Freie Universität Berlin

FB Mathematik und Informatik

Numerische Mathematik/Scientific Computing

Einführung in die Numerische Mathematik (Numerik I)

Ralf Kornhuber und Christof Schütte

unter Verwendung von Vorlesungen von
F. Bornemann (TU München) und H. Yserentant (Uni Tübingen)

1. Auflage: Sommersemester 2001
– korrigierte Fassung vom Mai 2019 –

Typeset und Layout: Dorothé Auth und Sabrina Nordt
Technisch überarbeitet: Stefan Vater

Inhaltsverzeichnis

1 Nichtlineare Gleichungssysteme

1.1 Fixpunktiteration

Wir kennen bereits Verfahren, die es erlauben, die Lösung *linearer* Gleichungssysteme in endlich vielen Schritten (bis auf Rundungsfehler) *exakt* auszurechnen (vgl. z.B. Skript CoMa I). Im Falle *nichtlinearer Gleichungssysteme* gelingt dies im allgemeinen nicht mehr. Wir müssen uns mit *iterativen Verfahren* zufriedengeben, mit denen man eine Lösung in endlich vielen Schritten nur *bis auf eine vorgegebene Genauigkeit* berechnen kann. Wir beginnen mit einem einfachen Beispiel.

Beispiel:
Gesucht ist eine Nullstelle $x^* \in \mathbb{R}$ von

$$F(x) = x(x-1).$$

Offenbar sind $x_1^* = 0$ und $x_2^* = 1$ Lösungen (Achtung: keine Eindeutigkeit!). Wir wollen nun ein Iterationsverfahren zur näherungsweisen Berechnung einer Nullstelle x^* konstruieren. Dazu betrachten wir anstelle der ursprünglichen nichtlinearen Gleichung $F(x) = 0$ das äquivalente Problem

$$x^* \in \mathbb{R}: \quad \phi(x^*) = x^* \tag{1.1}$$

wobei

$$\phi(x) = F(x) + x = x^2$$

gesetzt ist. Jede Lösung von (1.1) heißt *Fixpunkt* von ϕ. Wir wollen nun eine Folge $\{x_k\}$ von Näherungslösungen konstruieren, die gegen einen Fixpunkt x^* von ϕ konvergiert. Wir versuchen es einfach mit der *Fixpunktiteration*

$$x_{k+1} = \phi(x_k), \qquad x_0 \in \mathbb{R} \text{ geeignet.} \qquad \triangleleft$$

Für $x_0 = 0.4$ erhält man die Folge $x_1 = 0.16$, $x_2 = 0.0256$, $x_3 = 0.0007$, ..., welche offenbar gegen x_1^* konvergiert. Die Wahl $x_0 = 1.1$ liefert hingegen die Folge $x_1 = 1.21$, $x_2 = 1.4641$, $x_3 = 2.1436$, ..., welche offenbar divergiert. Die Vorgehensweise ist in Abbildung 1.1 illustriert. Je nach Startwert erhält man also Konvergenz oder Divergenz. Wir wollen verstehen, woran das liegt.

Satz 1.1 *Sei $I = [a, b]$ und $\phi : I \to \mathbb{R}$ eine Abbildung mit den beiden folgenden Eigenschaften.*

$$\phi(x) \in I \qquad \forall x \in I. \tag{1.2}$$

$$|\phi(x) - \phi(y)| \leq q|x - y| \qquad \forall x, y \in I, \quad q \in [0, 1). \tag{1.3}$$

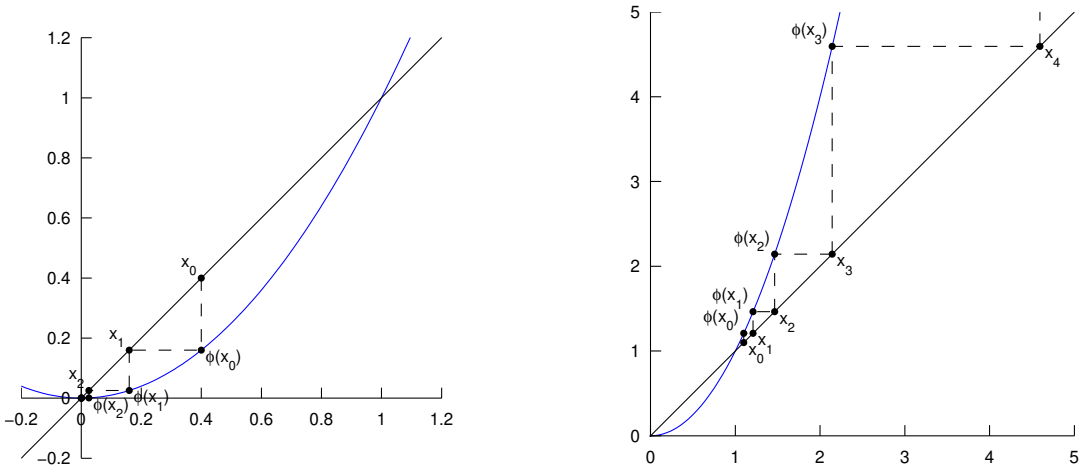


Abbildung 1.1: Funktionsweise der Fixpunktiteration am Beispiel $x_{k+1} = x_k^2$: Konvergenz (links) und Divergenz (rechts)

Dann besitzt ϕ genau einen Fixpunkt $x^* \in I$ und die Folge

$$x_{k+1} = \phi(x_k)$$

konvergiert für jeden Startwert $x_0 \in I$ gegen x^*. Die Fehlerreduktion *erfolgt gemäß*

$$|x^* - x_{k+1}| \leq q|x^* - x_k|.$$

Weiter gelten die a priori *Fehlerabschätzung*

$$|x^* - x_k| \leq \frac{q^k}{1-q}|x_1 - x_0|$$

und die a posteriori *Fehlerabschätzung*

$$|x^* - x_{k+1}| \leq \frac{q}{1-q}|x_{k+1} - x_k|.$$

Beweis:

a) Durchführbarkeit.
Wegen (1.2) ist $x_k \in I \ \forall k \in \mathbb{N}$, falls $x_0 \in I$. Damit ist die Fixpunktiteration und auch die resultierende Folge $\{x_k\} \subset I$ wohldefiniert.

b) Konvergenz.
Als erstes zeigen wir, daß $\{x_k\}$ eine Cauchy-Folge ist. Wegen (1.3) haben wir

$$|x_{k+1} - x_k| = |\phi(x_k) - \phi(x_{k-1})| \leq q|x_k - x_{k-1}| \leq q^k|x_1 - x_0|$$

und ebenso

$$|x_{k+i+1} - x_{k+i}| \leq q|x_{k+1+(i-1)} - x_{k+(i-1)}| \leq q^i|x_{k+1} - x_k|.$$

Es gilt daher für $k, j \geq 0$

$$|x_{k+j} - x_k| \leq \sum_{i=0}^{j-1} |x_{k+i+1} - x_{k+i}| \leq |x_{k+1} - x_k| \sum_{i=0}^{j-1} q^i \leq \frac{q^k}{1-q}|x_1 - x_0|.$$

Es gibt also zu jedem $\varepsilon > 0$ ein $k_0 \in \mathbb{N}$, so daß $|x_{k+j} - x_k| \leq \varepsilon \ \forall j \geq 0, k > k_0$. Damit ist $\{x_k\}$ Cauchy-Folge. Da $\mathbb{R}$ vollständig ist, gibt es ein $x^* \in \mathbb{R}$ mit

$$\lim_{k\to\infty} x_k = x^*.$$

Da $I \subset \mathbb{R}$ abgeschlossen ist, gilt $x^* \in I$.

c) x^* ist Lösung.
Aus (1.3) folgt insbesondere die Stetigkeit von ϕ. Daher gilt $\phi(x_k) \to \phi(x^*)$. Andererseits haben wir $\phi(x_k) = x_{k+1} \to x^*$. Aus der Eindeutigkeit des Grenzwerts folgt $\phi(x^*) = x^*$.

d) Eindeutigkeit des Fixpunkts.
Seien x^*, y^* zwei verschiedene Fixpunkte von ϕ. Dann folgt aus (1.3)

$$|x^* - y^*| = |\phi(x^*) - \phi(y^*)| \leq q|x^* - y^*|$$

offenbar $q \geq 1$. Widerspruch.

e) Fehlerreduktion.

$$|x^* - x_{k+1}| = |\phi(x^*) - \phi(x_k)| \leq q|x^* - x_k|.$$

f) a posteriori Fehlerabschätzung.
Die Abschätzung folgt aus

$$|x^* - x_{k+1}| \leq |\phi(x^*) - \phi(x_{k+1})| + |\phi(x_{k+1}) - \phi(x_k)| \leq q|x^* - x_{k+1}| + q|x_{k+1} - x_k|.$$

g) a priori Fehlerabschätzung.

$$|x^* - x_k| \leq \frac{q}{1-q}|x_k - x_{k-1}| \leq \frac{q}{1-q}q^{k-1}|x_1 - x_0|. \qquad \square$$

Bemerkung:
Ist $\phi \in C^1(I)$ und $q = \sup_{z\in I} |\phi'(z)|$, so gilt nach dem Mittelwertsatz mit einem $\xi \in I$

$$|\phi(x) - \phi(y)| = |\phi'(\xi)||x - y| \leq q|x - y| \qquad \forall x, y \in I. \qquad \triangleleft$$

Beispiel:
Wir kommen auf unser Eingangsbeispiel zurück. Dazu betrachten wir $I = [-r, r]$ mit $r < \frac{1}{2}$ und $\phi(x) = x^2$. Dann folgt

$$|x^2 - y^2| = |x + y||x - y| \leq 2r|x - y| \qquad \forall x, y \in I,$$

also gilt (1.3) mit $q = 2r < 1$. Außerdem haben wir

$$|\phi(x)| \leq |x| \leq r \qquad \forall x \in I,$$

also (1.2). Das erklärt die Konvergenz der Fixpunktiteration für $x_0 = 0.4$ gegen $x_1^* = 0$. Übrigens liegt für alle $x_0 < 1$ Konvergenz der Fixpunktiteration gegen $x_1^* = 0$ vor: Die Voraussetzungen von Satz 1.1 sind dafür *hinreichend*, aber *nicht notwendig*. ◁

Manche mögen es gleich gemerkt haben: Satz 1.1 ist ein Spezialfall des *Banachschen Fixpunktsatzes:*

Satz 1.2 *Sei B ein Banachraum mit Norm $\|\cdot\|$, $U \subset B$ abgeschlossen (und nichtleer) und $\phi : U \to B$ eine Abbildung mit den beiden folgenden Eigenschaften.*

$$\phi(x) \in U \qquad \forall x \in U. \tag{1.4}$$

$$\|\phi(x) - \phi(y)\| \leq q\|x - y\| \qquad \forall x, y \in U, \quad q \in [0, 1). \tag{1.5}$$

Dann besitzt ϕ genau einen Fixpunkt $x^* \in U$ und die Folge

$$x_{k+1} = \phi(x_k)$$

konvergiert für jeden Startwert $x_0 \in U$ gegen x^*. Die Fehlerreduktion *erfolgt gemäß*

$$\|x^* - x_{k+1}\| \leq q\|x^* - x_k\|.$$

Weiter gelten die a priori *Fehlerabschätzung*

$$\|x^* - x_k\| \leq \frac{q^k}{1 - q}\|x_1 - x_0\|$$

und die a posteriori *Fehlerabschätzung*

$$\|x^* - x_{k+1}\| \leq \frac{q}{1 - q}\|x_{k+1} - x_k\|.$$

Der Beweis von Satz 1.2 ist wortwörtlich der gleiche wie der Beweis von Satz 1.1 (Vertrauen ist gut, Kontrolle ist besser). Die Eigenschaft (1.5) gibt Anlass zu folgenden Definitionen.

Definition 1.3 *Sei B ein Banachraum. Eine Abbildung $\phi : U \subset B \to B$ heißt Lipschitz-stetig auf U mit Lipschitz-Konstante L, falls gilt*

$$\|\phi(x) - \phi(y)\| \leq L\|x - y\| \quad \forall x, y \in U. \tag{1.6}$$

ϕ heißt kontrahierend auf U, falls $L \in [0, 1)$.

Definition 1.4 *Sei B ein Banachraum. Eine Folge $\{x_k\} \subset B$ konvergiert linear mit Konvergenzrate q gegen x^*, falls gilt*

$$\|x^* - x_{k+1}\| \leq q\|x^* - x_k\| \quad \forall k \geq 0.$$

Als erste Anwendung von Satz 1.2 betrachten wir die *iterative Lösung linearer Gleichungssysteme*. Überraschenderweise sind iterative Verfahren den sogenannten *direkten* Methoden (z.B. Gaußscher Algorithmus) in wichtigen Fällen tatsächlich überlegen. (Stichwort: diskretisierte partielle Differentialgleichungen).
Vorgelegt sei also das lineare Gleichungssystem

$$Ax = b, \qquad A \in \mathbb{R}^{n,n},\ b \in \mathbb{R}^n. \tag{1.7}$$

Als erstes haben wir unser Gleichungssystem $F(x) = b - Ax = 0$ auf Fixpunktgestalt zu bringen. Die Wahl

$$\phi(x) = b - Ax + x$$

funktioniert meistens nicht (keine Kontraktion). Wir geben daher einen allgemeineren Zugang an. Dazu wählen wir $M \in \mathbb{R}^{n,n}$, regulär, mit der Eigenschaft

$$My = r \qquad \text{ist für alle } r \in \mathbb{R}^n \text{ mit } \mathcal{O}(n) \text{ Punktoperationen lösbar.} \tag{1.8}$$

Durch Addition von Mx erhält man die Fixpunktgestalt

$$Mx = F(x) + Mx = (M - A)x + b$$

und die zugehörige Fixpunktiteration

$$Mx_{k+1} = (M - A)x_k + b, \qquad x_0 \in \mathbb{R}^n. \tag{1.9}$$

In jedem Iterationsschritt hat man also wieder ein lineares Gleichungssystem zu lösen. Nach Voraussetzung ist das aber mit optimalem Aufwand (Ordnung $\mathcal{O}(n)$) möglich.
Mit Hilfe von Satz 1.2 wollen wir nun eine Bedingung an M herleiten, welche die Konvergenz des iterativen Verfahrens (1.9) garantieren. Die Fixpunktiteration (1.9) ist äquivalent zu

$$x_{k+1} = \phi(x_k), \qquad \phi(x) = (I - M^{-1}A)x + M^{-1}b, \tag{1.10}$$

wobei $I \in \mathbb{R}^{n,n}$ die Einheitsmatrix bezeichnet. Mit $\|\cdot\|$ bezeichnen wir sowohl eine *Vektornorm* auf $\mathbb{R}^n$ als auch die *zugehörige Matrixnorm* auf $\mathbb{R}^{n,n}$ (vgl. z.B. Skript CoMa I). Wir fordern nun zusätzlich zu (1.8), daß M auch der Bedingung

$$\|I - M^{-1}A\| = q < 1 \tag{1.11}$$

genügt. Dann sind die Voraussetzungen des Banachschen Fixpunktsatzes 1.2 mit $B = U = \mathbb{R}^n$ und ϕ aus (1.10) erfüllt. Die Lösung x^* des linearen Gleichungssystems (1.7) ist also eindeutig bestimmt (insbesondere ist A regulär!) und die Folge x_k konvergiert für jeden Startwert $x_0 \in \mathbb{R}^n$ mit der Konvergenzrate q gegen x^*.
Jede Vorschrift zur Wahl von M charakterisiert ein iteratives Verfahren für lineare Gleichungssysteme. Die einfachste Wahl $M = I$ führt auf das sogenannte *Richardson-Verfahren*. Die Wahl

$$M = \operatorname{diag}(A)$$

liefert das *Jacobi-Verfahren* (nach einer Arbeit von Carl Gustav Jacobi (1845)). Wie wir wissen (siehe z.B. Skript CoMa I), kann man gestaffelte Gleichungssysteme mit optimalem Aufwand (d.h. $n^2/2$ Punktop.) lösen. Das legt die Wahl

$$M_{ij} = \begin{cases} A_{ij}, & \text{falls } i \geq j \\ 0, & \text{sonst} \end{cases}$$

nahe. Man erhält das *Gauß-Seidel-Verfahren* (nach Carl Friedrich Gauß (1819–1823) und Phillip Ludwig Seidel (1874), übrigens ein Student von Jacobi).
Offenbar ist (1.8) in den obigen drei Fällen erfüllt. Ob auch (1.11) gilt, hängt jeweils von der Matrix A ab! Wir verweisen beispielsweise auf Deuflhard und Hohmann [1, Abschnitt 8.1] oder Stoer und Bulirsch [5, Kapitel 8]. Erheblich trickreichere iterative Verfahren, die sogenannten Mehrgittermethoden, werden später im Zusammenhang mit elliptischen partiellen Differentialgleichungen eine Rolle spielen.

Bislang haben wir nur lineare Konvergenz kennengelernt. Um höhere Konvergenzgeschwindigkeiten zu quantifizieren, führen wir jetzt die Begriffe *superlineare Konvergenz* und *Konvergenzordnung* ein.

Definition 1.5 *Sei B ein Banachraum und $\{x_k\} \subset B$. Die Folge $\{x_k\}$ heißt superlinear konvergent gegen x^*, falls es eine Folge $q_k \geq 0$ mit $\lim_{k\to\infty} q_k = 0$ gibt, so daß gilt*

$$\|x_{k+1} - x^*\| \leq q_k \|x_k - x^*\|.$$

Sei $\{x_k\}$ konvergent gegen x^*. Dann heißt $\{x_k\}$ konvergent mit der Ordnung $p \geq 2$, falls es eine von k unabhängige Konstante $C \geq 0$ gibt, so daß gilt

$$\|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^p. \tag{1.12}$$

Im Falle $p = 2$ sprechen wir von quadratischer Konvergenz.

Konvergenz mit Ordnung $p = 1$ ist dasselbe wie lineare Konvergenz (vgl. Definition 1.4). Die durch $x_{k+1} = x_k^2$ erzeugte Folge aus unserem Eingangsbeispiel konvergiert übrigens für $x_0 < 1$ quadratisch gegen $x_1^* = 0$. Das ist ein glücklicher Zufall. Nach Satz 1.1 bzw. Satz 1.2 konvergiert eine Fixpunktiteration nur linear. Ein berühmtes Verfahren, das unter gewissen Voraussetzungen quadratische Konvergenz liefert, diskutieren wir im nächsten Abschnitt.

1.2 Newton-Verfahren

Wir betrachten zunächst die skalare Gleichung

$$x^* \in \mathbb{R}: \qquad F(x^*) = 0 \tag{1.13}$$

mit einer stetig differenzierbaren Funktion $F : \mathbb{R} \to \mathbb{R}$.
Um eine Folge von Näherungslösungen zu berechnen, approximieren wir (1.13) durch eine Folge „einfacherer“ Probleme gleicher Bauart (vgl. iterative Verfahren für lineare Systeme im vorigen Abschnitt). Dazu ersetzen wir F durch eine „einfachere“ Funktion und berechnen deren Nullstelle. Ist $x_0 \in \mathbb{R}$ gegeben, so ist bekanntlich die *Tangente*

$$p(x) = F(x_0) + F'(x_0)(x - x_0)$$

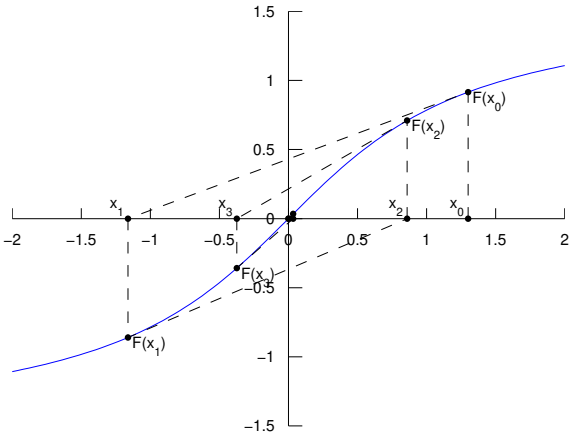


Abbildung 1.2: Funktionsweise des Newton-Verfahrens am Beispiel der Funktion $F(x) = \arctan(x)$.

eine gute Approximation von F, zumindest in einer genügend kleinen Umgebung von x_0. Unter der Voraussetzung $F'(x_0) \neq 0$ errechnet man die Nullstelle der Tangente p zu

$$x_1 = x_0 - \frac{F(x_0)}{F'(x_0)}.$$

Sukzessive Anwendung dieser Vorschrift liefert das *Newton-Verfahren*

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)}, \qquad x_0 \in \mathbb{R} \text{ geeignet.} \tag{1.14}$$

Beispiel:
Wir wollen das Newton-Verfahren auf die Funktion

$$F(x) = \arctan(x)$$

anwenden. Die ersten drei Iterationsschritte zum Startwert $x_0 = 1.3$ sind in Abbildung 1.2 veranschaulicht.
Zahlenwerte für $x_0 = 1.3$ und $x_0 = 1.4$ finden sich in der folgenden Tabelle.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1.3	-1.1616	0.8589	-0.3742	0.0342	-2.6240e-05	1.2045e-14	0
1.4	-1.4136	1.4501	-1.5506	1.8471	-2.8936	8.7103	-103.2498

Offenbar konvergieren für $x_0 = 1.3$ die Iterierten mit wachsender Geschwindigkeit gegen die Lösung $x^* = 0$: Ab $k = 3$ verdoppelt sich die Anzahl der gültigen Stellen in jedem Schritt. Das bedeutet *quadratische Konvergenz*! Für $x_0 = 1.4$ sieht die Sache anders aus: Die Iterierten divergieren. ◁

Bemerkung:
Das Newton-Verfahren ist eine Fixpunktiteration $x_{k+1} = \phi(x_k)$ für

$$\phi(x) = x - \frac{F(x)}{F'(x)}.$$

Wir können also Satz 1.1 anwenden und erhalten folgendes Konvergenzkriterium. Unter den Voraussetzungen $F \in C^2(\mathbb{R})$ und

$$\sup_{z\in\mathbb{R}} |\phi'(z)| = \sup_{z\in\mathbb{R}} \left| F(z)\frac{F''(z)}{F'(z)^2} \right| = q < 1$$

konvergiert daher das Newton-Verfahren für jeden Startwert $x_0 \in \mathbb{R}$. ◁

Leider erklärt das obige Resultat nicht die lokal wachsende Konvergenzgeschwindigkeit. Bevor wir dazu kommen, wollen wir das Newton-Verfahren auf ein System

$$x^* \in D: \qquad F(x) = 0 \quad , \quad F : D \subset \mathbb{R}^n \to \mathbb{R}^n \tag{1.15}$$

von nichtlinearen Gleichungen erweitern. Die Ableitung (Jacobi-Matrix)

$$F'(x_0) = \begin{pmatrix} \frac{\partial}{\partial x_1}F_1(x_0) & \cdots & \frac{\partial}{\partial x_n}F_1(x_0) \\ \vdots & & \vdots \\ \frac{\partial}{\partial x_1}F_n(x_0) & \cdots & \frac{\partial}{\partial x_n}F_n(x_0) \end{pmatrix} \in \mathbb{R}^{n,n}$$

von F an der Stelle x_0 ist bekanntlich charakterisiert durch

$$\|F(x) - (F(x_0) + F'(x_0)(x - x_0))\| = \mathcal{O}\big(\|x - x_0\|\big) \quad \text{für} \quad x \to x_0.$$

Damit ist die Tangentialebene

$$p(x) = F(x_0) + F'(x_0)(x - x_0)$$

wieder eine gute Approximation von $F(x)$, zumindest in einer genügend kleinen Umgebung von x_0. Unter der Voraussetzung, daß $F'(x_k)$ jeweils regulär ist, hat die Iterationsvorschrift des Newton-Verfahrens für nichtlineare Systeme wieder die Gestalt

$$x_{k+1} = x_k - F'(x_k)^{-1}F(x_k), \qquad x_0 \in \mathbb{R}^n \text{ geeignet.}$$

Um die Berechnung von $F'(x_k)^{-1}$ zu vermeiden, verwendet man die äquivalente Formulierung

$$x_{k+1} = x_k + \Delta x_k, \quad F'(x_k)\Delta x_k = -F(x_k), \qquad x_0 \in \mathbb{R}^n \text{ geeignet.}$$

In jedem Iterationsschritt hat man also anstelle des ursprünglichen, nichtlinearen Problems ein lineares Gleichungssystem mit der Koeffizientenmatrix $F'(x_k)$ zu lösen.

Beispiel:
Vorgelegt sei das Gleichungssystem

$$\begin{aligned} \sin(\xi) - \eta &= 0 \\ \xi - \cos(\eta) &= 0 \end{aligned}$$

also

$$\begin{aligned} F_1(\xi,\eta) &= \sin(\xi) - \eta \\ F_2(\xi,\eta) &= \xi - \cos(\eta) \end{aligned} .$$

Man erhält

$$F'(x) = \begin{pmatrix} \cos(\xi) & -1 \\ 1 & \sin(\eta) \end{pmatrix}, \qquad x = \begin{pmatrix} \xi \\ \eta \end{pmatrix}.$$

Achtung: $F'(x)$ kann singulär sein, z.B. für $x = (\pi, \frac{1}{2}\pi)$! Die Iterierten zu $x_0 = (3,3)^T$ finden sich in der folgenden Tabelle.

k	ξ_k	η_k
1	-1.16898713819790	4.26838599329955
2	-7.26977629911835	-3.30627645454922
3	-1.87916601032632	2.13896869859183
4	3.42480564811751	-2.56261488791645
5	1.44984477398723	1.61684138641047
6	0.67129464906329	0.89875685831196
7	0.77538096829107	0.70350160297372
8	0.76818082842510	0.69484618466670
9	0.76816915690064	0.69481969089595
10	0.76816915673680	0.69481969073079
11	0.76816915673680	0.69481969073079

Ab Schritt $k = 7$ beobachten wir wieder eine Verdopplung der gültigen Stellen in jedem Iterationsschritt, also *quadratische Konvergenz*. ◁

Bemerkung:
Offenbar sind (1.15) und

$$x^* \in D: \quad AF(x^*) = 0 \tag{1.16}$$

für jede reguläre Skalierungsmatrix $A \in \mathbb{R}^{n,n}$ äquivalent. Man spricht von *Affin-Invarianz*. Diese *Struktureigenschaft* wird durch das Newton-Verfahren erhalten! Anwendung auf (1.16) liefert nämlich

$$\big((AF(x_k))'\big)^{-1} AF(x_k) = F'(x_k)^{-1} F(x_k) = -\Delta x_k.$$

Konsequenterweise sollten also auch alle Konvergenzaussagen affin-invariant formuliert werden. ◁

Dies berücksichtigen wir gleich bei der Formulierung der Voraussetzungen unseres Konvergenzsatzes.

Satz 1.6 *Sei $D \subset \mathbb{R}^n$ offen und $F : D \to \mathbb{R}^n$. Es existiere eine Nullstelle $x^* \in D$ von F. Für alle $x \in D$ sei F differenzierbar. Die Jacobi-Matrix $F'(x)$ sei invertierbar für alle $x \in D$. Für alle $x \in D$ und $v \in \mathbb{R}^n$ mit $x + sv \in D$ für alle $s \in [0,1]$ gelte die (affin-invariante) Lipschitz-Bedingung*

$$\|F'(x)^{-1}\big(F'(x+sv) - F'(x)\big)v\| \le s\omega\|v\|^2 \tag{1.17}$$

mit Lipschitz-Konstante $\omega \ge 0$.
Es sei schließlich $x_0 \in B_\rho(x^*) = \{x \in \mathbb{R}^n \mid \|x - x^*\| < \rho\}$, wobei $\rho > 0$ so gewählt ist, daß die Bedingungen

$$\rho < \tfrac{2}{\omega} \quad \textit{und} \quad B_\rho(x^*) \subseteq D$$

erfüllt sind.
Dann ist x^* die einzige Nullstelle von F in $B_\rho(x^*)$. Die Folge der Newton-Iterierten $\{x_k\}$ liegt in $B_\rho(x^*)$ und konvergiert quadratisch gegen x^*. Insbesondere gilt $\lim_{k\to\infty} x_k = x^*$ und

$$\|x_{k+1} - x^*\| \le \frac{\omega}{2}\|x_k - x^*\|^2, \qquad k = 0, 1, \dots .$$

Beweis:
Die Fixpunktabbildung

$$\phi(x) = x - F'(x)^{-1}F(x)$$

ist wohldefiniert für alle $x \in D$. Sei $x \in B_\rho(x^*) \subseteq D$ und $s \in [0,1]$. Dann gilt $x + s(x^* - x) \in B_\rho(x^*) \subseteq D \ \forall s \in [0,1]$ und

$$\frac{d}{ds}F(x + s(x^* - x)) = F'(x + s(x^* - x))(x^* - x).$$

Aus dem Hauptsatz der Differential- und Integralrechnung folgt daher

$$0 = F(x^*) = F(x) + \int_0^1 F'(x + s(x^* - x))(x^* - x)\, ds.$$

Daraus ergibt sich

$$\begin{aligned} x^* - \phi(x) &= x^* - x + F'(x)^{-1}F(x) \\ &= F'(x)^{-1}\big(F(x) + F'(x)(x^* - x)\big) \\ &= F'(x)^{-1}\left(-\int_0^1 F'(x + s(x^* - x))(x^* - x)\, ds + F'(x)(x^* - x)\right) \\ &= -\int_0^1 F'(x)^{-1}\big(F'(x + s(x^* - x)) - F'(x)\big)(x^* - x)\, ds. \end{aligned}$$

Aus der Lipschitz-Bedingung und $\rho < \frac{2}{\omega}$ erhalten wir

$$\begin{aligned} \|x^* - \phi(x)\| &\le \int_0^1 \|F'(x)^{-1}\big(F'(x + s(x^* - x)) - F'(x)\big)(x^* - x)\|\, ds \\ &\le \int_0^1 s\omega\|x^* - x\|^2\, ds = \tfrac{1}{2}\omega\|x^* - x\|^2 < q\|x^* - x\| \qquad q := \tfrac{1}{2}\omega\rho < 1. \end{aligned} \tag{1.18}$$

Wegen $x_0 \in B_\rho(x^*)$ folgt daraus induktiv $x_k \subset B_\rho(x^*)$ und

$$\|x^* - x_k\| \le q^k\|x^* - x_0\| \to 0.$$

Auch die quadratische Konvergenz von $\{x_k\}$ folgt direkt aus (1.18).
Im Widerspruch zur Behauptung nehmen wir an, daß $\tilde{x}^* \in B_\rho(x^*)$ eine weitere Nullstelle von F ist. Einsetzen von $x = \tilde{x}^*$ in (1.18) führt dann auf $q > 1$. Widerspruch. □

Bemerkung:
Da die quadratische Konvergenz „gute“ Startwerte $x_0 \in B_\rho(x^*)$ voraussetzt, spricht man von *lokal* quadratischer Konvergenz des Newton-Verfahrens. Wir haben schon am Eingangsbeispiel gesehen, daß für „schlechte“ Startwerte überhaupt keine Konvergenz vorzuliegen braucht. ◁

Bemerkung:
Varianten des obigen Satzes liefern auch die Existenz der oben angenommenen Lösung x^*.◁

In der Praxis ist es oft nicht möglich, Startwerte zu finden, die auch nur die Konvergenz des Newton-Verfahrens gewährleisten. Um den Konvergenzbereich zu vergrößern, betrachten wir das *gedämpfte Newton-Verfahren*

$$x_{k+1} = x_k + \lambda_k \Delta x_k, \qquad F'(x_k)\Delta x_k = -F(x_k) \tag{1.19}$$

mit einem geeigneten *Dämpfungsparameter* $\lambda_k \in (0, 1]$. Die Wahl von λ_k sollte idealerweise so erfolgen, daß für alle $x_0 \in D$ Konvergenz vorliegt und sich darüberhinaus im Falle $x_k \in B_\rho(x^*)$ automatisch $\lambda_k = 1$ ergibt. So bliebe die lokal quadratische Konvergenz erhalten.
Wir beschreiben eine *affin-invariante Dämpfungsstrategie*, die auf dem sogenannten *natürlichen Monotonietest* beruht.

Algorithmus 1.7 (Dämpfungsstrategie)
gegeben: $\Delta x_k = -F'(x_k)^{-1}F(x_k)$

1. *setze:* $\lambda_k = 1$
2. *berechne* $\bar{\Delta} x_k = -F'(x_k)^{-1}F(x_k + \lambda_k \Delta x_k)$
3. *natürlicher Monotonietest:*
   falls $\|\bar{\Delta} x_k\| \leq (1 - \frac{\lambda_k}{2})\|\Delta x_k\|$, *akzeptiere* λ_k.
   andernfalls setze $\lambda_k = \lambda_k/2$ *und gehe zu Schritt 2.*

Beachte, daß in Schritt 2 jeweils ein lineares Gleichungssystem mit der Koeffizientenmatrix $F'(x_k)$ gelöst werden muß. Hat man Δx_k über eine LR-Zerlegung von $F'(x_k)$ berechnet, so ist dies mit optimalem Aufwand ($\mathcal{O}(n)$) möglich.

Beispiel:
Wir betrachten wieder unser skalares Eingangsbeispiel $F(x) = \arctan(x)$. Die Iterierten des gedämpften Newton-Verfahrens für verschiedene Startwerte zeigt die folgende Tabelle.

k	x_k	λ_k	x_k	λ_k	x_k	λ_k	x_k	λ_k
0	1.4000	0.5	5.0000	0.125	10.000	0.0625	100.00	0.0078
1	-0.0068	1.	0.5364	1.	0.7135	1.	-21.949	0.0312
2	2.1048e-07	1.	-0.0976	1.	-0.2217	1.	1.0620	0.5
3	-6.2469e-21	1.	6.1913e-04	1.	0.0072	1.	0.1944	1.
4	0	1.	-1.5821e-10	1.	-2.4854e-07	1.	-0.0049	1.
5			0	1.	1.0217e-20	1.	7.6666e-08	1.
6					0	1.	-2.9117e-22	1.
7							0	1.

Offenbar wird der Konvergenzbereich erheblich erweitert! Man sieht, daß die mit x_0 wachsenden Inversen $|F'(x_0)^{-1}|$ durch immer kleinere Dämpfungsparameter kompensiert werden müssen. Andererseits wird in der Nähe der Lösung nicht mehr gedämpft und die lokal quadratische Konvergenz bleibt erhalten. ◁

Literatur

[1] P. Deuflhard and A. Hohmann. *Numerische Mathematik I.* de Gruyter, 4. Auflage, 2008. Wir kennen das Buch schon aus der CoMa. Unsere Darstellung folgt im wesentlichen den Abschnitten 4.1 und 4.2. Ein Vergleich lohnt auf alle Fälle. In Abschnitt 8.1 wird auf iterative Verfahren für symmetrische, positiv definite Gleichungssysteme eingegangen.

[2] C.T. Kelley. *Iterative Methods for Linear and Nonlinear Equations.* SIAM, 1995. Eine kompakte Darstellung, die aber trotzdem deutlich über den Stoffumfang allgemeiner Einführungen in die Numerische Mathematik hinausgeht. Zum Beispiel erfährt man, wie eine Sekantenmethode für nichtlineare Systeme aussieht oder worauf man achten muß, wenn man beim Newton–Verfahren die linearen Teilprobleme ihrerseits iterativ löst.

[3] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* SIAM, 2000. Ein Reprint des 1970 erstmals erschienen Standardwerks. Alle grundlegenden Techniken werden ausführlich vorgestellt. Was verständlicherweise fehlt, sind moderne, vom kontinuierlichen Problem her motivierte Lösungsansätze für diskretisierte Differentialgleichungen. Aber das ist eine andere Geschichte und die soll ohnehin ein andermal erzählt werden.

[4] J. Stoer. *Numerische Mathematik I.* Springer, 10. Auflage, 2007. Ein Standardwerk zur Numerischen Mathematik, das zusammen mit Band II vor ca. 30 Jahren Maßstäbe gesetzt hat. Die Konvergenz des Newton–Verfahrens wird in Abschnitt 5.3 analysiert. Beachte, daß die Voraussetzungen leicht abweichen (Affin–Invarianz?).

[5] J. Stoer and R. Bulirsch. *Numerische Mathematik II.* Springer, 5. Auflage, 2005. Mehr über iterative Verfahren für lineare Gleichungssysteme kann man in Kapitel 8 erfahren.

2 Bestapproximation und lineare Ausgleichsprobleme

Ein Beispiel für *Bestapproximation* ist die „bestmögliche“ Approximation einer gegebenen Funktion $f \in C[a,b]$ durch eine „einfache“ Funktion $u \in U \subset C^1[a,b]$. Genauer gesagt soll $u \in U$ so bestimmt werden, daß der *Abstand von u zu f unter allen Kandidaten aus U minimal wird*. Der Abstand von u zu f wird durch eine geeignete *Norm* gemessen. Die resultierende Bestapproximationsaufgabe sieht dann so aus:

$$u \in U: \quad \|f-u\| \le \|f-v\| \quad \forall v \in U. \tag{2.1}$$

Die Wahl von U und $\|\cdot\|$ haben wir noch frei. Im Falle der Maximums-Norm

$$\|v\|_\infty = \max_{x\in[a,b]} |v(x)|, \quad v \in C[a,b]$$

spricht man von *Tschebyscheff-Approximation*. Eine andere Wahl könnte

$$\|v\|_2 = \left(\int_a^b v(x)^2 \, dx \right)^{1/2}, \quad v \in C[a,b]$$

sein. Dann erhält man die sogenannte *L^2-Approximation*. Eine naheliegende Wahl für U ist

$$\mathcal{P}_n = \{v \in C[a,b] \mid v \text{ ist Polynom vom Grad } \le n\}.$$

$\mathcal{P}_n$ ist bekanntlich ein $n+1$-dimensionaler Unterraum von $C[a,b]$.

Beispiel:
Es sei $f(x) = x^2$, $[a,b] = [-1,1]$ und $n = 0$. Gesucht ist eine Lösung von (2.1) für $U = \mathcal{P}_0$. Die entsprechende Tschebyscheff-Approximation ist offenbar $p_\infty(x) \equiv \frac{1}{2}$. Zur Berechnung der L^2-Approximation $p_2 \in \mathbb{R}$ reicht es, wegen der strengen Monotonie der Wurzelfunktion, die Funktion

$$g(p) = \int_{-1}^{1} (x^2 - p)^2 \, dx = 2(\tfrac{1}{5} - \tfrac{2}{3}p + p^2), \qquad p \in \mathbb{R},$$

zu minimieren. Aus $g'(p_2) = 0$ folgt $p_2 = \frac{1}{3} \neq p_\infty$. ◁

Wir betrachten ein weiteres Beispiel für „einfache“ Funktionen: Zu einem vorgegebenen *Gitter*

$$a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$$

definieren wir die *stückweise linearen Funktionen*

$$\mathcal{S}_n = \{v \in C[a,b] \mid v|_{[x_{i-1},x_i]} \text{ ist linear } \forall i = 1,\ldots,n\}, \tag{2.2}$$

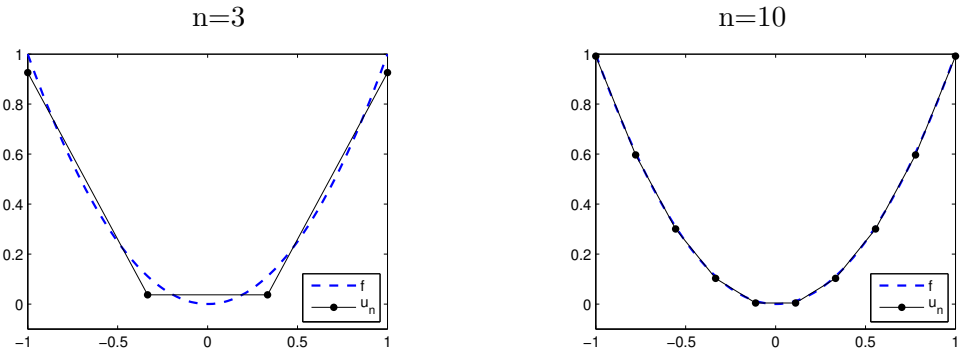


Abbildung 2.1: Bestapproximation der Funktion $f(x) = x^2$ mit linearen finiten Elementen auf verschiedenen Gittern

auch *lineare finite Elemente* genannt. Bei Wahl der Norm $\|\cdot\|_2$ erhalten wir die Bestapproximationsaufgabe

$$u_n \in \mathcal{S}_n : \qquad \|f - u_n\|_2 \leq \|f - v\|_2 \qquad \forall v \in \mathcal{S}_n.$$

Beispiel:
Wieder sei $f(x) = x^2$. Durch $x_i = -1 + 2ih$ und $h = \frac{1}{n}$ definieren wir ein Gitter. Abbildung 2.1 zeigt f im Vergleich zur Bestapproximation u_n für $n = 3$ und $n = 10$. Offenbar ist u_n nicht einfach die stückweise lineare Interpolation von $(x_i, f(x_i))$, $i = 1, \ldots, n$. ◁

Auf ein sogenanntes *Ausgleichsproblem* führt folgende Situation: Gegeben seien m Meßpunkte $(t_i, b_i) \in \mathbb{R}^2$, $i = 1, \ldots, m$ (Zustände b_i an den Stellen t_i). Für $t \neq t_i$ soll der Zustand durch eine geeignete Modellfunktion $\varphi(t; x_1, \ldots, x_n)$ beschrieben werden. Dabei sollen die $n \leq m$ (*fitting*-) Parameter $x_j \in \mathbb{R}$ so bestimmt werden, daß

$$\varphi(t_i; x_1, \ldots, x_n) \approx b_i \qquad \forall i = 1, \ldots, m.$$

Im Sinne einer präziseren Formulierung dieses Wunsches definieren wir die Vektoren

$$\underline{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m \qquad \underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

und die vektorwertige Funktion

$$\underline{\varphi} : \mathbb{R}^n \to \mathbb{R}^m, \quad \underline{\varphi}(\underline{x}) = \begin{pmatrix} \varphi(t_1; \underline{x}) \\ \vdots \\ \varphi(t_m; \underline{x}) \end{pmatrix}.$$

Nun soll der Parametervektor $\underline{x} \in \mathbb{R}^n$ so bestimmt werden, daß $\|\underline{b} - \underline{\varphi}(x)\|$ minimal wird.

Die Wahl der Norm $\|\cdot\|$ haben wir noch frei. Aus CoMa I kennen wir schon die *p-Normen*

$$\|\underline{y}\|_p = \left(\sum_{i=1}^{m} |y_i|^p\right)^{1/p},$$
$$\|\underline{y}\|_\infty = \max_{i=1,\ldots,m} |y_i| .$$

Je nach Wahl von p hat das resultierende Ausgleichsproblem einen anderen Namen:
$p = 1$: L^1-Ausgleichsrechnung
$p = 2$: Methode der kleinsten Fehlerquadrate (Gauß $\sim$ 1800)
$p = \infty$: Tschebyscheff-Ausgleichsrechnung

Die Verwendung der $\|\cdot\|_2$-Norm ist wegen ihrer Bezüge zur Wahrscheinlichkeitsrechnung von besonderer Bedeutung. Für ausführliche Erläuterungen verweisen wir auf Deuflhard und Hohmann [1, Abschnitt 3.1].
Sind die Fitting-Funktionen $\varphi(t_i; \underline{x})$ linear in den Parametern x_i, gilt also

$$\varphi(t_i; \underline{x}) = a_1(t_i)x_1 + \cdots + a_n(t_i)x_n, \quad i = 1, \ldots, m,$$

oder in Matrixschreibweise

$$\underline{\varphi}(\underline{x}) = A\underline{x}, \qquad A = \begin{pmatrix} a_1(t_1) & \cdots & a_n(t_1) \\ \vdots & & \vdots \\ a_1(t_m) & \cdots & a_n(t_m) \end{pmatrix} \in \mathbb{R}^{m\times n},$$

so spricht man von einem *linearen Ausgleichsproblem*

$$\underline{x} \in \mathbb{R}^n : \quad \|\underline{b} - A\underline{x}\| \le \|\underline{b} - A\underline{y}\| \quad \forall \underline{y} \in \mathbb{R}^n. \tag{2.3}$$

Ist zufällig $n = m$, so ist (2.3) äquivalent zu $A\underline{x} = \underline{b}$. In der Praxis liegt allerdings oft der Fall $n < m$ vor.

Beispiel:
Wir wollen eine Ausgleichsgerade $p(t)$ durch die Punkte (t_i, t_i^2) mit $t_i = ih$, $h = \frac{1}{m}$, legen. Also ist $n = 2$,

$$\varphi(t, ; \underline{x}) = x_1 + tx_2 \qquad \forall \underline{x} = (x_1, x_2)^T \in \mathbb{R}^2$$

und $b_i = t_i^2$, $i = 1, \ldots, m$. Für $m = 8$ zeigt Abbildung 2.2 das Ergebnis $p(t) = t - 0.1428571$ im Vergleich mit (t_i, b_i). Die Berechnung erfolgte mit der Methode der kleinsten Quadrate, die wir später kennenlernen werden. ◁

2.1 Bestapproximation in normierten Räumen und Prähilberträumen

Sei V ein normierter, linearer Raum über $\mathbb{R}$ und $U \subset V$ ein endlichdimensionaler Unterraum. Im vorigen Abschnitt haben wir die Beispiele

$$V = C[a, b], \qquad \|\cdot\| = \|\cdot\|_p,\ p = 2, \infty, \qquad U = \mathcal{P}_n, \mathcal{S}_n$$

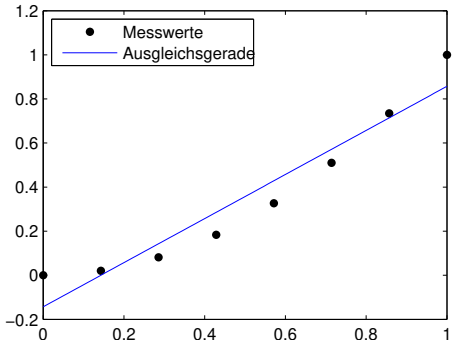


Abbildung 2.2: Approximation von Meßwerten mit der Methode der kleinsten Quadrate

kennengelernt.
Sei $f \in V$ fest gewählt. Wir betrachten wieder die allgemeine Bestapproximationsaufgabe

$$u \in U: \quad \|u - f\| \leq \|v - f\| \quad \forall v \in U. \tag{2.4}$$

Zunächst die gute Nachricht:

Satz 2.1 (Existenz) *Zu jedem $f \in V$ existiert eine Lösung der Approximationsaufgabe* (2.4).

Beweis:
Wir definieren die Abbildung $g : U \to \mathbb{R}$ durch $g(v) = \|v - f\|$. Aus der Dreiecksungleichung folgt

$$|g(v) - g(w)| = |\, \|v - f\| - \|w - f\| \,| \leq \|v - w\| \quad \forall v, w \in U.$$

Also ist g stetig. Sei $B = \{v \in U \mid \|v\| \leq 2\|f\|\} \subset U$. Aus $v \notin B$, also $\|v\| > 2\|f\|$ folgt

$$g(v) = \|v - f\| \geq \|v\| - \|f\| > \|f\| = g(0).$$

Offenbar ist $0 \in B$. Also kann kein Minimum von g außerhalb von B liegen. Damit ist die Aufgabe

$$u \in B: \qquad g(u) \leq g(v) \qquad \forall v \in B$$

zu (2.4) äquivalent. Wegen $g(v) \geq 0$ ist

$$\inf_{v \in B} g(v) \geq 0 > -\infty.$$

$B \subset U$ ist abgeschlossen und beschränkt. Daher ist B kompakt, denn U ist endlichdimensional. Eine stetige Funktion nimmt bekanntlich auf einer kompakten Menge ihr Infimum an. Also existiert ein $u \in B$ mit der Eigenschaft

$$g(u) = \inf_{v \in B} g(v). \qquad \square$$

Nun die schlechte Nachricht: Der Beweis ist nicht konstruktiv, d.h. wir erfahren nichts darüber, wie eine Lösung u berechnet werden kann. Im allgemeinen ist die Lösung auch nicht eindeutig, wie folgendes einfache Beispiel zeigt.

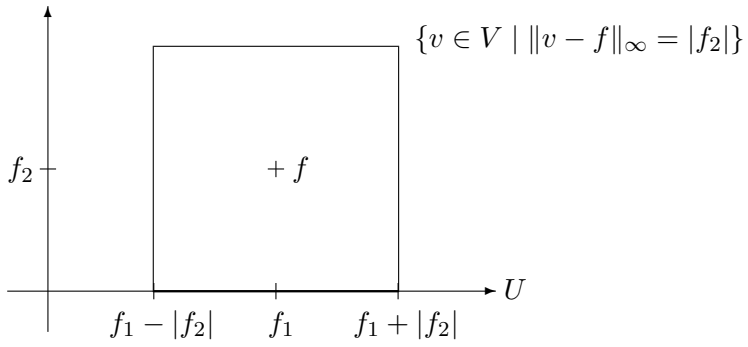


Abbildung 2.3: Unendlich viele Bestapproximationen

Beispiel:
Sei $V = \mathbb{R}^2$, $\|\cdot\| = \|\cdot\|_\infty$, $U = \{v = (v_1, v_2) \in \mathbb{R}^2 \mid v_2 = 0\}$ und $f = (f_1, f_2)$. Dann ist nicht nur $u = (f_1, 0) \in U$ Lösung, sondern jedes $u = (u_1, 0)$ mit $u_1 \in I = [f_1 - |f_2|, f_1 + |f_2|]$, denn

$$\|u - f\|_\infty = \max\{|u_1 - f_1|, |f_2|\} = |f_2| = \min_{v \in U} \|v - f\|_\infty \qquad \forall u_1 \in I. \qquad \triangleleft$$

Satz 2.2 (Eindeutigkeit) *Ist der Raum V strikt konvex, d.h.*

$$\frac{1}{2}\|v + w\| < 1 \qquad \forall v, w \in V \text{ mit } v \neq w, \|v\| = \|w\| = 1, \tag{2.5}$$

so ist die Approximationsaufgabe (2.4) *für jedes $f \in V$ eindeutig lösbar.*

Beweis:
Seien $u_1 \neq u_2$ Lösungen von (2.4), also

$$\gamma := \|u_1 - f\| = \|u_2 - f\| \leq \|v - f\| \qquad \forall v \in V.$$

Im Falle $\gamma = 0$ folgt sofort $u_1 = u_2$. Sei also $\gamma \neq 0$. Setzt man in (2.5) $v = \frac{1}{\gamma}(u_1 - f) \neq w = \frac{1}{\gamma}(u_2 - f)$ ein und multipliziert dann die Ungleichung mit γ, so folgt

$$\frac{1}{2}\|u_1 - f + u_2 - f\| < \gamma.$$

Wir setzen nun $u^* = \frac{1}{2}(u_1 + u_2)$ und erhalten

$$\|u^* - f\| = \frac{1}{2}\|u_1 - f + u_2 - f\| < \gamma = \|u_1 - f\| = \|u_2 - f\|.$$

Das steht aber im Widerspruch zur Bestapproximation von u_1 und u_2. $\square$

Definition 2.3 *Sei V ein reeller, linearer Raum. Eine Abbildung $(\cdot,\cdot) : V \times V \to \mathbb{R}$ heißt Skalarprodukt auf V, falls für alle u, v, $w \in V$ und alle α, $\beta \in \mathbb{R}$ die folgenden Bedingungen erfüllt sind:*

$$\begin{aligned} (u,v) &= (v,u) && \textit{(symmetrisch)} \\ (\alpha u + \beta v, w) &= \alpha(u,w) + \beta(v,w) && \textit{(linear)} \\ (v,v) &\geq 0,\ (v,v) = 0 \Leftrightarrow v = 0 && \textit{(positiv definit)} \end{aligned} \tag{2.6}$$

Versehen mit der Norm

$$\|v\| = \sqrt{(v,v)}, \quad v \in V,$$

heißt V dann Prähilbertraum.

Beispiel:

- Der $\mathbb{R}^n$ versehen mit dem *euklidischen Skalarprodukt*

$$(x,y) = \sum_{i=1}^{n} x_i y_i$$

  ist ein Prähilbertraum. Die zugehörige Norm ist $\|\cdot\|_2$.

- Der lineare Raum $C[a,b]$ versehen mit dem *L^2-Skalarprodukt*

$$(v,w) = \int_a^b v(x)w(x)\,dx$$

  ist ein Prähilbertraum. Die zugehörige Norm ist $\|\cdot\|_2$. ◁

Bemerkung:
Ein vollständiger Prähilbertraum heißt Hilbertraum. Der $\mathbb{R}^n$ versehen mit dem euklidischen Skalarprodukt ist ein Hilbertraum. Der lineare Raum $C[a,b]$ versehen mit dem L^2-Skalarprodukt ist nicht vollständig, also kein Hilbertraum. ◁

Der Hilbertraum $\mathbb{R}^2$ mit dem euklidischen Skalarprodukt ist konvex, wie Abbildung 2.4 zeigt. Dieser Sachverhalt lässt sich verallgemeinern.

Satz 2.4 *Ein Prähilbertraum ist strikt konvex.*

Beweis:
Übung. □

Satz 2.5 *In einem Prähilbertraum existiert eine eindeutig bestimmte Lösung u von* (2.4).

Beweis:
Die Behauptung folgt direkt aus Satz 2.1, Satz 2.2 und Satz 2.4. □

In einem Prähilbertraum steht eine sogenannte *Variationsformulierung* des Minimierungsproblems (2.4) zur Verfügung, die wir später zur praktischen Berechnung von u verwenden werden.

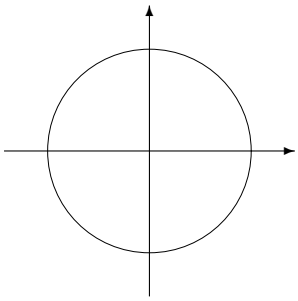


Abbildung 2.4: Die Einheitskugel im $\mathbb{R}^2$

Satz 2.6 *Sei V ein Prähilbertraum. Dann ist das Bestapproximationsproblem* (2.4) *äquivalent zu der sogenannten Normalengleichung*

$$u \in U: \qquad (u-f,v) = 0 \qquad \forall v \in U. \tag{2.7}$$

Beweis:
Sei u eine Lösung von (2.4). Wir zeigen, daß u dann auch (2.7) löst. Sei $v \in U$ und $t > 0$ beliebig aber fest gewählt. Ausmultiplizieren liefert

$$\|u-f\|^2 \le \|u+tv-f\|^2 = \|u-f\|^2 + 2t(v,u-f) + t^2\|v\|^2,$$

also

$$2t(v,u-f) + t^2\|v\|^2 \ge 0. \tag{2.8}$$

Division durch $t > 0$ und Grenzübergang $t \to 0$ liefert

$$(v,u-f) \ge 0.$$

Offenbar gilt (2.8) auch für beliebiges $t < 0$. Division durch $t < 0$ und Grenzübergang $t \to 0$ ergibt dann

$$(v,u-f) \le 0.$$

Insgesamt haben wir also $(v,u-f) = 0$ gezeigt. Da $v \in U$ beliebig war, ist u Lösung von (2.7).
Sei nun umgekehrt u Lösung von (2.7). Wir zeigen, daß u dann auch (2.4) löst. Sei $v \in U$ beliebig aber fest gewählt. Dann gilt

$$\|u+v-f\|^2 = \|u-f\|^2 + 2(v,u-f) + \|v\|^2 = \|u-f\|^2 + \|v\|^2 \ge \|u-f\|^2.$$

Damit ist u Lösung von (2.4). □

Beispiel:
Wir betrachten das Beispiel $V = \mathbb{R}^2$, $\|\cdot\| = \|\cdot\|_2$, $U = \{v = (v_1, v_2) \in \mathbb{R}^2 \mid v_2 = 0\}$ und $f = (f_1, f_2)$. Die Normalengleichung hat dann eine anschauliche Interpretation: *Der Fehler $f - u$ der Bestapproximation steht senkrecht auf dem Unterraum U.* ◁

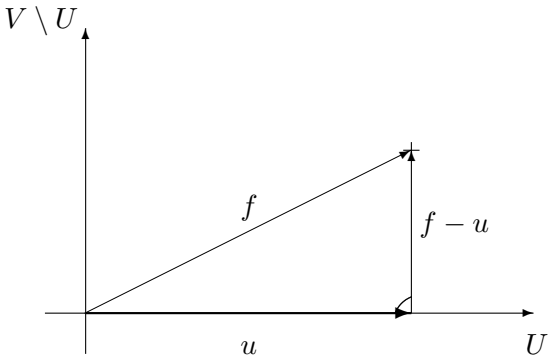


Abbildung 2.5: Orthogonalität bei Bestapproximation in Prähilberträumen

Dieser Sachverhalt motiviert folgende Definition.

Definition 2.7 *Sei V ein normierter Raum und $P : V \to V$ eine lineare Abbildung mit der Eigenschaft*

$$P^2 = P.$$

Dann heißt P Projektion.
Sei V ein Prähilbertraum und $P : V \to V$ eine Projektion mit der Eigenschaft

$$(v, (I - P)f) = 0 \quad \forall f \in V,\ \forall v \in R(P) = \{Pw \mid w \in V\}.$$

Dann heißt P Orthogonalprojektion auf $R(P)$.

Beispiel:
Sei $V = \mathbb{R}^2$. Dann ist

$$Px = \begin{pmatrix} x_1 \\ 0 \end{pmatrix}, \qquad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

eine Orthogonalprojektion bezüglich des euklidischen Skalarprodukts. ◁

Beispiel:
Sei V ein Prähilbertraum. Nach Satz 2.5 existiert zu jedem $f \in V$ eine eindeutig bestimmte Bestapproximation $u \in U$. Die durch

$$V \ni f \mapsto u = Pf \in U$$

definierte Abbildung P ist eine Orthogonalprojektion von V auf U (Übung). ◁

Satz 2.8 *Sei V ein Prähilbertraum und $P : V \to U = R(P)$ eine Orthogonalprojektion. Dann ist*

$$u = Pf$$

die Bestapproximation von f aus U.

Beweis:
Übung. □

Satz 2.9 *Sei V ein normierter Raum und $P \neq 0$ eine Projektion. Dann gilt $\|P\| \geq 1$.*
Sei V ein Prähilbertraum und $P \neq 0$ eine Projektion. Dann ist P genau dann eine Orthogonalprojektion, wenn $\|P\| = 1$ gilt.

Beweis:
Ist P unbeschränkt, also $\|P\| = \sup_{v \neq 0} \frac{\|Pv\|}{\|v\|} = \infty$, so ist die Aussage trivial. Andernfalls folgt aus $P^2 = P$

$$\|P\| = \|P^2\| \leq \|P\|^2$$

und Division durch $\|P\| > 0$ liefert die Behauptung.
Sei P eine Orthogonalprojektion auf $R(P)$. Wir zeigen $\|P\| = 1$ und haben dazu nur noch $\|P\| \leq 1$ nachzuweisen. Sei $f \in V$, $f \neq 0$, beliebig aber fest gewählt und $u = Pf$, $w = (I - P)f$. Dann gilt nach Voraussetzung $(u, w) = 0$ und daher

$$\|f\|^2 = \|u + w\|^2 = \|u\|^2 + 2(u, w) + \|w\|^2 = \|u\|^2 + \|w\|^2 \qquad \text{(Pythagoras)}.$$

Daraus folgt

$$\|Pf\|^2 = \|u\|^2 \leq \|u\|^2 + \|w\|^2 = \|f\|^2,$$

also $\|P\| \leq 1$.
Sei nun umgekehrt $P : V \to R(P)$ eine Projektion mit $\|P\| = 1$. Wir zeigen $(v, (I-P)f) = 0$ $\forall f \in V$, $v \in R(P)$. Sei also $v \in R(P) \subset V$, $f \in V$ und $w = (I - P)f \in R(I - P)$. Dann gilt $Pv = v$ nebst $Pw = Pf - P^2 f = 0$ und daher

$$\|0 - v\| = \|Pw - v\| = \|Pw - Pv\| = \|P(w - v)\| \leq \|w - v\|.$$

Da $f \in V$ und damit $w \in R(I - P)$ beliebig war, ist $0 \in R(I - P)$ die Bestapproximation von v. Aus Satz 2.6 folgt dann die Normalengleichung $(0 - v, w) = 0$ $\forall w \in R(I - P)$. Gleichbedeutend ist $(v, (I - P)f) = 0$ $\forall v \in V$ und das wollten wir zeigen. □

Wie angekündigt, wollen wir jetzt aus der Normalengleichung eine Berechnungsvorschrift für u herleiten. Dazu wählen wir eine Basis $\{\varphi_1, \ldots, \varphi_n\}$ von U. Dann hat u die Darstellung

$$u = \sum_{j=1}^{n} u_j \varphi_j$$

mit *unbekannten Koeffizienten* $u_j \in \mathbb{R}$. Einsetzen dieser Darstellung in die Normalengleichung und Wahl von $v = \varphi_i$, $i = 1, \ldots, n$ liefert n *lineare Gleichungen*

$$\sum_{j=1}^{n} (\varphi_j, \varphi_i)\, u_j = (f, \varphi_i), \quad i = 1, \ldots, n.$$

Setzt man

$$\begin{aligned} A &= (a_{ij})_{i,j=1}^{n} \in \mathbb{R}^{n,n}, & a_{ij} &= (\varphi_j, \varphi_i), \\ \underline{b} &= (b_j)_{j=1}^{n} \in \mathbb{R}^{n}, & b_j &= (f, \varphi_j), \\ \underline{u} &= (u_j)_{j=1}^{n} \in \mathbb{R}^{n}, \end{aligned} \tag{2.9}$$

so erhält man das lineare Gleichungssystem

$$A\underline{u} = \underline{b} \tag{2.10}$$

zur Berechnung des Koeffizientenvektors $\underline{u}$ und damit der Bestapproximation u.

Bemerkung:
Die Koeffizientenmatrix A heißt *Gramsche Matrix*. Wegen $a_{ij} = (\varphi_j, \varphi_i) = (\varphi_i, \varphi_j) = a_{ji}$ ist A *symmetrisch*. Darüberhinaus ist A *positiv definit* (Übung). ◁

Für die Lösung von (2.10) wäre es besonders angenehm, wenn $a_{ij} = (\varphi_j, \varphi_i) = 0$ für $i \neq j$ gelten würde, denn dann könnte man die Lösung $u_j = b_j/a_{jj}$ direkt hinschreiben.

Definition 2.10 *Sei V ein Prähilbertraum mit Skalarprodukt $(\cdot,\cdot)$ und $U \subset V$ endlichdimensional. Eine Basis $(\varphi_i)_{i=1}^n$ von U mit der Eigenschaft*

$$(\varphi_i, \varphi_j) = 0 \quad \forall i, j = 1, \ldots, n,\ i \neq j,$$

heißt Orthogonalbasis bezüglich $(\cdot,\cdot)$.

Beispiel:
Sei $U = \mathbb{R}^2$. Dann ist $\{e_1, e_2\}$ mit

$$e_1 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad e_2 = \begin{pmatrix} 0 \\ 9 \end{pmatrix}$$

eine Orthogonalbasis von U bezüglich des euklidischen Skalarprodukts. ◁

Beispiel:
Sei $U = \mathcal{P}_n \subset V = C[-1,1]$ mit dem L^2-Skalarprodukt $(v,w) = \int_{-1}^1 v(x)w(x)\,dx$. Dann bilden die

$$\psi_i(x) = \frac{i!}{(2i)!}\frac{d^i}{dx^i}(x^2-1)^i, \quad i = 0, \ldots, n,$$

eine Orthogonalbasis von $\mathcal{P}_n$ mit führenden Koeffizienten 1. Bis auf einen Normierungsfaktor ($\frac{1}{2^i i!}$ statt $\frac{i!}{(2i)!}$) sind das die sogenannten *Legendre-Polynome*. ◁

Es stellt sich die Frage, wie man zu einer Orthogonalbasis kommt. Aus einer vorliegenden Basis kann man mit dem sogenannten *Gram-Schmidtschen Orthogonalisierungsverfahren* eine Orthogonalbasis berechnen (vgl. z.B. Werner [7, Satz V.4.2]). Im Falle von Polynomen $U = \mathcal{P}_n$ geben wir eine Berechnungsvorschrift an.

Satz 2.11 *Zu jedem Skalarprodukt $(\cdot,\cdot)$ auf $\mathcal{P}_n$ gibt es eindeutig bestimmte Orthogonalpolynome $\psi_i \in \mathcal{P}_i$, $i = 0, \ldots, n$, mit führendem Koeffizienten eins. Sie genügen der Drei-Term-Rekursion*

$$\psi_i(x) = (x + \alpha_i)\psi_{i-1}(x) + \beta_i\psi_{i-2}(x), \qquad i = 1, 2, \ldots, n, \tag{2.11}$$

mit den Anfangswerten

$$\psi_{-1} \equiv 0, \qquad \psi_0 \equiv 1, \qquad \beta_1 = 0,$$

und den Koeffizienten

$$\alpha_i = -\frac{(x\psi_{i-1}, \psi_{i-1})}{(\psi_{i-1}, \psi_{i-1})}, \qquad \beta_i = -\frac{(\psi_{i-1}, \psi_{i-1})}{(\psi_{i-2}, \psi_{i-2})}.$$

Beweis:
Der Induktionsbeweis findet sich bei Deuflhard und Hohmann [1, Satz 6.2]. □

Dieser Satz lässt sich auf abstrakte Prähilberträume verallgemeinern (vgl. Satz 6.4 in [1]).

2.2 Approximation stetiger Funktionen

2.2.1 Tschebyscheff-Approximation durch Polynome

Wir betrachten den Spezialfall $V = C[a,b]$, $\|\cdot\| = \|\cdot\|_\infty$ und $U = \mathcal{P}_n$ der allgemeinen Bestapproximationsaufgabe (2.4), also

$$p \in \mathcal{P}_n : \quad \|p - f\|_\infty \le \|q - f\|_\infty \quad \forall q \in \mathcal{P}_n \tag{2.12}$$

zu gegebenem $f \in C[a,b]$.
Nach Satz 2.1 existiert eine Lösung p von (2.12). Allerdings liefert der allgemeine Satz 2.2 nicht die Eindeutigkeit, denn $C[a,b]$ mit Norm $\|\cdot\|_\infty$ ist nicht strikt konvex (Übung). Hier haben wir es aber mit einem *Spezialfall* zu tun! Die speziellen Eigenschaften von $C[a,b]$ und vor allem von $\mathcal{P}_n$ kann man ausnutzen (Stichworte: Haarscher Raum, Tschebyscheffscher Alternantensatz (1853)). So erhält man:

Satz 2.12 *Die Lösung von* (2.12) *ist eindeutig bestimmt.*

Beweis:
Wir verweisen auf eine ausführliche Darstellung in Hämmerlin und Hoffmann [3, Kapitel 4 §4]. □

Beispiel:
Sei $V = C[-1,1]$, $f(x) = x^{n+1}$ und $U = \mathcal{P}_n$. Dann ist (2.12) offenbar äquivalent zu

$$\omega \in \mathcal{P}_{n+1}^{(1)} : \quad \|\omega\|_\infty \le \|q\|_\infty \quad \forall q \in \mathcal{P}_{n+1}^{(1)}, \tag{2.13}$$

wobei

$$\mathcal{P}_{n+1}^{(1)} = \{p \in \mathcal{P}_{n+1} \mid p = x^{n+1} - q,\ q \in \mathcal{P}_n\}$$

gesetzt ist. Die Lösung von (2.13) ist gegeben durch (siehe [3, Kapitel 4, §4.7])

$$\omega = \frac{1}{2^n} T_{n+1}.$$

Dabei sind

$$T_n(x) = \cos\big(n \arccos(x)\big), \quad n = 0, 1, \ldots$$

die sogenannten *Tschebyscheff-Polynome 1. Art*. Eine äquivalente Charakterisierung ist durch die *3-Term-Rekursion*

$$T_0(x) = 1, \quad T_1(x) = x,$$
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n = 1, 2, \ldots$$

gegeben. Abbildung 2.6 zeigt links die Tschebyscheff-Polynome T_1, T_2 und T_3 und rechts die Lösung von (2.13) für $n = 2$ im Vergleich mit x^3. Beachte, daß auch die Tschebyscheff-Approximation eine ungerade Funktion ist. ◁

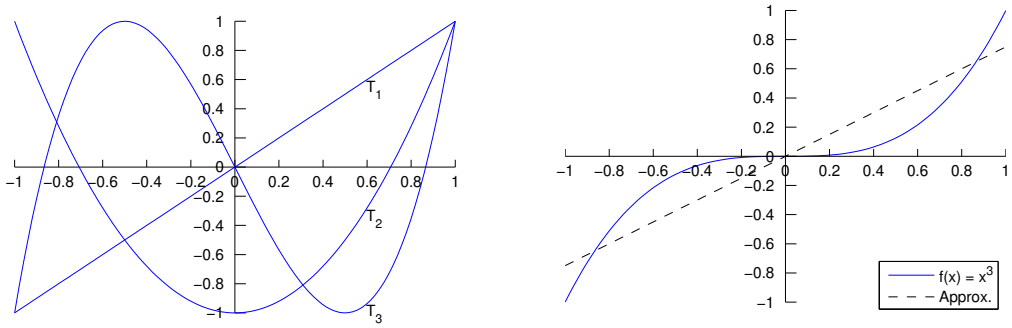


Abbildung 2.6: Tschebyscheff-Polynome T_1, T_2 und T_3 (links) und Tschebyscheff-Approximation von $f(x) = x^3$ in $\mathcal{P}_2$ (rechts).

2.2.2 L^2-Approximation

Der lineare Raum $V = C[0, 1]$ versehen mit dem Skalarprodukt

$$(v, w) = \int_0^1 v(x)w(x)\, dx \tag{2.14}$$

ist ein Prähilbertraum. Wir betrachten zunächst die Approximation durch Polynome, also $U = \mathcal{P}_n$. Nach Satz 2.5 ist die entsprechende Bestapproximationsaufgabe eindeutig lösbar und nach Satz 2.6 äquivalent zur Normalengleichung

$$p \in \mathcal{P}_n : \quad (p, q) = (f, q) \quad \forall q \in \mathcal{P}_n.$$

Um p tatsächlich ausrechnen zu können, müssen wir eine Basis $(\varphi_i)_{i=0}^n$ von $\mathcal{P}_n$ wählen.

Naheliegende Wahl: Die *Monome*

$$\varphi_i(x) = x^i, \quad i = 0, \ldots, n.$$

Die zugehörige Gramsche Matrix hat dann die Koeffizienten

$$(\varphi_i, \varphi_j) = \int_0^1 x^{i+j}\, dx = \frac{1}{i + j + 1}.$$

Das ist gerade die sogenannte *Hilbertmatrix*. Die Kondition der Hilbertmatrix wächst sehr schnell mit n (MATLAB-Routine `cond`)! Damit werden bei der Lösung des zugehörigen Gleichungssystems (vorsichtig ausgedrückt) Schwierigkeiten entstehen.

Kluge Wahl: Die *Legendre-Polynome*

$$\psi_i(x) = \frac{1}{2^i i!} \frac{d^i}{dx^i}(x^2-1)^i, \quad i = 0, \ldots, n,$$

bilden bekanntlich eine Orthogonalbasis von $\mathcal{P}_n \subset C[-1,1]$. Wie kann man aus den Legendre-Polynomen eine Orthogonalbasis $(\varphi_i)_{i=0}^n$ von $\mathcal{P}_n \subset C[0,1]$ bezüglich des zugehörigen Skalarprodukts (2.14) bestimmen (Übung)?

Beispiel:
Es sei $f(x) = x^3$ und $n = 2$. Eine Orthogonalbasis von $\mathcal{P}_2$ ist gegeben durch

$$\varphi_0(x) = 1, \quad \varphi_1(x) = 2x - 1, \quad \varphi_2(x) = 3(2x-1)^2 - 1.$$

Wir erhalten die Gramsche Matrix

$$A = \begin{pmatrix} (\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & (\varphi_0,\varphi_2) \\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & (\varphi_1,\varphi_2) \\ (\varphi_2,\varphi_0) & (\varphi_2,\varphi_1) & (\varphi_2,\varphi_2) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{4}{5} \end{pmatrix}$$

und die rechte Seite

$$\underline{b} = \begin{pmatrix} (f,\varphi_0) \\ (f,\varphi_1) \\ (f,\varphi_2) \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ \frac{3}{20} \\ \frac{1}{10} \end{pmatrix}.$$

Aus dem linearen Gleichungssystem $A\underline{p} = \underline{b}$ erhalten wir die Koeffizienten

$$\underline{p} = \begin{pmatrix} \frac{1}{4} \\ \frac{9}{20} \\ \frac{1}{8} \end{pmatrix}$$

und damit die L^2-Approximation

$$p(x) = \frac{1}{4} + \frac{9}{20}\,(2x-1) + \frac{1}{8}\,(3(2x-1)^2 - 1).$$

In Abbildung 2.7 ist die L^2-Approximation p im Vergleich zu $f(x) = x^3$ (gestrichelt) dargestellt. ◁

Als nächstes betrachten wir die Approximation durch lineare finite Elemente. Wir setzen also $U = \mathcal{S}_n$ mit $\mathcal{S}_n$ aus (2.2). Zur Berechnung der Lösung u aus der Normalengleichung

$$u \in \mathcal{S}_n: \quad (u,v) = (f,v) \quad \forall v \in \mathcal{S}_n$$

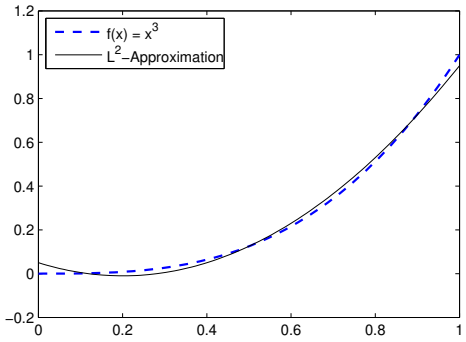


Abbildung 2.7: L^2-Approximation von $f(x) = x^3$ in $\mathcal{P}_2$

haben wir eine Basis von $\mathcal{S}_n$ zu wählen. Jedes Element φ_i der sogenannten *Knotenbasis* ist charakterisiert durch die Interpolationseigenschaft

$$\varphi_i \in \mathcal{S}_n : \quad \varphi_i(x_k) = \delta_{ik} \quad \forall k = 0, \ldots, n \quad \text{(Kronecker-}\delta\text{)}. \tag{2.15}$$

Setzt man

$$h_i = x_i - x_{i-1}, \quad i = 1, \ldots, n,$$

so gilt

$$\begin{aligned}
\varphi_0(x) &= \begin{cases} 1 - \frac{1}{h_1}(x - x_0) & \text{falls } x \in [x_0, x_1] \\ 0 & \text{sonst,} \end{cases} \\
\varphi_i(x) &= \begin{cases} 1 + \frac{1}{h_i}(x - x_i) & \text{falls } x \in [x_{i-1}, x_i] \\ 1 - \frac{1}{h_{i+1}}(x - x_i) & \text{falls } x \in [x_i, x_{i+1}] \\ 0 & \text{sonst,} \end{cases} \qquad i = 1, \ldots, n-1 \\
\varphi_n(x) &= \begin{cases} 1 + \frac{1}{h_n}(x - x_n) & \text{falls } x \in [x_{n-1}, x_n] \\ 0 & \text{sonst.} \end{cases}
\end{aligned} \tag{2.16}$$

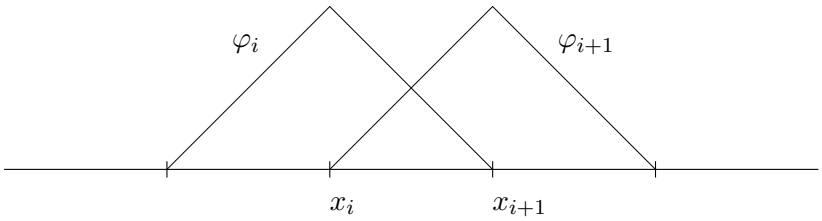


Abbildung 2.8: Knotenbasis von $\mathcal{S}_n$

Man nennt die Basisfunktionen φ_i oft *Dach-* oder *Hütchenfunktionen*. Mit Blick auf Abbildung 2.8 ist klar warum. Die Knotenbasis hat zwei Vorteile: Erstens erhält man mit den

Koeffizienten u_i der Knotenbasisdarstellung

$$u = \sum_{i=0}^{n} u_i \varphi_i$$

wegen $u_i = u(x_i)$ direkt die Werte an den Gitterpunkten (hier Knoten). Daher der Name. Zweitens gilt $(\varphi_i, \varphi_j) = 0$ falls $|i - j| \geq 2$, denn die Knotenbasisfunktionen haben einen *lokalen Träger*. Nur in der Diagonalen und in den beiden Nebendiagonalen der Gramschen Matrix A (vgl. (2.9)) stehen also von 0 verschiedene Elemente. Eine solche Matrix heißt *Tridiagonalmatrix*.
Wir wollen nun auch die Koeffizienten in der Diagonalen und den beiden Nebendiagonalen ausrechnen. Wir beginnen mit den Diagonalelementen. Für $i = 1, \ldots, n-1$ erhält man

$$\begin{aligned} a_{ii} = (\varphi_i, \varphi_i) &= \int_{x_{i-1}}^{x_{i+1}} \varphi_i(x)^2 \, dx \\ &= \int_{x_{i-1}}^{x_i} \left(1 + \frac{x - x_i}{h_i}\right)^2 dx + \int_{x_i}^{x_{i+1}} \left(1 - \frac{x - x_i}{h_{i+1}}\right)^2 dx \\ &= h_i \int_0^1 t^2 \, dt - h_{i+1} \int_1^0 t^2 \, dt = \frac{1}{3}(h_i + h_{i+1}) \end{aligned}$$

und es gilt

$$a_{00} = (\varphi_0, \varphi_0) = \frac{1}{3} h_1, \quad a_{nn}(\varphi_n, \varphi_n) = \frac{1}{3} h_n.$$

Ist $i = 0, \ldots, n-1$ so gilt

$$\begin{aligned} a_{i,i+1} = (\varphi_i, \varphi_{i+1}) &= \int_{x_i}^{x_{i+1}} \varphi_i(x) \varphi_{i+1}(x) \, dx \\ &= \int_{x_i}^{x_{i+1}} \left(1 - \frac{x - x_i}{h_{i+1}}\right) \left(1 + \frac{x - x_{i+1}}{h_{i+1}}\right) dx \\ &= \int_{x_i}^{x_{i+1}} \left(1 - \frac{x - x_i}{h_{i+1}}\right) \frac{x - x_i}{h_{i+1}} \, dx \\ &= h_{i+1} \int_0^1 (1 - t) t \, dt = \frac{1}{6} h_{i+1}. \end{aligned}$$

Für $i = 1, \ldots, n$ ergibt sich daraus

$$a_{i-1,i} = (\varphi_{i-1}, \varphi_i) = \frac{1}{6} h_i.$$

Zur Vereinfachung der Schreibweise definieren wir

$$u_{-1} = u_{n+1} = h_0 = h_{n+1} = 0.$$

Insgesamt erhalten wir dann die Normalengleichungen

$$\frac{1}{6} h_i u_{i-1} + \frac{1}{3}(h_i + h_{i+1}) u_i + \frac{1}{6} h_{i+1} u_{i+1} = (f, \varphi_i), \quad i = 0, \ldots, n.$$

Multipliziert man die i-te Gleichung jeweils mit $\frac{6}{h_i+h_{i+1}}$, so ergibt sich zur Berechnung der Unbekannten

$$\underline{u} = \begin{pmatrix} u_0 \\ \vdots \\ u_n \end{pmatrix}$$

das lineare Gleichungssystem

$$M\underline{u} = \underline{\beta} \tag{2.17}$$

mit Koeffizientenmatrix

$$M = \begin{pmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \lambda_{n-1} \\ & & & \mu_n & 2 \end{pmatrix}, \quad \mu_i = \frac{h_i}{h_i + h_{i+1}}, \quad \lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}} \tag{2.18}$$

und rechter Seite

$$\underline{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_n \end{pmatrix}, \quad \beta_i = \frac{6}{h_i + h_{i+1}}(f, \varphi_i). \tag{2.19}$$

Man kann zeigen, daß sich eine LR-Zerlegung von A mit $\mathcal{O}(n)$ Punktoperationen berechnen lässt (vgl. z.B. Hämmerlin und Hoffmann [3, Kapitel 2 §1.4]). Damit ist das zugehörige lineare Gleichungssystem mit optimalem Aufwand lösbar! Dazu kommt, daß M *unabhängig von* n sehr gut konditioniert ist.

Satz 2.13 *Es gilt*

$$\kappa_\infty(M) = \|M\|_\infty \|M^{-1}\|_\infty \leq 3.$$

Beweis:
Wegen

$$|\mu_i| + |\lambda_i| = \mu_i + \lambda_i = 1, \quad i = 0, \ldots, n \tag{2.20}$$

ist

$$\|M\|_\infty = \max_{i=0,\ldots,n} \sum_{j=0}^{n} |m_{ij}| = \max_{i=0,\ldots,n} (\mu_i + 2 + \lambda_i) = 3.$$

Da wir M^{-1} nicht kennen, ist die Abschätzung von $\|M^{-1}\|_\infty$ etwas schwieriger. Sei $z \in \mathbb{R}^{n+1}$ beliebig aber fest gewählt. Aus

$$(Mz)|_i = \mu_i z_{i-1} + 2z_i + \lambda_i z_{i+1}, \quad i = 0, \ldots, n,$$

und (2.20) folgt die Abschätzung

$$|z_i| \leq \frac{1}{2}\|Mz\|_\infty + \frac{1}{2}\|z\|_\infty, \quad i = 0, \ldots, n.$$

Daraus ergibt sich

$$\|z\|_\infty \leq \frac{1}{2}\|Mz\|_\infty + \frac{1}{2}\|z\|_\infty$$

und damit

$$\|z\|_\infty \leq \|Mz\|_\infty.$$

Setzt man $z = M^{-1}y$ so erhält man schließlich

$$\|M^{-1}\|_\infty = \max_{\substack{y\in\mathbb{R}^{n+1}\\ y\neq 0}} \frac{\|M^{-1}y\|_\infty}{\|y\|_\infty} = \max_{\substack{z\in\mathbb{R}^{n+1}\\ z\neq 0}} \frac{\|z\|_\infty}{\|Mz\|_\infty} \leq 1. \qquad \square$$

2.3 Methode der kleinsten Fehlerquadrate

Wir betrachten das *lineare Ausgleichsproblem* (2.3) im Falle $\|\cdot\| = \|\cdot\|_2$, also

$$x \in \mathbb{R}^n : \quad \|b - Ax\|_2 \leq \|b - Av\|_2 \quad \forall v \in \mathbb{R}^n \tag{2.21}$$

für gegebene Matrix $A \in \mathbb{R}^{m\times n}$, $m \geq n$ und $b \in \mathbb{R}^m$.

Bemerkung:
Ist x eine Lösung von (2.21), so ist $u = Ax \in R(A) = \{Ay \mid y \in \mathbb{R}^n\} \subseteq \mathbb{R}^m$ eine Lösung von

$$u \in R(A) : \quad \|b - u\|_2 \leq \|b - v\|_2 \quad \forall v \in R(A) \tag{2.22}$$

und umgekehrt. $\triangleleft$

Damit lässt sich das lineare Ausgleichsproblem (2.21) als Bestapproximationsaufgabe (2.4) mit $V = \mathbb{R}^m$, $\|\cdot\| = \|\cdot\|_2$ und $U = R(A)$ formulieren. Aus Satz 2.5 folgt die Existenz einer eindeutig bestimmten Lösung $u \in R(A)$ von (2.22), welche durch die Normalengleichung

$$u \in R(A) : \quad (u, v) = (b, v) \quad \forall v \in R(A) \tag{2.23}$$

charakterisiert ist (euklidisches Skalarprodukt in $\mathbb{R}^m$). Eigentlich interessiert uns aber eine Lösung $x \in \mathbb{R}^n$ von (2.21).

Satz 2.14 *Es sei $m \geq n$. Der Vektor $x \in \mathbb{R}^n$ ist genau dann Lösung des linearen Ausgleichsproblems* (2.21)*, wenn x der Normalengleichung*

$$A^T Ax = A^T b \tag{2.24}$$

genügt. Ist A injektiv, so ist x eindeutig bestimmt.

Beweis:
Sei x Lösung von (2.21). Wir zeigen, daß dann x Lösung von (2.24) ist. Da x (2.21) löst, ist $u = Ax$ Lösung von (2.22), genügt also der Normalengleichung (2.23). Einsetzen von $u = Ax$ und $v = Ay$, $y \in \mathbb{R}^n$, beliebig, liefert

$$(Ax, Ay) = (u, v) = (b, v) = (b, Ay)$$

und daher

$$(A^TAx, y) = (A^Tb, y) \quad \forall y \in \mathbb{R}^n.$$

Wählt man $y = e_i$ (i-ter Einheitsvektor in $\mathbb{R}^n$), $i = 1, \ldots, n$, so folgt (2.24). Sei umgekehrt x Lösung von $A^TAx = A^Tb$. Wir zeigen, daß x dann Lösung von (2.21) ist. Da x Lösung von (2.24) ist, genügt $u = Ax$ der Normalengleichung (2.23), denn die obigen Schlüsse sind umkehrbar. Damit ist u Lösung von (2.22) und somit x Lösung von (2.21). Nach Satz 2.5 ist u eindeutig bestimmt. Ist $A : \mathbb{R}^n \to R(A) \subset \mathbb{R}^m$ injektiv, so existiert genau ein $x \in \mathbb{R}^n$ mit $u = Ax$. □

Bekanntlich ist A genau dann injektiv, wenn $N(A) = \{0\}$ (Kern von A) und $\dim R(A) = n$ vorliegt oder, gleichbedeutend, wenn A maximalen Spaltenrang n hat.

Bemerkung:
Es sei $n = m$ und A regulär. Dann gilt

$$\kappa_2(A^TA) = \|A^TA\|_2 \|(A^TA)^{-1}\|_2 = \|A\|_2^2 \|A^{-1}\|_2^2 = \kappa_2(A)^2,$$

denn

$$\|A^TA\|_2 = \lambda_{\max}((A^TA)^2)^{\frac{1}{2}} = \lambda_{\max}(A^TA) = \|A\|_2^2, \ \|(A^TA)^{-1}\|_2 = \lambda_{\min}(A^TA)^{-1} = \|A^{-1}\|_2^2.$$

Ist A schlecht konditioniert, so ist also A^TA *sehr* schlecht konditioniert. ◁

Beispiel:
Sei

$$A = \begin{pmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \end{pmatrix}, \quad A^TA = \begin{pmatrix} 1+\varepsilon^2 & 1 \\ 1 & 1+\varepsilon^2 \end{pmatrix}.$$

Ist ε größer als die kleinste darstellbare Zahl, aber $\varepsilon < \sqrt{eps}$ (*eps* ist die Maschinengenauigkeit), so gilt

$$\tilde{A} = rd(A) = \begin{pmatrix} 1 & 1 \\ rd(\varepsilon) & 0 \\ 0 & rd(\varepsilon) \end{pmatrix} \text{ regulär}, \quad \widetilde{A^TA} = rd(A^TA) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \text{ singulär}.$$

Man spricht vom *fast rangdefekten Fall*. ◁

Ist A injektiv, so ist A^TA symmetrisch und positiv definit, d.h.

$$(A^TAx, x) \geq 0 \ \forall x \in \mathbb{R}^n, \quad (A^TAx, x) = 0 \Leftrightarrow x = 0.$$

Eine Variante des Gaußschen Algorithmus, das *Cholesky-Verfahren* (vgl. Deuflhard und Hohmann [1, Kapitel 1.4]), ist daher auf (2.24) anwendbar. Bekanntlich gibt es beim Gaußschen Algorithmus und ähnlich auch beim Cholesky-Verfahren *Stabilitätsprobleme* im Falle schlecht konditionierter Koeffizientenmatrizen. Wir beschreiben daher etwas aufwendigere, aber stabilere Methoden zur Lösung der Normalengleichung (2.24).

2.3.1 Orthogonalisierungsverfahren

Die Grundidee stammt von Gene Golub (vgl. Golub und van Loan [2, Kapitel 5]): Statt $A^TAx = A^Tb$ zu lösen, versuchen wir das lineare Ausgleichsproblem (2.21) *direkt anzugehen.*

Bemerkung:
Die euklidische Norm $\|\cdot\|_2$ ist invariant unter *Orthogonaltransformationen* $Q \in \mathbb{R}^{m\times m}$, das sind $Q \in \mathbb{R}^{m\times m}$ mit $Q^TQ = I$, denn

$$\|b - Ax\|_2^2 = (Q^TQ(b-Ax), b-Ax) = (Q(b-Ax), Q(b-Ax)) = \|Q(b-Ax)\|_2^2.$$

Anstelle von (2.21) können wir also auch

$$x \in \mathbb{R}^n: \quad \|Q(b-Ax)\|_2 \le \|Q(b-Av)\|_2 \quad \forall v \in \mathbb{R}^n$$

lösen. Dabei haben wir die Wahl einer Orthogonaltransformation $Q \in \mathbb{R}^{m\times m}$ *frei*! ◁

Satz 2.15 *Sei $m \ge n$, $b \in \mathbb{R}^m$ und $A \in \mathbb{R}^{m\times n}$ mit $\dim R(A) = n$. Ferner sei $Q \in \mathbb{R}^{m\times m}$ eine Orthogonaltransformation mit der Eigenschaft*

$$Q^TA = \begin{pmatrix} R \\ 0 \end{pmatrix}, \quad Q^Tb = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

Dabei ist $R \in \mathbb{R}^{n\times n}$ eine obere Dreiecksmatrix und $b_1 \in \mathbb{R}^n$, $b_2 \in \mathbb{R}^{m-n}$. Dann ist

$$x = R^{-1}b_1$$

die Lösung von (2.21) *und es gilt*

$$\|b - Ax\|_2 = \min_{v\in\mathbb{R}^n} \|b - Av\|_2 = \|b_2\|_2.$$

Beweis:
Sei $v \in \mathbb{R}^n$ beliebig. Dann gilt

$$\begin{aligned} \|b - Av\|_2^2 &= \|Q^Tb - Q^TAv\|_2^2 \\ &= \left\| \begin{pmatrix} b_1 - Rv \\ b_2 \end{pmatrix} \right\|_2^2 = \|b_1 - Rv\|_2^2 + \|b_2\|_2^2 \\ &\ge \|b_2\|_2^2. \end{aligned}$$

Einsetzen von $v = x = R^{-1}b_1$ liefert

$$\|b - Ax\|_2^2 = \|b_2\|_2^2.$$

Zusammen erhält man die Behauptung. □

Definition 2.16 *Die Produktdarstellung*

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix} \quad R \in \mathbb{R}^{n\times n} \text{ ist obere Dreiecksmatrix,}$$

heißt QR-Zerlegung von A.

Bemerkung:
Eine QR-Zerlegung ist *stabil*, denn $\kappa_2(A) = \kappa_2(R)$. ◁

Es bleibt nur noch eine kleine Frage offen: *Wie kann man eine QR-Zerlegung berechnen?*

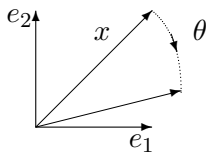
2.3.2 Givens-Rotationen und Householder-Reflexionen

Orthogonale Abbildungen setzen sich aus *Drehungen* und *Spiegelungen* zusammen (siehe z.B. Kowalsky [4, 6. Kapitel, Satz 24.3]).

Beispiel:

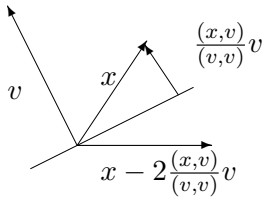
- Im $\mathbb{R}^2$ haben Drehungen die Gestalt

$$Q = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}.$$



- Die Spiegelung (Reflexion) eines Vektors $x \in \mathbb{R}^2$ an einer Geraden mit Normalenvektor $v \in \mathbb{R}^2$ lassen sich mit Hilfe der Orthogonalprojektion $\frac{(x,v)}{(v,v)}v$ von x auf v formulieren,

$$Qx = x - 2\frac{(v,x)}{(v,v)}v.$$



◁

Wir erinnern uns: Bei Durchführbarkeit des Gaußschen Algorithmus lieferten die Eliminationsmatrizen $(I - G_k)$ wegen

$$R = (I - G_{n-1}) \cdots (I - G_1)A, \quad L = (I - G_1)^{-1} \cdots (I - G_{n-1})^{-1}$$

eine LR-Zerlegung von A. Anstelle der Gaußschen Eliminationsmatrizen $(I - G_k)$ wollen wir nun Orthogonaltransformationen verwenden.

Givens-Rotationen. Es sei $s^2 + c^2 = 1$ und $1 \leq l < k \leq n$. Dann heißt eine Orthogonaltransformation $Q_{lk} \in \mathbb{R}^{m\times m}$ der Gestalt

$$Q_{lk} = \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & c & & & s & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & -s & & & & c & \\ & & & & & & & & 1 \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \rightarrow \quad l\text{-te Zeile} \\ \\ \\ \rightarrow \quad k\text{-te Zeile} \end{matrix}$$

Givens-Rotation. Im Falle $s = \sin\theta$, $c = \cos\theta$ bewirkt Q_{lk} gerade eine Drehung in der Ebene span $\{e_l, e_k\}$ um den Winkel θ.
Sei $a \in \mathbb{R}^m$ gegeben. Dann gilt

$$(Q_{lk}a)|_i = \begin{cases} ca_l + sa_k & \text{für } i = l \\ -sa_l + ca_k & \text{für } i = k \\ a_i & \text{sonst.} \end{cases}$$

Insbesondere ist $(Q_{lk}a)|_k = 0$ falls $a_l = a_k = 0$. Sei nun $a_k \neq 0$. Dann wollen wir Q_{lk} abhängig von a so bestimmen, dass $(Q_{lk}a)|_k = 0$ gilt. Wir wollen also a_k mittels Q_{lk} „eliminieren“. Dazu berechnen wir $c, s \in \mathbb{R}$ aus dem linearen Gleichungssystem

$$\begin{array}{lll} ca_l + sa_k & = r & = (Q_{kl}a)|_l \\ ca_k - sa_l & = 0 & = (Q_{kl}a)|_k \end{array}, \qquad r = \sqrt{a_l^2 + a_k^2} \neq 0. \tag{2.25}$$

Die Lösung ist

$$c = \frac{a_l}{r}, \quad s = \frac{a_k}{r}.$$

Sei nun $A = (A_1|\cdots|A_n)$. Der Vektor $A_j \in \mathbb{R}^m$ bezeichnet also die j-te Spalte von $A \in \mathbb{R}^{m\times n}$. Wir eliminieren zunächst sukzessive die Subdiagonalelemente von A_1. Dazu bestimmen wir nach obiger Vorschrift Givens-Rotationen $Q^{(1)}_{k-1,k}$, $k = m, \ldots, 2$, so daß gilt

$$Q^{(1)}_{12} \cdots Q^{(1)}_{m-1,m} A_1 = \begin{pmatrix} * \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

und setzen

$$A^{(1)} = Q^{(1)}A \qquad Q^{(1)} = Q^{(1)}_{12} \cdots Q^{(1)}_{m-1,m}.$$

Beispiel:
Sei $A \in \mathbb{R}^{4\times 3}$. Dann sieht die Elimination der Subdiagonalelemente der ersten Spalte wie folgt aus.

$$A \xrightarrow{Q_{34}} \begin{pmatrix} & & \\ & & \\ * & * & * \\ 0 & * & * \end{pmatrix} \xrightarrow{Q_{23}} \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & & \\ & & \end{pmatrix} \xrightarrow{Q_{12}} \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & & \\ 0 & & \end{pmatrix}$$

Dabei werden nur die mit $*$ gekennzeichneten Koeffizienten neu berechnet. ◁

Auf die gleiche Weise eliminieren wir nun die Subdiagonalelemente der zweiten Spalte $A_2^{(1)}$ von $A^{(1)} = (A_1^{(1)}|\cdots|A_n^{(1)})$ mit Givens-Rotationen $Q_{k-1,k}^{(2)}$, $k = m, \ldots, 3$, und erhalten

$$A^{(2)} = Q^{(2)}A^{(1)} \qquad Q^{(2)} = Q_{23}^{(2)} \cdots Q_{m-1,m}^{(2)}.$$

Da $A_{1,k-1}^{(1)} = A_{1k}^{(1)} = 0$, $\forall k = m, \ldots, 3$, bleiben die Nullen in der ersten Spalte erhalten! Induktiv erhält man

$$Q^T A = \begin{pmatrix} R \\ 0 \end{pmatrix} \qquad Q^T = Q^{(n-1)} \cdots Q^{(1)}.$$

Abschließend wollen wir den Aufwand abschätzen. Unter der Voraussetzung $m \approx n$ werden

$$\mathcal{O}(\tfrac{4}{3}n^3) \text{ Punktoperationen und } \mathcal{O}(\tfrac{1}{2}n^2) \text{ Quadratwurzeln}$$

benötigt (Übung). Das ist etwa der *vierfache Aufwand* der Gauß-Elimination angewandt auf die Normalengleichung (2.24). Das ist der Preis für bessere Stabilitätseigenschaften! Im Falle $m \gg n$ benötigt man

$$\mathcal{O}(2mn^2) \text{ Punktoperationen und } \mathcal{O}(mn) \text{ Quadratwurzeln.}$$

Householder-Reflexionen. Als nächstes wollen wir *Spiegelungen* (Reflexionen) verwenden, um eine QR-Zerlegung zu berechnen. Dazu benötigen wir das sogenannte *dyadische Produkt*

$$vw^T = \begin{pmatrix} v_1w_1 & \cdots & v_1w_n \\ \vdots & & \vdots \\ v_nw_1 & \cdots & v_nw_n \end{pmatrix} \in \mathbb{R}^{m\times m}, \quad v, w \in \mathbb{R}^m.$$

Das dyadische Produkt von v, w ist gerade das Matrixprodukt von $v \in \mathbb{R}^{m\times 1}$ und $w^T \in \mathbb{R}^{1\times m}$. Umgekehrt ist das euklidische Skalarprodukt

$$(v, w) = v^T w \in \mathbb{R}, \quad v, w \in \mathbb{R}^m,$$

gerade das Matrixprodukt von $v^T \in \mathbb{R}^{1\times m}$ mit $w \in \mathbb{R}^{m\times 1}$.
Sei $v \neq 0 \in \mathbb{R}^m$. Dann heißt

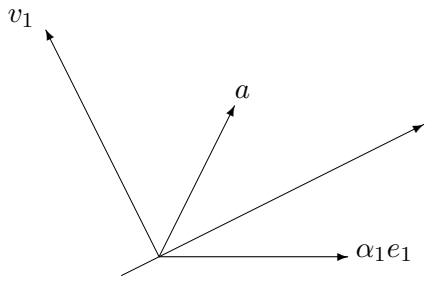
$$Q = Q(v) = I - 2\frac{vv^T}{v^T v}$$

die zu v gehörige *Householder-Reflexion*.

Sei $a \in \mathbb{R}^m$ gegeben. Wir nehmen an, daß $a \neq \alpha e_1$, $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^m$, $\forall \alpha \in \mathbb{R}$. Dann wollen wir $v_1 \in \mathbb{R}^m$ abhängig von a so bestimmen, daß die zugehörige Householder-Reflexion $Q_1 = Q(v_1)$ die Bedingung

$$Q_1 a = Q(v_1)a = \alpha_1 e_1 \tag{2.26}$$

erfüllt. Wir wollen also die Koeffizienten $a_2, \ldots, a_n$ von a mittels Q_1 „eliminieren“.



Um v_1 zu finden, gehen wir folgendermaßen vor. Aus $Q_1 a = a - 2\frac{(v_1,a)}{(v_1,v_1)}v_1 = \alpha_1 e_1$ folgt unmittelbar, daß v_1 und $a - \alpha_1 e_1$ linear abhängig sein müssen. Die Längenerhaltung liefert $\|a\|_2 = \|Q_1 a\|_2 = \|\alpha_1 e_1\|_2 = |\alpha_1|$. Diese beiden Beobachtungen motivieren die Wahl

$$v_1 = a - \alpha_1 e_1, \quad \alpha_1 = \|a\|_2.$$

Wir zeigen nun, daß der so gewählte Vektor v_1 tatsächlich die gewünschte Eigenschaft (2.26) hat. Es gilt nämlich

$$(v_1, v_1) = (a - \alpha_1 e_1, a - \alpha_1 e_1) = \|a\|_2^2 - 2(\alpha_1 e_1, a) + \alpha_1^2 = 2(a - \alpha_1 e_1, a) = 2(v_1, a)$$

und daher

$$Q_1 a = a - 2\frac{(v_1, a)}{(v_1, v_1)}v_1 = a - v_1 = \alpha_1 e_1.$$

Sei nun wieder $A = (A_1|\cdots|A_n)$. Der Vektor $A_j \in \mathbb{R}^m$ bezeichnet also die j-te Spalte von $A \in \mathbb{R}^{m\times m}$. Wir eliminieren zunächst die Subdiagonalelemente von A_1. Dazu verwenden wir die Householder-Reflexion

$$Q_1 = Q(v_1), \quad v_1 = A_1 - \alpha_1 e_1, \quad \alpha_1 = \|A_1\|_2,$$

mit der Eigenschaft

$$Q_1 A_1 = \alpha_1 e_1$$

und setzen

$$A^{(1)} = Q_1 A.$$

Die Matrix $A^{(1)} = (A_1^{(1)}|\cdots|A_n^{(1)})$ hat die Spalten $A_j^{(1)} \in \mathbb{R}^m$. Zur Elimination der Subdiagonalelemente von $A_2^{(1)}$ verwenden wir die Householder-Reflexion

$$Q_2 = Q(v_2), \quad v_2 = \begin{pmatrix} 0 \\ \tilde{v}_2 \end{pmatrix},$$

wobei

$$\tilde{v}_2 = \tilde{A}_2^{(1)} - \alpha_2 \tilde{e}_1 \in \mathbb{R}^{m-1}, \quad \alpha_2 = \|\tilde{A}_2^{(1)}\|_2$$

und

$$\tilde{A}_2^{(1)} = \begin{pmatrix} A_{22}^{(1)} \\ A_{32}^{(1)} \\ \vdots \\ A_{m2}^{(1)} \end{pmatrix} \in \mathbb{R}^{m-1}, \quad \tilde{e}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{m-1}$$

gesetzt ist. Wegen $(e_1, v_2) = 0$ gilt

$$Q_2(\alpha_1 e_1) = \alpha_1 e_1.$$

Damit bleibt die erste Spalte von $A^{(1)}$ bei Multiplikation mit Q_2 erhalten und somit gilt

$$A^{(2)} = Q_2 A^{(1)} = \begin{pmatrix} * & * & * & * \\ 0 & * & \vdots & \vdots \\ \vdots & 0 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & * & * \end{pmatrix},$$

wobei „$*$“ eventuell von Null verschiedene Elemente bezeichnet.
Nach n Schritten folgt

$$Q^T A = \begin{pmatrix} R \\ 0 \end{pmatrix}, \quad Q^T = Q_{n-1} \cdots Q_1.$$

Abschließend wollen wir den Aufwand abschätzen. Unter der Voraussetzung $m \approx n$ werden

$$\mathcal{O}(\tfrac{2}{3}n^3) \text{ Punktoperationen}$$

benötigt (Übung). Das ist weniger als der *halbe Aufwand* der Givens-Rotationen und etwa der *doppelte Aufwand* der Gauß-Elimination (bei deutlich besseren Stabilitätseigenschaften). Im Falle $m \gg n$ benötigt man

$$\mathcal{O}(mn^2) \text{ Punktoperationen.}$$

Sieht man von weitergehenden Stabilitätsüberlegungen ab, so sind also Householder-Reflexionen den Givens-Rotationen vorzuziehen.

Beispiel:
Wir betrachten das lineare Gleichungssystem

$$Wx = b$$

mit der *Wilkinson-Matrix*

$$W = (w_{ij})_{i,j=1}^n, \quad w_{ij} = \begin{cases} 1 & \text{falls } i = j \text{ oder } j = n \\ -1 & \text{falls } i > j \\ 0 & \text{sonst.} \end{cases}$$

Zu einer vorgegebenen exakten Lösung $x = (x_i)_{i=1}^n$ mit Zufallszahlen $x_i \in [0,1]$ (Matlab-Zufallsgenerator `rand`) bestimmen wir die rechte Seite $b = Wx$ und berechnen nun Lösungen x_{LR} mittels LR-Zerlegung und x_{QR} mittels QR-Zerlegung, jeweils für $n = 1, \ldots, 50$. Dabei nutzen wir die Matlab-Programme `[L, R, P] = lu(W)` und `[L, R, P] = qr(W)`.

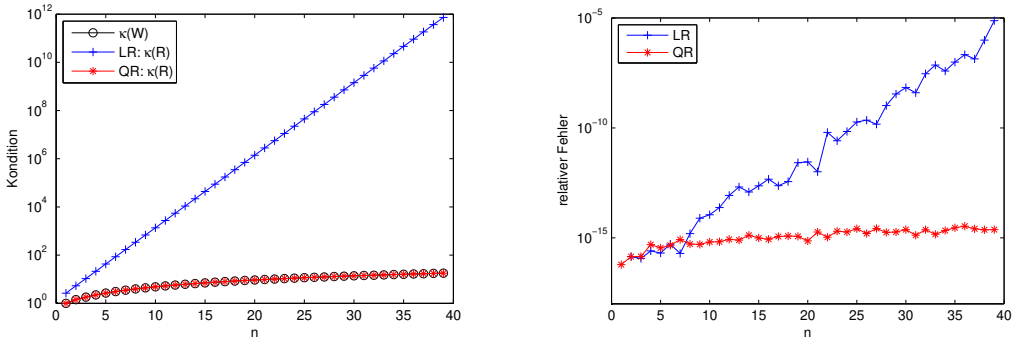


Abbildung 2.9: Vergleich von LR- und QR-Zerlegung der Wilkinson-Matrix

Wir wissen schon aus CoMa, daß der Gaußsche Algorithmus für die Wilkinson-Matrix instabil ist. Abbildung 2.9 zeigt dann auch links die moderat wachsende Kondition von W neben der explodierenden Kondition von R_{LR} (mit LR-Zerlegung erzeugt). Wie nach Konstruktion zu erwarten, ist die Kondition von R_{QR} (mit QR-Zerlegung erzeugt) von $\kappa(W)$ nicht zu unterscheiden.
Die schlechte Kondition von R_{LR} spiegelt sich direkt im relativen Fehler der Lösung x_{LR} wieder: Für $n = 50$ hat man keine gültige Stelle mehr! Demgegenüber stimmt x_{QR} bis auf Maschinengenauigkeit mit der exakten Lösung x überein. Der Mehraufwand für die QR-Zerlegung lohnt also in diesem Fall! ◁

Bemerkung:
Die QR-Zerlegung lässt sich auch zur Lösung von Eigenwertproblemen $Ax = \lambda x$ mit symmetrischer Koeffizientenmatrix $A \in \mathbb{R}^{n\times n}$ verwenden. Grundlage ist die Beobachtung, daß bei der Transformation $\tilde{A} = Q^T AQ$ mit einer orthogonalen Matrix $Q \in \mathbb{R}^{n\times n}$ die Eigenwerte invariant bleiben. Für Einzelheiten zum QR-Verfahren zur Eigenwertbestimmung verweisen wir auf Deuflhard und Hohmann [1, Kapitel 5.3], Stoer und Bulirsch [6, Kapitel 6.6.6] oder Golub und van Loan [2, Kapitel 8.8]. ◁

Literatur

[1] P. Deuflhard and A. Hohmann. *Numerische Mathematik I.* de Gruyter, 4. Auflage, 2008. Lineare Ausgleichsprobleme und Orthogonalisierungsverfahren sind in Kapitel 3 beschrieben. Wer wissen möchte, wie man nichtlineare Ausgleichsprobleme durch Kombination von Fehlerquadratmethode und Newton–Verfahren angeht, sollte in Kapitel 4 weiterlesen. Orthogonalisierungsverfahren für Eigenwertprobleme sind Gegenstand von Kapitel 5.

[2] G. Golub and C. van Loan. *Matrix Computations.* The Johns Hopkins Univ. Press, 3. Auflage, 1996. Seit seinem ersten Erscheinen im Jahre 1989 eines der Standardwerke zur

numerischen linearen Algebra. Nicht nur Orthogonalisierungsverfahren werden ausführlich dargestellt.

[3] G. Hämmerlin and K.-H. Hoffmann. *Numerische Mathematik*. Springer, 4. Auflage, 2004. Ein eher analytisch orientiertes Lehrbuch. Wer mehr über Approximation wissen möchte, sollte in Kapitel 4 weiterlesen, wo das Thema sehr ausführlich abgehandelt wird.

[4] H.J. Kowalsky and O. Michler. *Lineare Algebra*. de Gruyter, 12. Auflage, 2003. Dieses Buch werden Sie schon aus der Grundvorlesung *Lineare Algebra* kennen.

[5] J. Stoer. *Numerische Mathematik I*. Springer, 10. Auflage, 2007. Mit Kapitel 4 kann man seine Kenntnisse über lineare Gleichungssysteme auffrischen und erweitern.

[6] J. Stoer and R. Bulirsch. *Numerische Mathematik II*. Springer, 5. Auflage, 2005. Eine recht ausführliche Darstellung von Verfahren zur Lösung von Eigenwertproblemen findet sich in Kapitel 6.

[7] D. Werner. *Funktionalanalysis*. Springer, 6. Auflage, 2007. Ein sehr empfehlenswertes Lehrbuch zur Einführung in die Funktionalanalysis. In Kapitel V finden sich weiterführende Betrachtungen über Prä–Hilberträume und Hilberträume.

3 Interpolation

Es sei $f \in C[a,b]$ gegeben. Oft ist man nicht unbedingt an der *bestmöglichen* Approximation von f interessiert, sondern auch mit einer „guten“ Approximation zufrieden. Wir betrachten daher in diesem Kapitel die Interpolationsaufgabe

$$u_n \in U_n : \quad u_n(x_k) = f(x_k) \quad \forall k = 0, \ldots, n. \tag{3.1}$$

Dabei ist $U_n \subset C[a,b]$, $n = 1, 2, \ldots$, eine Folge von Teilräumen „einfacher“ Ansatzfunktionen und das *Gitter*

$$\Delta = \{x_0, \ldots, x_n \mid a \leq x_0 < x_1 < \cdots < x_n \leq b\}$$

„geeignet“ gewählt. Als Teilraum U_n von Ansatzfunktionen werden wir Polynome und gewisse stückweise polynomiale Funktionen, sogenannte *Splines*, verwenden.
Als erstes klären wir natürlich jeweils Existenz, Eindeutigkeit und Berechnung der Lösung u_n. Wir hoffen, durch Lösung der Interpolationsaufgabe (3.1) eine „gute“ Approximation zu erhalten. Deshalb werden wir jeweils die Approximationseigenschaften der Räume U_n untersuchen. Wir wollen insbesondere wissen, ob

$$\lim_{n\to\infty} \|f - u_n\|_\infty = 0$$

gilt. Darüberhinaus sind wir an der *Konvergenzgeschwindigkeit*, also an Fehlerabschätzungen der Form

$$\|f - u_n\|_\infty \leq C n^{-q}$$

mit einer von n unabhängigen Zahl $C > 0$ und (möglichst großem) $q > 0$, interessiert.
Wir betrachten nur die ∞–Norm, weil man aus

$$\|v\|_2 \leq (b-a)\|v\|_\infty \quad \forall v \in C[a,b]$$

direkt entsprechende Aussagen für die L^2–Norm erhält. Umgekehrt wäre das nicht der Fall.

3.1 Polynominterpolation

Als erstes betrachten wir die *Polynominterpolation*, also

$$U_n = \mathcal{P}_n = \{v \in C[a,b] \mid v \text{ ist Polynom vom Grad } \leq n\}.$$

Aus der allgemeinen Interpolationsaufgabe (3.1) erhält man dann das bereits aus der CoMa bekannte Problem

$$p_n \in \mathcal{P}_n : \quad p_n(x_k) = f(x_k) \quad \forall k = 0, \ldots, n. \tag{3.2}$$

Wir erinnern kurz an grundlegende Resultate zur Polynominterpolation. Einzelheiten finden sich im CoMa–Skript.

Satz 3.1 *Die Interpolationsaufgabe* (3.2) *hat eine eindeutig bestimmte Lösung* p_n. *Die Lagrange–Darstellung des Interpolationspolynoms* p_n *ist*

$$p_n = \sum_{k=0}^{n} f(x_k) L_k. \tag{3.3}$$

Dabei sind die Lagrange–Polynome $L_k \in \mathcal{P}_n$ *zum Gitter* Δ *gegeben durch*

$$L_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j}, \quad k = 0, \ldots, n.$$

Die Newton'sche Darstellung des Interpolationspolynoms p_n *ist*

$$p_n(x) = \sum_{k=0}^{n} f[x_0, \ldots, x_k] \prod_{i=0}^{k-1} (x - x_i). \tag{3.4}$$

Dabei sind die dividierten Differenzen ($(k-i)$*-ter Ordnung*) $f[x_i, \ldots, x_k]$, $0 \leq i \leq k \leq n$, *rekursiv definiert durch*

$$\begin{aligned} f[x_i] &= f(x_i), && i = 0, \ldots, n, \\ f[x_i, \ldots, x_k] &= \frac{f[x_{i+1}, \ldots, x_k] - f[x_i, \ldots, x_{k-1}]}{x_k - x_i}, && 0 \leq i < k \leq n. \end{aligned} \tag{3.5}$$

Es sei $f \in C^{n+1}[a,b]$. *Dann gibt es zu jedem* $x \in [a,b]$ *ein* $\xi(x) \in (a,b)$, *so daß die Fehlerformel*

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{k=0}^{n} (x - x_k) \tag{3.6}$$

gilt. Daraus folgt

$$\| f - p_n \|_\infty \leq \frac{1}{(n+1)!} \| f^{(n+1)} \|_\infty \| \omega \|_\infty, \qquad \omega(x) = \prod_{k=0}^{n} (x - x_k).$$

Wir werden in diesem Abschnitt zunächst überraschende Zusammenhänge zur Taylor'schen Formel aufdecken und dann die Approximationseigenschaften des Interpolationspolynoms untersuchen.

3.1.1 Hermite-Interpolation und Taylor'sche Formel

Eine bekannte Möglichkeit, eine gegebene Funktion $f \in C^{n+1}[a,b]$ durch ein Polynom zu approximieren, beruht auf der *Taylor'schen Formel*

$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x - x_0)^{n+1}$$

mit $\xi(x) = x_0 + \theta(x - x_0)$, $\theta \in (0, 1)$. Das Polynom $q_n \in \mathcal{P}_n$,

$$q_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k,$$

ist offenbar Lösung der folgenden *Hermite'schen Interpolationsaufgabe*

$$q_n \in \mathcal{P}_n : \quad q_n^{(k)}(x_0) = f^{(k)}(x_0) \quad \forall k = 0, \ldots, n.$$

Hermite'sche Interpolationsaufgaben sind dadurch charakterisiert, daß außer Funktionswerten auch gewisse Ableitungen reproduziert werden sollen. Die *Lagrange'sche Restglieddarstellung*

$$f(x) - q_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)^{n+1} \tag{3.7}$$

erinnert stark an die Fehlerformel (3.6) der Polynominterpolation. Auch zwischen q_n und p_n gibt es Analogien: Die dividierte Differenz k-ter Ordnung $f[x_0, \ldots, x_k]$ im Newton'schen Interpolationspolynom p_n entspricht der Ableitung $f^{(k)}(x_0)$ an der Stelle $x_0 = \cdots = x_k$.
Den engen Zusammenhang zwischen dividierten Differenzen und Ableitungen beschreibt folgender Satz.

Satz 3.2 (Hermite–Genocchi–Formel) *Es sei $f \in C^k[a, b]$ und $k \geq 1$. Dann gilt*

$$f[x_0, \ldots, x_k] = \int\limits_{\Sigma^k} f^{(k)}(x_0 + \textstyle\sum_{i=1}^{k} s_i(x_i - x_0))\, ds \tag{3.8}$$

mit dem k-dimensionalen Einheitssimplex

$$\Sigma^k = \left\{ s = (s_1, \ldots, s_k) \in \mathbb{R}^k \mid 0 \leq s_i \ \forall i = 1, \ldots, k, \ \textstyle\sum_{i=1}^{k} s_i \leq 1 \right\} \subset \mathbb{R}^k.$$

mit dem Volumen

$$| \Sigma^k | = \tfrac{1}{k!}$$

Beweis:
Der Beweis erfolgt durch vollständige Induktion über k.
a) Induktionsanfang: $k = 1$.
Offenbar ist $\Sigma^1 = [0, 1] \subset \mathbb{R}^1$. Variablentransformation $z = x_0 + s_1(x_1 - x_0)$ nebst Hauptsatz der Differential– und Integralrechnung liefern

$$\begin{aligned} \int_0^1 f^{(1)}(\overbrace{x_0 + s_1(x_1 - x_0)}^{z})\, ds_1 &= \frac{1}{x_1 - x_0} \int_{x_0}^{x_1} f^{(1)}(z)\, dz \\ &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f[x_0, x_1] \end{aligned}$$

und damit die Behauptung für $k = 1$.
b) Induktionsannahme.

Wir nehmen an, daß die Hermite–Genocchi–Formel (3.8) für ein fest gewähltes $k \geq 1$ gilt.
c) Induktionsschluß.
Wir zeigen, daß unter der Induktionsannahme b) die Formel auch für $k+1$ richtig ist. Ähnlich wie beim Induktionsanfang a) berechnet man mit Hilfe der Variablentransformation

$$z = x_0 + \textstyle\sum_{i=1}^{k} s_i(x_i - x_0) + s_{k+1}(x_{k+1} - x_0),$$

des Satzes von Fubini, des Hauptsatzes der Differential– und Integralrechnung, der Induktionsannahme b) und schließlich der rekursiven Definition der dividierten Differenzen (3.5)

$$\begin{aligned}
&\int_{\Sigma^{k+1}} f^{(k+1)}\big(x_0 + \sum_{i=1}^{k+1} s_i(x_i - x_0)\big)\, d(s, s_{k+1}) \\
&= \int_{\Sigma^k} \int_0^{1-\sum_{i=1}^k s_i} f^{(k+1)}\big(\overbrace{x_0 + \sum_{i=1}^{k} s_i(x_i - x_0) + s_{k+1}(x_{k+1} - x_0)}^{z}\big)\, ds_{k+1}\, ds \\
&= \frac{1}{x_{k+1} - x_0} \int_{\Sigma^k} \int_{x_0+\sum_{i=1}^k s_i(x_i - x_0)}^{x_{k+1}+\sum_{i=1}^k s_i(x_i - x_{k+1})} f^{(k+1)}(z)\, dz\, ds \\
&= \frac{1}{x_{k+1} - x_0} \left(\int_{\Sigma^k} f^{(k)}\big(x_{k+1} + \sum_{i=1}^{k} s_i(x_i - x_{k+1})\big)\, ds - \int_{\Sigma^k} f^{(k)}\big(x_0 + \sum_{i=1}^{k} s_i(x_i - x_0)\big)\, ds \right) \\
&= \frac{1}{x_{k+1} - x_0} \big(f[x_{k+1}, x_1, \ldots, x_k] - f[x_0, \ldots, x_k] \big) = f[x_0, \ldots, x_{k+1}].
\end{aligned}$$

Das ist gerade die Behauptung. □

Man kann die Hermite–Genocchi–Formel etwas kürzer schreiben, indem man

$$s_0 = s_0(s_1, \ldots, s_k) = 1 - \sum_{i=1}^{k} s_i$$

einführt. Dann gilt

$$f[x_0, \ldots, x_k] = \int_{\Sigma^k} f^{(k)}\big(\textstyle\sum_{i=0}^{k} s_i x_i\big)\, ds. \tag{3.9}$$

Ist f genügend oft differenzierbar, so ist die rechte Seite der Hermite–Genocchi–Formel (3.9) auch im Falle *zusammenfallender (konfluenter) Stützstellen*

$$a \leq x_0 \leq \cdots \leq x_k \leq b$$

wohldefiniert. Wir nutzen diese Tatsache, um Definition (3.5) auf konfluente Stützstellen zu erweitern.

Definition 3.3 *Es sei $f \in C^k[a,b]$ und $a \le x_0 \le \cdots x_k \le b$. Dann setzen wir*

$$f[x_0,\ldots,x_k] = \int\limits_{\Sigma^k} f^{(k)}\big(\textstyle\sum_{i=0}^k s_i x_i\big)\, ds.$$

Ist $x_0 = \cdots = x_k$, so folgt

$$\sum_{i=0}^k s_i x_i = x_0 \sum_{i=0}^k s_i = x_0.$$

Definition 3.3 liefert daher in diesem Fall

$$f[x_0,\ldots,x_k] = \frac{1}{k!} f^{(k)}(x_0),$$

denn es gilt ja

$$\int_{\Sigma^k} ds = \frac{1}{k!}.$$

Bei Definition 3.3 handelt es sich um eine *stetige Erweiterung* von Definition (3.5), wie folgendes Resultat zeigt.

Satz 3.4 *Es seien $f \in C^k[a,b]$ und $\{x_i^\nu\} \subset [a,b]$, $i = 0,\ldots,k$, Folgen mit der Eigenschaft*

$$\lim_{\nu\to\infty} x_i^\nu = x^* \in [a,b], \quad i = 0,\ldots,k.$$

Dann gilt

$$\lim_{\nu\to\infty} f[x_0^\nu,\ldots,x_k^\nu] = f[x^*,\ldots,x^*] = \frac{f^{(k)}(x^*)}{k!}.$$

Beweis:
Da $[a,b]$ kompakt ist, ist $f^{(k)}$ gleichmäßig stetig auf $[a,b]$. Daher ist Integration und Limesbildung vertauschbar und mit der Hermite–Genocchi–Formel (3.9)) folgt

$$\begin{aligned}\lim_{\nu\to\infty} f[x_0^\nu,\ldots,x_k^\nu] &= \lim_{\nu\to\infty} \int\limits_{\Sigma^k} f^{(k)}\big(\textstyle\sum_{i=0}^k s_i x_i^\nu\big)\, ds \\ &= \int\limits_{\Sigma^k} \lim_{\nu\to\infty} f^{(k)}\big(\textstyle\sum_{i=0}^k s_i x_i^\nu\big)\, ds = \frac{f^{(k)}(x^*)}{k!}.\end{aligned}$$

□

Wir verallgemeinern nun den Mittelwertsatz der Differentialrechnung.

Satz 3.5 *Es sei $a \le x_0 \le \cdots \le x_k \le b$ und $f \in C^k[a,b]$. Dann gibt es ein $\xi \in [x_0,x_k]$ mit der Eigenschaft*

$$f[x_0,\ldots,x_k] = \frac{1}{k!} f^{(k)}(\xi).$$

Beweis:
Aus Definition 3.3 und der Monotonie des Integrals folgt

$$\frac{1}{k!}\ \min_{s\in\Sigma^k} f^{(k)}\left(\sum_{i=0}^{k} s_i x_i\right) \le f[x_0,\ldots,x_k] \le \frac{1}{k!}\ \max_{s\in\Sigma^k} f^{(k)}\left(\sum_{i=0}^{k} s_i x_i\right).$$

Sei

$$f^{(k)}(x_{\min}) = \min_{s\in\Sigma^k} f^{(k)}\left(\sum_{i=0}^{k} s_i x_i\right), \quad f^{(k)}(x_{\max}) = \max_{s\in\Sigma^k} f^{(k)}\left(\sum_{i=0}^{k} s_i x_i\right).$$

Dann ist $x_0 \le x_{\min}, x_{\max} \le x_k$ und

$$\frac{1}{k!} f^{(k)}(x_{\min}) \le f[x_0,\ldots,x_k] \le \frac{1}{k!} f^{(k)}(x_{\max}).$$

Die Behauptung folgt nun aus dem Zwischenwertsatz. □

Mit Blick auf Definition 3.3 ist das zugehörige Newton–Polynom

$$p_n(x) = \sum_{k=0}^{n} f[x_0,\ldots,x_k] \prod_{i=0}^{k-1}(x - x_i) \tag{3.10}$$

auch für konfluente Stützstellen $a \le x_0 \le x_1 \le \cdots \le x_n \le b$ wohldefiniert. In diesem Fall ist p_n Lösung einer entsprechenden *Hermite'schen Interpolationsaufgabe*.
Wir illustrieren diesen Sachverhalt anhand eines Beispiels.

Folgerung 3.6 *Es sei $f \in C^{k_0}[a,b]$ und*

$$a \le x_0 = \cdots = x_{k_0} < x_{k_0+1} < \ldots < x_n \le b.$$

Dann ist das zugehörige Newton–Polynom (3.10) *die eindeutig bestimmte Lösung der Hermite'schen Interpolationsaufgabe*

$$p_n \in \mathcal{P}_n: \quad \begin{array}{rcll} p_n^{(k)}(x_0) & = & f^{(k)}(x_0) & \forall k = 0,\ldots,k_0 \\ p_n(x_k) & = & f(x_k) & \forall k = k_0+1,\ldots,n \end{array} \tag{3.11}$$

Es sei $f \in C^{n+1}[a,b]$. Dann gibt es zu jedem $x \in [a,b]$ ein $\xi(x) \in [a,b]$, so daß die Fehlerformel

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\ (x - x_0)^{k_0+1} \prod_{k=k_0+1}^{n} (x - x_k) \tag{3.12}$$

gilt.

Beweis:

a) Eindeutigkeit.

Dazu nehmen wir an, daß es zwei Lösungen $p_n \ne q_n$ von (3.11) gibt. Dann hat das Polynom $Q = p_n - q_n \in \mathcal{P}_n$ zumindest die k_0+1–fache Nullstelle x_0 und die $n-k_0$ einfachen Nullstellen $x_{k_0+1},\ldots,x_n$. Insgesamt sind das $n+1$ Nullstellen (ihrer Vielfachheit nach gezählt). Nach dem Fundamentalsatz der Algebra ist daher $Q \equiv 0$. Widerspruch!

b) Lösung durch das Newton–Polynom.

Nun wollen wir bestätigen, daß p_n aus (3.10) tatsächlich Lösung von (3.11) ist. Es gilt nach Definition 3.3

$$\begin{aligned} p_n(x) &= \sum_{k=0}^{n} f[x_0,\ldots,x_k] \prod_{i=0}^{k-1}(x-x_i) \\ &= \sum_{k=0}^{k_0} \frac{f^{(k)}(x_0)}{k!}(x-x_0)^k + \sum_{k=k_0+1}^{n} f[x_0,\ldots,x_k](x-x_0)^{k_0+1} \prod_{i=k_0+1}^{k-1}(x-x_i). \end{aligned}$$

Die Hermite–Bedingungen $p_n^{(k)}(x_0) = f^{(k)}(x_0)$, $k = 0,\ldots,k_0$, lassen sich nun durch Einsetzen überprüfen.

Um die übrigen Interpolationsbedingungen einzusehen, wählen wir Folgen $\{x_k^\nu\}$, $k = 0,\ldots,k_0$, mit den Eigenschaften

$$\lim_{\nu\to\infty} x_k^\nu = x_0, \quad k = 0,\ldots,k_0, \qquad x_0^\nu < x_1^\nu < \cdots < x_{k_0}^\nu \quad \nu = 1,2,\ldots.$$

Außerdem setzen wir einfach

$$x_k^\nu = x_k, \quad k = k_0+1,\ldots,n, \quad \nu = 1,2,\ldots.$$

Dann löst

$$p_n^\nu(x) = \sum_{k=0}^{n} f[x_0^\nu,\ldots,x_k^\nu] \prod_{i=0}^{k-1}(x-x_i^\nu)$$

die Interpolationsaufgabe

$$p_n^\nu \in \mathcal{P}_n: \quad p_n^\nu(x_k^\nu) = f(x_k^\nu), \quad \forall k = 0,\ldots,n.$$

Wegen Satz 3.4 gilt

$$\|p_n - p_n^\nu\|_\infty \to 0 \quad \nu\to\infty.$$

Daraus ergibt sich mit

$$|p_n(x_k) - f(x_k)| \le \|p_n - p_n^\nu\|_\infty \to 0, \quad k = k_0+1,\ldots,n,$$

die Behauptung.

c) Fehlerformel.

Eine Möglichkeit wäre, die Fehlerformel aus Satz 3.1 nebst Satz 3.4 zu verwenden. Wir geben einen direkten Beweis, der auf der Hermite–Genocchi–Formel beruht. Es sei $x \in [a,b]$, $x \ne x_k$ $\forall k = 0,\ldots,n$. Dann fügen wir zu den $n+1$ Knoten x_k noch den Knoten x hinzu. Das entsprechende Interpolationspolynom ist

$$p_{n+1}(z) = p_n(z) + f[x_0,\ldots,x_n,x](z-x_0)\cdots(z-x_n).$$

Einsetzen von $z = x$ liefert

$$p_{n+1}(x) = f(x) = p_n(x) + f[x_0, \ldots, x_n, x](x - x_0) \cdots (x - x_n).$$

Aus Satz 3.5 folgt nun die Existenz von $\xi(x) \in [a, b]$ mit

$$f[x_0, \ldots, x_n, x] = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \quad . \qquad \square$$

Im Extremfall $x_0 = \cdots = x_n$ erhalten wir gerade das Taylor–Polynom q_n nebst Lagrange'scher Restglieddarstellung (3.7).
Die Argumentation aus Folgerung 3.6 lässt sich direkt auf andere Hermite'sche Interpolationsaufgaben übertragen.

Beispiel:
Wir betrachten die Hermite'sche Interpolationsaufgabe

$$p_3 \in \mathcal{P}_3 : \quad \begin{array}{rcll} p_3^{(k)}(0) & = & 1 & k = 0, 1 \\ p_3^{(k)}(1) & = & -1 & k = 0, 1 \ . \end{array}$$

Zur Bestimmung der Lösung setzen wir

$$x_0 = x_1 = 0, \quad x_2 = x_3 = 1$$

und

$$f[x_0] = f[x_1] = 1, \qquad f[x_0, x_1] = 1$$
$$f[x_2] = f[x_3] = -1, \qquad f[x_2, x_3] = -1 \ .$$

Dann berechnen wir das folgende Neville–Tableau wie üblich gemäß der Rekusion (3.5).

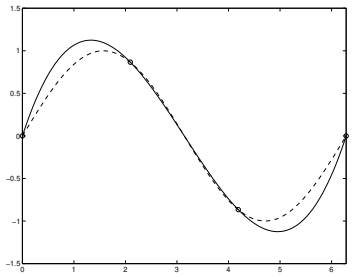
x_k	$f[x_k]$	$f[x_{k-1}, x_k]$	$f[x_{k-2}, \ldots, x_k]$	$f[x_{k-3}, \ldots, x_k]$
0	1			
0	1	1		
1	-1	$(-1-1)/1 = -2$	$(-2-1)/1 = -3$	
1	-1	-1	$(-1+2)/1 = 1$	$(1+3)/1 = 4$

Die Lösung ist dann

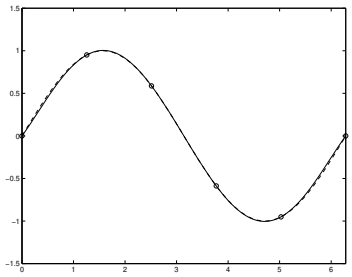
$$\begin{aligned} p_3(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, \ldots, x_2](x - x_0)(x - x_1) \\ &\quad + f[x_0, \ldots, x_3](x - x_0)(x - x_1)(x - x_2) \\ &= 1 + x - 3x^2 + 4x^2(x - 1). \end{aligned} \qquad \triangleleft$$

3.1.2 Approximationseigenschaften des Interpolationspolynoms

Wir wollen nun das Verhalten des *Interpolationsfehlers* $\|f - p_n\|_\infty$ für $n \to \infty$ untersuchen. Zur Einstimmung betrachten wir zwei Beispiele.



$n = 3$



$n = 5$

Abbildung 3.1: Konvergenz des Interpolationspolynoms für $f(x) = \sin(x)$

Beispiel:
Wir interpolieren die Funktion $f(x) = \sin(x)$ auf dem Intervall $[a, b] = [0, 2\pi]$ an den äquidistanten Stützstellen $x_k = kh$, $k = 0, \ldots, n$ zur Schrittweite $h = 2\pi/n$ für $n = 3, 5$. Die Ergebnisse im Vergleich mit $f(x) = \sin(x)$ (gestrichelt) sind in Abbildung 3.1 zu sehen.
Wir beobachten

$$\|f - p_n\|_\infty \to 0.$$

Mit Hilfe der Fehlerformel (3.6) und der *Stirling'schen Formel*

$$n! \approx \sqrt{2\pi} \left(\frac{n}{e}\right)^n$$

lässt sich diese Beobachtung auch theoretisch absichern. Aus

$$\|f - p_n\|_\infty \leq (2\pi)^{n+1} \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \approx \frac{\sqrt{2\pi}}{2\pi\sqrt{n+1}} \left(\frac{2\pi e}{n+1}\right)^{n+1}$$

folgt sogar exponentielle Konvergenz! ◁

Unsere Beobachtung lässt sich verallgemeinern: Ist f eine *ganze Funktion*, d.h. konvergiert die Potenzreihenentwicklung von f in ganz $\mathbb{C}$, so gilt

$$\|f - p_n\| \to 0 \quad n \to \infty$$

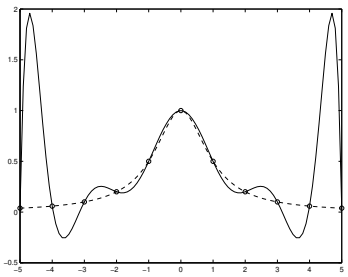
für *jede* Folge von Stützstellen (vgl. Hämmerlin und Hoffmann [4, Kapitel 5, §4.3]). Mit anderen Funktionen kann man böse Überraschungen erleben, wie das nächste Beispiel zeigt.

Beispiel: (Runge 1901)
Wir interpolieren die Funktion $f(x) = \frac{1}{1+x^2}$ auf dem Intervall $[a, b] = [-5, 5]$ an den äquidistanten Stützstellen $x_k = kh$, $k = 0, \ldots, n$ zur Schrittweite $h = 10/n$ für $n = 10, 20$. Die Ergebnisse im Vergleich mit $f(x) = \frac{1}{1+x^2}$ (gestrichelt) sind in Abbildung 3.2 zu sehen.
Wir beobachten

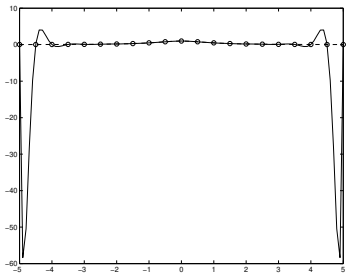
$$\|f - p_n\|_\infty \to \infty, \quad n \to \infty.$$ ◁

Wir betrachten nun die Folge von Interpolationsproblemen

$$p_n \in \mathcal{P}_n : \quad p_n(x_{nk}) = f(x_{nk}) \quad \forall k = 0, \ldots, n, \quad n \in \mathbb{N} \tag{3.13}$$



$n = 10$



$n = 20$

Abbildung 3.2: Divergenz des Interpolationspolynoms für $f(x) = \frac{1}{1+x^2}$

zu einer Folge von Stützstellen

$$a \leq x_{n0} < x_{n1} < \cdots < x_{n(n-1)} < x_{nn} \leq b, \quad n \in \mathbb{N} \tag{3.14}$$

und $f \in C[a,b]$. Bevor wir die Approximationseigenschaften der Polynominterpolation näher untersuchen, erinnern wir an ein weiteres Resultat aus dem CoMa–Skript.

Satz 3.7 *Der Interpolationsoperator*

$$C[a,b] \ni f \to \phi(f) = p_n \in \mathcal{P}_n$$

ist eine Projektion, d.h. ϕ ist linear und $\phi^2 = \phi$. Weiter gilt

$$\|\phi\|_\infty = \Lambda_n$$

mit der Lebesgue–Konstanten

$$\Lambda_n = \left\| \sum_{k=0}^{n} |L_k| \right\|_\infty = \max_{x\in[a,b]} \sum_{k=0}^{n} |L_k(x)|, \quad L_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{n} \frac{x - x_j}{x_k - x_j}, \quad k = 0, \ldots, n.$$

Als nächstes klären wir den Zusammenhang zwischen Interpolations– und Approximationsfehler.

Satz 3.8 *Es sei p_n^* Lösung der Bestapproximationsaufgabe* (2.12)*, also*

$$E_n(f) = \|f - p_n^*\|_\infty = \min_{q\in\mathcal{P}_n} \|f - q\|_\infty \quad f \in C[a,b]$$

der zugehörige Approximationsfehler und p_n die Lösung von (3.13)*. Dann gilt:*

$$\|f - p_n\|_\infty \leq (1 + \Lambda_n) E_n(f) \quad \forall f \in C[a,b].$$

Beweis:

$$\begin{aligned} \|f - p_n\|_\infty &\leq \|f - p_n^*\|_\infty + \|p_n - p_n^*\|_\infty \\ &= \|f - p_n^*\|_\infty + \|\phi(f - p_n^*)\|_\infty \\ &\leq (1 + \Lambda_n) E_n(f) \end{aligned}$$

Satz 3.8 besagt, daß der Interpolationsfehler maximal um den Faktor $1+\Lambda_n$ schlechter ist als der Approximationsfehler. Somit ist ein kleines Λ_n (hängt von den Stützstellen ab!) sowohl von der Stabilität als auch von der Approximationsgüte her wünschenswert.
Wir untersuchen die beiden Einflußfaktoren Λ_n und $E_n(f)$ im einzelnen. Zunächst die gute Nachricht:

Satz 3.9 (Weierstraß 1885) *Für jedes $f \in C[a,b]$ gilt*

$$E_n(f) \to 0 \quad n \to \infty.$$

Beweis:
Im Zentrum des Beweises stehen die *Bernstein–Polynome* (Bernstein 1912)

$$B_n(x) = \sum_{k=0}^{n} \binom{n}{k} x^k (1-x)^{n-k} f\left(\frac{k}{n}\right).$$

Es gilt nämlich

$$E_n(f) \leq \|B_n - f\|_\infty \to 0$$

für jedes $f \in C[a,b]$. Ein ausführlicher Beweis findet sich bei Hämmerlin und Hoffmann [4, Kapitel 4, §2.2]. □

Nun die schlechte Nachricht:

Satz 3.10 (Faber 1914) *Zu jeder Folge von Stützstellen* (3.14) *gibt es eine Funktion $f \in C[a,b]$, so daß gilt*

$$\liminf_{n\to\infty} \|f - p_n\|_\infty > 0.$$

Es kann also passieren, daß der Interpolationsfehler nicht gegen Null geht. Zusammen mit Satz 3.8 bedeutet das, daß für jede Folge von Stützstellen (3.14)

$$\Lambda_n \to \infty \quad n \to \infty$$

gelten muß. Es gibt genauere Aussagen:

Satz 3.11 (Brutman, de Boor, Pinkus 1978) *Für jede Folge von Stützstellen gilt*

$$\frac{2}{\pi}\log(n+1) + \frac{2}{\pi}\left(\gamma + \log\left(\frac{4}{\pi}\right)\right) \leq \Lambda_n.$$

Dabei bezeichnet $\gamma = \lim_{n\to\infty}(\sum_{i=1}^{n} \frac{1}{i} - \log(n))$ die Euler–Mascheroni–Konstante.

Die untere Schranke in Satz 3.11 lässt sich auch bei *bestmöglicher* Wahl von Stützstellen nicht unterschreiten. Es kann aber durchaus viel schlimmer kommen.

Satz 3.12 (Turetskii 1940) *Für die Folge äquidistanter Stützstellen $x_{nk} = a + kh$, $k = 0,\ldots,n$, zur Schrittweite $h = \frac{b-a}{n}$ gilt*

$$\Lambda_n \approx e^{-1}\frac{2^{n+1}}{n\log(n)}\ .$$

Λ_n *wächst also exponentiell.*

Das hat eine höchst praktische Konsequenz:

Folgerung 3.13 *Wenn irgend möglich, sind äquidistante Stützstellen zu vermeiden!*

Es stellt sich die Frage nach einer geschickten Wahl der Stützstellen. Sieht man vom Einfluß der Funktion f auf den Fehler ab, so legt es die Fehlerformel (3.6) nahe, x_{nk} so zu bestimmen, daß

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

mit

$$\omega_{n+1}(x) = \prod_{k=0}^{n}(x - x_{nk})$$

möglichst klein wird. Mit anderen Worten soll ω_{n+1} Lösung des Minimierungsproblems

$$\omega \in \mathcal{P}_{n+1}^{(1)}: \quad \|\omega\|_\infty \le \|q\|_\infty \quad \forall q \in \mathcal{P}_{n+1}^{(1)}$$

mit

$$\mathcal{P}_{n+1}^{(1)} = \{p \in \mathcal{P}_{n+1} \mid p = x^{n+1} - q,\ q \in \mathcal{P}_n\}$$

sein. Dieses Minimierungproblem kennen wir schon aus Abschnitt 2.2.1. Im Falle $[a,b] = [-1,1]$ ist die Lösung gerade das skalierte Tschebyscheff–Polynom 1. Art, also

$$\omega_{n+1} = \frac{1}{2^n} T_{n+1}, \quad T_n = \cos(n \arccos(x)), \quad n = 0, 1, \ldots$$

Die resultierenden Stützstellen sind gerade die Nullstellen von T_{n+1}, nämlich

$$x_{nk} = \cos\left(\frac{2(n-k)+1}{2(n+1)}\pi\right), \quad k = 0, \ldots, n.$$

Im Falle $[a,b] \neq [-1,1]$ erhält man die Tschebyscheff–Stützstellen durch lineare Transformation.

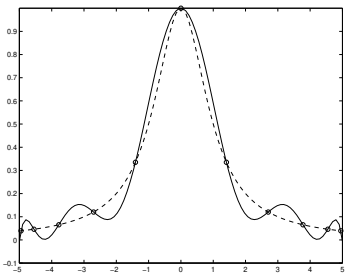
Satz 3.14 (Rivlin 1974) *Für die Tschebyscheff–Stützstellen gilt*

$$\frac{2}{\pi}\log(n+1) + \frac{2}{\pi}\left(\gamma + \log\left(\frac{8}{\pi}\right)\right) \le \Lambda_n \le 1 + \frac{2}{\pi}\log(n+1)$$

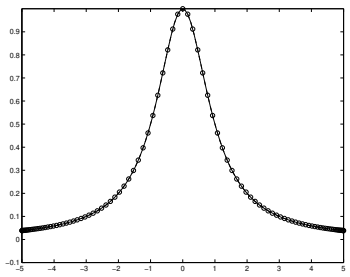
Bedenkt man, daß $\frac{2}{\pi}(\gamma + \log(\frac{8}{\pi})) \approx 0.9625$ gilt, so wird klar, daß die Tschebyscheff–Stützstellen tatsächlich nahezu optimal sind.

Beispiel:
Wir betrachten nochmals die Funktion $f(x) = \frac{1}{1+x^2}$ auf dem Intervall $[a,b] = [-5,5]$. Anstelle äquidistanter Stützstellen wählen wir aber diesmal die Tschebyscheff–Stützstellen

$$x_k = 5\cos\left(\frac{2(n-k)+1}{2(n+1)}\pi\right), \quad k = 0, \ldots, n.$$



$n = 10$



$n = 100$

Abbildung 3.3: Interpolation von $f(x) = \frac{1}{1+x^2}$ an Tschebyscheff–Stützstellen

Die entsprechenden Resultate für $n = 10, 100$ sind in Abbildung 3.3 dargestellt. Dabei ist $f(x) = \frac{1}{1+x^2}$ wieder gestrichelt gezeichnet.
Wir beobachten nun auf einmal Konvergenz der Polynominterpolation! ◁

3.2 Spline-Interpolation

Das Kürzel CAGD (*computer–aided graphic design*) steht für die Graphik–gestützte Entwicklung z.B. in der Autoindustrie oder im Schiffbau. Eine wichtige Teilaufgabe ist dabei die Interpolation vorgegebener Punkte durch eine glatte Kurve oder Fläche. Dasselbe Problem taucht bei einer Vielzahl weiterer graphischer Anwendungen, z.B. bei der Entwicklung von Video-Spielen, auf.
Wir haben gesehen, daß die Approximationsgüte der Polynominterpolation sehr stark von der Glattheit von f (siehe Abbildung 3.1) und der Lage der Stützstellen (siehe Abbildungen 3.2, 3.3) abhängt. Um eine robustere Approximation zu erreichen, betrachten wir nun *stückweise polynomiale Ansatzfunktionen.*

3.2.1 Stückweise lineare Interpolation

Die stückweise lineare Interpolation beruht auf der Wahl

$$U_n = \mathcal{S}_n = \{v \in C[a,b] \mid v|_{[x_{k-1},x_k]} = p_k,\ p_k \in \mathcal{P}_1\ \forall k = 1,\dots,n\}.$$

Die zugehörige Interpolationsaufgabe

$$u_n \in \mathcal{S}_n : \quad u_n(x_k) = f(x_k) \quad \forall k = 0,\dots,n$$

hat die Lösung

$$u_n = \sum_{k=0}^{n} f(x_k)\varphi_k.$$

Die Knotenbasis φ_k, $k = 0,\dots,n$, ist in (2.15) definiert.

Satz 3.15 *Es sei $a = x_0$, $x_n = b$ und $f \in C^2[a,b]$.*
Dann gilt die Fehlerabschätzung

$$\|f - u_n\|_\infty \le h^2\ \frac{\|f''\|_\infty}{8}, \tag{3.15}$$

wobei

$$h = \max_{k=1,\ldots,n} h_k, \quad h_k = x_k - x_{k-1},$$

gesetzt ist.

Beweis:
Die Fehlerformel (3.6) liefert für $k = 1, \ldots, n$

$$\begin{aligned} \max_{x\in[x_{k-1},x_k]} |f(x) - u_n(x)| &\leq \max_{x\in[x_{k-1},x_k]} \frac{|f''(x)|}{2} \max_{x\in[x_{k-1},x_k]} |(x - x_{k-1})(x - x_k)| \\ &= h_k^2 \max_{x\in[x_{k-1},x_k]} \frac{|f''(x)|}{8} \end{aligned}$$

und damit die Behauptung. □

Bemerkung:
Offenbar gilt

$$\min_{v\in\mathcal{S}_n} \|f - v\|_\infty \leq \|f - u_n\|_\infty = \mathcal{O}(h^2).$$

Der Approximationsfehler von $\mathcal{S}_n$ verhält sich also wie $\mathcal{O}(h^2)$. ◁

Beachte, daß die Polynominterpolation für $f \in C^2[a, b]$ nicht notwendig konvergiert, geschweige denn einer Fehlerabschätzung der Form (3.15) genügt. Die größere Robustheit stückweise linearer Approximation bestätigt auch folgendes Beispiel.

Beispiel:
Ein Schiffbauer möchte die Spanten unbedingt im äquidistanten Abstand anbringen. Um die Form des Schiffes festzulegen, gilt es, die folgenden Punkte (x_k, f_k) zu interpolieren.

x_k	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
f_k	0	0.3	0.7	0.88	0.96	0.98	0.99	0.999	1	1	1

Polynominterpolation liefert die befürchteten Überschwinger, während bei der stückweise linearen Interpolation nichts davon zu sehen ist (Abbildung 3.4). Im Gegensatz zur Polynominterpolation zeigen sich allerdings unerwünschte Ecken an den Stützstellen. Erwünscht sind aber glatte Übergänge wie zum Beispiel in Abbildung 3.5. Die Verbindung von robustem Interpolationsverhalten *und* glatten Übergängen erreicht man durch sogenannte *Spline–Interpolation.* ◁

3.3 Kubische Spline–Interpolation

Mit Blick auf Satz 3.15 gehen wir von nun an davon aus, daß das Gitter

$$\Delta = \{x_0, \ldots, x_n \mid a \leq x_0 < \ldots < x_n \leq b\}$$

den Bedingungen

$$a = x_0, \quad x_n = b$$

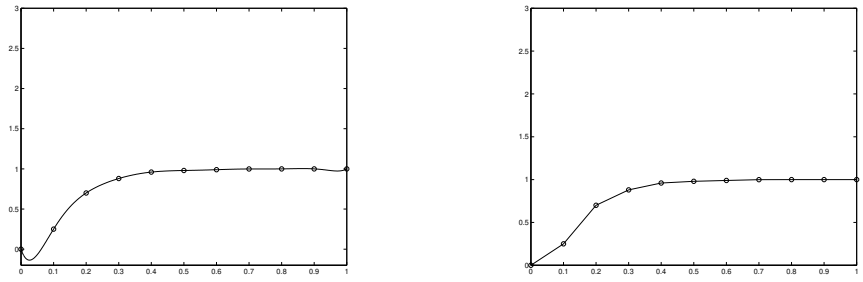


Abbildung 3.4: Überschwinger oder Ecken

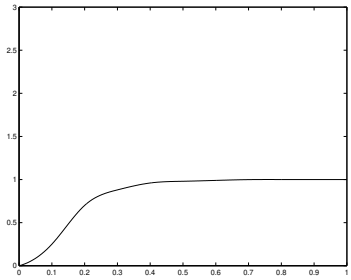


Abbildung 3.5: Kubische Spline–Interpolation

genügt. Wir definieren

$$I_k = [x_{k-1}, x_k] \ , \quad h_k = x_k - x_{k-1}, \quad k = 1, \ldots, n.$$

Um glatte Übergänge zu gewährleisten, bauen wir *die Stetigkeit von Ableitungen* in die Ansatzfunktionen ein.

Definition 3.16 *Die Splines m-ter Ordnung zum Gitter* Δ *sind gegeben durch*

$$\mathcal{S}_\Delta^m = \left\{ v \in C^{m-1}[a,b] \mid v|_{I_k} = p_k \in \mathcal{P}_m \right\}. \tag{3.16}$$

Im Falle $m = 3$ *spricht man von kubischen Splines.*

Bemerkung:
$\mathcal{S}_\Delta^m$ ist ein linearer Raum über $\mathbb{R}$. ◁

Beispiel:

- Es gilt
$$\mathcal{S}_\Delta^1 = \mathcal{S}_n.$$

- Es sei $\Delta = \{x_0 = -1, x_1 = 0, x_2 = 1\}$ und
$$v(x) = \begin{cases} p_1(x) = -x(x+1) & \text{für } x \in I_1 = [x_0, x_1] \\ p_2(x) = +x(x-1) & \text{für } x \in I_2 = [x_1, x_2] \end{cases}.$$
Dann ist $v \in \mathcal{S}_\Delta^2$. ◁

Wir betrachten die Interpolationsaufgabe

$$s_n \in \mathcal{S}_\Delta^m : \quad s_n(x_k) = f(x_k) \quad \forall k = 0, \ldots, n. \tag{3.17}$$

Bemerkung:
Jede Funktion $v \in \mathcal{S}_\Delta^m$ ist durch n Polynome m-ten Grades p_k, $k = 1, \ldots, n$, mit insgesamt $n(m+1)$ Koeffizienten festgelegt. Aus $v \in C^{m-1}$ erhält man die $(n-1)m$ *Übergangsbedingungen*

$$p_k^{(j)}(x_k) = p_{k+1}^{(j)}(x_k) \quad j = 0, \ldots, m-1, \quad k = 1, \ldots, n-1.$$

Die resultierenden Gleichungen für die Koeffizienten von p_k, $k = 1, \ldots, n-1$, sind linear unabhängig (Übung). Damit ist

$$\dim \mathcal{S}_\Delta^m = n(m+1) - (n-1)m = n + m.$$

Die Lösung $s_n \in \mathcal{S}_\Delta^m$ der Interpolationsaufgabe (3.17) genügt zusätzlich noch den $n+1$ Interpolationsbedingungen. Im Falle $m > 1$ ist die Lösung von (3.17) wegen

$$\dim \mathcal{S}_\Delta^m - (n+1) = m - 1$$

also *nicht* eindeutig bestimmt. Wir können noch $m-1$ zusätzliche Bedingungen stellen. ◁

Wir beschreiben nun einen überraschenden Zusammenhang zwischen Spline–Interpolation und Kontinuumsmechanik, der auf Schoenberg (1946) zurückgeht. Im Falle kubischer Splines ($m = 3$) kann man nämlich die zusätzlichen $m - 1 = 2$ Freiheitsgrade so wählen, daß $s_n \in \mathcal{S}_\Delta^3$ nicht nur die Interpolationsaufgabe (3.17), sondern gleichzeitig noch ein Minimierungsproblem löst.

Motivation:
Spline bedeutet auf englisch „längliches Stück Holz oder Metall“. In der Mechanik heißen solche Gegenstände *Balken*. Befestigt man einen Balken drehbar in den Punkten $(x_k, f(x_k))$ und fixiert ihn in $x_0 = a$ bzw. $x_n = b$ in Richtung $f'(a)$ bzw. $f'(b)$, so stellt sich eine Biegelinie

$$\psi \in C_\Delta^2[a,b] = \{v \in C^2[a,b] \mid v(x_k) = f(x_k)\ \forall k = 0, \ldots, n\}$$

mit den Randbedingungen

$$\psi'(a) = f'(a), \quad \psi'(b) = f'(b) \tag{3.18}$$

ein. Dabei wird die aufgenommene Energie

$$\mathcal{E}(\psi) = \int_a^b \frac{(\psi'')^2}{(1 + (\psi')^2)^3}\, dx$$

unter allen möglichen Auslenkungen $v \in H$,

$$H = \{v \in C_\Delta^2[a,b] \mid v'(a) = f'(a),\ v'(b) = f'(b)\},$$

minimal. Also löst ψ das Minimierungsproblem

$$\psi \in H : \quad \mathcal{E}(\psi) \le \mathcal{E}(v) \quad \forall v \in H. \tag{3.19}$$

Unter der Voraussetzung *kleiner Verzerrungen* $\psi' \approx 0$ ist

$$\mathcal{E}(\psi) \doteq \int_a^b (\psi'')^2 \, dx = \|\psi''\|_2^2.$$

Damit erhält das Minimierungsproblem (3.19) die Gestalt

$$\psi \in H: \quad \|\psi''\|_2 \le \|v''\|_2 \quad \forall v \in H. \tag{3.20}$$

◁

Wir beginnen mit einer wichtigen Extremaleigenschaft interpolierender, kubischer Splines.

Lemma 3.17 *Es sei* $s_n \in \mathcal{S}_\Delta^3$ *Lösung von* (3.17) *und* $v \in C_\Delta^2[a,b]$. *Dann folgt aus der Bedingung*

$$s_n''(x)\big(v'(x) - s_n'(x)\big)\Big|_a^b = 0 \tag{3.21}$$

die Abschätzung

$$\|s_n''\|_2 \le \|v''\|_2$$

Beweis:
Wir zeigen, daß unter der Voraussetzung (3.21) die Orthogonalität

$$(s_n'', v'' - s_n'') = 0 \tag{3.22}$$

bezüglich des L^2–Skalarprodukts

$$(v, w) = \int_a^b v(x)w(x) \, dx$$

vorliegt. Aus (3.22) ergibt sich die Behauptung dann nämlich sofort mit dem Satz des Pythagoras:

$$\|v''\|_2^2 = \|s_n''\|_2^2 + \|v'' - s_n''\|_2^2 \ge \|s_n''\|_2^2.$$

Um (3.22) zu beweisen, berechnen wir

$$\begin{aligned}
\int_a^b s_n''(v'' - s_n'') \, dx &= \sum_{k=1}^n \int_{x_{k-1}}^{x_k} s_n''(x)(v'' - s_n'') \, dx \\
&= \sum_{k=1}^n \left(s_n''(v' - s_n')\Big|_{x_{k-1}}^{x_k} - \int_{x_{k-1}}^{x_k} s_n'''(v' - s_n') \, dx \right) \\
&= s_n''(v' - s_n')\Big|_a^b - \sum_{k=1}^n d_k \int_{x_{k-1}}^{x_k} (v' - s_n') \, dx,
\end{aligned}$$

wobei $d_k = s_n'''|_{I_k} = \text{const.}$ gesetzt ist. Weiter folgt

$$\begin{aligned}
\int_a^b s_n''(v'' - s_n'') \, dx &= s_n''(v' - s_n')\Big|_a^b - \sum_{k=1}^n d_k (v - s_n)\Big|_{x_{k-1}}^{x_k} \\
&= s_n''(v' - s_n')\Big|_a^b = 0,
\end{aligned}$$

denn $v(x_k) = s(x_k)$, $k = 0, \ldots, n$. □

Wir können nun unser angestrebtes Existenz– und Eindeutigkeitsresultat formulieren.

Satz 3.18 *Es existiert genau eine Lösung $s_n \in S^3_\Delta$ der Interpolationsaufgabe* (3.17) *mit den*

$$\textit{vollständigen Randbedingungen:} \quad s_n'(a) = f'(a) \quad s_n'(b) = f'(b). \tag{3.23}$$

Darüberhinaus ist s_n Lösung von (3.20)*, d.h. $s_n \in H$ und*

$$\|s_n''\|_2 \leq \|v''\|_2 \quad \forall v \in H. \tag{3.24}$$

Beweis:
Wir beginnen mit der Eindeutigkeit und zeigen zunächst, daß im Falle

$$f'(a) = f(x_0) = f(x_1) = \cdots = f(x_n) = f'(b) = 0 \tag{3.25}$$

die Funktion $s_n \equiv 0$ die einzige Lösung ist. Wegen $v \equiv 0 \in C^2_\Delta[a,b]$ und $v'(x) - s_n'(x)|_a^b = 0$ folgt aus Lemma 3.17

$$\|s_n''\|_2 \leq \|v''\|_2 = 0 \,.$$

Also ist $s_n'' = 0$ und s_n somit linear. Da aber $s_n(a) = s_n'(a) = 0$ gilt, muss $s_n \equiv 0$ sein. Nun zeigen wir die Eindeutigkeit im allgemeinen Fall. Seien s_n, $\tilde{s}_n$ Lösungen. Dann ist $s_n - \tilde{s}_n$ Lösung zu den homogenen Daten (3.25). Das bedeutet aber $s_n - \tilde{s}_n \equiv 0$.
Wir kommen nun zur Existenz einer Lösung. Aus der eben gezeigten Eindeutigkeit folgt die Injektivität der lineare Abbildung $A : \mathcal{S}^3_\Delta \to \mathbb{R}^{n+3}$, definiert durch

$$\mathcal{S}^3_\Delta \ni s_n \to As_n = (s_n'(a), s_n(x_0), s_n(x_1), \ldots, s_n(x_n), s_n'(b)) \in \mathbb{R}^{n+3}.$$

Wegen $\dim \mathcal{S}^3_\Delta = n+3$ muß A nach einem Satz aus der linearen Algebra auch surjektiv sein. Das ist aber gerade die Existenz.
Es fehlt noch die Extremaleigenschaft von s_n. Da $s_n \in \mathcal{S}^3_\Delta \subset C^2[a,b]$ die Interpolationsbedingungen nebst Randbedingungen (3.23) erfüllt, ist $s_n \in H$. Die Extremaleigenschaft (3.24) folgt dann aus Lemma 3.17, da man (3.21) direkt aus den Randbedingungen (3.18) und (3.23) erschließt. □

Bemerkung:
Die Aussagen von Satz 3.18 bleiben gültig, wenn man die Randbedingungen (3.18) und (3.23) an $v = \psi, s_n$ entweder durch

$$\text{natürliche Randbedingungen:} \quad v''(a) = v''(b) = 0 \tag{3.26}$$

oder, falls f periodisch mit Periode $b - a$ ist, durch

$$\text{periodische Randbedingungen:} \quad v'(a) = v'(b) \quad v''(a) = v''(b) \tag{3.27}$$

ersetzt (Übung). ◁

3.3.1 Berechnung der vollständigen kubischen Splineinterpolation

Zu jeder Funktion $f \in C^2[a,b]$ gibt es nach Satz 3.18 genau eine Funktion $s_n = \phi_n(f) \in \mathcal{S}_\Delta^3$, welche die Interpolationsaufgabe (3.17) löst und vollständigen Randbedingungen (3.23) genügt. Damit ist der Spline-Interpolationsoperator

$$C^2[a,b] \ni f \to \phi_n(f) \in \mathcal{S}_\Delta^3$$

wohldefiniert.

Bemerkung:
ϕ_n ist eine Projektion.
Aus (3.24) folgt die Extremaleigenschaft

$$\|(\phi_n f)''\|_2 \le \|f''\|_2 \quad \forall f \in C^2[a,b]. \tag{3.28}$$

◁

Zur Berechnung von $s_n = \phi_n(f)$ machen wir den folgenden Taylor–Ansatz für die Polynome auf den einzelnen Intervallen $I_{k+1} = [x_k, x_{k+1}]$

$$s_n(x) = a_k + b_k(x-x_k) + \frac{c_k}{2!}(x-x_k)^2 + \frac{d_k}{3!}(x-x_k)^3, \qquad x \in I_{k+1}, \quad k = 0,\dots,n-1. \tag{3.29}$$

Wir haben die unbekannten Koeffizienten a_k, b_k, c_k und d_k des Polynomes p_{k+1} zu berechnen.

Satz 3.19 *Die zweiten Ableitungen $s_n''(x_k)$, $k = 0,\dots,n$, seien bekannt. Setzt man $c_n = s_n''(x_n)$, so gilt*

$$a_k = f(x_k), \quad k = 0,\dots,n-1, \tag{3.30}$$

$$c_k = s_n''(x_k), \quad k = 0,\dots,n, \tag{3.31}$$

$$d_k = \frac{c_{k+1} - c_k}{h_{k+1}}, \quad k = 0,\dots,n-1 \tag{3.32}$$

und rekursiv erhält man

$$b_0 = f'(a), \quad b_{k+1} = b_k + \frac{1}{2}(c_{k+1} + c_k)h_{k+1}, \quad k = 0,\dots,n-2. \tag{3.33}$$

Beweis:
Die Bestimmungsgleichung (3.30) folgt direkt aus den Interpolationsbedingungen. Gleichung (3.31) folgt ebenfalls direkt durch Einsetzen. Die Stetigkeit von s_n'' in x_{k+1}, $k = 0,\dots,n-2$, liefert

$$c_k + d_k h_{k+1} = s_n''|_{I_{k+1}}(x_{k+1}) = s_n''|_{I_{k+2}}(x_{k+1}) = c_{k+1}, \quad k = 0,\dots,n-2.$$

und es gilt ja

$$c_n = s_n''(x_n) = c_{n-1} + d_{n-1}h_n.$$

Damit ist (3.32) gezeigt. Ebenso erhalten wir aus der Stetigkeit von s_n' in x_{k+1} sofort

$$b_k + c_k h_{k+1} + \tfrac{1}{2} d_k h_{k+1}^2 = s_n'|_{I_{k+1}}(x_{k+1}) = s_n'|_{I_{k+2}}(x_{k+1}) = b_{k+1}, \quad k = 0, \ldots, n-2.$$

Einsetzen von (3.32) liefert die Rekursionsvorschrift (3.33) und $b_0 = f'(a)$ folgt durch Einsetzen der linken Randbedingung. □

Unser nächstes Ziel ist es also, eine Berechnungsvorschrift für die zweiten Ableitungen $s_n''(x_k)$ zu finden. Dazu definieren wir eine lineare Abbildung

$$P : C[a,b] \to S_n = \{v \in C[a,b] \mid v \text{ ist linear auf } I_k \; \forall k = 1, \ldots, n\}$$

auf folgende Weise. Sei $g \in C[a,b]$ gegeben.

$$\text{Wähle } w \in C^2[a,b] \text{ mit } g = w''. \quad \text{Setze } Pg = (\phi_n w)''. \tag{3.34}$$

Seien $w_1, w_2 \in C^2[a,b]$ mit $w_1'' = w_2'' = g$. Dann ist $w_1 - w_2$ linear und daher

$$Pw_1'' - Pw_2'' = P(w_1'' - w_2'') = (\phi_n(w_1 - w_2))'' = (w_1 - w_2)'' = 0.$$

Damit ist Pg unabhängig von der Wahl von w. P ist also wohldefiniert.

Bemerkung:
Für alle $f \in C^2[a,b]$ gilt

$$P(f'') = (\phi_n f)'' = s_n''.$$

Um die $s_n''(x_k)$ zu erhalten, konzentrieren uns also nun darauf, P auszuwerten. ◁

Wegen

$$P^2 g = P(Pg) = P\big((\phi_n w)''\big) = \big(\phi_n(\phi_n w)\big)'' = (\phi_n w)'' = Pg$$

ist P eine Projektion. Nach Satz 2.9 ist $\|P\|_2 \geq 1$.
Aus der Extremaleigenschaft (3.28) folgt

$$\|Pg\|_2 = \|(\phi_n w)''\|_2 \leq \|w''\|_2 = \|g\|_2$$

also

$$\|P\|_2 = \max_{\substack{g \in C[a,b] \\ g \neq 0}} \frac{\|Pg\|_2}{\|g\|_2} \leq 1.$$

Insgesamt gilt $\|P\|_2 = 1$. Nach Satz 2.9 ist P damit eine *Orthogonalprojektion*. Aus Satz 2.8 folgt weiter, daß dann $u = Pg$ Lösung der *Bestapproximationsaufgabe*

$$u \in \mathcal{S}_n : \quad \|u - g\|_2 \leq \|v - g\|_2 \quad \forall v \in \mathcal{S}_n$$

ist. Die Berechnung der Lösung dieses Problems haben wir in Abschnitt 2.2.2 ausführlich behandelt. Zur Berechnung von $P(f'')$ machen wir wieder den Ansatz

$$Pf'' = s_n'' = \sum_{k=0}^{n} c_k \varphi_k.$$

Die Knotenbasis $\{\varphi_k\}$ von $\mathcal{S}_n$ ist in (2.16) definiert. Dann können wir die unbekannten Koeffizienten

$$c = \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix}$$

aus dem linearen Gleichungssystem (vgl. (2.17))

$$Mc = \beta \tag{3.35}$$

mit Koeffizientenmatrix M (vgl. (2.18))

$$M = \begin{pmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \lambda_{n-1} \\ & & & \mu_n & 2 \end{pmatrix} \qquad \mu_i = \frac{h_i}{h_i + h_{i+1}}, \quad \lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}}$$

und rechter Seite β (vgl. (2.19))

$$\beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_n \end{pmatrix}, \qquad \beta_i = \frac{6}{h_i + h_{i+1}}(f'', \varphi_i).$$

berechnen. Die rechte Seite kann man noch weiter ausrechnen. Für $i = 1, \ldots, n-1$ erhält man

$$\begin{aligned}
(f'', \varphi_i) &= \int_{x_{i-1}}^{x_{i+1}} f''(x)\varphi_i(x)\,dx \\
&= f'\varphi_i\Big|_{x_{i-1}}^{x_{i+1}} - \int_{x_{i-1}}^{x_i} f'(x)\varphi_i'(x)\,dx - \int_{x_i}^{x_{i+1}} f'(x)\varphi_i'(x)\,dx \\
&= -\frac{1}{h_i}\Big(f(x_i) - f(x_{i-1})\Big) + \frac{1}{h_{i+1}}\Big(f(x_{i+1}) - f(x_i)\Big) \\
&= f[x_i, x_{i+1}] - f[x_i, x_{i-1}] \\
&= (h_i + h_{i+1}) f[x_{i-1}, x_i, x_{i+1}]
\end{aligned}$$

Außerdem ist mit $x_{-1} = x_0$ und $h_0 = 0$

$$\begin{aligned}
(f'', \varphi_0) &= \int_{x_0}^{x_1} f''(x)\varphi_0(x)\,dx = f'\varphi_0\Big|_{x_0}^{x_1} - \int_{x_0}^{x_1} f'(x)\varphi_0'(x)\,dx \\
&= -f'(x_0) + \frac{1}{h_1}\big(f(x_1) - f(x_0)\big) = -f'(x_0) + f[x_0, x_1] \\
&= (h_0 + h_1) f[x_{-1}, x_0, x_1].
\end{aligned}$$

Analog berechnet man mit $x_{n+1} = x_n$ und $h_{n+1} = 0$

$$(f'', \varphi_n) = (h_n + h_{n+1}) f[x_{n-1}, x_n, x_{n+1}].$$

Insgesamt erhalten wir also

$$\beta_i = 6f[x_{i-1}, x_i, x_{i+1}], \quad i = 0, \ldots, n.$$

Bemerkung:
Das Gleichungssystem (3.35) kann mittels LR–Zerlegung mit optimalem Aufwand gelöst werden. Es gilt $\kappa_\infty(M) \leq 3$ (vgl. Satz 2.13). ◁

Obwohl wir unsere Aufgabe, eine Berechnungsvorschrift für die unbekannten Koeffizienten a_k, b_k, c_k und d_k in (3.29) anzugeben damit erledigt haben, bleibt noch ein kleiner Schönheitsfehler: Bei der rekursiven Berechnung der b_k können sich Rundungsfehler aufschaukeln. Das lässt sich vermeiden.

Satz 3.20 *Die Lösung der Rekursion* (3.33) *ist*

$$b_k = f[x_k, x_{k+1}] - h_{k+1}\frac{2c_k + c_{k+1}}{6}, \quad k = 0, \ldots, n-1. \tag{3.36}$$

Beweis:
Wir führen den Beweis mit vollständiger Induktion über k.
Induktionsanfang.
Die erste Gleichung von (3.35) ist

$$2c_0 + c_1 = 6f[x_{-1}, x_0, x_1] = \frac{6}{h_1}(f'(a) + f[x_0, x_1]).$$

Daraus erschließt man

$$f[x_0, x_1] - h_1\frac{2c_0 + c_1}{6} = f'(a) = b_0.$$

Induktionsvoraussetzung.
Wir nehmen an, daß Formel (3.36) für ein $k \geq 0$ richtig ist.
Induktionsschluß.
Ausschreiben der $(k+1)$–ten Gleichung von (3.35) liefert nach Multiplikation mit $h_{k+1}+h_{k+2}$ und Division durch 6

$$\tfrac{1}{6}h_{k+1}c_k + \tfrac{1}{3}(h_{k+1} + h_{k+2})c_{k+1} + \tfrac{1}{6}h_{k+2}c_{k+2} = f[x_{k+1}, x_{k+2}] - f[x_k, x_{k+1}].$$

Umordnen ergibt

$$f[x_{k+1}, x_k] - h_{k+2}\frac{2c_{k+1} + c_{k+2}}{6} = \frac{1}{2}(c_{k+1} + c_k)h_{k+1} + \left(f[x_k, x_{k+1}] - h_{k+1}\frac{2c_k + c_{k+1}}{6}\right).$$

Durch Einsetzen der Induktionsvoraussetzung erhält man

$$f[x_{k+1}, x_k] - h_{k+2}\frac{2c_{k+1} + c_{k+2}}{6} = \frac{1}{2}(c_{k+1} + c_k)h_{k+1} + b_k = b_{k+1}$$

und damit die Behauptung. □

3.3.2 Approximationseigenschaften vollständiger kubischer Splines

Wir wollen nun den *Approximationsfehler*

$$\|f - \phi_n f\|_\infty$$

abschätzen. Dabei wird uns die in (3.34) definierte Projektion $P : C[a,b] \to \mathcal{S}_n$ wieder gute Dienste leisten. Wir wissen schon, daß $\|P\|_2 = 1$ gilt. Um Abschätzungen in der Norm $\|\cdot\|_\infty$ zu gewinnen, benötigen wir aber Kenntnis von $\|P\|_\infty$.

Lemma 3.21 *Es gilt*

$$\|P\|_\infty = \sup_{\substack{g \in C[a,b] \\ g \neq 0}} \frac{\|Pg\|_\infty}{\|g\|_\infty} \leq 3. \tag{3.37}$$

Ist $g \in C^2[a,b]$, so genügt P der Fehlerabschätzung

$$\|g - Pg\|_\infty \leq h^2 \frac{\|g''\|_\infty}{2} \quad h = \max_{k=1,\ldots,n} h_k. \tag{3.38}$$

Beweis:
Es sei $g \in C[a,b]$, $w \in C^2[a,b]$, $w'' = g$ und

$$Pg = (\phi_n w)'' = \sum_{k=0}^{n} c_k \varphi_k.$$

Nach Definition ist

$$\|Pg\|_\infty = \max_{x \in [a,b]} \left| \sum_{k=0}^{n} c_k \varphi_k(x) \right| \leq \max_{k=0,\ldots,n} |c_k| = \|c\|_\infty.$$

Zur Vereinfachung der Schreibweise setzen wir $c_{-1} = c_{n+1} = 0$. Die k–te Gleichung von $Mc = \beta$ (vgl. (3.35)) lautet dann

$$\mu_k c_{k-1} + 2c_k + \lambda_k c_{k+1} = \beta_k, \quad k = 0, \ldots, n.$$

Umordnen ergibt wegen $|\mu_k| + |\lambda_k| = \mu_k + \lambda_k = 1$

$$2|c_k| \leq |\beta_k| + |\mu_k||c_{k-1}| + |\lambda_k||c_{k+1}| \leq \|\beta\|_\infty + \|c\|_\infty$$

und damit

$$\|c\|_\infty \leq \|\beta\|_\infty.$$

Nun ist aber nach Satz 3.5

$$|\beta_k| = 6f|[x_{k-1}, x_k, x_{k+1}]| = 6\frac{|f''(\xi)|}{2!} \leq 3\|f''\|_\infty = 3\|g\|_\infty, \quad k = 0, \ldots, n.$$

Insgesamt folgt also

$$\|Pg\|_\infty \leq \|c\|_\infty \leq \|\beta\|_\infty \leq 3\|g\|_\infty$$

und damit die Abschätzung (3.37).
Wir kommen zur Fehlerabschätzung (3.38). Es sei $g \in C^2[a,b]$ und

$$u_n = \sum_{k=0}^{n} g(x_k)\varphi_k \in \mathcal{S}_n$$

die stückweise lineare Interpolation von g. Aus $u_n \in \mathcal{S}_n$ folgt $Pu_n = u_n$. Aus Satz 3.15 kennen wir die Fehlerabschätzung

$$\|g - u_n\|_\infty \leq h^2 \frac{\|g''\|_\infty}{8}.$$

In Verbindung mit (3.37) liefert schließlich die Dreiecksungleichung

$$\|g - Pg\|_\infty \leq \|g - u_n\|_\infty + \|P(u_n - g)\|_\infty \leq 4\|g - u_n\|_\infty \leq h^2 \frac{\|g''\|_\infty}{2}. \qquad \square$$

Nun sind wir soweit.

Satz 3.22 *Es sei $f \in C^4[a,b]$. Dann gilt die Fehlerabschätzung*

$$\|f - \phi_n f\|_\infty \leq \frac{h^4}{16}\|f^{(4)}\|_\infty.$$

Beweis:
Wir setzen $g = f''$ und $e = f - \phi_n f$. Dann erhält man unter Verwendung von Lemma 3.21

$$\|e''\|_\infty = \|f'' - (\phi_n f)''\|_\infty = \|f'' - P(f'')\|_\infty \leq h^2 \frac{\|f^{(4)}\|_\infty}{2}. \tag{3.39}$$

Unser Ziel ist es nun $\|e\|_\infty$ durch $\|e''\|_\infty$ abzuschätzen. Sei $x \in [x_k, x_{k+1}]$. Das Polynom p_2 interpoliere die Funktion e in den Stützstellen x_k, x, x_{k+1}. Die Newton'sche Darstellung von p_2 ist

$$p_2(z) = e[x_k] + e[x_k, x_{k+1}](z - x_k) + e[x, x_k, x_{k+1}](z - x_k)(z - x_{k+1}),$$

wobei

$$e[x_k] = f(x_k) - \phi_n f(x_k) = 0 \quad \text{und} \quad e[x_{k+1}] = f(x_{k+1}) - \phi_n f(x_{k+1}) = 0$$

und daher auch

$$e[x_k, x_{k+1}] = \frac{1}{h_{k+1}}\big(e[x_{k+1}] - e[x_k]\big) = 0.$$

p_2 interpoliert e an der Stelle $z = x$ und es gilt daher

$$e(x) = p_2(x) = e[x, x_k, x_{k+1}](x - x_k)(x - x_{k+1}).$$

Aus Satz 3.5 erhalten wir die Existenz von $\xi(x) \in [x_k, x_{k+1}]$ mit

$$e(x) = \frac{1}{2!}e''(\xi(x))(x - x_k)(x - x_{k+1}).$$

Mit (3.39) folgt daraus für alle $k = 0, \ldots, n-1$

$$\max_{x\in[x_k,x_{k+1}]} |e(x)| \leq \frac{1}{2}\|e''\|_\infty \max_{x\in[x_k,x_{k+1}]} |(x-x_k)(x-x_{k+1})| \leq \; h_{k+1}^2 \frac{\|e''\|_\infty}{8} \leq h_{k+1}^4 \frac{\|f^{(4)}\|_\infty}{16}$$

und damit die Behauptung. □

Bemerkung:
Bei Hall (1968) findet sich die schärfere Abschätzung

$$\|f - \phi_n f\|_\infty \leq \frac{5}{384} h^4 \|f^{(4)}\|_\infty. \qquad \triangleleft$$

Bemerkung:
Ist $f \in C^4[a,b]$ periodisch mit Periode $b-a$, so gilt für die kubische Spline–Interpolation mit *periodischen Randbedingungen* die Fehlerabschätzung

$$\|f - \phi_n f\|_\infty \leq \frac{h^4}{16} \|f^{(4)}\|_\infty.$$

Da das Randverhalten von f bei der *natürlichen Spline–Interpolation* nicht berücksichtigt wird, hat man in diesem Fall nur

$$\|f - \phi_n f\|_\infty = \mathcal{O}(h^2). \qquad \triangleleft$$

Literatur

[1] C. de Boor. *A Practical Guide to Splines.* Springer, 1978. Die klassische Monographie zur Polynom– und Spline–Interpolation.

[2] P. Deuflhard and A. Hohmann. *Numerische Mathematik I.* de Gruyter, 4. Auflage, 2008. Zur Vertiefung des Stoffes lohnt es Kapitel 7 anzuschauen. Neben bekannten Dingen, wie der Hermite–Genocchi-Formel, findet man neue Stichworte, wie Bézier–Techniken oder Trigonometrische Interpolation.

[3] R.A. DeVore and G.G. Lorentz. *Constructive Approximation.* Springer, 1993. Eine umfassende Darstellung der Approximationstheorie für Funktionen einer reellen Variablen von derzeit führenden Experten. Aber Vorsicht: Nur für Hartgesottene, die sich dafür interessieren, wohin es weitergeht. Beispielsweise führt die Frage nach der bestmöglichen Approximation einer Funktion f mit genau n Elementen aus $\bigcup_{k=1}^{\infty} \mathcal{S}_k$ auf sogenannte *Besov–Räume*. Das sind Funktionenräume, bei denen die meisten aussteigen, für die Sobolev–Räume noch das Normalste der Welt sind. Um Sobolev–Räume geht es aber erst im nächsten Semester...

[4] G. Hämmerlin and K.-H. Hoffmann. *Numerische Mathematik.* Springer, 4. Auflage, 2004. In Kapitel 5 findet man eine ausführliche Darstellung der Polynominterpolation nebst weiteren Verbindungen zur Analysis und interessanten historischen Kommentaren.

4 Numerische Quadratur

Unsere Aufgabe ist die Berechnung des Integrals

$$I(f) = \int_a^b f(x)\, dx, \quad f \in C[a,b], \tag{4.1}$$

mit $a, b \in \mathbb{R}$, $a < b$. Offenbar ist $I : C[a,b] \to \mathbb{R}$ eine lineare Abbildung. Wegen

$$\|I\|_\infty = \sup_{\substack{f \in C[a,b] \\ f \neq 0}} \frac{|I(f)|}{\|f\|_\infty} \leq (b-a)$$

ist I beschränkt. Wir untersuchen die relative Kondition von I.

Satz 4.1 *Für alle $f, \Delta f \in C[a,b]$ mit $I(f) \neq 0$ gilt*

$$\frac{|I(f) - I(f+\Delta f)|}{|I(f)|} \leq \kappa(I(f)) \frac{\|\Delta f\|_\infty}{\|f\|_\infty}$$

mit

$$\kappa(I(f)) = (b-a)\frac{\|f\|_\infty}{|I(f)|}.$$

Beweis:
Die Behauptung folgt aus der Linearität von I. □

Beispiel:
Für jedes $n = 0, 1, \ldots$ hat die Funktion

$$f_n(x) = \frac{(2n+1)\pi}{2} \sin\big((2n+1)\pi x\big)$$

die Eigenschaft

$$I(f_n) = \frac{(2n+1)\pi}{2} \int_0^1 \sin\big((2n+1)\pi x\big)\, dx = 1.$$

Andererseits gilt

$$\|f_n\|_\infty = \frac{(2n+1)\pi}{2}.$$

Insgesamt folgt

$$\kappa(I(f_n)) = \frac{(2n+1)\pi}{2} \to \infty \quad \text{für} \quad n \to \infty. \qquad \triangleleft$$

Zur Approximation von $I(f)$ wollen wir Quadraturformeln

$$I_N(f) \approx I(f)$$

konstruieren. Die Anzahl N der zur Berechnung von $I_N(f)$ benötigten f–Auswertungen ist das Maß für den *Aufwand*,

$$\text{Aufwand von } I_N(f) = \text{Anzahl der } f\text{–Auswertungen} = N.$$

Wir erwarten, daß $I(f)$ mit wachsendem Aufwand beliebig genau approximiert wird, also

$$\lim_{N\to\infty} I_N(f) = I(f).$$

Die Quadraturformel ist von *q–ter Ordnung*, falls

$$|I(f) - I_N(f)| = \mathcal{O}(N^{-q}).$$

Darüberhinaus sind wir an möglichst *effizienten* Verfahren interessiert. Es soll also

$$\begin{aligned} \text{Effizienz von } I_N(f) &= \text{Genauigkeit von } I_N(f)/\text{Aufwand von } I_N(f) \\ &= |I(f) - I_N(f)|^{-1}/N \end{aligned}$$

möglichst groß sein. Beachte, daß die Effizienz von f abhängt. Eine gegebene Quadraturformel kann also für die eine Funktion f effizient sein und für die andere nicht.
In der CoMa haben wir bereits eine Strategie kennengelernt, wie man Quadraturformeln konstruieren kann. Ausgehend von dem Gitter

$$\Delta = \{a = z_0 < z_1 < \cdots < z_{m-1} < z_m = b\}, \qquad V_k = [z_{k-1}, z_k],$$

zerlegen wir zunächst das Integral

$$I(f) = \sum_{k=1}^{m} \int_{V_k} f(x)\, dx.$$

Auf jedem Teilintervall approximieren wir f durch das Interpolationspolynom $p_{nk} \in \mathcal{P}_n$,

$$p_{nk} \in \mathcal{P}_n : \quad p_{nk}(x_{ik}) = f(x_{ik}) \quad \forall i = 0, \ldots, n,$$

zu gewissen Stützstellen

$$z_{k-1} \le x_{0k} < \cdots < x_{nk} \le z_k.$$

Dazu sind $n+1$ f–Auswertungen nötig. Die Approximation von f führt zu einer Approximation der Teilintegrale

$$\int_{V_k} f(x)\, dx \approx \int_{V_k} p_{nk}(x)\, dx = \sum_{i=0}^{n} f(x_{ik})\lambda_{ik} h_k, \quad h_k = z_k - z_{k-1},$$

mit Gewichten

$$\lambda_{ik} = \frac{1}{h_k} \int_{V_k} L_{ik}(x)\, dx, \quad L_{ik}(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_{jk}}{x_{ik} - x_{jk}}.$$

Durch Aufsummieren erhält man die Quadraturformel

$$I_\Delta(f) = \sum_{k=1}^{m} \sum_{i=0}^{n} \lambda_{ik} f(x_{ik})\, h_k. \tag{4.2}$$

mit dem Aufwand $N \leq m(n+1)$. Offenbar ist die Abbildung $I_\Delta : C[a,b] \to \mathbb{R}$ linear. Es gilt

$$|I_\Delta(f)| \leq \sum_{k=1}^{m} \sum_{i=0}^{n} |\lambda_{ik}||f(x_{ik})|\, h_k \leq \gamma(b-a)\|f\|_\infty$$

mit

$$\gamma = \max_{k=1,\ldots,m} \sum_{i=0}^{n} |\lambda_{ik}|.$$

Damit ist $\|I_\Delta\|_\infty \leq \gamma(b-a)$. In Analogie zu Satz 4.1 ermitteln wir die *diskrete Kondition* von I_Δ.

Satz 4.2 *Für alle $f, \Delta f \in C[a,b]$ mit $I_\Delta(f) \neq 0$ gilt*

$$\frac{|I_\Delta(f) - I_\Delta(f + \Delta f)|}{|I_\Delta(f)|} \leq \kappa(I_\Delta(f)) \frac{\|\Delta f\|_\infty}{\|f\|_\infty} \tag{4.3}$$

mit

$$\kappa(I_\Delta(f)) = \gamma(b-a) \frac{\|f\|_\infty}{|I_\Delta(f)|}.$$

Ist die Quadraturformel I_Δ von positivem Typ, d.h. gilt $\lambda_i \geq 0$, $\forall i = 1, \ldots, n$, so ist $\gamma = 1$, sonst $\gamma \geq 1$.

Beweis:
Die Konditionsabschätzung (4.3) folgt direkt aus $\|I_\Delta\|_\infty \leq \gamma(b-a)$. Für $p_0 \equiv 1$ und beliebiges $k = 1, \ldots, m$ gilt

$$h_k = \int_{V_k} p_0(x)\, dx = \sum_{i=0}^{n} 1 \cdot \lambda_{ik} h_k = h_k \sum_{i=0}^{n} \lambda_{ik},$$

denn $p_0 \in \mathcal{P}_n$ wird durch Interpolation exakt reproduziert. Ist I_Δ von positivem Typ, so folgt daraus $\gamma = 1$. Ansonsten erhält man $\gamma \geq 1$ aus der Dreiecksungleichung. □

Im Falle $\gamma = 1$ wird die Fehlerempfindlichkeit der Integration durch Diskretisierung nicht verstärkt. In diesem Sinne sind Quadraturformeln von positivem Typ *stabil*. Aus diesem Grund sind wir im folgenden nur an Verfahren von positivem Typ interessiert.

Im Falle eines äquidistanten Gitters

$$\Delta_h = \{z_k = a + kh \mid k = 0, \ldots, m\}, \quad h = \frac{b-a}{m}, \tag{4.4}$$

schreiben wir

$$I_h(f) = I_{\Delta_h}(f).$$

Für jedes feste n ist $I_h(f)$ genau dann von q–ter Ordnung, wenn

$$|I(f) - I_h(f)| = \mathcal{O}(h^q)$$

gilt.
Die aus der CoMa bekannten *summierten Newton–Côtes–Formeln* erhält man durch Wahl äquidistanter Stützstellen

$$x_{ik} = z_{k-1} + i\frac{h_k}{n}, \quad i = 0, \ldots, n.$$

In Tabelle 4.1 ist wie üblich $h = \max_{k=1,\ldots,m} h_k$ gesetzt und der Fehler nur bis auf eine von f und h unabhängige Konstante angegeben.

n	Gewichte	Fehler	Name
1	$\frac{1}{2}\ \frac{1}{2}$	$h^2\|f^{(2)}\|_\infty$	sum. Trapezregel
2	$\frac{1}{6}\ \frac{4}{6}\ \frac{1}{6}$	$h^4\|f^{(4)}\|_\infty$	sum. Simpson–Regel, Keplersche Faßregel
3	$\frac{1}{8}\ \frac{3}{8}\ \frac{3}{8}\ \frac{1}{8}$	$h^4\|f^{(4)}\|_\infty$	sum. 3/8 Regel
4	$\frac{7}{90}\ \frac{32}{90}\ \frac{12}{90}\ \frac{32}{90}\ \frac{7}{90}$	$h^6\|f^{(6)}\|_\infty$	sum. Milne–Regel
5	$\frac{19}{288}\ \frac{75}{288}\ \frac{50}{288}\ \frac{50}{288}\ \frac{75}{288}\ \frac{19}{288}$	$h^6\|f^{(6)}\|_\infty$	—
6	$\frac{41}{840}\ \frac{216}{840}\ \frac{27}{840}\ \frac{272}{840}\ \frac{27}{840}\ \frac{216}{840}\ \frac{41}{840}$	$h^8\|f^{(8)}\|_\infty$	sum. Weddle–Regel

Tabelle 4.1: Newton–Côtes–Formeln

Ist man an einer *effizienten* Lösung des Quadraturproblems (4.1) interessiert, so bleibt noch einiges zu tun:
f sei eine glatte Funktion, d.h. „oft“ differenzierbar mit „schwach“ wachsender Norm $\|f^{(q)}\|_\infty$ für wachsendes q. Dann ist es sinnvoll, ein äquidistantes Gitter (4.4) zu wählen. In diesem Fall wächst die Genauigkeit der Newton–Côtes–Formeln für $n = 2, 4, 6$ exponentiell, der Aufwand aber nur linear. Damit steigt die Effizienz mit wachsender Ordnung. Leider ist bei der Weddle–Regel Schluß. Bei Interpolation mit Polynomen siebter und höherer Ordnung treten negative Gewichte auf. Die entsprechenden Quadraturformeln sind nicht von positivem Typ, also numerisch uninteressant. Wir wollen

- Verfahren beliebig hoher Ordnung von positivem Typ

konstruieren.
f sei keine glatte Funktion, d.h. zeige lokal „stark“ variierendes Verhalten. Dann ist es nicht sinnvoll, Verfahren höherer Ordnung zu verwenden. Hohe Genauigkeit kann durch Reduktion der Schrittweiten h_k erreicht werden. Um die Effizienz zu erhöhen, gilt es, die Lage der Gitterpunkte z_k *automatisch* an das lokale Verhalten von f anzupassen. Zu diesem Zweck wollen wir

- adaptive Multilevel–Verfahren

entwickeln.

4.1 Gauß-Christoffel-Quadratur

Aus Kapitel 3 rührt bereits ein gewisses Mißtrauen gegenüber äquidistanter Polynominterpolation höherer Ordnung her. Dieses wird durch folgendes Beispiel bestätigt.

Beispiel:
Wir betrachten das Integral

$$\int_0^1 \frac{1}{\log(2)(1+x)}\, dx = 1.$$

Durch Interpolation an den Stützstellen

$$x_{0k} = z_{k-1},\ \ x_{1k} = z_{k-1} + \tfrac{1}{2}\big(1 - \tfrac{1}{5}\sqrt{5}\big)h_k,\ \ x_{2k} = z_{k-1} + \tfrac{1}{2}\big(1 + \tfrac{1}{5}\sqrt{5}\big)h_k,\ \ x_{3k} = z_k \quad (4.5)$$

erhält man die zugehörigen Gewichte

$$\lambda_0 = \tfrac{1}{12},\ \lambda_1 = \tfrac{5}{12},\ \lambda_2 = \tfrac{5}{12},\ \lambda_3 = \tfrac{1}{12}.$$

Die resultierende Quadraturformel erfordert genauso wie die 3/8–Regel $N = 3m + 1$ f–Auswertungen. Ein Vergleich (äquidistantes Gitter) zeigt aber erheblich besseres Konvergenzverhalten. Wie in Abbildung 4.1 zu sehen ist, fällt der Fehler mit wachsendem m deutlich schneller (gestrichelte Linie). Auf dem rechten Bild ist das Produkt Fehler $\cdot$ (Aufwand)^q mit $q = 4$ für 3/8– und $q = 6$ für die modifizierte Formel dargestellt. Offenbar haben wir durch geschickte Wahl der *zwei* inneren Stützstellen x_{1k}, x_{2k} die Ordnung gegenüber der 3/8–Formel um *zwei* verbessert. ◁

Wir wollen feststellen, ob wir durch geschickte Wahl der $n+1$ Stützstellen x_{ik}, $i = 0, \ldots, n$, die Mindestordnung $n + 1$ der summierten Newton–Côtes–Formeln auf $2(n + 1)$ steigern können. Dabei wird auch klar werden, warum Simpson–, Milne– und Weddle–Regel jeweils eine Ordnung besser sind als erwartet.

Wir betrachten das äquidistante Gitter Δ_h mit Gitterpunkten

$$z_k = a + kh, \qquad k = 0, \ldots, m, \qquad V_k = [z_{k-1}, z_k], \quad h = \frac{b-a}{m}.$$

Das nächste Lemma motiviert die weitere Vorgehensweise.

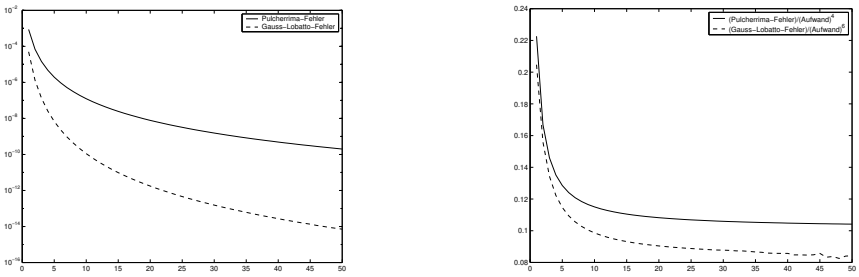


Abbildung 4.1: Ordnungsverbesserung durch Wahl der Stützstellen

Lemma 4.3 *Gegeben seien α, $\beta \in \mathbb{R}$ mit $\alpha < \beta$. Die Stützstellen*

$$\alpha \le x_0 < \cdots < x_n \le \beta$$

seien so gewählt, daß mit den Gewichten

$$\mu_k = \frac{1}{\beta - \alpha} \int_\alpha^\beta L_k(x)\, dx, \qquad L_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}, \qquad k = 0, ..., n \tag{4.6}$$

die Bedingung

$$\int_\alpha^\beta p(x)\, dx = (\beta - \alpha) \sum_{i=0}^{n} \mu_i p(x_i) \qquad \forall p \in \mathcal{P}_{2n+1} \tag{4.7}$$

erfüllt ist. Dann ist die Quadraturformel

$$I_h(f) = \sum_{k=1}^{m} \sum_{i=0}^{n} \mu_i f(x_{ik})\, h$$

mit

$$x_{ik} = z_{k-1} + \frac{h}{\beta - \alpha}(x_i - \alpha), \qquad i = 0, \ldots, n, \quad k = 1, \ldots, m,$$

für alle $f \in C^{2(n+1)}[\alpha, \beta]$ von der Ordnung $2(n+1)$.

Beweis:
Sei $p \in \mathcal{P}_{2n+1}$. Dann gilt

$$\begin{aligned} \int_{V_k} p(x)\, dx &= \int_{z_{k-1}}^{z_k} p(x)\, dx = \frac{h}{\beta - \alpha} \int_\alpha^\beta p\big(z_{k-1} + \tfrac{h}{\beta-\alpha}(x - \alpha)\big)\, dx \\ &= \sum_{i=0}^{n} \mu_i p(x_{ik})\, h. \end{aligned} \tag{4.8}$$

Polynome der Ordnung $2n + 1$ werden also von I_h über V_k exakt integriert.
Nun approximieren wir $f|_{V_k}$ durch das Hermite–Polynom $p \in \mathcal{P}_{2n+1}$, welches neben den $n+1$ Interpolationsbedingungen

$$p(x_{ik}) = f(x_{ik}), \qquad i = 0, \ldots, n,$$

noch den $n+1$ Ableitungsbedingungen

$$p^{(j)}(x_{0k}) = f^{(j)}(x_{0k}), \qquad j = 1, \ldots, n+1,$$

genügt. Dann gilt nach Folgerung 3.6 die Fehlerabschätzung

$$\|f - p\|_{\infty,V_k} \leq \frac{1}{(2(n+1))!} \|f^{(2(n+1))}\|_{\infty,V_k} h^{2(n+1)}.$$

Zusammen mit (4.8) führt dies auf die Fehlerabschätzung

$$\left| \int_{V_k} f(x)\, dx - \sum_{i=0}^{n} \mu_i p(x_{ik})\, h \right| = \left| \int_{V_k} f(x) - p(x)\, dx \right| \leq \frac{h}{(2(n+1))!} \|f^{(2(n+1))}\|_{\infty,V_k} h^{2(n+1)}.$$

Summation über k liefert die Behauptung. □

Wir suchen also Stützstellen x_i mit der Eigenschaft (4.7). Das nächste Lemma zeigt uns, wo wir suchen sollen.

Lemma 4.4 *Genügen die Stützstellen*

$$\alpha \leq x_0 < \cdots < x_n \leq \beta$$

der Exaktheits–Bedingung (4.7)*, so ist das Polynom* $p_{n+1} \in \mathcal{P}_{n+1}$*,*

$$p_{n+1}(x) = (x - x_0) \cdots (x - x_n)$$

orthogonal zu $\mathcal{P}_n$ *bezüglich des* L^2*–Skalarprodukts, d.h. es gilt*

$$(p_{n+1}, q) = 0 \qquad \forall q \in \mathcal{P}_n \tag{4.9}$$

mit

$$(v, w) = \int_{\alpha}^{\beta} v(x) w(x)\, dx.$$

Beweis:
Sei $q \in \mathcal{P}_n$. Dann ist $p_{n+1} \cdot q \in \mathcal{P}_{2n+1}$ und aus (4.7) folgt

$$(p_{n+1}, q) = \int_{\alpha}^{\beta} p_{n+1}(x) q(x)\, dx = (\beta - \alpha) \sum_{k=0}^{n} \mu_k p_{n+1}(x_k) q(x_k) = 0.$$ □

Bemerkung:
Ist $\alpha = -1$ und $\beta = 1$, so erfüllt das Legendre–Polynom

$$p_{n+1}(x) = \frac{1}{2^{n+1}(n+1)!} \left(\frac{d}{dx} \right)^{n+1} (x^2 - 1)^{n+1}$$

die Orthogonalitätsbedingung (4.9). ◁

Nach Satz 2.11 sind Polynome $p_{n+1} \in \mathcal{P}_{n+1}$ mit der Orthogonalitätseigenschaft (4.9) bis auf einen Normierungsfaktor eindeutig bestimmt. Damit sind insbesondere deren Nullstellen eindeutig bestimmt. Wir zeigen nun in Umkehrung von Lemma 4.4, daß diese Nullstellen auf eine Quadraturformel führen, die für alle $p \in \mathcal{P}_{2n+1}$ exakt ist.
Eine notwendige Voraussetzung klärt folgendes Lemma.

Lemma 4.5 *Ein Polynom $p_{n+1} \in \mathcal{P}_{n+1}$ mit der Orthogonalitätseigenschaft* (4.9) *hat $n+1$ verschiedene Nullstellen*

$$x_0, \ldots, x_n \in [\alpha, \beta].$$

Beweis:
Wir zeigen zunächst, daß p_{n+1} keine mehrfachen Nullstellen hat. Im Widerspruch dazu nehmen wir an, daß p_{n+1} mindestens eine doppelte Nullstelle x_0 hat, also

$$p_{n+1}(x) = (x - x_0)^2 q(x)$$

mit $q \in \mathcal{P}_{n-1} \subset \mathcal{P}_n$ vorliegt. Aus der Orthogonalität (4.9) folgt dann

$$0 = (p_{n+1}, q) = \int_\alpha^\beta (x - x_0)^2 q(x)^2 \, dx$$

und da $(x - x_0)^2 q(x)^2 \geq 0$ ist, muss $q \equiv 0$ sein. Das ist ein Widerspruch zu $p_{n+1} \not\equiv 0$. Als nächstes zeigen wir, dass alle $n+1$ verschiedenen Nullstellen von p_{n+1} reell sind und im Intervall $[\alpha, \beta]$ liegen. Im Widerspruch dazu nehmen wir an, daß p_{n+1} nur $n_0 < n+1$ Nullstellen $x_0, \ldots, x_{n_0-1} \in [\alpha, \beta]$ hat. Im Falle $n_0 = 0$ setzen wir $q \equiv 1$ und sonst

$$q(x) = (x - x_0) \cdots (x - x_{n_0-1}).$$

Da p_{n+1} und $q \in \mathcal{P}_{n_0} \subset \mathcal{P}_n$ dieselben Nullstellen in $[\alpha, \beta]$ haben, wechselt das Produkt $p_{n+1} \cdot q$ auf $[\alpha, \beta]$ nicht das Vorzeichen. Ohne Beschränkung der Allgemeinheit sei

$$p_{n+1} \cdot q(x) \geq 0 \qquad \forall x \in [\alpha, \beta].$$

Da weder $p_{n+1} \equiv 0$ noch $q \equiv 0$ ist, gilt $p_{n+1} \cdot q \not\equiv 0$. Daraus folgt

$$\int_\alpha^\beta p_{n+1} \cdot q(x) \, dx > 0$$

im Widerspruch zur Orthogonalität (4.9). □

Nun fahren wir die Ernte ein.

Satz 4.6 *Seien*

$$\alpha \leq x_0 < \cdots < x_n \leq \beta$$

die Nullstellen eines Polynoms $p_{n+1} \in \mathcal{P}_{n+1}$ mit der Orthogonalitätseigenschaft (4.9). *Die Gewichte $\mu_k \in \mathbb{R}$ seien gemäß* (4.6) *berechnet. Dann genügt die resultierende Quadraturformel der Exaktheitsbedingung* (4.7).

Beweis:
Sei zunächst $p \in \mathcal{P}_n$. Dann gilt wegen (4.6)

$$\int_\alpha^\beta p(x)\,dx = \sum_{k=0}^{n} p(x_k) \int_\alpha^\beta L_k(x)\,dx = (\beta-\alpha)\sum_{k=0}^{n} \mu_k p(x_k).$$

Die Quadraturformel ist also exakt für alle $p \in \mathcal{P}_n$.
Sei nun $p \in \mathcal{P}_{2n+1}$. Polynomdivision durch das zu $\mathcal{P}_n$ orthogonale Polynom p_{n+1} ergibt

$$p = p_{n+1} \cdot q + r,$$

mit q, $r \in \mathcal{P}_n$. Dann ist

$$\begin{aligned}
\int_\alpha^\beta p(x)\,dx &= \int_\alpha^\beta p_{n+1}(x)q(x)\,dx + \int_\alpha^\beta r(x)\,dx \\
&= \overbrace{(p_{n+1}, q)}^{=0} + (\beta-\alpha)\sum_{k=0}^{n} \mu_k r(x_k) \\
&= (\beta-\alpha)\sum_{k=0}^{n} \mu_k \big(r(x_k) + \overbrace{p_{n+1}(x_k)q(x_k)}^{=0}\big) \\
&= (\beta-\alpha)\sum_{k=0}^{n} \mu_k p(x_k).
\end{aligned}$$

□

Bemerkung:
Als Konsequenz aus Lemma 4.4 und Satz 4.6 existiert genau ein Satz von Stützstellen x_k, für den die Exaktheitsbedingung (4.7) erfüllt ist. Diese Stützstellen heißen *Gauß–Knoten*. Sie lassen sich aus den Nullstellen der Legendre–Polynome berechnen. Die resultierenden Quadraturformeln heißen *Gauß'sche Quadraturformeln*. ◁

Satz 4.7 *Gauß'sche Quadraturformeln sind von positivem Typ.*

Beweis:
Sei $k_0 = 0, \ldots, n$. Da $L_{k_0} \in \mathcal{P}_n$ folgt $L_{k_0}^2 \in \mathcal{P}_{2n} \subset \mathcal{P}_{2n+1}$. Somit wird $L_{k_0}^2$ exakt integriert und $u_{k_0} > 0$ folgt aus

$$0 < \int_\alpha^\beta L_{k_0}^2\,dx = (\beta-\alpha)\sum_{k=0}^{n} \mu_k L_{k_0}^2(x_k) = (\beta-\alpha)\mu_{k_0}.$$

□

Unsere bisherigen Ergebnisse lassen sich direkt auf Integrale der Form

$$\int_\alpha^\beta f(x)\,\omega(x)\,dx$$

mit einer *Gewichtsfunktion* ω, d.h. einer Funktion mit der Eigenschaft

$$\omega(x) > 0 \qquad \forall x \in (\alpha, \beta),$$

übertragen. Diese Tatsache ist vor allem für *uneigentliche Integrale*, also für $\alpha = -\infty$ und/oder $\beta = +\infty$, interessant, die wir hier aber nicht behandeln wollen. Wir verweisen dazu z.B. auf Deuflhard und Hohmann [1, Abschnitt 9.3] oder Hämmerlin und Hoffmann [2, Kapitel 7, §3.5].
Die entsprechenden Stützstellen, die sogenannten *Gauß–Christoffel–Knoten* $\alpha \geq x_0 < \cdots < x_n \leq \beta$ erhält man in diesem Fall als Nullstellen eines Orthogonalpolynoms p_{n+1} zum *gewichteten Skalarprodukt*

$$(v, w)_\omega = \int_\alpha^\beta v(x)w(x)\ \omega(x)\ dx.$$

Die Gewichte μ_k ergeben sich aus

$$\mu_k = \frac{1}{\beta - \alpha} \int_\alpha^\beta L_k(x)\ \omega(x)\ dx = \frac{1}{\beta - \alpha} \int_\alpha^\beta \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}\ \omega(x)\ dx.$$

Man kann die Gewichte μ_k auch direkt aus p_{n+1} berechnen (siehe z.B. Deuflhard und Hohmann [1, Abschnitt 9.3.2]).

Bemerkung:
Die resultierenden Quadraturformeln heißen *Gauß–Christoffel–Formeln.* ◁

Beispiel:
Im Falle $[\alpha, \beta] = [-1, 1]$ und

$$\omega(x) = \frac{1}{\sqrt{1 - x^2}}$$

erhält man als zugehöriges Orthonormalsystem die Tschebyscheff–Polynome. ◁

Wir kommen zu unserer ursprünglichen Integrationsaufgabe (4.1) zurück. Um Funktionsauswertungen bei der summierten Version zu sparen, wäre die Eigenschaft

$$x_0 = \alpha, \qquad x_n = \beta,$$

wünschenswert. Die Gauß–Knoten haben aber diese Eigenschaft nicht.

Satz 4.8 *Es sei $x_0 = \alpha$, $x_n = \beta$ und $n \geq 2$. Die Stützstellen $x_1, \ldots, x_{n-1}$ seien die Nullstellen eines Polynoms $p_{n-1} \in \mathcal{P}_{n-1}$ mit der Eigenschaft*

$$(p_{n-1}, q)_\omega = 0 \qquad \forall q \in \mathcal{P}_{n-2}$$

für

$$\omega(x) = (x - \alpha)(\beta - x)$$

und μ_k, $k = 0, \ldots, n$, seien die nach (4.6) ohne Gewichtsfunktion berechneten Gewichte. Dann erfüllt die resultierende Quadraturformel die Exaktheitsbedingung

$$\int_\alpha^\beta p(x)\ dx = (\beta - \alpha) \sum_{i=0}^{n} \mu_i p(x_i) \qquad \forall p \in \mathcal{P}_{2n-1}. \tag{4.10}$$

Beweis:
Übung. □

Bemerkung:
Die Quadraturformeln aus Satz 4.8 heißen *Gauß–Lobatto–Formeln.* ◁

Mit Lemma 4.3 folgt aus (4.10), daß die zugehörige summierte Gauß–Lobatto–Formel von $2n$–ter Ordnung ist. Durch Festlegung von $x_0 = \alpha$ und $x_n = \beta$ haben wir also gegenüber der entsprechenden Gauß'schen Formel zwei Ordnungen verloren.

Beispiel:
Wir wollen die Gauß–Lobatto–Formeln für $n = 2, 3$ ausrechnen. Dabei wählen wir $\alpha = -1$, $\beta = 1$ und berechnen die Orthogonalpolynome p_{n-1}, $n = 2, 3$, zur resultierenden Gewichtsfunktion $\omega(x) = 1 - x^2$. Aus der Drei–Term–Rekursion (2.11) erhalten wir

$$p_1(x) = x, \qquad p_2(x) = x^2 - \tfrac{1}{5}.$$

Damit ist die Gauß–Lobatto–Formel $2 \cdot 2 = 4$–ter Ordnung gerade die Simpson–Regel. Daher die höhere Ordnung. Für $n = 3$ erhält man die Stützstellen (4.5) der Gauß–Lobatto–Formel 6–ter Ordnung aus dem Beispiel vom Beginn dieses Abschnitts. ◁

4.2 Klassische Romberg–Quadratur

Wir betrachten eine Funktion $T : [0, h_0] \to \mathbb{R}$. Wir wollen den Funktionswert $T(0)$ approximieren, indem wir das Interpolationspolynom p_n,

$$p_n \in \mathcal{P}_n : \qquad p(h_j) = T(h_j) \qquad \forall j = 1, \ldots, n, \tag{4.11}$$

zu gewissen Stützstellen

$$0 < h_n < \cdots < h_1 < h_0$$

berechnen und, in der Hoffnung, daß

$$p_n(0) \approx T(0)$$

gilt, dann an der Stelle $h = 0$ auswerten. Da $h = 0$ außerhalb des Intervalls $[h_n, h_0]$ liegt, nennt man dieses Verfahren *Extrapolation.*

Beispiel:
Wir betrachten die Funktion $T(h) = \frac{\sin h}{h}$. Bekanntlich gilt $T(0) = 1$. Wir wählen $h_j = 2^{-j}$, $j = 0, \ldots, n$ und n=0,...,7. Die Berechnung von $p_n(0)$ erfolgt mit dem *Algorithmus von Aitken–Neville* (siehe z.B. CoMa–Skript). Abbildung 4.2 zeigt den Fehler $|T(0) - p_n(0)|$ in Abhängigkeit von n. Obwohl der erste Schritt sogar eine leichte Verschlechterung bringt, erhält man schließlich eine deutlich bessere Fehlerreduktion durch Extrapolation (gestrichelt) im Vergleich zur schlichten Auswertung von $T(h_n)$.
Als zweites Beispiel betrachten wir $T(h) = \sqrt{h}$ (Abbildung 4.3). In diesem Fall bringt Extrapolation (gestrichelt) keine nennenswerte Verbesserung gegenüber $T(h_n)$. Man beachte den Skalenunterschied von 11 Zehnerpotenzen gegenüber Abbildung 4.2! Wie üblich wollen wir verstehen, was los ist. ◁

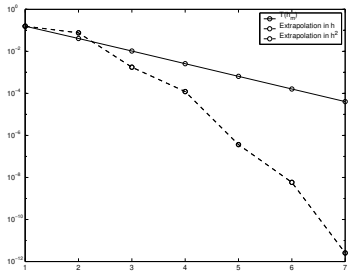


Abbildung 4.2: Extrapolation zur Approximation von $T(h) = \frac{\sin h}{h}$ in $h = 0$.

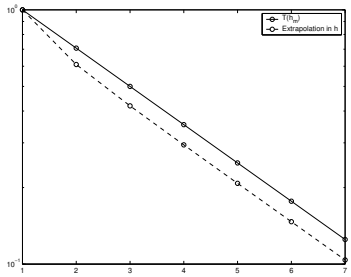


Abbildung 4.3: Extrapolation zur Approximation von $T(h) = \sqrt{h}$ in $h = 0$.

Satz 4.9 *Die Abbildung T besitze die* asymptotische Entwicklung

$$T(h) = T(0) + \sum_{k=1}^{n+1} \tau_k h^k + r_{n+2}(h) h^{n+2}, \qquad h \in [0, h_0], \tag{4.12}$$

mit

$$\|r_{n+2}\|_\infty \leq C.$$

Dann genügt die Extrapolation $p_n(0)$ mit p_n aus (4.11) *der Fehlerabschätzung*

$$|T(0) - p_n(0)| \leq |\tau_{n+1}| h_0 h_1 \cdots h_n + C \sum_{j=0}^{n} h_j^{n+2} |L_j(0)|. \tag{4.13}$$

Dabei bezeichnen

$$L_j(h) = \prod_{\substack{i=0 \\ i \neq j}}^{n} \frac{h - h_i}{h_j - h_i}$$

die bekannten Lagrange–Polynome.

Beweis:
Zur Vorbereitung notieren wir zunächst, daß

$$\sum_{j=0}^{n} h_j^m L_j(0) = \begin{cases} 1 & \text{für } m = 0 \\ 0 & \text{für } m = 1, \ldots, n \\ (-1)^n h_0 h_1 \cdots h_n & \text{für } m = n+1 \end{cases}. \tag{4.14}$$

Ein Beweis von (4.14) findet sich bei Deuflhard und Hohmann [1, Lemma 9.23].
Nun kommen wir zum Beweis von (4.13). Unter Verwendung von (4.12) und (4.14) erhält man

$$\begin{aligned} |T(0) - p_n(0)| &= |T(0) - \sum_{j=0}^{n} T(h_j) L_j(0)| \\ &= |T(0) - \sum_{j=0}^{n} T(0) L_j(0) - \sum_{j=0}^{n} \sum_{k=1}^{n+1} \tau_k h_j^k L_j(0) - \sum_{j=0}^{n} h_j^{n+2} r_{n+2}(h_j) L_j(0)| \\ &\leq |\tau_{n+1}| h_0 h_1 \cdots h_n + C \sum_{j=0}^{n} h_j^{n+2} |L_j(0)|. \end{aligned}$$

□

Bemerkung:
Eine asymptotische Entwicklung von $T(h)$ ist nichts anderes als eine Taylor–Entwicklung von $T(h)$ um $h = 0$. ◁

Bemerkung:
Ignoriert man den Restterm in (4.13), so wird also durch Hinzunahme jeder neuen Stützstelle h_n der Fehler um den Faktor h_n reduziert. Man „gewinnt eine Ordnung in h_n“. ◁

Beispiel:
Wir kommen auf unsere einführenden Beispiele zurück. Bekanntlich ist $T(h) = \sqrt{h}$ in $h = 0$ nicht differenzierbar. Damit existiert keine Taylor–Entwicklung um $h = 0$ und die mangelhafte Genauigkeitsausbeute beim Extrapolieren wird verständlich.
Dagegen hat $T(h) = \frac{\sin h}{h}$ bekanntlich folgende Taylor–Entwicklung um $h = 0$

$$T(h) = \frac{\sin h}{h} = 1 - \frac{h^2}{3!} + \frac{h^4}{5!} - \frac{h^6}{7!} + \frac{h^8}{9!} \cdots . \tag{4.15}$$

Damit existiert für jedes $n \geq 0$ eine asymptotische Entwicklung der Form (4.12) und Satz 4.9 ist anwendbar. ◁

Es fällt auf, daß in (4.15) nur geradzahlige Exponenten auftreten. Dieser Umstand lässt sich wie folgt zur Verbesserung unseres Extrapolationsverfahrens nutzen. Definiert man

$$G(h) = 1 - \frac{h}{3!} + \frac{h^2}{5!} - \frac{h^3}{7!} + \frac{h^4}{9!} \cdots ,$$

so gilt offenbar die Beziehung

$$G(h^2) = T(h).$$

Approximiert man nun G anstelle von T und zwar durch Interpolation an den Stützstellen $h_0^2, h_1^2, \ldots, h_n^2$, so erhält man das Extrapolationspolynom $p_n \in \mathcal{P}_n$ aus

$$p_n \in \mathcal{P}_n : \qquad p_n(h_j^2) = G(h_j^2) = T(h_j) \qquad \forall j = 0, \ldots, n.$$

Nun braucht man in Satz 4.9 nur durchgängig die Stützstellen h_j durch die neuen Stützstellen h_j^2 zu ersetzen, um die Fehlerabschätzung

$$|T(0) - p_n(0)| = |G(0) - p_n(0)| = |\tau_{n+1}| h_0^2 h_1^2 \cdots h_n^2 + \mathcal{O}(h^{2(n+2)})$$

zu erhalten. Anstelle von einer Ordnung gewinnt man auf diese Weise zwei Ordnungen mit jeder neuen Stützstelle und das bei gleicher Anzahl von T–Auswertungen. Die folgende Abbildung zeigt, daß die Fehlerreduktion durch Extrapolation in h^2 tatsächlich nochmals drastisch beschleunigt wird.

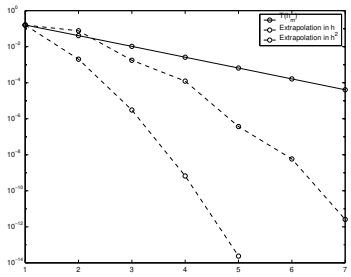


Abbildung 4.4: Extrapolation in h^2 zur Approximation von $\frac{\sin h}{h}$ in $h = 0$.

Wir wollen nun unsere frisch erworbenen Kenntnisse über Extrapolation auf unsere Quadraturaufgabe (4.1) anwenden. Wir nehmen dabei an, daß f eine glatte Funktion ist. Daher gehen wir von einem äquidistanten Gitter

$$z_k = a + kh, \qquad h = \frac{b-a}{m},$$

aus. Wir betrachten eine Quadraturformel $I_h(f)$ der Gestalt (4.2) für festes n. Offenbar lässt sich $T(h) := I_h(f)$ als Funktion

$$(0, h_0] \ni h \to T(h) = I_h(f) \in \mathbb{R}$$

interpretieren. Unser Ziel ist die Approximation von

$$T(0) = I(f) = \int_a^b f(x)\, dx.$$

Ausgehend von der *summierten Trapez–Regel*

$$T(h) = I_h(f) := \tfrac{1}{2}\big(f(a) + f(b)\big)h + \sum_{i=1}^{m-1} f(z_i)h$$

erhalten wir durch Extrapolation die Quadraturformel

$$E_h(f) := p_n(0) \approx T(0) = I(f).$$

Wir wollen die Eigenschaften von $E_h(f)$, insbesondere den Diskretisierungsfehler $|E_h(f) - I(f)|$ in Abhängigkeit von den benötigten Funktionsauswertungen untersuchen.

Als erstes müssen wir klären, ob eine asymptotische Entwicklung von T existiert. Dabei spielen die sogenannten *Bernoulli–Polynome* $B_k \in \mathcal{P}_k$ eine entscheidende Rolle. Sie sind rekursiv definiert durch

$$B_0(x) \equiv 1, \qquad B_k'(x) = kB_{k-1}(x), \quad \int_0^1 B_k(x)\,dx = 0, \quad k = 1, \dots \ .$$

Man erhält

$$B_1(x) = x - \tfrac{1}{2}, \quad B_2(x) = x^2 - x + \tfrac{1}{6}, \quad B_3(x) = x^3 - \tfrac{3}{2}x^2 + \tfrac{1}{2}x, \dots \ .$$

Die *Bernoulli'schen Zahlen* sind definiert durch

$$B_k = B_k(0), \qquad k = 0, \dots, \ .$$

Es gilt

$$B_1 = -\frac{1}{2}, \qquad B_{2k+1} = 0 \qquad \forall k = 1, 2, \dots \ .$$

Für geradzahlige Indizes erhält man

$$B_0 = 1,\ B_2 = \tfrac{1}{6},\ B_4 = -\tfrac{1}{30},\ B_6 = \tfrac{1}{42},\ \dots,\ B_{18} = \tfrac{43867}{798},\ B_{20} = -\tfrac{174611}{30}, \dots$$

und

$$|B_{2k}| \approx (2k)!, \qquad k \to \infty.$$

Zentrales Hilfsmittel bei der Herleitung einer asymptotischen Entwicklung für die summierte Trapezregel ist die *Euler-MacLaurin'sche Summenformel*. Ein Beweis findet sich z.B. bei Stoer [3, Abschnitt 3.2]

Satz 4.10 *Es sei $m, n \in \mathbb{N}$ und $g \in C^{2(n+2)}[0, m]$. Dann gibt es ein $\xi \in [0, m]$, so daß*

$$\begin{aligned} \sum_{k=0}^{m} g(k) \ &= \ \int_0^m g(t)\,dt \\ &\quad + \tfrac{1}{2}\big(g(0) + g(m)\big) + \sum_{k=1}^{n+1} \frac{B_{2k}}{(2k)!}\big(g^{(2k-1)}(m) - g^{(2k-1)}(0)\big) \\ &\quad + \frac{B_{2(n+2)}}{(2(n+2))!} m g^{(2(n+2))}(\xi). \end{aligned}$$

Bemerkung:
Die Reihenentwicklung stammt von Euler (1736) und MacLaurin (1742), das Restglied von Poisson (1823). ◁

Die gesuchte asymptotische Entwicklung ist eine Folgerung aus Satz 4.10.

Satz 4.11 (Asymptotische Fehlerentwicklung) *Es sei $f \in C^{2(n+2)}[a, b]$. Dann gilt*

$$T(h) = I(f) + \tau_2 h^2 + \cdots + \tau_{2(n+1)} h^{2(n+1)} + r_{2(n+2)}(h) h^{2(n+2)} \tag{4.16}$$

mit den Koeffizienten

$$\tau_{2k} = \frac{B_{2k}}{(2k)!}\Big(f^{(2k-1)}(b) - f^{(2k-1)}(a)\Big).$$

und dem Restglied

$$r_{2(n+2)}(h) = \frac{B_{2(n+2)}}{(2(n+2))!}(b-a)f^{(2(n+2))}(\xi)$$

mit geeignetem $\xi \in [a,b]$. Insbesondere ist

$$|r_{2(n+2)}(h)| \le \frac{B_{2(n+2)}}{(2(n+2))!}(b-a)\|f^{(2(n+2))}\|_\infty =: c\|f^{(2(n+2))}\|_\infty.$$

Beweis:
Setze $g(t) = f(a+th)$. Dann gilt mit der Euler–MacLaurin'schen Summenformel

$$\begin{aligned}
T(h) &= h\left(\sum_{k=0}^{m} g(k) - \tfrac{1}{2}\big(g(0)+g(m)\big)\right)\\
&= h\int_0^m g(t)\,dt + \sum_{k=1}^{n+1}\frac{B_{2k}}{(2k)!}h\big(g^{(2k-1)}(m) - g^{(2k-1)}(0)\big)\\
&\quad + \frac{B_{2(n+2)}}{(2(n+2))!}\overbrace{mh}^{b-a} g^{(2(n+2))}(\bar\xi)\\
&= \int_a^b f(x)\,dx + \sum_{k=1}^{n+1}\frac{B_{2k}}{(2k)!}h^{2k}\big(f^{(2k-1)}(b) - f^{(2k-1)}(a)\big)\\
&\quad + \frac{B_{2(n+2)}}{(2(n+2))!}(b-a)h^{2(n+2)}f^{(2(n+2))}(\xi)
\end{aligned}$$

mit $\bar\xi \in [0,m]$ und $\xi = a + \bar\xi h \in [a,b]$. □

Bemerkung:
Die Reihe

$$I(f) + \sum_{k=1}^{n}\tau_{2k}h^{2k}$$

braucht für $n \to \infty$ für kein $h > 0$ zu konvergieren. Für eine ausführliche Diskussion des Nutzens divergenter Reihenentwicklungen verweisen wir auf Deuflhard und Hohmann [1, Beispiel 9.18]. ◁

In der asymptotischen Fehlerentwicklung (4.16) tauchen nur Potenzen von h^2 auf. Das macht die Extrapolation der summierten Trapezregel so attraktiv.

Satz 4.12 *Es sei $f \in C^{2(n+2)}[a,b]$,*

$$h_n < h_{n-1} < \cdots < h_0 \tag{4.17}$$

eine Folge von Schrittweiten und $p_n \in \mathcal{P}_n$ definiert durch

$$p_n \in \mathcal{P}_n : \qquad p_n(h_j^2) = T(h_j) \qquad \forall j = 0, \ldots, n.$$

Dann gilt die Fehlerabschätzung

$$|I(f) - p_n(0)| \leq |\tau_{2(n+1)}| h_0^2 h_1^2 \cdots h_n^2 + c\|f^{(2(n+2))}\|_\infty \sum_{j=0}^{n} h_j^{2(n+2)} |L_j(0)| \tag{4.18}$$

mit $\tau_{2(n+1)}$ und c aus Satz 4.11.

Beweis:
Die Behauptung folgt direkt aus Satz 4.9 und Satz 4.11. □

Folgerung 4.13 *Unter den Voraussetzungen von Satz 4.12 gilt für $h = h_0 = (b-a)/m$ die Fehlerabschätzung*

$$|I(f) - p_n(0)| \leq C\|f^{(2(n+1))}\|_\infty h^{2(n+1)}.$$

Wir kommen zur algorithmischen Realisierung der Extrapolation der Trapezregel. Zunächst erinnern wir an das Verfahren von Aitken–Neville zur Berechnung von $p_n(0)$ (siehe z.B. CoMa). Wir benennen $T_{j0} = T(h_j)$ und berechnen

$$p_n(0) = T_{nn}$$

mit dem Algorithmus von Aitken-Neville. Dazu bestimmen wir T_{jk} aus dem folgenden Tableau:

$$\begin{array}{lccccccccc}
T(h_0) = T_{00} & & & & & & & & & \\
& \ddots & & & & & & & & \\
T(h_1) = T_{10} & \cdots & T_{11} & & & & & & & \\
& \ddots & & \ddots & & & & & & \\
T(h_2) = T_{20} & \cdots & T_{21} & \cdots & T_{22} & & & & & \\
\vdots & & \vdots & & \vdots & \ddots & & & & \\
\vdots & & \vdots & & \vdots & \cdots & T_{n-1,n-1} & & & \\
\vdots & \ddots & & & & \ddots & & \ddots & & \\
T(h_n) = T_{n0} & \cdots & T_{n1} & \cdots & \cdots & \cdots & T_{n,n-1} & \cdots & T_{nn}
\end{array}$$

Die einzelnen T_{jk} werden in folgender Weise rekursiv berechnet

$$\begin{aligned}
T_{jk} &= \frac{1}{h_j^2 - h_{j-k}^2}\Big((0 - h_{j-k}^2)T_{j,k-1} - (0 - h_j^2)T_{j-1,k-1}\Big) \\
&= T_{j,k-1} - \frac{h_j^2}{h_j^2 - h_{j-k}^2}(T_{j,k-1} - T_{j-1,k-1}) \\
&= T_{j,k-1} + \frac{T_{j,k-1} - T_{j-1,k-1}}{\left(\frac{h_{j-k}}{h_j}\right)^2 - 1}.
\end{aligned}$$

Achtung: Mit wachsendem n und kleiner Schrittweite droht Auslöschung!

Wir kommen nun zur Schrittweitenfolge h_j aus (4.17). Durch eine geschickte Wahl wollen wir folgende Ziele erreichen.

- Die Anzahl N der f–Auswertungen soll möglichst klein sein.
- Die Quadraturformel $E_h(f) = p_n(0)$ soll von positivem Typ sein.

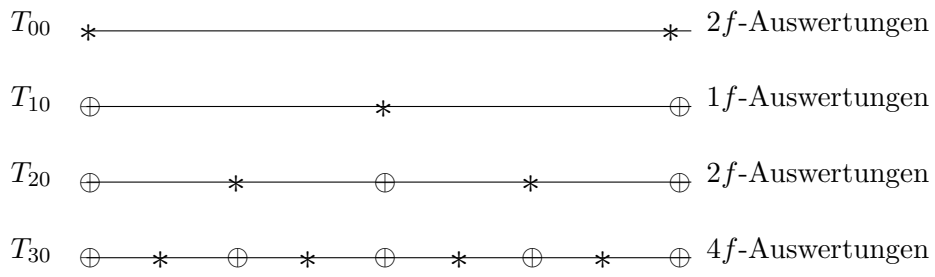
Wir betrachten von nun an die sogenannte *Romberg–Folge*

$$h_j = 2^{-j} h_0, \quad j = 0, \ldots, n, \qquad h_0 = \frac{b-a}{m}. \tag{4.19}$$

Im Falle $j = n$ haben wir $m_n + 1 = 2^n + 1$ Gitterpunkte. Die Auswertung der summierten Trapezregel $I_{h_n}(f) = T(h_n)$ erfordert $2^n + 1$ f–Auswertungen, also

$$N(T_{h_n}) = \text{Aufwand}(T_{h_n}(f)) = 2^n + 1.$$

Zur Auswertung $E_{h_n}(f) = T_{nn} = p_n(0)$ müssen wir $T_{00} = T(h_0), \ldots, T_{n0} = T(h_n)$ berechnen. Wir zählen die dazu benötigten f–Auswertungen.



Induktiv erhält man:

$$N(E_{h_n}) = \text{Aufwand}(E_{h_n}(f)) = 2 + 1 + \cdots + 2^{n-1} = 2 + \frac{2^n - 1}{2 - 1} = 2^n + 1.$$

Damit hat die extrapolierte Trapezregel $E_h(f)$ den gleichen Aufwand wie die Trapezregel $I_h(f)$ selbst (gemessen in f–Auswertungen).

Offenbar sind die T_{jk} Linearkombinationen der $T_{j0} = T(h_j)$. Die $T(h_j)$ aber sind ihrerseits Linearkombinationen der Funktionswerte $f(z_{kj})$, $z_{kj} = a + kh_j$. Insgesamt ist also im Falle der Romberg–Folge

$$E_h(f) = T_{nn} = \sum_{k=0}^{n} \mu_{kn} f(z_{kn}) \tag{4.20}$$

mit gewissen Gewichten μ_{kn}.

Satz 4.14 *Bei Verwendung der Romberg–Folge* (4.19) *ist das resultierende Extrapolationsverfahren von positivem Typ, d.h. die Koeffizienten μ_{kn} in* (4.20) *sind positiv.*

Zum Abschluß wollen wir unser Verfahren ausprobieren.

Beispiel:
Wir betrachten das Integral

$$\frac{1}{2\arctan(1)} \int_{-1}^{1} \frac{1}{1+x^2}\, dx = 1.$$

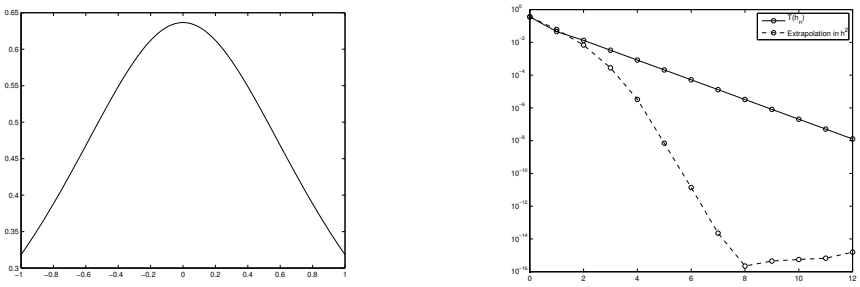


Abbildung 4.5: Extrapolierte Trapezregel für $f(x) = \frac{1}{2\arctan(1)} \frac{1}{1+x^2}$.

Die Abbildung 4.5 zeigt links die zu integrierende Funktion $f(x) = \frac{1}{2\arctan(1)} \frac{1}{1+x^2}$. Auf der rechten Seite ist der Diskretisierungsfehler der extrapolierten Trapezregel $E_{h_n}(f)$ zur Romberg-Folge $h_j = h_0 2^{-j}$, $j = 0, \ldots, n$, mit $h_0 = 1$ (gestrichelt) im Vergleich mit dem Fehler der summierten Trapezregel $T_{h_n}(f)$ für $n = 0, 1, \ldots, 12$ dargestellt. Die Extrapolation wird natürlich entsprechend Satz 4.12 in h_j^2 durchgeführt. In Übereinstimmung mit unseren theoretischen Untersuchungen beobachten wir einen großen Genauigkeitsgewinn durch Extrapolation. Wie lässt sich der Anstieg des Fehlers ab $n = 8$ erklären?

Nun verändern wir den Integranden etwas und betrachten das Integral

$$\frac{\gamma}{2\arctan(\gamma)} \int_{-1}^{1} \frac{1}{1+(\gamma x)^2}\, dx = 1.$$

Den Fall $\gamma = 1$ haben wir eben untersucht. Jetzt wählen wir $\gamma = 500$. Der entsprechende Integrand ist in Abbildung 4.6 links zu sehen. Der Vergleich von Extrapolation $E_{h_n}(f)$ (gestrichelt) mit der summierten Trapezregel $T_{h_n}(f)$ zeigt, daß in diesem Fall die Extrapolation eher schadet als nutzt! Der Grund liegt in dem stark variierenden Verhalten des Integranden in $x = 0$. Nach unserer theoretischen Analyse kann dies zu einem starken Anwachsen des Restglieds führen, welches alle h–Potenzen zunichte macht.
Die Funktion

$$f(x) = \frac{1}{1+(\gamma x)^2}, \qquad \gamma \gg 1, \tag{4.21}$$

ist in diesem Sinne *nicht glatt*. Verfahren höherer Ordnung sind hier fehl am Platze. ◁

4.3 Adaptive Multilevel–Quadratur

In diesem Abschnitt gehen wir davon aus, daß die zu integrierende Funktion f *lokal stark variierendes Verhalten* zeigt (vgl. z.B. (4.21)). Wie wir gesehen haben, sind dann Verfahren

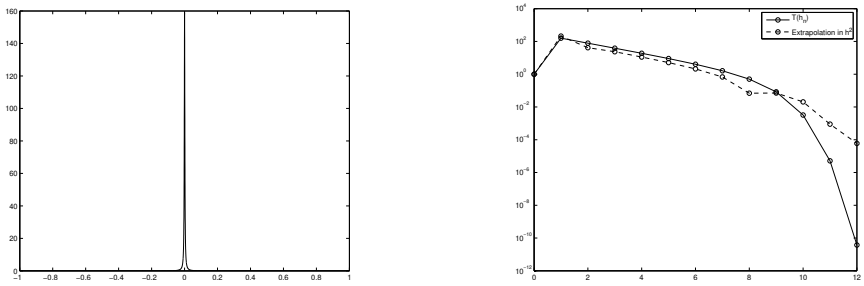


Abbildung 4.6: Extrapolierte Trapezregel für $f(x) = \frac{\gamma}{2\arctan(10)} \frac{1}{1+(\gamma x)^2}$, $\gamma = 500$.

höherer Ordnung nicht sinnvoll. Wir beschränken uns daher auf die summierte Trapezregel

$$T_\Delta(f) = \sum_{k=1}^{m} T_{V_k}(f), \quad T_{V_k}(f) = \tfrac{1}{2}\big(f(z_{k-1}) + f(z_k)\big)h_k, \quad V_k = [z_{k-1}, z_k], \quad h_k = z_k - z_{k-1},$$

zu einem Gitter

$$\Delta = \{a = z_0 < z_1 < \cdots < z_{m-1} < z_m = b\}.$$

Gesucht ist ein Gitter Δ^* mit folgenden Eigenschaften

- Für ein vorgegebenes TOL ist die Genauigkeitsbedingung

$$|I(f) - T_{\Delta^*}(f)| \le TOL \tag{4.22}$$

  erfüllt.

- Die Anzahl der Gitterpunkte $m^* + 1$ =Aufwand$(T_{\Delta^*}(f))$ soll möglichst klein sein.

Die Grundidee der *Mehrgitter–Quadratur* zur approximativen Bestimmung von Δ^* besteht darin, ein geeignet gewähltes Startgitter Δ_0 rekursiv zu verfeinern, bis die Genauigkeitsbedingung (4.22) erfüllt ist. Das genaue Vorgehen ist in Abbildung 4.7 beschrieben. Demnach setzt sich ein Mehrgitterverfahren zur Quadratur aus folgenden Bausteinen zusammen:

- Wahl eines Ausgangsgitters Δ_0,
- Auswertung einer Quadraturformel $T_{\Delta_j}(f)$
- Fehlerkontrolle
- Verfeinerungsstrategie

Wir konzentrieren uns zunächst auf den 4. Baustein: die Verfeinerungsstrategie.

Uniforme Verfeinerung. Die *uniforme* Verfeinerung von Δ_j basiert auf der Halbierung *jedes* Teilintervalls

$$V_k^{(j)} = [z_{k-1}^{(j)}, z_k^{(j)}], \qquad k = 1, \ldots, m_j,$$

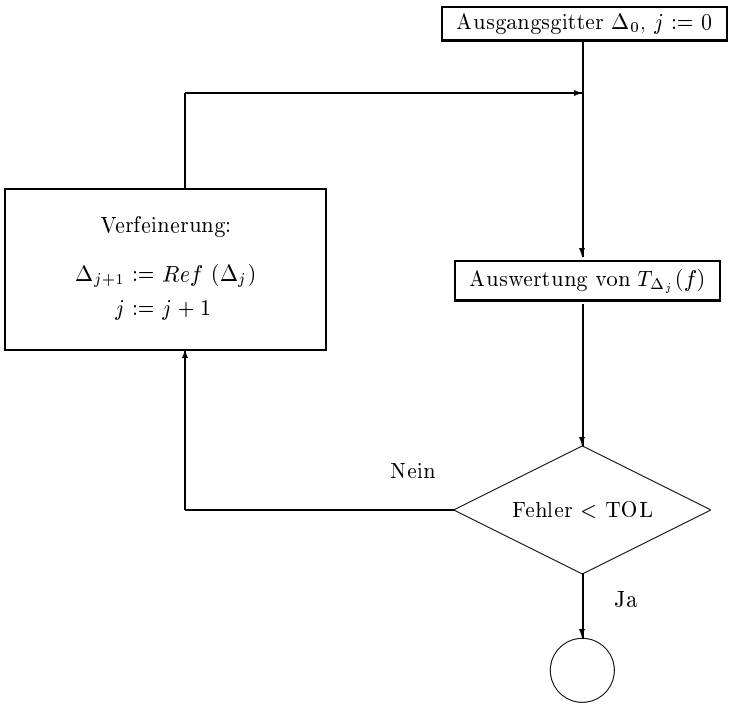


Abbildung 4.7: Mehrgitter–Quadratur

also

$$\Delta_{j+1} = Ref_{\text{uniform}}(\Delta_j) = \Delta_j \cup \left\{ \tfrac{1}{2}\left(z_{k-1}^{(j)} + z_k^{(j)}\right) \mid k = 1, \ldots, m_j \right\}.$$

Die uniforme Verfeinerung zielt nur auf das Erreichen der vorgegebenen Genauigkeit TOL. Tatsächlich ist unabhängig vom Startgitter Δ_0 die Genauigkeitsbedingung (4.22) nach endlich vielen Schritten erfüllt (sofern genügend Speicherplatz verfügbar ist). Auf der anderen Seite wird gar nicht erst versucht, das zweite Ziel der Mehrgitter–Quadratur zur erreichen, nämlich die Anzahl der Gitterpunkte klein zu halten. Alle Teilintervalle $V_k^{(j)}$ werden halbiert, auch dann, wenn sie keinen nennenswerten Beitrag zum Gesamtfehler leisten. Auf diese Weise werden überflüssige Gitterpunkte eingefügt, welche den Aufwand erhöhen, ohne die Genauigkeit zu verbessern. Damit sinkt die Effizienz.

Adaptive Verfeinerung. Um Gitterpunkte zu sparen, soll ein Teilintervall $V_k^{(j)}$ dann und nur dann halbiert werden, wenn der *lokale Diskretisierungsfehler*

$$e^*_{V_k^{(j)}}(f) = I_{V_k^{(j)}}(f) - T_{V_k^{(j)}}(f), \qquad I_{V_k^{(j)}}(f) = \int_{V_k^{(j)}} f(x)\, dx,$$

genügend groß ist, wenn also

$$\left| e^*_{V_k^{(j)}}(f) \right| \geq \eta \tag{4.23}$$

ausfällt. Dabei ist η ein geeigneter Schwellwert.
Leider ist der lokale Diskretisierungsfehler im allgemeinen nicht bekannt. Wir müssen daher auf eine geeignete Schätzung zurückgreifen. Die Auswertung der *a priori Fehlerschätzung*

$$\tfrac{1}{12}\min_{x\in V_k^{(j)}}|f''(x)|h_k^3 \le |e_{V_k^{(j)}}(f)| \le \tfrac{1}{12}\max_{x\in V_k^{(j)}}|f''(x)|h_k^3$$

ist zu aufwendig (Berechnung der zweiten Ableitung, Minimierungsprobleme).
Der nächste Satz liefert eine *a posteriori Fehlerschätzung.*

Satz 4.15 *Die Simpson–Regel*

$$S_{V_k^{(j)}} = \frac{h_k}{6}\left(f(z_{k-1}^{(j)}) + 4f(z_{k-\frac{1}{2}}^{(j)}) + f(z_k^{(j)})\right), \qquad z_{k-\frac{1}{2}}^{(j)} = \frac{1}{2}\left(z_{k-1}^{(j)} + z_k^{(j)}\right) \tag{4.24}$$

genüge der Bedingung

$$|I_{V_k^{(j)}}(f) - S_{V_k^{(j)}}(f)| \le q|e^*_{V_k^{(j)}}(f)|, \qquad 0 \le q < 1. \tag{4.25}$$

Dann hat die a posteriori Fehlerschätzung

$$e_{V_k^{(j)}}(f) = S_{V_k^{(j)}}(f) - T_{V_k^{(j)}}(f) \tag{4.26}$$

die Eigenschaft

$$(1+q)^{-1}|e_{V_k^{(j)}}(f)| \le |e^*_{V_k^{(j)}}(f)| \le (1-q)^{-1}|e_{V_k^{(j)}}(f)|. \tag{4.27}$$

Beweis:
Wir zeigen nur die rechte Abschätzung. Aus der Dreiecksungleichung und (4.25) folgt

$$\begin{aligned} |e^*_{V_k^{(j)}}(f)| &= |I_{V_k^{(j)}}(f) - T_{V_k^{(j)}}(f)| \le |S_{V_k^{(j)}}(f) - I_{V_k^{(j)}}(f)| + |S_{V_k^{(j)}}(f) - T_{V_k^{(j)}}(f)| \\ &\le q|e^*_{V_k^{(j)}}(f)| + |e_{V_k^{(j)}}(f)|. \end{aligned}$$

Bemerkung:
Die Bedingung (4.25) heißt *Saturationsbedingung.* Sie ist erfüllt, wenn die Simpson–Regel eine bessere Approximation als die Trapez–Regel liefert. Das muß nicht unbedingt der Fall sein (Übung.)
Bekanntlich gilt aber für $f \in C^4[a,b]$

$$|I_{V_k^{(j)}}(f) - S_{V_k^{(j)}}(f)| \le \tfrac{1}{90}\max_{x\in V_k^{(j)}}|f^{(4)}(x)|h_k^5.$$

Liegt umgekehrt

$$ch_k^3 \le |I_{V_k^{(j)}}(f) - T_{V_k^{(j)}}(f)|$$

vor, so ist die Saturationsbedingung (4.25) für genügend kleine h_k erfüllt. Es gilt sogar

$$q = q(h_k) \to 0, \qquad h_k \to 0.$$

Mit Blick auf (4.27) wird die Fehlerschätzung (4.26) mit immer kleinerem h_k also immer besser: Der Fehlerschätzer (4.26) ist *asymptotisch exakt.* ◁

Mit Blick auf (4.23) wird nun jedes Teilintervall $V_k^{(j)}$ verfeinert, für das

$$|e_{V_k^{(j)}}(f)| \geq \eta$$

ausfällt, also

$$\Delta_{j+1} = Ref_{\text{adaptiv}}(\Delta_j) = \Delta_j \cup \left\{ \tfrac{1}{2}\left(z_{k-1}^{(j)} + z_k^{(j)}\right) \mid |e_{V_k^{(j)}}(f)| \geq \eta,\ k = 1, \ldots, m_j \right\}.$$

Wir haben noch den Schwellwert η festzulegen. Eine mögliche Wahl ist die sogenannte *Maximumsstrategie*

$$\eta = \max_{k=1,\ldots,m_j} |e_{V_k^{(j)}}(f)|$$

In jedem Verfeinerungsschritt werden also solche Intervalle halbiert, in denen der maximale geschätzte Fehler auftritt. Eine verfeinerte Heuristik zur Bestimmung des Schwellwerts, welche auf Extrapolationsideen beruht, findet sich bei Deuflhard und Hohmann [1, 9.7.1]. Anders als bei uniformer Verfeinerung ist nicht ohne weiters klar, ob die adaptive Mehrgitter–Quadratur konvergiert.

Satz 4.16 *Es sei $f \in C^2[a,b]$ und die Saturationsbedingung* (4.25) *sei für alle $k = 1, \ldots, m_j$, $j = 0, \ldots$ erfüllt. Dann gibt es zu jedem TOL ein J, so daß $T_{\Delta_J}(f)$ der Genauigkeitsbedingung* (4.22) *genügt.*

Beweis:
Der Einfachheit halber führen wir den Beweis unter der Annahme, daß $q = 0$ oder gleichbedeutend $e_{V_k^{(j)}} = e^*_{V_k^{(j)}}$ gilt. Die Argumentation lässt sich auf $q \in [0,1)$ erweitern.
Wir zeigen, daß mit $h_k^{(j)} = z_k^{(j)} - z_{k-1}^{(j)}$

$$h^{(j)} = \max_{k=1,\ldots,m_j} h_k^{(j)} \to 0, \qquad j \to \infty.$$

Dazu reicht es nachzuweisen, daß aus

$$|e^*_{V_{k^*}^{(j)}}| \geq \varepsilon > 0$$

folgt, daß $h_{k_0}^{(j)}$ nach endlich vielen Schritten halbiert wird. Nach j Verfeinerungsschritten ist

$$|e^*_{V_k^{(j)}}| \leq c\|f''\|_\infty h^{(j)} \qquad \forall k = 1, \ldots, m_j$$

Der maximale lokale Fehler tritt in mindestens einem Intervall $V_{k_0}^{(j)}$, $k_0 \in \{1, \ldots, m_j\}$, auf. Nach Halbierung dieser Intervalle ist der Fehler auf den resultierenden Teilintervallen durch $\frac{1}{8}c\|f''\|_\infty h^{(j)}$ beschränkt. Entweder ist nun die Menge

$$M_1 = \{k \mid e^*_{V_k^{(j)}}| > \tfrac{1}{8}c\|f''\|_\infty h^{(j)}\}$$

leer, oder sie wird durch Verfeinerung um mindestens ein weiters Element reduziert. Damit ist nach (höchstens) m_j Schritten

$$|e^*_{V_k^{(j+m_j)}}| \leq \tfrac{1}{8}c\|f''\|_\infty h^{(j)}.$$

Mit geeignetem $s \in \mathbb{N}$ erhält man nach endlich vielen Schritten

$$|e^*_{V_k^{(\tilde{j})}}| \leq \tfrac{1}{8^s} c \|f''\|_\infty h^{(j)} \leq \varepsilon \qquad \forall k = 1, \ldots, m_{\tilde{j}}.$$

Dann muß insbesondere $h^{(j)}_{k_0}$ halbiert worden sein. □

Damit ist unsere adaptive Verfeinerungsstrategie fertig und soll ausprobiert werden.

Beispiel:
Wir betrachten wieder das Integral

$$\frac{\gamma}{2\arctan(\gamma)} \int_{-1}^{1} \frac{1}{1+(\gamma x)^2}\, dx = 1 \tag{4.28}$$

und zwar wieder im Fall $\gamma = 500$. Als Ausgangsgitter wählen wir

$$\Delta_0 = \{a = z_0^0, z_1^0 = b\} \tag{4.29}$$

mit $m_0 = 2$ Gitterpunkten. Wie oben beschrieben, bestimmen wir ausgehend von Δ_0 durch adaptive Verfeinerung die Gitterfolge $\Delta_1, \Delta_2, \ldots, \Delta_j$ mit $j = 250$. Abbildung 4.8 zeigt links den Integranden. Jeder adaptiv gewählte Gitterpunkt $z_k^{(j)}$ ist durch einen Kreis markiert. Auf der rechten Seite sieht man die Entwicklung des Diskretisierungsfehler in Abhängigkeit von der Anzahl der Funktionsauswertungen. Beachte, daß bei dem adaptiven Verfahren die zur Fehlerschätzung benötigten Funktionsauswertungen an den Intervallmittelpunkten $z^{(j)}_{k-\frac{1}{2}}$ mitgezählt werden. Als Vergleich dient die Trapezregel auf einem uniform verfeinerten Gitter (gestrichelt), die sich im letzten Beispiel des vorigen Abschnitt gegen die Extrapolation behauptet hatte.

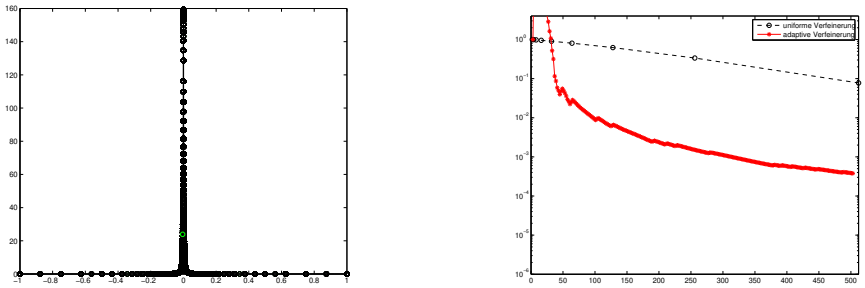


Abbildung 4.8: Uniforme und adaptive Verfeinerung

Es zeigt sich, daß in diesem Beispiel die adaptive Verfeinerung deutlich überlegen ist. Erwartungsgemäß verstärkt sich diese Überlegenheit, wenn man γ weiter vergrößert. ◁

Wir kommen nun zum letzten Baustein unserer Mehrgitter–Quadratur, der *Kontrolle des Diskretisierungsfehlers*

$$e^*_{\Delta_j}(f) = I(f) - T_{\Delta_j}(f)$$

Bis jetzt sind wir immer davon ausgegangen, daß die exakte Lösung $I(f)$ bekannt ist. Das ist natürlich in der Praxis gerade nicht der Fall. Wir wollen daher auch den Diskretisierungsfehler $e^*_{\Delta_j}(f)$ schätzen und gehen dabei genau wie beim lokalen Diskretisierungsfehler vor.

Satz 4.17 *Die summierte Simpson–Regel*

$$S_{\Delta_j} = \sum_{k=1}^{m_j} S_{V_k^{(j)}}$$

mit $S_{V_k^{(j)}}$ *aus* (4.24) *genüge der Bedingung*

$$|I(f) - S_{\Delta_j}(f)| \leq q|e^*_{\Delta_j}(f)|, \qquad 0 \leq q < 1. \tag{4.30}$$

Dann hat die a posteriori Fehlerschätzung

$$e_{\Delta_j}(f) = S_{\Delta_j}(f) - T_{\Delta_j}(f) \tag{4.31}$$

die Eigenschaft

$$(1+q)^{-1}|e_{\Delta_j}(f)| \leq |e^*_{\Delta_j}(f)| \leq (1-q)^{-1}|e_{\Delta_j}(f)|. \tag{4.32}$$

Beweis:
Der Beweis ist wortwörtlich derselbe wie für Satz 4.15. □

Bemerkung:
Analog zum lokalen Fall ist die Saturationsbedingung (4.30) für genügend kleine $h = \max_{k=1,\ldots,m_j} h_k$ erfüllt, wenn die Bedingung

$$ch^2 \leq |I(f) - T_{\Delta_j}(f)|$$

vorliegt, wenn also die summierte Trapezregel nicht zufällig genauer ist als erwartet. Die a posteriori Fehlerschätzung (4.31) ist dann asymptotisch exakt. ◁

Beispiel:
Wir betrachten wieder den Integranden f aus (4.28) mit $\gamma = 500$ und dieselbe adaptiv gewählte Gitterfolge $\Delta_0, \Delta_1, \Delta_2, \ldots, \Delta_j$ mit $j = 250$ wie im vorigen Beispiel.

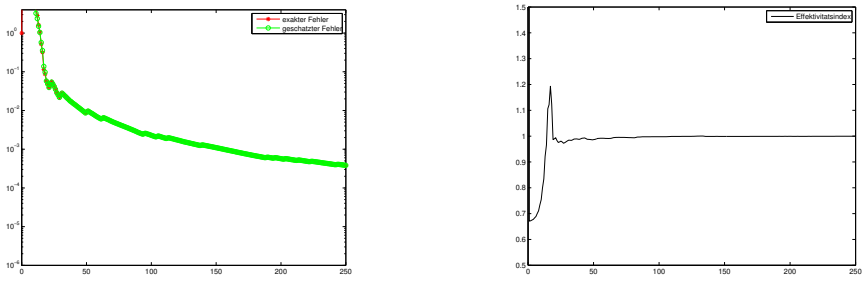


Abbildung 4.9: A posteriori Schätzung des Diskretisierungsfehlers

Abbildung 4.9 zeigt links die exakten Diskretisierungsfehler $e^*_{\Delta_i}(f)$ im Vergleich mit den Schätzungen $e_{\Delta_i}(f)$ aus (4.31) für $i = 0, \ldots, 250$. Nach anfänglichen Schwankungen sind exakte Fehler und Schätzungen nicht zu unterscheiden. Auf der rechten Seite ist der sogenannte *Effektivitätsindex*

$$\kappa_i = \frac{|e_{\Delta_i}(f)|}{|e^*_{\Delta_i}(f)|}, \qquad i = 0, \ldots, j = 126,$$

zu sehen. Es gilt $\kappa_{20} = 0.9865$ und zum Schluß ist $\kappa_{250} = 0.9998$, d.h. exakter Wert und Schätzung stimmen bis auf 3 Stellen überein. ◁

Damit haben wir alle Bausteine eines typischen adaptiven Mehrgitterverfahrens zur Quadratur zusammengestellt (vgl. Abbildung 4.7). In praktischen Anwendungen ersetzt man die exakte Abbruchbedingung (4.22) meist durch

$$|e_{\Delta_j}(f)| \leq \sigma \, TOL.$$

Dabei soll ein *Sicherheitsfaktor* $\sigma < 1$ den unbekannten Faktor $(1-q)$ kompensieren. Besser wäre es natürlich, auch noch $(1-q)$ zu schätzen. Es gibt entsprechende Techniken, auf die wir an dieser Stelle aber nicht weiter eingehen.
Zum Abschluß wollen wir kurz den Aufwand unserer adaptiven Mehrgitter–Quadratur untersuchen.

Satz 4.18 *Der Aufwand zur Berechnung von $T_{\Delta_j}(f)$ mit der oben beschriebenen adaptiven Mehrgitter–Quadratur inklusive a posteriori Schätzung des Diskretisierungsfehlers ist durch $2m_j + 1$ beschränkt.*

Beweis:
Außer der Auswertung in den $m_j + 1$ Gitterpunkten von Δ_j benötigen wir m_j Auswertungen in den Intervall–Mittelpunkten zur Schätzung des lokalen und globalen Diskretisierungsfehlers. Natürlich werden dabei die Werte von f aus jedem Verfeinerungszyklus aufgehoben und später wiederverwendet. □

Der Vorteil adaptiver Verfahren liegt darin, daß bei Integranden mit stark variierendem Glattheitsverhalten typischerweise

$$2m_{\text{adaptiv}} \ll m_{\text{uniform}}$$

ausfällt (vgl. obige Beispiele). Nur im glatten Fall, wenn a priori ein uniformes Gitter gewählt werden kann, sind adaptive Techniken nicht sinnvoll.

Bemerkung:
Bis jetzt haben wir nur die Trapezregel (Ordnung $p = 2$) verwendet und die Genauigkeit nur durch Einfügen von Gitterpunkten erhöht. Ist der Intergrand f in gewissen Teilintervallen glatt, so ist es besser, dort die Ordnung p zu erhöhen, z.B. durch Verwendung eines Extrapolationsverfahrens. In anderen Teilintervallen mag f nicht glatt sein. Dann ist dort nach wie vor die Halbierung der Schrittweite sinnvoll. Ein Kriterium, ob eine Erhöhung der Ordnung p oder eine Halbierung der Schrittweite h vorgenommen werden soll, ist das Kernstück derartiger *hp–Methoden.* ◁

Literatur

[1] P. Deuflhard and A. Hohmann. *Numerische Mathematik I.* de Gruyter, 4. Auflage, 2008. Obwohl unsere Darstellung sich thematisch eng an dieses Buch anlehnt, wird die geneigte Leserin immer wieder unterschiedliche Blickwinkel finden. Empfehlenswert ist vor allem die ausführliche Motivation wichtiger Konzepte, insbesondere von Extrapolation und Adaptivität.

[2] G. Hämmerlin and K.-H. Hoffmann. *Numerische Mathematik*. Springer, 4. Auflage, 2004. Neben einer Reihe von interessanten Bemerkungen zu speziellen Fragen findet sich insbesondere ein Abschnitt über mehrdimensionale Quadratur.

[3] J. Stoer. *Numerische Mathematik I*. Springer, 10. Auflage, 2007. Beim ersten Erscheinen des Buches waren Extrapolationsverfahren zur Quadratur noch ein brandaktuelles Thema, zu dem sowohl Stoer als auch Bulirsch wichtige Beiträge geleistet haben.

5 Anfangswertprobleme für gewöhnliche Differentialgleichungen

5.1 Mathematische Modelle zeitabhängiger Prozesse

5.1.1 Radioaktiver Zerfall und Populationsdynamik

Wir kennen die Anzahl der Atome $x_0 \in \mathbb{R}$ eines radioaktiven Materials zum Zeitpunkt $t = 0$ und interessieren uns für

$$x(t): \quad \text{Anzahl der Atome zum Zeitpunkt } t > 0.$$

Es sei

$$p\Delta t, \quad p \in (0,1)$$

die Wahrscheinlichkeit, daß ein Atom während eines „kleinen“ Zeitraums Δt zerfällt. Dann folgt die Bilanzgleichung

$$p\Delta t = \frac{x(t) - x(t+\Delta t)}{x(t)}.$$

In der Wahrscheinlichkeitstheorie nennt man den Ausdruck rechts vom Gleichheitszeichen *relative Häufigkeit*. Grenzübergang $\Delta t \to 0$ liefert die Differentialgleichung

$$x'(t) = -px(t), \quad t > 0. \tag{5.1}$$

Linearkombinationen von Lösungen dieser Differentialgleichung sind wieder Lösungen. Die Differentialgleichung nennt man daher *linear*.
Beachte, daß wir von jetzt an zulassen, daß $x(t)$ reelle Werte annimmt. Erst nach Rundung auf ganze Zahlen ist $x(t)$ im Sinne unserer Aufgabenstellung interpretierbar. Wir nehmen an, daß der damit verbundene Rundungsfehler vernachlässigbar ist. Diese Annahme nennt man

Kontinuumshypothese.

Offenbar ist die Kontinuumshypothese vertretbar, wenn $x(t) \gg 0$.
Der Zerfallsprozess wird also durch das *Anfangswertproblem* (AWP)

$$x'(t) = -px(t), \quad x(0) = x_0, \tag{5.2}$$

beschrieben. Durch Einsetzen finden wir, daß alle Funktionen der Form

$$x(t) = \alpha e^{-pt}, \quad \alpha \in \mathbb{R}, \tag{5.3}$$

Lösungen von 5.1 sind. Die Festlegung der Konstante α erfolgt durch die *Anfangsbedingung* in (5.2). Man erhält

$$x_0 = x(0) = \alpha e^{-p\cdot 0} = \alpha.$$

Damit ist

$$x(t) = x_0 e^{-pt}$$

eine Lösung von (5.2). Wir werden später sehen, daß es keine weiteren Lösungen gibt.

Beispiel:
Abbildung 5.1 zeigt die Lösung im Falle $p = 1$ und $x_0 = 10^6$. Achtung: Offenbar gilt $x(t) \to 0$ für $t \to \infty$. Für große t ist also $x(t) \gg 0$ sicherlich falsch und die Kontinuumshypothese nicht mehr haltbar. Damit ist unser Modell nur für ein beschränktes Zeitintervall physikalisch sinnvoll. ◁

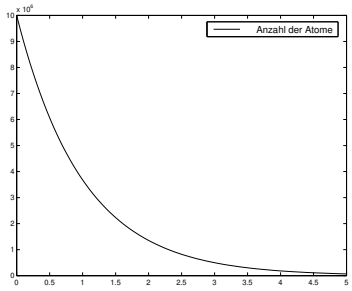


Abbildung 5.1: Radioaktiver Zerfall

Als nächstes betrachten wir das Wachstum einer Bakterienkultur. Die Vermehrung erfolgt durch Teilung. Wir kennen die Anzahl x_0 Bakterien zum Zeitpunkt $t = 0$ und wollen

$$x(t): \quad \text{Anzahl der Bakterien zum Zeitpunkt } t > 0$$

ermitteln. Es sei

$$p\Delta t \in [0, 1]$$

die Wahrscheinlichkeit, daß sich ein Bakterium während eines „kleinen“ Zeitintervalls Δt teilt. Dann folgt

$$x(t + \Delta t) - x(t) = p\Delta t\, x(t). \tag{5.4}$$

Grenzübergang $\Delta t \to 0$ liefert

$$x'(t) = px(t), \quad t > 0.$$

Wir erhalten also das AWP

$$x'(t) = px(t), \quad x(0) = x_0. \tag{5.5}$$

In gleicher Weise wie zuvor kommt man auf die Lösung

$$x(t) = x_0 e^{pt}.$$

Offenbar wächst $x(t)$ exponentiell mit wachsender Zeit t (vgl. Abbildung 5.2). Das ist unrealistisch. Der Grund für diesen *Modellfehler* liegt darin, daß wir die Sterblichkeit und Konkurrenz der Bakterien, z.B. um Nahrung, nicht berücksichtigt haben. Zur Verbesserung unseres Modells nehmen wir an, daß Konkurrenz zweier Bakterien innerhalb des „kleinen" Zeitintervalls Δt zum *Absterben* von

$$\Delta x_{kon} = kx(t)^2 \Delta t, \quad k > 0,$$

Bakterien führt. Beachte, daß eine ortsabhängige Konzentration der Bakterien zu einem ortsabhängigen Wahrscheinlichkeit der Konkurrenz zweier Bakterien führen würde. Wir haben einfach angenommen, daß die Konzentration der Bakterien *nicht* vom Ort abhängt. Diese Annahme nennt man auch

Durchmischungshypothese.

Anstelle von (5.4) erhalten wir nun

$$x(t + \Delta t) - x(t) = p\Delta t x(t) - kx(t)^2 \Delta t.$$

und daraus durch Grenzübergang $\Delta t \to 0$ die Differentialgleichung

$$x'(t) = px(t) - kx(t)^2. \tag{5.6}$$

Linearkombinationen von Lösungen von (5.6) sind i.a. *nicht* wieder Lösungen: Die Differentialgleichung (5.6) ist *nichtlinear*.

Beispiel:
Abbildung 5.2 zeigt die Lösung des einfachen Modells für $x_0 = 10^6$ und $p = 1$ im Vergleich mit der Lösung des verfeinerten Modells (5.6) für $x(0) = x_0 = 10^6$, $p = 1$ und $k = 2 \cdot 10^{-8}$. Die Lösung des verfeinerten Modells scheint nicht nur beschränkt, sondern für $t \geq t_0 = 9$ sogar konstant zu sein. ◁

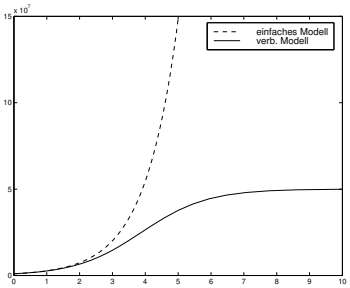


Abbildung 5.2: Einfaches und verfeinertes Modell einer Bakterienkultur

Die Lösung $x(t)$ ist *im Gleichgewicht* oder, gleichbedeutend, *stationär* für $t \geq t_0$, falls es keine zeitliche Änderung gibt, also

$$x'(t) \equiv 0, \quad \forall t \geq t_0.$$

Einsetzen in die Differentialgleichung (5.6) liefert

$$0 \equiv px - kx^2.$$

Mögliche Gleichgewichtslösungen sind also

$$x \equiv 0 \quad \text{und} \quad x \equiv \frac{p}{k}.$$

In unserem obigen Beispiel nähert sich die Lösung $x(t)$ gerade der Gleichgewichtslösung $x \equiv \frac{p}{k} = 5 \cdot 10^7$.

Bemerkung:
Die weitere mathematische Analyse unseres Populationsmodells hätte sich u.a. folgenden Fragen zu widmen:

- Existiert für alle $x_0 \in \mathbb{R}$ eine eindeutig bestimmte Lösung?
- Kann man zeigen, daß die Lösung immer beschränkt ist?
- Bleibt die Lösung positiv?
- Strebt jede Lösung von (5.6) für $t \to \infty$ gegen eine stationäre Lösung? ◁

5.1.2 Newton'sche Mechanik

Wir betrachten ein Objekt im $\mathbb{R}^d$ mit der Masse $m > 0$, zum Beispiel ein Flugzeug. Auf das Objekt wirkt zu jedem Zeitpunkt $t > 0$ eine Kraft $F(t)$ ein. Es sei

$$\begin{aligned} x(t) &: \text{ Ort des Objektes zur Zeit } t > 0, \\ v(t) = x'(t) &: \text{ Geschwindigkeit des Objektes zur Zeit } t > 0, \\ a(t) = v'(t) &: \text{ Beschleunigung.} \end{aligned}$$

Die Ableitungen sind jeweils komponentenweise zu verstehen. Das *Newton'sche Gesetz der Impulserhaltung* lautet

$$\text{Kraft} = \text{Masse} \times \text{Beschleunigung}$$

oder als Formel

$$F(t) = m\, a(t), \quad t > 0. \tag{5.7}$$

Wir betrachten zwei Spezialfälle von (5.7).

Freies Teilchen. Ist

$$F \equiv 0,$$

wirkt also keine Kraft auf das Objekt, so spricht man von einem *freien Teilchen*. Nach Division durch $m > 0$ folgt dann aus (5.7)

$$a(t) = v'(t) = x''(t) = 0.$$

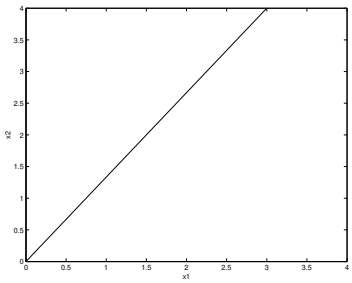


Abbildung 5.3: Freies Teilchen

Wir erhalten also das folgende *System von d Differentialgleichungen zweiter Ordnung*

$$x''(t) = 0, \quad t > 0. \tag{5.8}$$

Da jede der d Differentialgleichungen linear ist, ist (5.8) eine *lineares System* von Differentialgleichungen. Wir können (5.8) auch als äquivalentes *System von* $2d$ *linearen Differentialgleichungen 1. Ordnung* schreiben, nämlich

$$\begin{aligned} x'(t) &= v(t) \\ v'(t) &= 0 \end{aligned} \tag{5.9}$$

Für jedes $\alpha, \beta \in \mathbb{R}^d$ ist

$$x(t) = \alpha t + \beta, \quad v(t) = \alpha$$

eine Lösung von (5.9). Um die Parameter $\alpha, \beta \in \mathbb{R}^d$ zu bestimmen, brauchen wir $2d$ Anfangsbedingungen. Folgende Bedingungen sind naheliegend:

$$\begin{aligned} x(0) = x_0 &: \text{ Ort zum Zeitpunkt } t = 0, \\ v(0) = v_0 &: \text{ Geschwindigkeit zum Zeitpunkt } t = 0. \end{aligned} \tag{5.10}$$

Wir erhalten dann die Lösung

$$x(t) = v_0 t + x_0, \quad v(t) = v_0,$$

und werden später sehen, daß es keine weiteren Lösungen gibt.

Beispiel:
Im Falle

$$x_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad v_0 = \begin{pmatrix} 3 \\ 4 \\ 0 \end{pmatrix}$$

ist $x_3(t) = 0 \ \forall t > 0$. Abbildung 5.3 zeigt die (etwas eintönige) Bahn des Teilchens in der (x_1, x_2)–Ebene für $t \in [0, 1]$. ◁

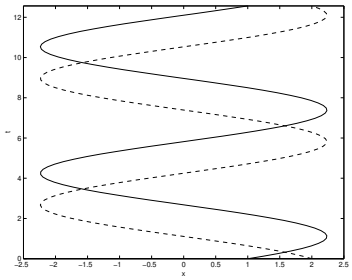


Abbildung 5.4: Harmonische Schwingung

Harmonischer Oszillator. Wir nehmen an, daß unser Objekt an einer Feder befestigt ist, deren Masse gegenüber m zu vernachlässigen ist. Wir wählen unser Koordinatensystem so, daß $x = 0$ der Ruhelage entspricht. Dann beschreibt $x_1(t)e_1$ die Auslenkung aus der Ruhelage $x = 0$. Wir nehmen an, daß wir es mit einer Feder zu tun haben, die dem *Hooke'sche Gesetz* genügt. Das Hooke'sche Gesetz lautet

$$x(t) \text{ ist proportional zur Rückstellkraft } -F(t)$$

oder

$$F(t) = -Dx(t). \tag{5.11}$$

Die Proportionalitätskonstante $D > 0$ heißt *Federkonstante*. Einsetzen von (5.11) in die Newton'sche Bewegungsleichung (5.7) liefert

$$mx'' = -Dx \tag{5.12}$$

oder gleichbedeutend

$$\begin{array}{rcl} x' & = & v \\ mv' & = & -Dx \end{array} \tag{5.13}$$

Für die Anfangsbedingungen (5.10) erhält man die Lösung

$$\begin{array}{rcl} x(t) & = & x_0 \cos \omega t + \frac{v_0}{\omega} \sin \omega t \\ v(t) & = & -\omega x_0 \sin \omega t + v_0 \cos \omega t \end{array} \qquad \omega = \sqrt{D/m}$$

Das ist eine *harmonische Schwingung*.

Beispiel:
Wir betrachten den Fall $m = D = 1$ und

$$x_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad v_0 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}$$

Dann ist $x_2(t) = x_3(t) = v_2(t) = v_3(t) = 0$ für alle $t \geq 0$ und Abbildung 5.4 zeigt $x_1(t)$ und $v_1(t)$ (gestrichelt) für $t \in [0, 4\pi]$. ◁

Für weitere Beispiele verweisen wir auf die reichhaltige Literatur, z.B. Deuflhard und Bornemann [1, Kapitel 1].

5.2 Existenz, Eindeutigkeit und Kondition

Vorgelegt sei das AWP

$$y'(t) = F(y,t),\ t \in (T_0, T_1], \quad y(T_0) = y_0, \tag{5.14}$$

mit gegebener Funktion

$$F : \mathbb{R}^{d+1} \to \mathbb{R}^d,$$

gegebenen Anfangsdaten

$$y_0 \in \mathbb{R}^d$$

und der gesuchten, einmal stetig differenzierbarer Funktion

$$y : [T_0, T_1] \to \mathbb{R}^d, \quad \text{kurz} \quad y \in C^1[T_0, T_1].$$

Ist $F(y,t) = g(t)$ *unabhängig von* y, so gilt

$$y(t) = y_0 + \int_{T_0}^{t} g(s)\, ds, \quad t \in [T_0, T_1]. \tag{5.15}$$

Die rechte Seite von (5.15) lässt sich numerisch mit Hilfe von bereits bekannten Quadraturformeln auswerten. Im allgemeinen erhält man

$$y(t) = y_0 + \int_{T_0}^{t} F\big(s, y(s)\big)\, ds, \quad t \in [T_0, T_1]. \tag{5.16}$$

Anfangswertprobleme lassen sich also als Quadraturaufgabe mit i.a. unbekanntem Integranden auffassen. Von dieser Interpretation werden wir später noch intensiv Gebrauch machen.

Definition 5.1 *Das AWP* (5.14) *heißt* autonom*, falls* F *nicht von* t *abhängt. Andernfalls heißt* (5.14) nicht–autonom.

In Abschnitt 5.1.1 haben wir nur autonome Systeme kennengelernt.

Bemerkung:
Sind $x \in C^1[T_0, T_1]$ und $\tilde{x} \in C^1[T_0+T, T_1+T]$ eindeutig bestimmte Lösungen der autonomen AWPe

$$x'(t) = f(x(t)),\ t \in (T_0, T_1], \quad x(T_0) = x_0,$$

und

$$\tilde{x}'(t) = f(\tilde{x}(t)),\ t \in (T_0 + T, T_1 + T], \quad \tilde{x}(T_0 + T) = x_0,$$

so gilt

$$\tilde{x}(t) = x(t - T) \quad \forall t \in [T_0 + T, T_1 + T]. \tag{5.17}$$

Autonome Systeme sind also *translationsinvariant*. ◁

Jedes nicht–automome AWP lässt sich autonomisieren:

Satz 5.2 *Sei $x : [T_0, T_1] \to \mathbb{R}^{d+1}$ eine Lösung des AWPs*

$$x'(t) = f(x(t)),\ t \in (T_0, T_1], \quad x(T_0) = x_0, \tag{5.18}$$

mit

$$\mathbb{R}^{d+1} \ni x \to f(x) = \begin{pmatrix} F(x) \\ 1 \end{pmatrix} \in \mathbb{R}^{d+1}, \quad x_0 = \begin{pmatrix} y_0 \\ T_0 \end{pmatrix} \in \mathbb{R}^{d+1}.$$

Ist dann y eine Lösung von (5.14)*, so ist*

$$x(t) = \begin{pmatrix} y(t) \\ t \end{pmatrix} \tag{5.19}$$

eine Lösung von (5.18)*. Ist umgekehrt x Lösung von* (5.18)*, so liefert* (5.19) *eine Lösung y von* (5.14)*.*

Beweis:
Übung. □

Mit Blick auf Satz 5.2 und die Translationseigenschaft (5.17) betrachten wir im folgenden nur autonome Systeme der Gestalt

$$x'(t) = f(x(t)),\ t \in (0, T], \quad x(0) = x_0, \tag{5.20}$$

mit

$$f : \mathbb{R}^d \to \mathbb{R}^d, \quad x_0 \in \mathbb{R}^d,$$

$d \in \mathbb{N}$ und $T \in \mathbb{R}$, $0 < T$. Als erstes untersuchen wir Existenz und Eindeutigkeit.

Satz 5.3 (Picard/Lindelöf) *Sei f Lipschitz–stetig. Dann existiert zu jedem Anfangswert $x_0 \in \mathbb{R}^d$ eine eindeutig bestimmte Lösung $x \in C^1[0, \infty)$ des AWPs* (5.20) *für alle $T > 0$.*

Beweis:
Ein Beweis findet sich beispielsweise bei Walter [3]. Unter der zusätzlichen Voraussetzung, daß T „hinreichend“ klein ist (Kontraktionsbedingung!), kann man den Beweis übrigens mit Hilfe des Banach'schen Fixpunktsatzes führen (Übung). □

Bemerkung:
Die Funktion f heißt *lokal Lipschitz–stetig*, falls für jede kompakte Teilmenge $K \subset \mathbb{R}^d$ eine Lipschitzkonstante L_K existiert, so daß die Abschätzung

$$\|f(x) - f(y)\| \le L_K \|x - y\| \quad \forall x, y \in K$$

erfüllt ist. Erinnerung: Gilt diese Abschätzung mit $\mathbb{R}^d$ anstelle von K, so liegt (globale) Lipschitz–Stetigkeit vor. Man kann Existenz und Eindeutigkeit auch im Falle von lokal Lipschitz–stetigem f zeigen, allerdings *nicht* für alle Zeiten $T > 0$. ◁

Beispiel:
Die Funktion

$$f(x) = x^2$$

ist zwar lokal, aber nicht global Lipschitz–stetig. Das entsprechende AWP

$$x'(t) = x(t)^2,\ t \in (0, T], \quad x = x_0,$$

hat die Lösung

$$x(t) = -\frac{x_0}{x_0 t - 1}.$$

Im Falle $x_0 \leq 0$ ist $x \in C^1[0, \infty)$. Ist aber $x_0 > 0$ so hat $x(t)$ bei $t_0 = 1/x_0$ eine Singularität. Unser Anfangsproblem ist dann nur für $T < t_0 < \infty$ lösbar. Man spricht von "blow up". ◁

Ist f nicht lokal Lipschitz–stetig, so braucht keine Eindeutigkeit vorzuliegen, wie das nächste Beispiel zeigt.

Beispiel:
Die Funktion

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

ist zwar stetig, aber nicht lokal Lipschitz–stetig in $x = 1$. Das entsprechende AWP

$$x' = -\frac{\sqrt{1 - x^2}}{x},\ t \in (0, 1], \quad x(0) = 1,$$

hat die zwei Lösungen

$$x_1(t) \equiv 1 \quad \text{und} \quad x_2(t) = \sqrt{1 - t^2},$$

denn

$$x_2'(t) = -\frac{t}{\sqrt{1 - t^2}} = -\frac{t}{x_2(t)} = -\frac{\sqrt{1 - x_2^2(t)}}{x_2(t)}.$$ ◁

Definition 5.4 *Das AWP* (5.20) *heißt* reversibel, *wenn das zugehörige Randwertproblem*

$$x'(t) = f(x(t)),\ x \in [0, T), \qquad x(T) = x_0, \tag{5.21}$$

für alle $x_0 \in \mathbb{R}^d$ *eine eindeutig bestimmte Lösung besitzt.*

Beispiel:
Das einfache Populationsmodell aus Abschnitt 5.1.1 ist reversibel. Das verfeinerte Populationsmodell ist nicht reversibel (Übung). ◁

Bemerkung:
Ist f global Lipschitz–stetig, so ist das AWP (5.20) reversibel (Übung). ◁

Einen umfassenderen Überblick über typische Eigenschaften des AWPs (5.20) und seine Lösungen findet man beispielweise bei Deuflhard und Bornemann [1, Kapitel 2].
Bevor wir die Kondition eines AWP untersuchen, führen wir noch sogenannte *Flußoperatoren* ein.

Definition 5.5 *Für ein $x_0 \in \mathbb{R}^d$ und $T_{-1} \leq 0 \leq T_1$ habe das Problem*

$$x'(t) = f(x(t)),\ t \in [T_{-1}, T_1], \quad x(0) = x_0, \tag{5.22}$$

die eindeutig bestimmte Lösung $x \in C^1[T_{-1}, T_1]$. Dann ist x_0 für jedes $t \in [T_{-1}, T_1]$ im Definitionsbereich $\mathcal{D}(\phi^t)$ des Flußoperators ϕ^t,

$$\phi^t : \mathcal{D}(\phi^t) \subset \mathbb{R}^d \to \mathbb{R}^d.$$

Für jedes $t \in [T_{-1}, T_1]$ ist der entsprechende Fluß $\phi^t x_0$ definiert durch

$$\phi^t x_0 = x(t) \in \mathbb{R}^d.$$

Beispiel:
Im Falle des einfachen Populationsmodells (5.5) ist der Flußoperator ϕ^t für alle $x_0 \in \mathbb{R}$ und alle $t \in \mathbb{R}$ definiert. Es gilt

$$\phi^t x_0 = e^{pt} x_0. \qquad \triangleleft$$

Bemerkung:
Für jedes feste x_0 ist $\phi^t x_0 : [T_{-1}, T_1] \to \mathbb{R}^d$ eine vektorwertige Abbildung. Nach Definition ist diese Abbildung differenzierbar und es gilt

$$\frac{d}{dt}\phi^t x_0 = x'(t) = f(x(t)) = f(\phi^t x_0). \tag{5.23}$$

$\triangleleft$

Satz 5.6 *Verknüpft man durch Hintereinanderausführung, so gilt mit* $\mathrm{id} = \phi^0$

$$\mathrm{id}\phi^t = \phi^t \mathrm{id} = \phi^t.$$

Im Falle $T_{-1} = 0$ existiere zu jedem Anfangswert $x_0 \in \mathbb{R}^d$ eine eindeutig bestimmte Lösung $x \in C^1[0, \infty)$ von (5.22) für alle $T_1 > 0$. Dann ist $\mathcal{D}(\phi^t) = \mathbb{R}^d$, $\forall t \geq 0$ und die Hintereinanderausführung von Flußoperatoren hat die Halbgruppeneigenschaft

$$\phi^{s+t} = \phi^s \phi^t \quad \forall s, t \geq 0.$$

Es existiere zu jedem Anfangswert $x_0 \in \mathbb{R}^d$ eine eindeutig bestimmte Lösung $x \in C^1(-\infty, \infty)$ von (5.22) für alle $T_{-1} < 0 < T_1$. Dann ist $\mathcal{D}(\phi^t) = \mathbb{R}^d$, $\forall t \in \mathbb{R}$ und zu jedem Flußoperator ϕ^t gibt es den inversen Flußoperator ϕ^{-t} mit der Eigenschaft

$$\phi^t \phi^{-t} = \mathrm{id}.$$

Die Menge der Flußoperatoren

$$\Phi = \{\phi^t \mid t \in \mathbb{R}\}$$

ist dann eine abelsche Gruppe bzgl. der Hintereinanderausführung.

Beweis:
Übung. □

Wir kommen nun zur Definition der Kondition des AWPs (5.20). Dabei nehmen wir der Einfachheit halber an, daß (5.20) für jeden Anfangswert $x_0 \in \mathbb{R}^d$ und alle $T > 0$ eindeutig lösbar ist.

Definition 5.7 *Sei $\|\cdot\|$ eine Vektornorm auf $\mathbb{R}^d$. Dann ist für jedes feste $t > 0$ die (punktweise) Kondition $\kappa(t)$ die kleinste Zahl $\kappa(t)$ mit der Eigenschaft*

$$\|\phi^t(x_0 + \Delta x_0) - \phi^t x_0\| \leq \kappa(t)\|\Delta x_0\| + o(\|\Delta x_0\|) \quad \textit{für } \Delta x_0 \to 0. \tag{5.24}$$

Die Kondition misst die Verstärkung eines Fehlers in den Anfangsdaten x_0.

Beispiel:
Wegen

$$|\phi^t(x_0 + \Delta x_0) - \phi^t x_0| = e^{pt}|\Delta x_0|$$

ist die Kondition des einfachen Populationsmodells (5.5)

$$\kappa(t) = e^{pt}. \qquad \triangleleft$$

Definitionsgemäß ist $\phi^t x$ für jedes feste t eine Funktion von $x \in \mathbb{R}^d$ mit Werten in $\mathbb{R}^d$. Wir nehmen an, daß diese Abbildung in x_0 differenzierbar ist. Aus der Definition der Ableitung $D\phi^t x_0 \in \mathbb{R}^{d,d}$ (Jacobi–Matrix) und der Dreiecksungleichung folgt sofort

$$\|\phi^t(x_0 + \Delta x_0) - \phi^t x_0\| \leq \|D\phi^t x_0\|\|\Delta x_0\| + o(\|\Delta x_0\|) \quad \text{für } \Delta x_0 \to 0.$$

Also ist

$$\kappa(t) \leq \|D\phi^t x_0\|.$$

Die Ableitung $W(t) = D\phi^t x_0$ nach x_0 heißt *Wronski–Matrix*.
Ist f stetig differenzierbar, so folgt aus (5.23) und der Kettenregel die Differentialgleichung

$$\frac{d}{dt}W(t) = D\frac{d}{dt}\phi^t x_0 = Df(\phi^t x_0) = f'(\phi^t x_0)W(t). \tag{5.25}$$

Zusammen mit der Anfangsbedingung

$$W(0) = I \qquad \text{(Einheitsmatrix in } \mathbb{R}^d) \tag{5.26}$$

erhält man ein lineares Anfangswertproblem zur Berechnung der Wronski–Matrix. Leider hängt die Koeffizientenmatrix $f'(\phi^t x_0)$ von der unbekannten Lösung $x(t) = \phi^t x_0$ ab. Damit hat man es mit einem gekoppelten AWP bestehend aus (5.20) und dem System (5.25), (5.26) zu tun.
Wir wollen nun eine besser zugängliche, dafür leider auch gröbere Abschätzung für die Kondition $\kappa(t)$ herleiten. Dabei setzen wir voraus, daß f global Lipschitz–stetig mit Lipschitz–Konstante L ist. Unter Verwendung von (5.23) und des Hauptsatzes der Differential– und Integralrechnung erhält man

$$\phi^t x_0 = x_0 + \int_0^t f(\phi^s x_0)\, ds.$$

Somit gilt

$$\phi^t(x_0 + \Delta x_0) - \phi^t x_0 = \Delta x_0 + \int_0^t \big(f(\phi^s(x_0 + \Delta x_0)) - f(\phi^s x_0)\big)\, ds.$$

Nun nehmen wir auf beiden Seiten die Norm und nutzen Dreiecksungleichung und Lipschitz–Stetigkeit von f. So ergibt sich

$$\begin{aligned} \|\phi^t(x_0 + \Delta x_0) - \phi^t x_0\| &\leq \|\Delta x_0\| + \int_0^t \|f(\phi^s(x_0 + \Delta x_0)) - f(\phi^s x_0)\| ds \\ &\leq \|\Delta x_0\| + L \int_0^t \|\phi^s(x_0 + \Delta x_0) - \phi^s x_0\| ds \end{aligned} \tag{5.27}$$

Um die rechte Seite in (5.27) weiter abzuschätzen, benötigen wir das sogenannte *Gronwall-Lemma:*

Lemma 5.8 *Sei* $g : [a, b] \to \mathbb{R}$ *stetig und* $\alpha, \beta > 0$.
Dann folgt aus

$$g(t) \leq \alpha + \beta \int_a^t g(s) ds \qquad \forall t \in [a, b] \tag{5.28}$$

die Abschätzung

$$g(t) \leq \alpha e^{\beta(t-a)} \qquad \forall t \in [a, b] \tag{5.29}$$

Beweis:
Sei $\varepsilon > 0$ beliebig, aber fest gewählt. Wir setzen

$$G(t) = (\alpha + \varepsilon) e^{\beta(t-a)}, \qquad G'(t) = \beta G(t).$$

Dann ist entweder

$$g(t) < G(t) \qquad \forall t \in [a, b] \tag{5.30}$$

oder

$$M = \{t \in [a, b] \mid g(t) \geq G(t)\} \neq \emptyset.$$

Wir nehmen an, daß der zweite Fall, also $M \neq \emptyset$ vorliegt. Aufgrund der Stetigkeit von g und G ist M abgeschlossen, also kompakt. Daher gibt es ein kleinstes Element $t_0 \in \mathbb{R} \cap M$ von M. Wegen

$$g(a) = \alpha < \alpha + \varepsilon = G(a)$$

ist $a \notin M$, also $a < t_0$. Da t_0 das kleinste Element von M ist, muß

$$g(t) < G(t) \quad \forall t \in [a, t_0)$$

sein. Daraus folgt mit $\beta > 0$ die Abschätzung

$$\begin{aligned} g(t_0) \leq \;& \alpha + \beta \int_a^{t_0} g(s)\, ds \\ < \;& \alpha + \beta \int_a^{t_0} G(s)\, ds = \alpha + \int_a^{t_0} G'(s)\, ds = \alpha + G(t_0) - G(a) = G(t_0) - \varepsilon \end{aligned}$$

im Widerspruch zu $t_0 \in M$! Also muß $M = \emptyset$ sein. Grenzübergang $\varepsilon \to 0$ in (5.30) liefert die Behauptung. □

Satz 5.9 *Es sei f Lipschitz–stetig mit Lipschitzkonstante L. Dann gilt*

$$\kappa(t) \leq e^{Lt} \qquad \forall t \geq 0.$$

Beweis:
Setzt man

$$g(t) = \|\phi^t(x_0 + \Delta x_0) - \phi^t x_0\|, \quad t \geq 0,$$

so folgt die Behauptung sofort aus (5.27) und Lemma 5.8. □

Bemerkung:
Die obere Schranke aus Satz 5.9 ist zwar scharf (vgl. das einfache Populationsmodell (5.5)), kann die Kondition aber stark überschätzen. So ist für die Zerfallsgleichung (5.2)

$$\kappa(t) = e^{-pt} \ll e^{pt}.$$ ◁

Neben der punktweisen Kondition $\kappa(t)$ betrachtet man auch die *intervallweise Kondition*

$$\kappa[0,t] = \max_{s \in [0,t]} \kappa(s) \tag{5.31}$$

Das AWP (5.20) ist *gut konditioniert*, falls $\kappa[0,T]$ „klein“ ist.

5.3 Euler–Verfahren

Wir wollen Diskretisierungsverfahren zur näherungsweisen Lösung des AWPs (5.20) entwickeln. Dabei setzen wir der Einfachheit halber voraus, daß $f \in C(\mathbb{R}^d)$ und daß für jeden Anfangswert $x_0 \in \mathbb{R}^d$ eine eindeutig bestimmte Lösung $x \in C^1(0,\infty)$ von (5.20) existiert. Nach Satz 5.3 ist dafür die globale Lipschitz–Stetigkeit von f hinreichend (aber nicht notwendig).
Wir wollen die unbekannte Lösung $x(t)$ durch eine stückweise lineare Funktion annähern. Dazu wählen wir zunächst ein Gitter

$$\Delta = \{0 = t_0 < t_1 < \ldots < t_n = T\}$$

mit den Schrittweiten

$$\tau_k = t_{k+1} - t_k, \quad k = 0, \ldots, n-1.$$

Die maximale Schrittweite bezeichnen wir mit

$$\tau_\Delta = \max_{k=0,\ldots,n-1} \tau_k.$$

Wegen $x'(0) = f(x(0)) = f(x_0)$ ist die Tangente an $x(t)$ in $t_0 = 0$ gegeben durch

$$\psi^\tau x_0 = x_0 + \tau f(x_0).$$

Einsetzen von $\tau = \tau_0$ liefert

$$x_1 = \psi^{\tau_0} x_0 = x_0 + \tau_0 f(x_0).$$

Auf die gleiche Weise können wir $x_2, \ldots, x_n$ berechnen und erhalten so das *explizite Euler–Verfahren*

$$x_{k+1} = \psi^{\tau_k} x_k, \quad k = 0, \ldots, n-1, \quad x_0 \in \mathbb{R}^d \text{ gegeben,}$$

mit

$$\psi^\tau x = x + \tau f(x), \quad x \in \mathbb{R}^d, \ \tau \geq 0. \tag{5.32}$$

Beachte, daß sich der neue Gitterfehler $\|x(t_{k+1}) - x_{k+1}\|$ an der Stelle t_{k+1} gemäß

$$\|x(t_{k+1}) - x_{k+1}\| \leq \|\phi^{\tau_k} x(t_k) - \psi^{\tau_k} x(t_k)\| + \|\psi^{\tau_k} x(t_k) - \psi^{\tau_k} x_k\| \tag{5.33}$$

aus zwei Beiträgen zusammensetzt: Dem

$$\text{Konsistenzfehler:} \quad \|\phi^{\tau_k} x(t_k) - \psi^{\tau_k} x(t_k)\|$$

durch Linearisierung im aktuellen Zeitschritt und der

$$\text{Fehlerfortpflanzung:} \quad \|\psi^{\tau_k} x(t_k) - \psi^{\tau_k} x_k\|$$

des alten Gitterfehlers.
Das Euler–Verfahren liefert die Gitterfunktion

$$x_\Delta(t_k) = x_k, \quad k = 0, \ldots, n.$$

Der *Diskretisierungsfehler* ist definiert durch

$$\|x - x_\Delta\|_\infty = \max_{k=0,\ldots,n} \|x(t_k) - x_k\|.$$

Natürlich hoffen wir, daß

$$\|x - x_\Delta\|_\infty \to 0 \quad \text{für } \tau_\Delta \to 0.$$

Beispiel:
Wir diskretisieren das AWP

$$x'(t) = x(t), \ t \in (0, 2], \quad x(0) = 1$$

mit der exakten Lösung $x(t) = e^x$ mit dem expliziten Euler–Verfahren. In Abbildung 5.5 sieht man links den neuen Konsistenzfehler nebst Fortpflanzung des Fehlers aus dem ersten Zeitschritt für die unrealistische große Schrittweite $\tau_\Delta = 0.5$. Auf der rechten Seite ist der Diskretisierungsfehler $\|x - x_\Delta\|_\infty$ in Abhängigkeit von der Schrittweite $\tau_\Delta = 1/n$ dargestellt. Wir beobachten

$$\|x - x_\Delta\|_\infty = \mathcal{O}(\tau_\Delta). \qquad \triangleleft$$

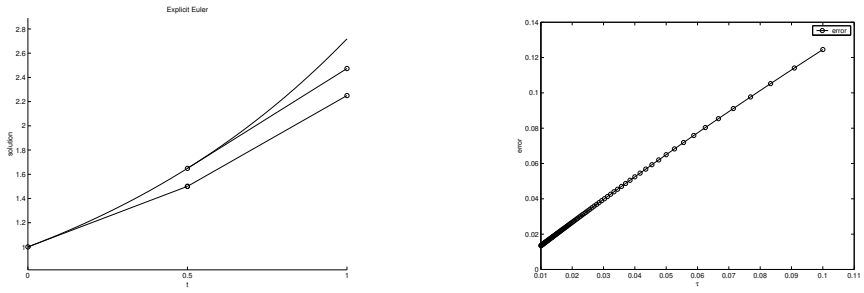


Abbildung 5.5: Explizites Euler–Verfahren

Das explizite Euler–Verfahren lässt sich auch als Diskretisierung der Ableitung x' durch den vorwärtsgenommenen Differenzenquotienten interpretieren. Auf diese Weise erhält man nämlich

$$\frac{x_{k+1} - x_k}{\tau_k} = f(x_k), \quad k = 0, \ldots, n-1,$$

und Auflösung nach x_{k+1} liefert (5.32).
Verwendet man stattdessen den rückwärtsgenommenen Differenzenquotienten, so ergibt sich das *implizite Euler–Verfahren*

$$x_{k+1} = \psi^{\tau_k} x_k, \quad k = 0, \ldots, n-1, \qquad \psi^\tau x = x + \tau f(\psi^\tau x), \quad x \in \mathbb{R}^d,\ \tau \in [0, \tau_0]. \tag{5.34}$$

Beachte, daß die Werte des diskreten Flußoperators diesmal nicht explizit gegeben sind. Die Auswertung von $\psi^{\tau_k} x$ erfordert die Lösung eines nichtlinearen Gleichungssystems.

Beispiel:
Wir betrachten das AWP

$$x'(t) = x(t)^2, \quad x(0) = x_0 < 0.$$

Anwendung des impliziten Euler–Verfahrens liefert im ersten Zeitschritt die quadratische Gleichung

$$x_1 = x_0 + \tau_0 x_1^2$$

mit den zwei Lösungen

$$x_1^\pm = \frac{1}{2\tau_0} \left(1 \pm \sqrt{1 - 4\tau_0 x_0}\right).$$

Es ist nicht klar, welche der beiden Lösungen man nehmen soll. ◁

Trotz dieser Schwierigkeiten sind implizite Diskretisierungen für AWPs mit Zusatzstruktur (Stichwort: dissipative Differentialgleichungen [1, Abschnitt 6.3.3]) genau die richtige Verfahrensklasse. Bevor wir dazu kommen, sind aber noch eine Reihe wichtigerer Fragen zu klären. Wir untersuchen also im folgenden ausschließlich explizite Verfahren und erwähnen implizite Verfahren nur am Rande. Ungeduldige verweisen wir auf Deuflhard und Bornemann [1, Kapitel 6].

5.4 Konsistenz von Einschrittverfahren

Bei der numerischen Lösung des AWPs (5.20) gehen wir aus von dem Gitter

$$\Delta = \{0 = t_0 < t_1 < \ldots < t_{n-1} < t_n = T\}$$

mit den zugehörigen Schrittweiten

$$\tau_k = t_{k+1} - t_k, \quad k = 0, \ldots, n-1, \quad \tau_\Delta = \max_{k=0,\ldots,n-1} \tau_k.$$

Durch eine Familie diskreter Flußoperatoren

$$\psi^\tau : \mathbb{R}^d \to \mathbb{R}^d, \quad \tau \geq 0, \tag{5.35}$$

wird ein *explizites Einschrittverfahren*

$$x_{k+1} = \psi^{\tau_k} x_k, \quad k = 0, \ldots, n-1, \quad x_0 \in \mathbb{R}^d \text{ gegeben}, \tag{5.36}$$

charakterisiert. Wir sprechen daher häufig kurz vom „Verfahren ψ^τ“. Die Anwendung eines diskreten Flußoperators ψ^τ auf ein $x \in \mathbb{R}^d$ ergibt den *diskreten Fluß* $\psi^\tau x$, $\tau \geq 0$.
Mit Blick auf (5.33) ist eine *notwendige* Voraussetzung für die Konvergenz der resultierenden Gitterfunktion

$$x_\Delta(t_k) = x_k, \quad k = 0, \ldots, n,$$

gegen die kontinuierliche Lösung $x(t)$, $t \in [0, T]$, daß der diskrete Flußoperator ψ^τ den exakten Flußoperator ϕ^τ mit immer kleiner werdender Schrittweite τ_Δ immer besser approximiert.

Definition 5.10 *Die Differenz von kontinuierlichem und diskretem Fluß*

$$\varepsilon(x, \tau) = \phi^\tau x - \psi^\tau x \quad x \in \mathbb{R}^d, \ \tau \geq 0,$$

heißt Konsistenzfehler. Das Verfahren ψ^τ heißt konsistent mit dem AWP (5.20), wenn

$$\lim_{\tau \to 0} \frac{\|\varepsilon(x, \tau)\|}{\tau} = 0 \quad \forall x \in \mathbb{R}^d.$$

Das Verfahren ψ^τ heißt konsistent mit Konsistenzordnung p, wenn für jedes Kompaktum $K \subset \mathbb{R}^n$ eine Konstante $C_K(f) > 0$ und ein $\tau^* > 0$ existieren, so daß gilt

$$\|\varepsilon(x, \tau)\| \leq C_K(f) \tau^{p+1} \quad \forall x \in K \quad \forall \tau \leq \tau^*. \tag{5.37}$$

Als Beispiel betrachten wir das explizite Euler–Verfahren.

Satz 5.11 *Es sei $f \in C^1(\mathbb{R}^d)$. Dann ist das explizite Euler–Verfahren (5.32) konsistent mit Konsistenzordnung $p = 1$.*

Beweis:
Seien $K \subset \mathbb{R}^d$ kompakt und $x \in K$ beliebig, aber fest gewählt. Der Beweis beruht auf einer Taylor–Entwicklung des Konsistenzfehlers $\varepsilon(x,\tau) = \phi^\tau x - \psi^\tau x$ in τ um $\tau = 0$. Bekanntlich gilt mit geeignetem $s = s(\tau) \in (0,1)$

$$\phi^\tau x = \phi^\tau x|_{\tau=0} + \tau \frac{d}{d\tau}\phi^\tau x|_{\tau=0} + \frac{\tau^2}{2}\frac{d^2}{d\tau^2}\phi^\theta x|_{\theta=s\tau}.$$

Offenbar ist

$$\phi^\tau x|_{\tau=0} = \phi^0 x = x.$$

Wir haben die Ableitungen $\frac{d}{d\tau}\phi^\tau x$ und $\frac{d^2}{d\tau^2}\phi^\tau$ zu berechnen. Aus (5.23) folgt

$$\frac{d}{d\tau}\phi^\tau x = f(\phi^\tau x),$$

also

$$\frac{d}{d\tau}\phi^\tau x|_{\tau=0} = f(\phi^0 x) = f(x).$$

Die Kettenregel liefert

$$\frac{d^2}{d\tau^2}\phi^\tau x = \frac{d}{d\tau} f(\phi^\tau x) = f'(\phi^\tau x)\frac{d}{d\tau}\phi^\tau x = f'(\phi^\tau x) f(\phi^\tau x),$$

also

$$\frac{d^2}{d\tau^2}\phi^\tau x|_{\tau=\theta} = f'(\phi^\theta x) f(\phi^\theta x).$$

Wegen $\psi^\tau x = x + \tau f(x)$ (vgl. (5.32)) erhält man

$$\phi^\tau x - \psi^\tau x = \frac{\tau^2}{2} f'(\phi^\theta x) f(\phi^\theta x)|_{\theta=s\tau} \quad s = s(\tau) \in (0,1). \tag{5.38}$$

Nun wählen wir ein weiteres Kompaktum $\hat{K}$ mit $K \subset \operatorname{int} \hat{K}$. Dann ist $\|x - y\| \geq \delta > 0$ für alle y auf dem Rand von $\hat{K}$. Die Lösung $z(t) = \phi^t x$ von (5.20) zum Anfangswert $x \in K$ ist stetig auf $[0,\infty)$. Daher existiert ein $\tau^* > 0$, so daß

$$z(t) = \phi^t x \in \hat{K} \quad \forall t \in [0,\tau^*]. \tag{5.39}$$

Nun folgt aus (5.38)

$$\|\phi^\tau x - \psi^\tau x\| \leq \frac{\tau^2}{2} \max_{t\in[0,\tau^*]} \|f'(\phi^t x) f(\phi^t x)\| \leq \frac{\tau^2}{2} \max_{y\in\hat{K}} \|f'(y) f(y)\| \quad \forall \tau \in [0,\tau^*].$$

Das ist gerade die Behauptung mit

$$C_K(f) = \frac{1}{2} \max_{y\in\hat{K}} \|f'(y) f(y)\|. \qquad \square$$

Aus Kapitel 4 wissen wir, daß im Falle glatter Integranden Quadraturformeln höherer Ordnung aus Effizienzgründen vorzuziehen sind. Bei der Diskretisierung von AWPs ist das ganz genauso. (Im Unterschied zur Quadratur kennen wir allerdings den Integranden $x'(t) = f(x(t))$ nicht). Wir wollen daher Diskretisierungsverfahren höherer Ordnung konstruieren. Als erstes betrachten wir die sogenannte

Methode der Taylor–Entwicklung. Kernstück des obigen Konsistenzbeweises für das explizite Euler–Verfahren ist (5.38): Der diskrete Euler–Fluß $\psi^\tau x = x + \tau f(x)$ besteht gerade aus den beiden führenden Termen der Taylorentwicklung von $\phi^\tau x$. Bei entsprechender Differenzierbarkeit von f erhalten wir Verfahren höherer Ordnung, indem wir einfach weitere Terme der Taylorentwicklung von $\phi^\tau x$ zum diskreten Flußoperator hinzunehmen.
Als Beispiel wollen wir auf diese Weise ein Taylor–Verfahren 2. Ordnung konstruieren. Es gilt offenbar

$$\phi^\tau x = x + \tau f(x) + \frac{\tau^2}{2} f'(x)f(x) + \frac{\tau^3}{3!}\frac{d^3}{d\tau^3}\phi^\theta x|_{\theta=s\tau}, \quad s = s(\tau) \in (0,1).$$

Ist $f \in C^2(\mathbb{R}^d)$, so beweist man genau wie oben, daß das Taylor–Verfahren

$$\psi^\tau x = x + \tau f(x) + \frac{\tau^2}{2} f'(x)f(x), \quad \tau \geq 0, \tag{5.40}$$

die Konsistenzordnung $p = 2$ hat.

Beispiel:
Wir betrachten das AWP

$$x' = e^{-x},\ t \in (0,1], \quad x(0) = 1,$$

mit der exakten Lösung $x = \log(t+e)$. Unser Taylor–Verfahren mit diskretem Flußoperator ψ^τ aus (5.40) hat dann die Gestalt

$$x_{k+1} = x_k + \tau e^{-x_k} - \frac{\tau^2}{2} e^{-2x_k}.$$

Wir wählen die konstante Schrittweite $\tau_\Delta = \frac{1}{n}$ und somit $t_k = k\tau_\Delta$, $k = 0, \ldots, n$. Abbildung 5.6 zeigt links die exakteLösung x im Vergleich mit den Näherungslösungen x_Δ^{Taylor} bzw. x_Δ^{Euler}, die wir mit dem obigen Taylor–Verfahren 2. Ordnung bzw. mit dem expliziten Euler–Verfahren zur Schrittweite $\tau = 1/5$ erhalten. Schon mit bloßem Auge erkennt man eine deutlich höhere Genauigkeit des Taylor–Verfahrens.

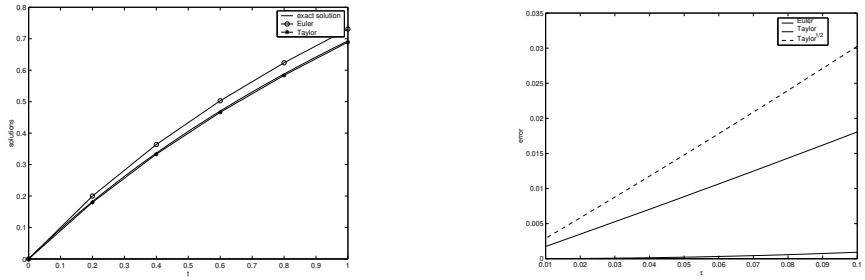


Abbildung 5.6: Explizites Euler–Verfahren und Taylor–Verfahren 2. Ordnung

Im rechten Bild sind die Diskretisierungsfehler $\|x - x_\Delta^{Taylor}\|_\infty$ und $\|x - x_\Delta^{Euler}\|_\infty$ in Abhängigkeit von τ_Δ abgebildet. Offenbar ist wieder

$$\|x - x_\Delta^{Euler}\|_\infty = \mathcal{O}(\tau_\Delta).$$

Die gestrichelten Linie zeigt $\|x - x_\Delta^{Taylor}\|_\infty^{1/2} = \mathcal{O}(\tau_\Delta)$, also

$$\|x - x_\Delta^{Taylor}\|_\infty = \mathcal{O}(\tau_\Delta^2).$$

Die höhere Konsistenzordnung spiegelt sich also direkt im Konvergenzverhalten wieder. ◁

Leider ist die Berechnung von f' nicht immer so einfach wie im obigen Beispiel. Im Systemfall $d > 1$ müssen d^2 skalare Ableitungen gebildet werden. Die Lage verschlimmert sich mit wachsender Ordnung. So erfordert das Taylor–Verfahren 3. Ordnung die Auswertung von

$$\frac{d^3}{d\tau^3}\phi^\tau x = \frac{d}{d\tau}\left(f'(\phi^\tau x)f(\phi^\tau x)\right) = f''(\phi^\tau x)(f(\phi^\tau x), f(\phi^\tau x)) + f'(\phi^\tau x)^2 f(\phi^\tau x),$$

also die Kenntnis von f''. Dazu müssen weitere d^3 skalare Ableitungen gebildet werden. Es gibt aber Verfahren beliebig hoher Ordnung, die völlig ohne Ableitungen auskommen. Taylor–Verfahren spielen daher in der Praxis keine Rolle.

5.5 Runge–Kutta–Verfahren

5.5.1 Allgemeine Form und klassische Beispiele

Grundlage für die Konstruktion von Runge–Kutta–Verfahren ist die Integraldarstellung (5.16). Für das AWP (5.20) lautet sie in Flußoperatorschreibweise

$$\phi^\tau x = x + \int_0^\tau f(\phi^s x)\, ds, \quad x \in \mathbb{R}^d. \tag{5.41}$$

Wir verfügen bereits über Quadraturformeln, die wir auf das Integral auf der rechten Seite anwenden können. Als Beispiel betrachten wir die Mittelpunktsregel

$$\int_0^\tau f(\phi^s x)\, ds \approx \tau f(\phi^{\tau/2} x). \tag{5.42}$$

Als nächstes müssen wir den unbekannten Wert $f(\phi^{\tau/2}x)$ durch eine geeignete Approximation ersetzen. Wir wählen dazu das explizite Euler–Verfahren

$$f(\phi^{\tau/2}x) \approx f(x + \tfrac{\tau}{2} f(x)). \tag{5.43}$$

Insgesamt erhalten wir das *Verfahren von Runge* (1895)

$$\psi^\tau x = x + \tau f\big(x + \tfrac{\tau}{2} f(x)\big), \quad x \in \mathbb{R}^d,\ \tau \geq 0. \tag{5.44}$$

Wir werden später sehen, daß das Verfahren von Runge tatsächlich die Konsistenzordnung $p = 2$ hat.

Bemerkung:
Das Verfahren von Runge ist explizit. Es benötigt in jedem Zeitschritt zwei f-Auswertungen und keine Auswertung von Ableitungen von f. Damit ist es bei gleicher Konsistenzordnung erheblich billiger als das Taylor–Verfahren (5.40). ◁

Kutta (1901) hatte die Idee, Runges Ansatz zu allgemeineren Schachtelungen von f-Auswertungen zu erweitern.

Definition 5.12 *Ein diskreter Flußoperator der Form*

$$\psi^\tau x = x + \tau \sum_{i=1}^{s} b_i k_i \tag{5.45}$$

mit

$$k_i = f\big(x + \tau \sum_{j=1}^{s} a_{ij} k_j\big), \quad i = 1, \ldots, s, \tag{5.46}$$

heißt Runge–Kutta–Verfahren s–ter Stufe.

Hinter dem Ansatz (5.45) steht wie z.B. in (5.42) eine Quadraturformel für das Integral (5.16). Die unbekannten Werte k_i an den Stützstellen werden entsprechend (5.46) durch sukzessive f–Auswertungen approximiert. Man nennt k_i die *i–te Stufe* des Runge–Kutta–Verfahrens. Wir setzen

$$b = \begin{pmatrix} b_1 \\ \vdots \\ b_s \end{pmatrix} \in \mathbb{R}^s, \quad \mathcal{A} = \begin{pmatrix} a_{11} & \cdots & a_{1s} \\ \vdots & & \vdots \\ a_{s1} & \cdots & a_{ss} \end{pmatrix} \in \mathbb{R}^{s,s}.$$

Damit ist ein Runge–Kutta–Verfahren durch das folgende Butcher–Schema charakterisiert (Butcher 64).

$$\begin{array}{c|c} & \mathcal{A} \\ \hline & b^T \end{array} \tag{5.47}$$

Bemerkung:
Ist

$$a_{ij} = 0 \quad \forall j \geq i, \tag{5.48}$$

so kann man die einzelnen Stufen k_i,

$$k_i = f\left(x + \tau \sum_{j=1}^{i-1} a_{ij} k_j\right), \quad i = 1, \ldots, s,$$

rekursiv auswerten. Runge–Kutta–Verfahren mit der Eigenschaft (5.48) sind also explizite Verfahren. ◁

Beispiel: (Explizite Runge–Kutta–Verfahren)

- Das explizite Euler–Verfahren (5.32) ist ein explizites Runge–Kutta–Verfahren 1. Stufe,

$$\psi^\tau x = x + \tau \cdot 1 \cdot k_1, \quad k_1 = f(x + 0 \cdot \tau k_1),$$

mit dem Butcher–Schema

$$\begin{array}{c|c} 0 \\ \hline 1 \end{array}$$

- Das Verfahren von Runge (5.44) ist ein explizites Runge–Kutta–Verfahren 2. Stufe

$$\psi^\tau x = x + \tau k_2, \quad k_1 = f(x), \quad k_2 = f\big(x + \tau\tfrac{1}{2}k_1\big)$$

mit dem Butcher–Schema

0	0	
1/2	0	
0	1	

- Das klassische *Runge–Kutta–Verfahren* ist 4–stufig, heißt daher Runge–Kutta–4 und ist charakterisiert durch das Butcher–Schema

0				
1/2	0			
0	1/2	0		
0	0	1	0	
1/6	1/3	1/3	1/6	

Der Übersichtlichkeit halber haben wir die Nullen oberhalb der Diagonalen von $\mathcal{A}$ weggelassen. Welche bekannte Quadraturformel steckt hinter diesem Verfahren (Übung)?◁

Beispiel:
Wir betrachten das AWP

$$x' = 1 + x^2,\ t \in (0,1], \quad x(0) = 0,$$

mit der exakten Lösung $x = \tan(x)$. Zur approximativen Lösung wollen wir das explizite Euler–Verfahren, das Verfahren von Runge und Runge–Kutta–4 auf äquidistanten Gittern verwenden. Indem wir das Intervall $[0,1]$ so lange wie nötig halbieren, bestimmen wir für jedes Verfahren eine zugehörige Schrittweite so, daß der entsprechende Diskretisierungsfehler kleiner als eine vorgegebene Schranke TOL ist.
Abbildung 5.7 zeigt links die exakte Lösung x im Vergleich mit den Näherungslösungen x_Δ^{Euler}, x_Δ^{Runge} und x_Δ^{RK4} zur Toleranz $TOL = 10^{-3}$. Beachte, daß Runge–Kutta–4 mit der Schrittweite $\tau_\Delta^{RK4} = 2^{-2}$ auskommt, während $\tau_\Delta^{Runge} = 2^{-6} \approx \big(\tau_\Delta^{RK4}\big)^2$ und $\tau_\Delta^{Euler} = 2^{-12} = \big(\tau_\Delta^{Runge}\big)^2$ ausfällt. Die höhere Konsistenzordnung zahlt sich also aus.

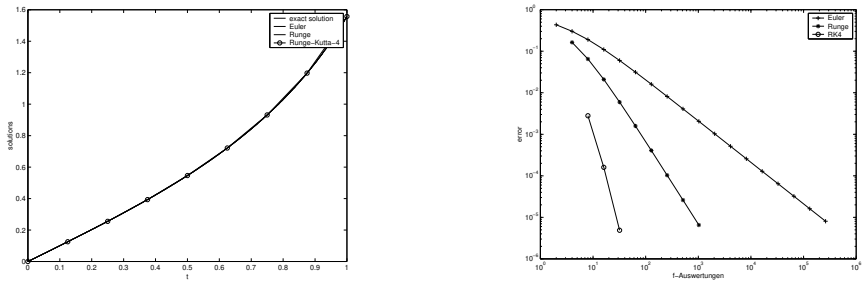


Abbildung 5.7: Euler, Runge und Runge–Kutta–4

Zum Erreichen der Genauigkeit $TOL = 10^{-5}$ benötigt Runge–Kutta–4 die Schrittweite $\tau_\Delta^{RK4} = 2^{-3}$, das Verfahren von Runge die Schrittweite $\tau_\Delta^{Runge} = 2^{-9}$ und das explizite Euler–Verfahren die Schrittweite $\tau_\Delta^{Euler} = 2^{-18}$. Erneut spiegelt sich die Konsistenzordnung in der Diskretisierungsgenauigkeit wieder. Im rechten Bild sieht man die Diskretisierungsfehler in Abhängigkeit von den benötigten f–Auswertungen. Im Vergleich mit den $N^{RK4} = 32$ f–Auswertungen von Runge–Kutta–4 benötigt das Verfahren von Runge $1024 = (N^{RK4})^2$ f–Auswertungen und das Euler–Verfahren gar $262144 \approx (N^{RK4})^4$ f–Auswertungen. Die höhere Konsistenzordnung spiegelt sich also auch in exponentiellem Effizienzgewinn wieder. ◁

Beispiel: (Implizite Runge-Kutta-Verfahren)

- Das implizite Euler–Verfahren (5.34) ist ein implizites Runge–Kutta–Verfahren 1. Stufe,

  $$\psi^\tau x = x + \tau k_1, \quad k_1 = f(x + \tau k_1)$$

  mit dem Butcher–Schema

  $$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

  In jedem Zeitschritt ist die Lösung eines nichtlinearen Gleichungssystems für k_1 erforderlich.

- Anwendung der Trapezregel aus Kapitel 4 auf das Integral (5.16) liefert

  $$\psi^\tau x = x + \tau \tfrac{1}{2}\big(f(x) + f(\psi^\tau x)\big).$$

  Dies ist ein implizites Runge–Kutta–Verfahren 2. Stufe, die *implizite Trapezregel*. Es gilt nämlich

  $$\psi^\tau x = x + \tau \tfrac{1}{2}\big(k_1 + k_2\big), \quad k_1 = f(x), \quad k_2 = f\big(x + \tau \tfrac{1}{2}(k_1 + k_2)\big).$$

  Daraus erhält man das Butcher–Schema

  $$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array}$$

  In jedem Zeitschritt ist die Lösung eines nichtlinearen Gleichungssystems für k_2 erforderlich. ◁

5.5.2 Systematische Entwicklung von Verfahren höherer Ordnung

Bislang sind wir eher zufällig auf Runge–Kutta–Verfahren gestoßen oder haben bereits bekannte Verfahren als solche identifiziert. In diesem Abschnitt wolle wir folgende Strategie zur systematischen Entwicklung von Runge–Kutta–Verfahren verfolgen:

- Taylor-Entwicklung des Konsistenzfehlers $\varepsilon(x, \tau) = \phi^\tau x - \psi^\tau x$ um $\tau = 0$.
- Elimination der führenden Terme durch entsprechende Wahl der Koeffizienten b und $\mathcal{A}$.

Wir illustrieren dieses Vorgehen am Beispiel der Runge–Kutta-Verfahren 1. Stufe, also

$$\psi^\tau x = x + \tau b_1 k_1, \quad k_1 = f(x + \tau a_{11} k_1),$$

mit zu bestimmenden Koeffizienten b_1 und a_{11}. Die Taylor-Entwicklung

$$\phi^\tau x = x + \tau f(x) + \frac{\tau^2}{2} f'(x) f(x) + \mathcal{O}(\tau^3)$$

kennen wir schon aus dem Beweis von Satz 5.11. Die Taylor-Entwicklung

$$\psi^\tau x = x + \tau \frac{d}{d\tau} \psi^\tau x|_{\tau=0} + \frac{\tau^2}{2} \frac{d^2}{d\tau^2} \psi^\tau x|_{\tau=0} + \mathcal{O}(\tau^3)$$

müssen wir noch berechnen. Es gilt

$$\frac{d}{d\tau} \psi^\tau x = b_1 k_1 + \tau b_1 \frac{d}{d\tau} k_1$$

und daher

$$\frac{d}{d\tau} \psi^\tau x|_{\tau=0} = b_1 f(x).$$

Weiter erhalten wir

$$\frac{d^2}{d\tau^2} \psi^\tau x = 2 b_1 \frac{d}{d\tau} k_1 + \tau b_1 \frac{d^2}{d\tau^2} k_1$$

und

$$\frac{d}{d\tau} k_1 = \frac{d}{d\tau} f(x + \tau a_{11} k_1) = f'\big(x + \tau a_{11} k_1\big)\big(a_{11} k_1 + \tau \frac{d}{d\tau} k_1\big).$$

Einsetzen von $\tau = 0$ liefert

$$\frac{d^2}{d\tau^2} \psi^\tau x|_{\tau=0} = 2 b_1 a_{11} f'(x) f(x).$$

Insgesamt ergibt sich die Taylor–Entwicklung

$$\phi^\tau x - \psi^\tau x = (1 - b_1)\tau f(x) + (1 - 2 b_1 a_{11}) \frac{\tau^2}{2} f'(x) f(x) + \mathcal{O}(\tau^3).$$

Unser Verfahren ist damit von 1. Ordnung, falls

$$b_1 = 1, \quad a_{11} \in \mathbb{R} \text{ beliebig.}$$

Zwei Vertreter dieser Klasse, nämlich das explizite und das implizite Euler–Verfahren, kennen wir schon. Durch Lösen des nichtlinearen Gleichungssystems

$$1 - b_1 = 0, \quad 1 - 2 b_1 a_{11} = 0$$

erhalten wir $b_1 = 1$ und $a_{11} = 1/2$. Die resultierende *implizite Mittelpunktsregel*

$$\psi^\tau x = x + \tau f\big(\tfrac{1}{2}(x + \psi^\tau x)\big) \tag{5.49}$$

ist somit ein implizites Runge–Kutta–Verfahren 1. Stufe mit Konsistenzordnung $p = 2$. Durch Ausrechnen der Terme dritter Ordnung der Taylorentwicklung lässt sich nachprüfen, daß diese nicht zusätzlich eliminiert werden können. Die maximale Konsistenzordnung expliziter Runge–Kutta–Verfahren der Stufe $s = 1$ ist also $p = 1$, während bei impliziten Runge–Kutta–Verfahren des Stufe $s = 1$ die Konsistenzordnung $p = 2$ erreicht werden kann.
Zur Herleitung von Runge–Kutta–Verfahren höherer Ordnung benötigt man höhere Ableitungen des Konsistenzfehlers $\varepsilon(x, \tau) = \phi^\tau x - \psi^\tau x$. Diese lassen sich systematisch mit Hilfe des sogenannten *Wurzelbaumverfahrens* (Butcher 1964) ermitteln (vgl. Deuflhard und Bornemann [1, Abschnitt 4.2.3]). Auf diese Weise erhält man folgendes Resultat.

Satz 5.13 *Ein Runge–Kutta–Verfahren s–ter Stufe hat genau dann für alle rechten Seiten $f \in C^p(\mathbb{R}^d)$ die Konsistenzordnung $p = 1$, wenn die Koeffizienten b, $\mathcal{A}$ der Bedingung*

$$\sum_{i=1}^{s} b_i = 1$$

genügen.
Das Verfahren hat genau dann die Konsistenzordnung $p = 2$, wenn zusätzlich die Bedingung

$$\sum_{i=1}^{s} b_i c_i = 1/2$$

mit

$$c_i = \sum_{j=1}^{s} a_{ij} \tag{5.50}$$

erfüllt ist.
Das Verfahren hat genau dann die Konsistenzordnung $p = 3$, wenn zusätzlich die Bedingungen

$$\begin{aligned} \sum_{i=1}^{s} b_i c_i^2 &= 1/3 \\ \sum_{i,j=1}^{s} b_i a_{ij} c_j &= 1/6 \end{aligned}$$

erfüllt sind.
Das Verfahren hat genau dann die Konsistenzordnung $p = 4$, wenn zusätzlich die Bedingungen

$$\begin{aligned} \sum_{i=1}^{s} b_i c_i^3 &= 1/4 \\ \sum_{i,j=1}^{s} b_i c_i a_{ij} c_j &= 1/8 \\ \sum_{i,j=1}^{s} b_i a_{ij} c_j^2 &= 1/12 \\ \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k &= 1/24 \end{aligned}$$

erfüllt sind.

Beweis:
Wir verweisen auf Deuflhard und Bornemann [1, Satz 4.17] □

Beispiel:
Beim Verfahren von Runge hat man

$$\sum_{i=1}^{2} b_i = 0 + 1 = 1$$

und wegen

$$c_1 = 0, \quad c_2 = 1/2$$

gilt auch

$$\sum_{i=1}^{2} b_i c_i = 0 \cdot 0 + 1 \cdot 1/2 = 1/2.$$

Das Verfahren von Runge hat nach Satz 5.13 also Konsistenzordnung $p = 2$. ◁

Mit wachsendem p wird auch diese Strategie etwas unbequem, denn die Anzahl N_p der Bedingungsgleichungen wächst exponentiell, wie folgende Tabelle zeigt:

p	1	2	3	4	5	6	7	8	9	10	20
N_p	1	2	4	8	17	37	85	200	486	1205	20247374

Für die *maximale* Konsistenzordnung von expliziten Runge–Kutta–Verfahren s–ter Stufe erhält man dieselben Schranken wie für $s = 1$.

Satz 5.14 *Es sei $f \in C^s(\mathbb{R}^d)$. Alle expliziten Runge–Kutta–Verfahren s–ter Stufe haben eine Konsistenzordnung $p \leq s$.*

Beweis:
Wir verweisen auf Deuflhard und Bornemann [1, Lemma 4.15]. □

Beachte, daß in Satz 5.14 *nicht* gesagt wird, ob es Verfahren mit größtmöglicher Konsistenzordnung $p = s$ gibt. Tatsächlich ist diese Frage noch offen. Die sogenannten *Butcher–Schranken* $s = s(p)$ bezeichnen zu jedem p die kleinste Stufe aller derzeit bekannten, expliziten Runge–Kutta–Verfahren mit Konsistenzordnung p. Die folgende Tabelle enthält einige Werte von $s(p)$.

p	1	2	3	4	5	6	7	8	≥ 9
$s(p)$	1	2	3	4	6	7	9	11	$\geq p + 3$

Für $p = 10$ wird der derzeitige Rekord von $s(p) = 17$ von Hairer (1978) gehalten. Die meisten dieser minimalen Konstruktionen haben allerdings nur theoretisches Interesse, weil praktisch brauchbare Verfahren noch anderen Kriterien genügen müssen. Beispiele für solche Kriterien werden wir später kennenlernen.

Bemerkung:
Alle impliziten Runge–Kutta–Verfahren s–ter Stufe haben eine Konsistenzordnung $p \leq 2s$ (vgl. Deuflhard und Bornemann [1, Lemma 6.34]). Darüberhinaus hat Butcher (1964) für jedes $s \in \mathbb{N}$ ein implizites Runge–Kutta–Verfahren der Ordnung $p = 2s$ konstruiert. Für implizite Runge–Kutta–Verfahren ist damit die Frage nach optimalen Butcher–Schranken erledigt. ◁

5.5.3 Diskrete Kondition

Bei der Abschätzung der Kondition eines AWPs (5.20) haben wir gesehen, daß die Lipschitz–Stetigkeit der rechten Seite f die Lipschitz–Stetigkeit des zugehörigen Flußoperators impliziert. Aus

$$\| f(x) - f(y) \| \leq L \| x - y \| \quad \forall x, y \in \mathbb{R}^d$$

folgt mit (5.27) und dem Gronwall–Lemma 5.8 nämlich

$$\| \phi^t x - \phi^t y \| \leq e^{tL} \| x - y \| \quad \forall x, y \in \mathbb{R}^d \; \forall t \geq 0. \tag{5.51}$$

und damit $\kappa(t) \leq e^{tL}$.
Wir zeigen nun ein entsprechendes Resultat für den diskreten Flußoperator ψ^τ.

Lemma 5.15 *Sei $f : \mathbb{R}^d \to \mathbb{R}^d$ Lipschitz–stetig mit Lipschitz–Konstante L. Dann existiert für jedes explizite Runge-Kutta-Verfahren ψ^τ eine Konstante γ, die nur von den Koeffizienten b, $\mathcal{A}$ des Verfahrens abhängt, so daß gilt*

$$\| \psi^\tau x - \psi^\tau y \| \leq e^{\tau\gamma L} \| x - y \| \quad \forall x, y \in \mathbb{R}^d \quad \forall \tau \geq 0. \tag{5.52}$$

Beweis:
Für den vollständigen Beweis verweisen wir auf Deuflhard und Bornemann [1, Lemma 4.22]. Wir betrachten hier nur den Fall $s = 2$, also

$$\psi^\tau x = x + \tau\big(b_1 k_1(x) + b_2 k_2(x)\big),$$

$$k_2(x) = f(x + \tau a_{21} k_1(x)), \quad k_1(x) = f(x).$$

Offenbar gilt

$$\psi^\tau x - \psi^\tau y = x - y + \tau b_1\big(k_1(x) - k_1(y)\big) + \tau b_2\big(k_2(x) - k_2(y)\big) \quad \forall x, y \in \mathbb{R}^d.$$

Aus der Lipschitz–Stetigkeit von f folgt

$$\|k_1(x) - k_1(y)\| = \|f(x) - f(y)\| \leq L\|x - y\|$$

und weiter

$$\begin{aligned}
\|k_2(x) - k_2(y)\| &= \|f\big(x + \tau a_{21} k_1(x)\big) - f\big(y + \tau a_{21} k_1(y)\big)\| \\
&\leq L\big(\|x - y\| + \tau|a_{21}|\|k_1(x) - k_1(y)\|\big) \\
&\leq L\big(\|x - y\| + \tau|a_{21}|L\|x - y\|\big) \\
&\leq L(1 + \tau L|a_{21}|)\|x - y\|.
\end{aligned}$$

Insgesamt ergibt sich mit

$$\gamma = \max\{|b_1| + |b_2|, \sqrt{2|b_2 a_{21}|}\} \tag{5.53}$$

die gewünschte Abschätzung

$$\begin{aligned}\|\psi^\tau x - \psi^\tau y\| &\leq \|x-y\| + \tau|b_1|\|k_1(x)-k_1(y)\| + \tau|b_2|\|k_2(x)-k_2(y)\| \\ &\leq \big(1+\tau(|b_1|+|b_2|)L + \tau^2|b_2a_{21}|L^2\big)\|x-y\| \\ &\leq \big(1+\tau\gamma L + \tfrac{1}{2}(\tau\gamma L)^2\big)\|x-y\| \\ &\leq e^{\tau\gamma L}.\end{aligned}$$

□

Man ist natürlich daran interessiert, die Störungsempfindlichkeit des kontinuierlichen Problems durch Diskretisierung möglichst wenig zu verschlechtern. Damit sind Verfahren mit der Eigenschaft $\gamma = 1$ besonders interessant (vgl. Satz 4.2, Kapitel 4).

Lemma 5.16 *Für jedes konsistente explizite, s-stufige Runge-Kutta-Verfahren gilt*

$$\gamma = \gamma(b, \mathcal{A}) \geq 1.$$

Hat das Verfahren die Konsistenzordnung $p = s$ und sind alle Koeffizienten b, $\mathcal{A}$ nicht–negativ, so gilt sogar

$$\gamma = \gamma(b, \mathcal{A}) = 1.$$

Beweis:
Im Falle $s = 2$ folgt aus der Konsistenzbedingung in Satz 5.13 und (5.53) sofort

$$\gamma \geq |b_1| + |b_2| \geq |b_1 + b_2| = 1.$$

Ist das Verfahren konsistent mit Ordnung $p = s = 2$, so folgt wieder aus Satz 5.13, daß

$$b_1 + b_2 = 1, \quad b_2 a_{21} = 1/2$$

gelten muß. Sind alle Koeffizienten b_1, b_2, a_{21} nicht–negativ, so liefert (5.53) sofort $\gamma = 1$. Für den allgemeinen Fall verweisen wir auf Deuflhard und Bornemann [1, Lemma 4.26]. □

5.5.4 Konvergenz expliziter Runge–Kutta–Verfahren

Vorgelegt sei ein Gitter

$$\Delta = \{0 = t_0 < t_1 \cdots < t_{n-1} < t_n = T\}$$

mit maximaler Schrittweite

$$\tau_\Delta = \max_{k=0,\ldots,n-1} \tau_k, \quad \tau_k = t_{k+1} - t_k,$$

und ein explizites Einschrittverfahren ψ^τ. Zu jedem Startwert $x_0 = x(0)$ erhalten wir aus

$$x_{k+1} = \psi^{\tau_k} x_k, \quad k = 0, \ldots, n-1, \tag{5.54}$$

die Gitterfunktion

$$x_\Delta(t_k) = x_k, \quad k = 0, \ldots, n,$$

von der wir hoffen, daß sie für $\tau_\Delta \to 0$ gegen die Lösung $x(t)$ von (5.20) konvergiert.

Definition 5.17 *Der Diskretisierungsfehler $\|x - x_\Delta\|_\infty$ ist definiert durch*

$$\|x - x_\Delta\|_\infty = \max_{k=0,\ldots,n} \|x(t_k) - x_k\|.$$

Das Verfahren ψ^τ heißt konvergent, wenn

$$\lim_{\tau_\Delta \to 0} \|x - x_\Delta\|_\infty = 0.$$

Das Verfahren ψ^τ heißt konvergent mit Konvergenzordnung p, wenn $C \geq 0$ und $\tau^* > 0$ existieren, so daß

$$\|x - x_\Delta\|_\infty \leq C\tau_\Delta^p \quad \forall \tau_\Delta \leq \tau^*.$$

Mit Blick auf (5.33) erscheint die Konsistenz von ψ^τ mit ϕ^τ notwendig für die Konvergenz des Verfahrens. *Zusätzlich dürfen sich Folgefehler nicht aufschaukeln!*

Satz 5.18 (Stabilität) *Das Einschrittverfahren genüge der Stabilitätsbedingung*

$$\| \psi^\tau x - \psi^\tau y \| \leq e^{\tau\gamma L} \| x - y \| \quad \forall x, y \in \mathbb{R}^d \quad \forall \tau \geq 0, \tag{5.55}$$

mit gewissen Konstanten γ, $L \geq 1$. Weiter seien $\varepsilon_k \in \mathbb{R}^d$, $k = 0, \ldots, n-1$, gegeben und die gestörte Gitterfunktion x_Δ^* definiert durch

$$x_{k+1}^* = \psi^{\tau_k} x_k^* + \varepsilon_k, \quad k = 0, \ldots, n-1, \quad x_0^* = x_0.$$

Dann gilt die Abschätzung

$$\|x_k^* - x_k\| \leq e^{t_k \gamma L} \sum_{i=0}^{k-1} \|\varepsilon_i\| \quad \forall k = 0, \ldots, n. \tag{5.56}$$

Beweis:
Der Beweis erfolgt mit vollständiger Induktion nach k.
Induktionsanfang $k = 0$. Es gilt

$$\|x_1^* - x_1\| = \|\psi^{\tau_0} x_0^* + \varepsilon_0 - \psi^{\tau_0} x_0\| = \|\varepsilon_0\| \leq e^{t_1 \gamma L} \|\varepsilon_0\|.$$

Induktionsannahme. Es gelte (5.56) für ein $k \geq 0$.

Induktionsschlußvon k auf $k+1$. Anwendung von (5.55) liefert

$$\begin{aligned}\|x_{k+1}^* - x_{k+1}\| &= \|\psi^{\tau_k} x_k^* + \varepsilon_k - \psi^{\tau_k} x_k\| \\ &\leq e^{\tau_k \gamma L}\|x_k^* - x_k\| + \|\varepsilon_k\| \\ &\leq e^{\tau_k \gamma L} e^{t_k \gamma L} \sum_{i=0}^{k-1} \|\varepsilon_i\| + e^{t_{k+1}\gamma L}\|\varepsilon_k\| \\ &= e^{t_{k+1}\gamma L} \sum_{i=0}^{k} \|\varepsilon_i\|\end{aligned}$$

und damit (5.56). □

Beachte, daß explizite Runge–Kutta–Verfahren nach Lemma 5.15 die Stabilitätsbedingung (5.55) erfüllen. Damit können wir nun den angestrebten Konvergenzsatz formulieren.

Satz 5.19 *Ein explizites Runge–Kutta–Verfahren ψ^τ mit Konsistenzordnung p ist konvergent mit der Konvergenzordnung p.*

Beweis:
Für alle $k = 0, \ldots, n$ gilt offenbar $x(t_k) \in K = \{x(t) \mid 0 \leq t \leq T\} \subset \mathbb{R}^d$ und K ist kompakt. Nach Definition 5.10 existieren daher eine Konstante $C_K(f)$ und ein $\tau^* > 0$, so daß für alle $\tau_\Delta \leq \tau^*$ der Konsistenzfehler

$$\varepsilon(x(t_k), \tau_k) = \phi^{\tau_k} x(t_k) - \psi^{\tau_k} x(t_k)$$

die Eigenschaft

$$\|\varepsilon(x(t_k), \tau_k)\| \leq C_K(f)\tau_k^{p+1} \quad \forall k = 0, \ldots, n-1$$

besitzt. Weiter gilt

$$x(t_{k+1}) = \phi^{\tau_k} x(t_k) = \psi^{\tau_k} x(t_k) + \varepsilon(x(t_k), \tau_k), \quad k = 0, \ldots, n-1.$$

Aus Satz 5.18 folgt schließlich mit $x_k^* = x(t_k)$ die Abschätzung

$$\|x - x_\Delta\|_\infty = \max_{k=0,\ldots,n} \|x(t_k) - x_k\| \leq e^{T\gamma L} \sum_{i=0}^{n-1} \|\varepsilon(x(t_i), \tau_i)\| \leq e^{T\gamma L} C_K(f) T \tau_\Delta^p$$

und damit die Behauptung. □

Grob gesprochen haben wir gezeigt

$$\textit{Konsistenz + Stabilität = Konvergenz.}$$

Diesen Zusammenhang findet man bei der Analyse von Diskretisierungen gewöhnlicher oder partieller Differentialgleichungen immer wieder. Allerdings hat man jeweils zu klären, was genau mit Konsistenz und Stabilität gemeint ist. Für explizite Einschrittverfahren haben wir das mit Definition 5.10 und der Stabilitätsbedingung (5.55) getan.

Bemerkung:
Wir haben mit Satz 5.19 bewiesen, daß wir zu jeder Toleranz $TOL > 0$ ein Gitter Δ mit genügend kleiner Schrittweite τ_Δ finden können, so daß

$$\|x - x_\Delta\|_\infty \leq TOL$$

ausfällt.
Achtung: Ist $L \gg 1$ oder/und $TOL \ll 1$ so kann eine Schrittweite τ_Δ erforderlich sein, die kleiner als die kleinste im Rechner darstellbare Zahl ist. Derart schlecht konditionierte Probleme lassen sich (mit einem Einschrittverfahren auf dem entsprechenden Rechner) nicht mit der gewünschten Genauigkeit lösen! ◁

5.6 Schrittweitensteuerung und eingebettete Runge-Kutta-Verfahren

Bisher haben wir das zugrundeliegende Gitter

$$\Delta = \{0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = T\}$$

einfach als gegeben hingenommen. Wie bei der Quadratur ist bei glatten Lösungen $x(t)$ die Wahl eines äquidistanten Gitters durchaus sinnvoll. Wie bei der Quadratur ändert sich die Lage, wenn große Gradienten auftreten. Das zeigt folgendes Beispiel.

Beispiel:
Wir betrachten das Anfangswertproblem

$$x' = \frac{1}{x},\ t \in (0,1], \quad x(0) = \varepsilon > 0 \tag{5.57}$$

mit der exakten Lösung $x(t) = \sqrt{2t + \varepsilon^2}$. Zur approximativen Lösung wollen wir wie auf Seite 111 das explizite Euler–Verfahren, das Verfahren von Runge und Runge–Kutta–4 auf äquidistanten Gittern verwenden. Indem wir das Intervall $[0, 1]$ so lange wie nötig halbieren, bestimmen wir für jedes Verfahren eine zugehörige Schrittweite so, daß der entsprechende Diskretisierungsfehler kleiner als eine vorgegebenen Schranke TOL ist.
Wir wählen den Parameter $\varepsilon = 10^{-3}$ und die Toleranz $TOL = 10^{-1}$. Abbildung 5.8 zeigt links die exakte Lösung x im Vergleich mit den Näherungslösungen x_Δ^{Euler}, x_Δ^{Runge} und x_Δ^{RK4}.

Auf der rechten Seite sieht man das Verhalten der Diskretisierungsfehler in Abhängigkeit von den benötigten f–Auswertungen. Anders als beim Beispiel mit glatter Lösung auf Seite 111 zahlt sich die höhere Konsistenzordnung diesmal nicht aus. Runge–Kutta–4 benötigt 16384 (!) f–Auswertungen, genausoviele wie das Euler–Verfahren. Sieger ist diesmal das Verfahren von Runge mit nur 1024 f–Auswertungen. ◁

Die Glattheit der Lösung $x(t)$, und damit auch die Glattheit des Integranden $f(x(\cdot))$ in

$$x(t) = x_0 + \int_0^t f(x(s))\, ds$$

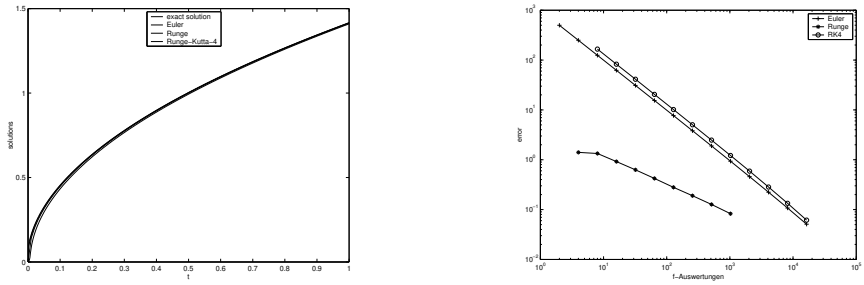


Abbildung 5.8: Euler, Runge und Runge–Kutta–4 für fast–singuläre Lösung

hat offenbar großen Einfluß auf die Wahl der Diskretisierung. Im Unterschied zur Quadratur ist der Integrand $f(x(\cdot))$ aber nicht bekannt. Außerdem kann man *nicht* von glatter rechter Seite f auf glatte Lösungen $x(t)$ schließen!

Beispiel:
Die Lösung des folgenden AWPs für die van der Pol'sche Differentialgleichung

$$x''(t) = 10(1 - x(t)^2)x'(t) - x(t) = 0,\ t \in (0, 20], \quad x(0) = 0,\ x'(0) = 1, \tag{5.58}$$

ist in Abbildung 5.9 dargestellt. Obwohl $f(x, x') = 20(1 - x^2)x' - x = 0$ recht harmlos

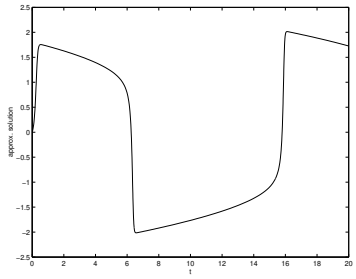


Abbildung 5.9: Lösung der van der Pol'schen Differentialgleichung

aussieht, weist die Lösung stark lokal variierendes Verhalten auf. ◁

Gerade im Bereich der chemischen Reaktionskinetik treten noch drastischere Fälle auf. Ein Beispiel ist das sogenannte *Oregonator–Modell* (vgl. [1, Abschnitt 1.2]). Damit wird adaptive Schrittweitenkontrolle unerlässlich.

Adaptive Kontrolle des Diskretisierungsfehlers. Ein naheliegendes Ziel besteht darin, zu einer gegebenen Toleranz TOL ein Gitter Δ_{TOL} mit möglichst kleiner Anzahl n_{TOL} von Gitterpunkten zu finden, so daß der Diskretisierungsfehler die Genauigkeitsbedingung

$$\|x - x_{\Delta_{TOL}}\|_\infty \leq TOL$$

erfüllt. Dabei gibt es folgende Schwierigkeit: Im Gegensatz zur Quadratur brauchen lokale Diskretisierungsfehler $\|x(t_k) - x_k\|$ in t_k *keine lokalen Ursachen* zu haben. Warum das so ist, wollen wir kurz erläutern.

Der lokale Diskretisierungsfehler in t_{k+1} lässt sich gemäß

$$\|x(t_{k+1}) - x_{k+1}\| \leq \|\phi^{\tau_k} x(t_k) - \phi^{\tau_k} x_k\| + \|\phi^{\tau_k} x_k - \psi^{\tau_k} x_k\|$$

abschätzen. Den Konsistenzfehler $\|\phi^{\tau_k} x_k - \psi^{\tau_k} x_k\|$ können wir durch Reduktion der aktuellen Schrittweite τ_k reduzieren. Bei dem Folgefehler $\|\phi^{\tau_k} x(t_k) - \phi^{\tau_k} x_k\|$ ist das *nicht* der Fall. Wir müssen eine Reihe von Zeitschritten verwerfen (vielleicht alle) und es mit kleineren Schrittweiten versuchen. Diese kann sich *erst später* vielleicht wieder als zu groß, vielleicht auch als zu klein herausstellen. Das macht die Kontrolle des Diskretisierungsfehlers im allgemeinen zu aufwendig.

Adaptive Kontrolle des Konsistenzfehlers. Für jedes t_k soll zu einer vorgegebenen Schranke TOL eine neue Schrittweite τ_k so bestimmt werden, daß

$$\|\varepsilon(x_k, \tau_k)\| = \|\phi^{\tau_k} x_k - \psi^{\tau_k} x_k\| \leq TOL \tag{5.59}$$

ausfällt. Dazu setzen wir von nun an voraus, daß für genügend kleine $\tau \geq 0$ die Entwicklung

$$\varepsilon(x_k, \tau) = \phi^{\tau} x_k - \psi^{\tau} x_k = c(x_k)\tau^{p+1} + \mathcal{O}(\tau^{p+2}), \quad c(x_k) \neq 0, \tag{5.60}$$

des Konsistenzfehlers $\varepsilon(x_k, \tau)$ vorliegt. Offenbar ist (5.60) erfüllt, wenn ψ^{τ} die Konsistenzordnung p hat und $\psi^{\tau} x_k$ nicht zufällig genauer ist als erwartet.

Beispiel:
Im Falle des expliziten Euler–Verfahrens ist bekanntlich (vgl. (5.38)).

$$\phi^{\tau} x_k - \psi^{\tau} x_k = \frac{1}{2} f'(x_k) f(x_k) \tau^2 + \mathcal{O}(\tau^3)$$

Bedingung (5.60) ist also erfüllt, wenn

$$c(x_k) = \frac{1}{2} f'(x_k) f(x_k) \neq 0$$

vorliegt. ◁

Wie schon bei der adaptiven Quadratur werden wir die Genauigkeitsanforderung (5.59) nur *asymptotisch exakt*, d.h. exakt für $\tau \to 0$ erfüllen. Der erste Schritt dazu ist folgender Satz.

Satz 5.20 *Es gelte*

$$\tau_k^{neu} = \tau_k^{alt} \sqrt[p+1]{\frac{\rho\, TOL}{\|\varepsilon(x_k, \tau_k^{alt})\|}}. \tag{5.61}$$

Dann folgt mit $\tau_k = \tau_k^{neu}$ die Abschätzung

$$\|\varepsilon(x_k, \tau_k)\| = \rho\big(1 + \mathcal{O}(\tau_k^{alt})\big) TOL + \mathcal{O}(\tau_k^{p+2}). \tag{5.62}$$

Ist der Sicherheitsfaktor ρ so gewählt, daß

$$\rho\big(1 + \mathcal{O}(\tau_k^{alt})\big) \leq 1,$$

so ist also die gewünschte Abschätzung (5.59) *bis auf Terme $(p+2)$–ter Ordnung erfüllt.*

Beweis:
Durch elementare Umformungen erhält man aus (5.61)

$$\frac{\rho\, TOL}{\|\varepsilon(x_k,\tau_k^{alt})\|} = \frac{(\tau_k^{neu})^{p+1}}{(\tau_k^{alt})^{p+1}} = \frac{\|\varepsilon(x_k,\tau_k^{neu}) + \mathcal{O}\big((\tau_k^{neu})^{p+2}\big)\|}{\|\varepsilon(x_k,\tau_k^{alt}) + \mathcal{O}\big((\tau_k^{alt})^{p+2}\big)\|}$$

und daraus

$$\rho\big(1+\mathcal{O}(\tau_k^{alt})\big)TOL \geq \rho\frac{\|\varepsilon(x_k,\tau_k^{alt}) + \mathcal{O}\big((\tau_k^{alt})^{p+2}\big)\|}{\|\varepsilon(x_k,\tau_k^{alt})\|}TOL \geq \|\varepsilon(x_k,\tau_k^{neu})\| - \mathcal{O}\big((\tau_k^{neu})^{p+2}\big).\ \square$$

Berechnet man die neue Schrittweite aus (5.61), so ist nach Satz 5.20 die gewünschte Fehlerabschätzung (5.59) bis auf Terme höherer Ordnung erfüllt, also asymptotisch exakt.
Der in (5.61) auftretende Konsistenzfehler $\varepsilon(x_k,\tau_k^{alt})$ ist unbekannt und muß durch eine geeignete Schätzung ersetzt werden. Dabei gehen wir genauso vor wie bei der adaptiven Quadratur (vgl. Satz 4.15).

Satz 5.21 *Es sei χ^τ ein Verfahren mit der Konsistenzordnung $p+1$. Dann gibt es zu jedem $q \in (0,1)$ ein $\tau^* > 0$, so daß die Saturationsbedingung*

$$\|\phi^\tau x_k - \chi^\tau x_k\| \leq q\|\varepsilon(x_k,\tau)\| \quad \forall \tau \leq \tau^* \tag{5.63}$$

erfüllt ist.
Aus (5.63) *folgt die a posteriori Fehlerabschätzung des Konsistenzfehlers*

$$(1+q)^{-1}\|\chi^\tau x_k - \psi^\tau x_k\| \leq \|\varepsilon(x_k,\tau)\| \leq (1-q)^{-1}\|\chi^\tau x_k - \psi^\tau x_k\| \quad \forall \tau \leq \tau^*. \tag{5.64}$$

Beweis:
Es sei $q \in (0,1)$ gegeben. Da χ^τ von $(p+1)$–ter Ordnung ist, gilt

$$\|\phi^\tau x_k - \chi^\tau x_k\| \leq C(x_k)\tau^{p+2}$$

mit geeignetem $C(x_k) \geq 0$. Da ψ^τ die Bedingung (5.60) mit $c(x_k) \neq 0$ erfüllt, gibt es ein $\hat\tau > 0$, so daß

$$\frac{1}{2}|c(x_k)|\tau^{p+1} \leq \|\varepsilon(x_k,\tau)\| \quad \forall \tau \leq \hat\tau.$$

Insgesamt erhält man

$$\|\phi^\tau x_k - \chi^\tau x_k\| \leq \frac{2C(x_k)}{|c(x_k)|}\tau\|\varepsilon(x_k,\tau)\| \quad \forall \tau \leq \hat\tau$$

und daraus die Saturationsbedingung (5.63).
Unter Voraussetzung von (5.63) erhält man die a posteriori Fehlerabschätzung (5.64) mit der Dreiecksungleichung. $\square$

Berechnet man in jedem Zeitschritt aus der Schrittweite $\tau_k^{alt} = \tau_{k-1}$ die neue Schrittweite $\tau_k = \tau_k^{neu}$ nach der Formel

$$\tau_k^{neu} = \tau_k^{alt}\sqrt[p+1]{\frac{\rho\, TOL}{\varepsilon_k}}, \quad \varepsilon_k = \|\chi^{\tau_k^{alt}}x_k - \psi^{\tau_k^{alt}}x_k\|, \tag{5.65}$$

so ist die Genauigkeitsbedingung (5.59) nach Satz 5.20 und Satz 5.21 bis auf Terme höherer Ordnung in τ_k^{alt} und τ_k^{neu} erfüllt.
Im ersten Zeitschritt muß man eine Schätzung $\tilde{\tau}_0$ für τ_0 vorgeben. Die Schätzung $\tilde{\tau}_0$ sollte klein genug sein, damit man von Beginn an in der Asymptotik (5.63) ist. Der Sicherheitsfaktor $\rho < 1$ in (5.65) sollte so gewählt sein, daß die vernachlässigten Terme höherer Ordnung kompensiert werden und man in der Asymptotik bleibt.
Zu große Schwankungen der Schrittweiten sind mit numerischen Schwierigkeiten wie `underflow` oder `overflow` verbunden. Deshalb erlauben wir in jedem Zeitschritt nur eine Verdoppelung oder Halbierung der alten Zeitschrittweite und geben eine maximale Schrittweite $\tau_{\max}$ vor. Für theoretische Betrachtungen mit Bezügen zu PID–Reglern verweisen wir auf Deuflhard und Bornemann [1, Abschnitt 5.2.1].
Geht man davon aus, daß die Saturationsbedingung (5.63) erfüllt ist, so ist es wegen

$$\|\phi^{\tau_k} x_k - \chi^{\tau_k} x_k\| \leq q \|\phi^{\tau_k} x_k - \psi^{\tau_k} x_k\| \approx TOL$$

sinnvoll, mit der genaueren Näherung $x_{k+1} = \chi^{\tau_k} x_k$ weiterzurechnen. In Umkehrung unserer ursprünglichen Intention wird damit χ^{τ_k} zum führenden Verfahren und ψ^{τ} dient nur noch zur adaptiven Schrittweitenkontrolle.
Wir erhalten folgenden Algorithmus.

Algorithmus 5.22 (adaptive Schrittweitensteuerung)

$t_0 := 0$
$\Delta := \{t_0 = 0\}$
$x_0 := x(0)$
$\tau_0 := \tilde{\tau}_0$
$k := 0$

while $(t_k < T)$
  $x := \chi^{\tau_k} x_k$
  $\varepsilon_k = \|\chi^{\tau_k} x_k - \psi^{\tau_k} x_k\|$
  $\tau := \min\{2\tau_k, \tau_{\max}, \sqrt[p+1]{\frac{\rho\, TOL}{\varepsilon_k}}\tau_k\}$
  $\tau := \max\{\tau_k/2, \tau\}$

  if $(\varepsilon_k \leq TOL)$ ***then*** *(Schritt akzeptieren)*
    $t_{k+1} := t_k + \tau_k$
    $\Delta := \Delta \cup \{t_{k+1}\}$
    $x_{k+1} := x$
    $\tau_{k+1} := \min\{\tau, T - t_{k+1}\}$
    $k := k + 1$
  else *(Schritt verwerfen)*
    $\tau_k := \tau$
  end if

end while

Man beachte, daß die **while**–Schleife keineswegs terminieren muß. Ein guter Algorithmus *sollte* sogar immer kleiner werdende Schrittweiten produzieren und sich auf diese Weise

„festfressen“, wenn die kontinuierliche Lösung gar nicht im gesamten Intervall $t \in [0, T]$ existiert (Blow–up). Für weitere Einzelheiten verweisen wir auf Deuflhard und Bornemann [1, Kapitel 5].

Zur adaptiven Schrittweitensteuerung benötigt man also jeweils ein Paar von Verfahren χ^τ und ψ^τ der Ordnung p und $p-1$. Um solche Paare kümmern wir uns jetzt.

Beispiel:
Das explizite Euler–Verfahren (5.32) und das Verfahren von Runge (5.44)

$$\begin{aligned} \chi^\tau x &= x + \tau f(x + \tfrac{1}{2}\tau k_1), \quad k_1 = f(x), \\ \psi^\tau x &= x + \tau k_1 \end{aligned} \tag{5.66}$$

sind ein Paar der Ordnung $p = 2$ und $p - 1 = 1$. ◁

Es fällt auf, daß $k_1 = f(x)$ in beiden Verfahren verwendet werden kann. Die Schrittweitenkontrolle durch das Euler–Verfahren erfordert daher *keine zusätzlichen Funktionsauswertungen.*

Definition 5.23 *Haben zwei Runge–Kutta–Verfahren ψ^τ und χ^τ der Ordnung $p-1$ und p die gleiche Koeffizientenmatrix $\mathcal{A}$, so heißt das Paar (χ^τ, ψ^τ) eingebettetes Runge-Kutta-Verfahren Runge-Kutta-$p(p-1)$.*

Da sich eingebettete Runge–Kutta–Verfahren χ^τ, ψ^τ nur in den Koeffizientenvektoren β, b unterscheiden, kann man das Paar kompakt als Butcher–Schema

	$\mathcal{A}$
p	β^T
$p-1$	b^T

schreiben.

Beispiel:
Das Paar (5.66) ist durch das Butcher–Schema

	0	0
	1/2	0
2	0	1
1	1	0

charakterisiert. ◁

Gelingt es also, zu einem vorliegenden Verfahren χ^τ höherer Ordnung ein Verfahren ψ^τ zu finden, so daß (χ^τ, ψ^τ) ein eingebettetes Runge–Kutta–Verfahren ist, so kostet die adaptive Schrittweitenkontrolle keine zusätzlichen Funktionsauswertungen. Wir wollen diese Vorgehensweise am Beispiel des klassischen Verfahrens Runge–Kutta–4 mit dem Butcher–Schema

0				
1/2	0			
0	1/2	0		
0	0	1	0	
1/6	1/3	1/3	1/6	

vorführen. Bekanntlich ist χ^τ = Runge–Kutta–4 vierstufig und von der Ordnung $p = 4$. Wir suchen nach einem vierstufigen Verfahren ψ^τ mit Ordnung $p = 3$, so daß (χ^τ, ψ^τ) ein eingebettetes Runge–Kutta–Verfahren wird. Damit sind die Koeffizienten $\mathcal{A}$ und

$$\beta_1 = 1/6, \quad \beta_2 = 1/3, \quad \beta_3 = 1/3, \quad \beta_4 = 1/6$$

dem obigen Runge–Kutta–4–Tableau zu entnehmen. Die Koeffizienten $b = (b_1, \ldots, b_4)^T$ sind so zu berechnen, daß zusammen mit den Koeffizienten $\mathcal{A}$ ein Verfahren der Ordnung 3 entsteht. Nach Satz 5.13 haben die Koeffizienten $b_1, \ldots, b_4$ dazu die Gleichungen

$$\begin{aligned} b_1 + b_2 + b_3 + b_4 &= 1 \\ b_2/2 + b_3/2 + b_4 &= 1/2 \\ b_2/4 + b_3/4 + b_4 &= 1/3 \\ b_3/4 + b_4/2 &= 1/6 \end{aligned}$$

zu erfüllen. Leider ist

$$b_1 = 1/6, \quad b_2 = 1/3, \quad b_3 = 1/3, \quad b_4 = 1/6$$

die eindeutig bestimmte Lösung dieses Gleichungssystems und unser Versuch ist gescheitert. Einen Ausweg weist der sogenannte *Fehlberg–Trick* (1970):
Wir erlauben zwar 5 Stufen, also

	0				
	1/2	0			
	0	1/2	0		
	0	0	1	0	
	a_{51}	a_{52}	a_{53}	a_{54}	0
4	1/6	1/3	1/3	1/6	0
3	b_1	b_2	b_3	b_4	b_5

verlangen aber, daß die 5. Stufe

$$k_5 = f\left(x_k + \tau_k \sum_{j=1}^{4} a_{5j} k_j\right)$$

identisch ist mit der ersten Stufe k_1^*,

$$k_1^* = f(x_{k+1}) = f\left(x_k + \tau_k \sum_{j=1}^{5} \beta_j k_j\right),$$

des *nächsten Zeitschritts*. Wegen $\beta_5 = 0$ ist das möglich. Man erhält damit sofort die Koeffizienten

$$a_{51} = \beta_1 = 1/6, \quad a_{52} = \beta_2 = 1/3, \quad a_{53} = \beta_3 = 1/3, \quad a_{54} = \beta_4 = 1/6.$$

Nun haben wir noch 5 (statt vorhin 4) Koeffizienten $b_1, \ldots, b_5$ zur Verfügung, um ein eingebettes Runge–Kutta–Verfahren 3. Ordnung zu konstruieren. Einsetzen der erweiterten

Matrix in die Bedingungen aus Satz 5.13 liefert das System

$$\begin{aligned} b_1 + b_2 + b_3 + b_4 + b_5 &= 1 \\ b_2/2 + b_3/2 + b_4 + b_5 &= 1/2 \\ b_2/4 + b_3/4 + b_4 + b_5 &= 1/3 \\ b_3/4 + b_4/2 + b_5/2 &= 1/6 \end{aligned}$$

Da in diesem System b_4 und b_5 vertauschbar sind, ist außer der bekannten Lösung $(\beta^T, 0) = (1/6, 1/3, 1/3, 1/6, 0)$ auch

$$b^T = (1/6, 1/3, 1/3, 0, 1/6)^T$$

eine Lösung. Das resultierende Paar Runge–Kutta–4(3) mit dem Butcher–Schema

	1/2				
	0	1/2			
	0	0	1		
	1/6	1/3	1/3	1/6	
4	1/6	1/3	1/3	1/6	0
3	1/6	1/3	1/3	0	1/6

heißt auch *Runge–Kutta–Fehlberg–Verfahren*. Als Fehlerschätzung erhält man in diesem Fall

$$\varepsilon_k = \|\chi^{\tau_k} x_k - \psi^{\tau_k} x_k\| = \frac{\tau_k}{6} \|k_4 - k_5\|.$$

Bemerkung:
Man beachte, daß n Zeitschritte des 5–stufigen Verfahrens Runge–Kutta–4(3) (wegen des Fehlberg–Tricks!) anstelle von $5n$ nur $4n+1$ Funktionsauswertungen benötigen. Man spricht daher von einem *effektiv* 4–*stufigen* Verfahren. ◁

Die wohl ausgereiftesten eingebetteten Verfahren vom Typ Runge–Kutta–p(p-1) sind von Dormand und Prince (1980,1981) konstruiert worden (vgl. den adaptiven Code DOPR–853). Darunter ist ein effektiv 6–stufiges Verfahren für $p = 5$ und ein 13–stufiges Verfahren für $p = 8$ (vgl. $s(8) = 11$).
Zum Abschluß wollen wir unser adaptives Runge–Kutta–Fehlberg–Verfahren ausprobieren.

Beispiel:
Wir betrachten wieder das AWP (5.57) mit dem Parameter $\varepsilon = 10^{-3}$. Zur näherungsweisen Lösung verwenden wir das eingebetteten Verfahren Runge–Kutta–4(3) mit der auf Seite 124 beschriebenen adaptiven Schrittweitenkontrolle. Wir starten mit der maximal erlaubten Schrittweite $\tilde{\tau}_0 = \tau_{\max} = 0.1$ und wählen die Toleranz $TOL = 10^{-4}$.
Abbildung 5.10 zeigt links die Näherungslösung zusammen mit dem erzeugten Gitter und rechts die Schrittweitenverteilung. Die Startschrittweite wird nicht akzeptiert, sondern bis auf eine Größenordnung von 10^{-5} abgesenkt, um den steilen Anstieg der Lösung x_Δ aufzulösen. Anschließend wird τ_k schnell bis zur maximalen Schrittweite $\tau_{\max}$ vergrößert. Auf diese Weise wird mit 119 f–Auswertungen ein Diskretisierungsfehler $\|x - x_\Delta\|_\infty = 0.0016$ erreicht. Das ist ein dramatischer Effizienzgewinn im Vergleich zum äquidistanten Gitter. ◁

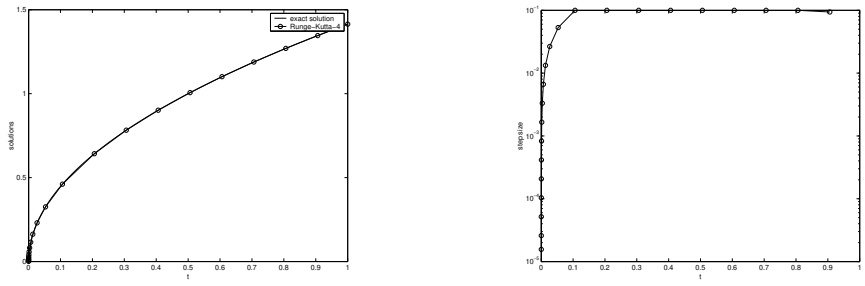


Abbildung 5.10: Adaptive Schrittweitenkontrolle: Fast–singuläre Lösung

Beispiel:
Das AWP (5.58) für die van der Pol'sche Differentialgleichung ist äquivalent zu dem System

$$\begin{aligned} x_1'(t) &= x_2(t) \\ x_2'(t) &= 10(1 - x_1(t)^2)x_2(t) - x_1(t) \end{aligned} \qquad t \in (0, 20]$$

mit den Anfangsbedingungen

$$x(0) = 0, \quad x_2(0) = 1.$$

Zur näherungsweisen Lösung verwenden wir wieder Runge–Kutta–4(3) mit adaptiver Schrittweitenkontrolle. Wir beginnen mit Startschrittweite $\tilde{\tau}_0 = \tau_{\max} = 2$ und wählen die lokale Toleranz $TOL = 10^{-4}$. Den Konsistenzfehler messen wir in der Maximumsnorm

$$\varepsilon_k = \|\chi^{\tau_k} x_k - \psi^{\tau_k} x_k\|_\infty = \frac{\tau_k}{6} \, \|k_4 - k_5\|_\infty.$$

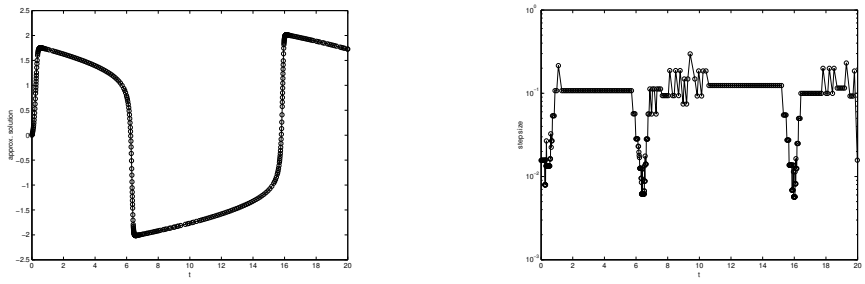


Abbildung 5.11: Adaptive Schrittweitenkontrolle: Van der Pol'sche Differentialgleichung

Abbildung 5.11 zeigt links die Näherungslösung zusammen mit dem adaptiv erzeugten Gitter Δ. Auf der rechten Seite ist die Schrittweitenverteilung zu sehen. Offenbar erkennt der Algorithmus die Bereiche starker Lösungsveränderungen und reduziert jeweils die Schrittweite. Anschließend wird die Schrittweite wieder vergrößert. Die dabei auftretenden Überschwinger sind nicht so schön. Sie können durch geschicktere Wahl der Norm in (5.65) verringert werden, aber dieses Thema würde den Rahmen einer Einführungsvorlesung sprengen. ◁

Literatur

[1] P. Deuflhard and F. Bornemann. *Numerische Mathematik II.* de Gruyter, 3. Auflage, 2008. Ein sehr empfehlenswertes Buch zur Einführung in die Numerik gewöhnlicher Differentialgleichungen. Eignet sich hervorragend zum Weiterlesen. Die verwendete Notation (erweiterter Phasenraum, Flußoperator) wirkt auf den Anfänger etwas spröde, erweist sich dem Fortgeschrittenen aber als äußerst elegant.

[2] E. Hairer, S. Nø rsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems.* Springer, 2. Auflage, 1993. Das Buch beschreibt den aktuellen Stand der Wissenschaft. Ein absolutes Muß für alle, die es genau wissen wollen.

[3] W. Walter. *Gewöhnliche Differentialgleichungen.* Springer, 7. Auflage, 2000. Ein empfehlenswertes Lehrbuch zur Einführung in die Theorie gewöhnlicher Differentialgleichungen.