 (i), indem Sie T affin auf D abbilden.

Lösung zu Aufgabe 38

Literaturverzeichnis

- [AS] M. ABRAMOWITZ, I.A. STEGUN Pocketbook of Mathematical Functions with Formulas, Verlag Harri Deutsch, Frankfurt/Main (1984).
- [A] R.A. ADAMS, "Sobolev Spaces", Pure Appl. Math. 65, Academic Press, New York, 1975.
- [Calvetti] D. CALVETTI, G.H. GOLUB, W.B. GRAGG UND L. REICHEL, Computation of Gauss-Kronrod rules. Math. Comp. 69, 1035–1052 (2000).
- [Cull] P. CULL, M. FLAHIVE UND R. ROBSON, Difference equations, Undergraduate Texts in Mathematics. Springer, New York (2005).
- [Cuyt/Wuytack] A. CUYT, L. WUYTACK, Nonlinear Methods in Numerical Analysis, North-Holland (Amsterdam).
- [Elayadi] S. ELAYADI An introduction to difference equations, Undergraduate Texts in Mathematics. Springer, New York (2005).
- [Golub] G. H. GOLUB, C. F. VAN LOAN, Matrix Computations, 3. ed., Hopkins Univ. Press, 1996.
- [H] W. HACKBUSCH, Iterative Lösung großer schwachbesetzter Gleichungssysteme, 2. Auflage, Teubner-Verlag, Stuttgart, 1993.
- [Hämmerlin/Hoffmann] G. HÄMMERLIN, K.-H. HOFFMANN, Numerische Mathematik, 4. Auflage, Springer-Verlag, Berlin u.a., 1994.
- [Heuser] H. Heuser: Funktionalanalysis. Teubner, 3. Auflage, 1992.
- [Kiefer] J. KIEFER, Optimum sequential search and approximation methods under minimum regularity assumptions. J. Soc. Ind. Appl. Math. 5, 105-136 (1957).
- [Kronrod] A.S. KRONROD, Nodes and weights of quadrature formulas. Sixteen-place tables. New York: Consultants Bureau. Authorized translation from the Russian (1965).
- [Laurie] D.P. LAURIE, Calculation of Gauss-Kronrod quadratur rules. Math. Comp. 1133-1145 (66) 1997.
- [Lebed] G. K. LEBED, Quadrature formulas with minimum error for certain classes of functions, Mathematical Notes 3, 368-373 (1968).
- [Marsden] M. J. MARSDEN, An identity for spline functions with applications to variation-diminishing spline approximation. J. Approx. Theory 3 (1970), 7-49.
- [Meyberg/Vachenaue] K. MEYBERG, P. VACHENAUER, Höhere Mathematik 1. Springer, Berlin (1999)
- [Plato] R. PLATO, Numerische Mathematik kompakt, Vieweg-Verlag.
- [Rivlin] T.J. RIVLIN, An Introduction to the Approximation of Functions, Blaisdell Publ., Waltham, MA, 1969.
- [Schönhage] A. SCHÖNHAGE, Approximationstheorie, de Gruyter, Berlin, 1971.

- [Schwarz] H. R. SCHWARZ, Numerische Mathematik, 4. Auflage, Teubner-Verlag, Stuttgart, 1997.
- [Stör/Bulirsch] J. STÖR, R. BULIRSCH, Numerische Mathematik 1 und 2, Springer-Verlag, Berlin u.a., 1994.
- [SW] K. STREHMEL, R. WEINER, Numerik gewöhnlicher Differentialgleichungen, Teubner-Verlag, Stuttgart, 1995.
- [St] A. H STROUD, Gaussian quadrature formulas, Prentice-Hall, Englewood Cliff (1966).
- [Sz] G. SZEGÖ, Orthogonal Polynomials, AMS 3. Auflage, 1967.
- [TS] W. TÖRNIG, P. SPELLUCCI, Numerische Mathematik für Ingenieure und Physiker, Band 1 & 2, Springer-Verlag, Berlin u.a.
- [W] W. WALTER, Gewöhnliche Differentialgleichungen, 6. Auflage, Springer-Verlag, Berlin u.a., 1996.