Skriptum zu NUM1
WS2014/2015

Inhaltsverzeichnis

Kapitel 1

Einleitung

In dieser Vorlesung beschäftigen wir uns mit mathematischen Methoden, die numerische Probleme lösen sollen. Damit unterscheiden sich diese Methoden stark von anderen Teilbereichen der Mathematik, bei denen das symbolische Rechnen (etwa in algebraischen Strukturen) im Vordergrund steht.

Zu denen in diesem Skriptum behandelten Themen gibt es eine Vielzahl von hervorragenden Lehrbüchern und Referenztexten; hier sollen nur die Standardwerke [Press *et al.*, 2007; Kreyszig, 2011; Burden and Faires, 2004] erwähnt werden, aus denen einige Ideen und Herleitungen übernommen wurden.

Da wir somit hauptsächlich mit Zahlen operieren und nicht mehr mit unbekannten Größen, die wir mit Variablen bezeichen, müssen wir diese Zahlenwerte auch aus dem zu lösenden Problem ablesen können. Allein im Modellierungsschritt, in dem wir das reelle Problem durch eine Abstraktion ersetzen, ergeben sich folgende Fehlermöglichkeiten:

Modellierungsfehler Bei der Abbildung eines physikalischen Vorgangs in ein Modell werden oftmals vereinfachende Annahmen getroffen: So wird etwa der Luftwiderstand bei Fallgeschwindigkeitsberechnungen oftmals vernachlässigt.

Messfehler Da Messgeräte nur eine begrenzte Genauigkeit haben, sind die damit gemessenen Werte nicht exakt.

Fehlerfortpflanzung Wenn eine Berechnung auf den Ergebnissen früherer Berechnungen beruht, setzen sich damit die dort gemachten Fehler fort.

Da Fehler dieser Art schon gemacht werden, bevor die Daten in die Berechnungen eingehen, sind sie von unserer Seite nicht mehr zu vermeiden. Bei Berechnungen ergeben sich allerdings ebenfalls Fehler: Wenn stetige durch diskrete Werte ersetzt werden (etwa bei der Approximation einer Ableitung durch den Differenzenquotienten), oder prinzipiell durch das Repräsentieren unendlich vieler reeller Zahlen im endlichen Zahlenbereich eines Computers.

Beispiel 1.1 Die Formel für die Oberfläche einer Kugel ist

$$A = 4r^2\pi.$$

Um mit dieser Formel einen Wert für die Erdoberfläche zu erhalten, müssen einige vereinfachende Annahmen getroffen werden:

- Die Erde wird als Kugel betrachtet.
- Der Erdradius wird mit $r \approx 6370\text{km}$ approximiert.
- Der Wert von $\pi \approx 3.141592$ ist nur eine Annäherung.

Zusätzlich werden bei der Berechnung von A im Computer Werte möglicherweise gerundet. □

Man sieht also, dass eine Vielzahl von Faktoren das genaue Rechnen am Computer erschweren. Es geht im Rahmen dieser Vorlesung nicht darum, genauere Verfahren vorzustellen, um eine höhere Rechengenauigkeit zu erreichen, sondern um auf diese Problematik hinzuweisen. Wir können uns dazu das Folgende überlegen: Wenn wir eine Funktion $f : \mathbb{R} \to \mathbb{R}$ an einer Stelle x auswerten wollen, dann müssen wir einerseits f im Computer implementieren und andererseits x im Computer repräsentieren können. Sei dazu $\hat{f}$ die Computerimplementierung von f und $\hat{x}$ die Repräsentation von x. Der Fehler, den wir bei der Berechnung von $f(x)$ durch $\hat{f}(\hat{x})$ machen, lässt sich schreiben als

$$\hat{f}(\hat{x}) - f(x) = \underbrace{\big(\hat{f}(\hat{x}) - f(\hat{x})\big)}_{\text{Berechnungsfehler}} + \underbrace{\big(f(\hat{x}) - f(x)\big)}_{\text{Datenfehler}}.$$

Der *Berechnungsfehler* gibt an, wie weit sich eine Funktion und ihre Implementierung auf denselben Daten unterscheiden; dieser Wert ist somit von den Daten unabhängig. Der *Datenfehler* wiederum hängt nur von dem Fehler in der Repräsentation der Daten ab und ist vom der Implementierung einer Berechnungsvorschrift unabhängig.

Die Bedeutung eines Fehlers ist meist abhängig von der Größe dieses Fehlers; allerdings ist dabei zu beachten, dass die Größe relativ zu den Daten gemessen werden muss. So ist bei einer GPS-Positionierung ein Fehler von 0.01m vernachlässigbar klein; bei einem Planungssystem für Augenchirurgie ist dieser Fehler etwa so groß wie das Objekt selbst!

Wir führen daher die Begriffe *absoluter* und *relativer* Fehler ein. Sei dazu wiederum $\hat{x}$ eine Approximation an x. Dann ist der absolute Fehler gleich

$$\big|\hat{x} - x\big|$$

und der relative Fehler gleich

$$\left|\frac{\hat{x} - x}{x}\right|.$$

Nach diesen einleitenden Bemerkungen betrachten wir nun einige Details hinter der Zahlendarstellung am Computer.

1.1 Gleitkommaarithmetik

Da am Computer nur endlich viele Zahlen dargestellt werden können, werden einige Zahlen (wie etwa 2) exakt repräsentiert, andere (wie etwa $\sqrt{2}$) nur approximiert. Die Zahlenrepräsentation im Computer besteht aus einer *Mantisse* und einem *Exponent*.

Wir betrachten exemplarisch die Darstellung von Gleitkommazahlen auf einem IBM-Großrechner. Die 32 bits eines normalen *floats* werden aufgeteilt in ein bit für das Vorzeichen, 7 bits für den Exponenten (zur Basis 16) und 24 bits für die Mantisse. Da man mit 7 bits nur Exponenten im Bereich $\{0, \ldots, 127\}$ darstellen kann (und damit keine sehr kleinen Zahlen, die ja negativen Exponenten haben), zieht man von diesem Wert noch $2^6 = 64$ ab und erhält damit einen Wertebereich von $\{-64, 63\}$. Das bit der Mantisse an Stelle k wird als Koeffizient von 2^{-k} verwendet.

Beispiel 1.2 Die Maschinenzahl

$x_1 =$

0	1000010	10110011000001000000000

repräsentiert eine positive Zahl mit Exponenten $(2^6 + 2^1 - 2^6) = 2$ und Mantisse

$$2^{-1} + 2^{-3} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-14} = 0.69927978515625$$

und somit die Dezimalzahl $0.69927978515625 \times 16^2 = 179.015625$.

Die nächstkleinere Maschinenzahl ist

$x_2 =$

0	1000010	10110011000000111111111

$= 179.0156097412109375,$

und die nächstgrößere ist

$x_3 =$

0	1000010	10110011000001000000001

$= 179.0156402587890625.$

Somit repräsentiert die Maschinenzahl x_1 nicht nur einen Wert, sondern das ganze Intervall $\left[\frac{x_1+x_2}{2}, \frac{x_1+x_3}{2}\right)$. □

Maschinenzahlen werden *normalisiert* dargestellt, damit jede Zahl eine eindeutige Darstellung hat. Dies erreicht man für eine Mantisse m und eine Basis β der Exponentialdarstellung durch die Forderung

$$\beta^{-1} \leq m \leq 1.$$

Bei einer Exponentenbasis von 16 bedeutet dies, dass zumindest eines der ersten vier bits 1 sein muss, um obige Ungleichung zu erfüllen. Bei binärer Exponentenbasis ($\beta = 2$) muss somit $m \geq \frac{1}{2}$ und damit das erste bit notwendigerweise 1 sein. Damit muss diese Information nicht mehr gespeichert werden, und die Genauigkeit der Zahlendarstellung wird somit erhöht.

Bei PCs wird eine Zahlendarstellung nach einem 1985 von der IEEE verabschiedeten Standard verwendet. So besteht eine 64-bit *double*-Zahl aus einem Vorzeichenbit, einem 11 bit langen Exponenten zur Basis 2 und 52 bit langen Mantisse. Zur

Darstellung von kleinen Zahlen wird vom Exponenten noch 1023 abgezogen. Der gültige Darstellungsbereich von double-Zahlen ist zwischen etwa 10^{-308} und 10^{308}.

In der folgenden Erläuterung von Runden und Gleitkommaarithmetik nehmen wir zur Vereinfachung an, dass im Computer die Zahlen in normalisierter Dezimalform, also als

$$\pm 0.d_1 d_2 \dots d_k \times 10^n$$

mit $d_i \in \{0, \dots, 9\}$ und $d_1 \neq 0$, gespeichert werden. Wenn im Computer eine Zahl gespeichert werden soll, die sich nicht in dieser Form ausdrücken lässt, dann wird sie durch die nächste solche Maschinenzahl repräsentiert.

Satz 1.1 Die zu einer positiven Dezimalzahl

$$x = 0.x_1 x_2 x_3 \cdots \times 10^n$$

nächste Maschinenzahl

$$d = 0.d_1 d_2 \dots d_k \times 10^n$$

ergibt sich als $d_1 = x_1, \dots, d_{k-1} = x_{k-1}$ und

$$d_k = \begin{cases} x_k + 1 & \text{wenn } x_{k+1} \geq 5 \\ x_k & \text{sonst.} \end{cases}$$

Dieser Vorgang wird *Runden* genannt. Der dabei gemachte Fehler heißt *Rundungsfehler*.

Die arithmetischen Operationen im Computer müssen auf den gerundeten Maschinenzahlen operieren. Sie dazu fl die Funktion, die eine reelle Zahl x auf ihre Gleitkommadarstellung $fl(x)$ konvertiert. Die "echten" (exakten) Operationen $+, -, \times, /$ werden dann durch folgendermaßen definierte Gleitkommaoperationen $\oplus, \ominus, \otimes, \oslash$ implementiert:

$$x \oplus y = fl(fl(x) + fl(y)) \qquad x \ominus y = fl(fl(x) - fl(y))$$
$$x \otimes y = fl(fl(x) \times fl(y)) \qquad x \oslash y = fl(fl(x)/fl(y))$$

Es werden also die exakten Operationen auf den Gleitkommazahlen durchgeführt, und das Ergebnis wiederum als Gleitkommazahl dargestellt.

Wir betrachten dazu zwei Beispiele.

Beispiel 1.3 Seien $x = 1/3$ und $y = 5/7$. In auf 5 Stellen gerundeter Gleitkommadarstellung sind $fl(x) = 0.33333 \times 10^0$ und $fl(y) = 0.71429 \times 10^0$. Für die Gleitkommaoperationen erhält man folgende Ergebnisse und Fehler.

	Resultat	Echtes Resultat	Abs. Fehler	Rel. Fehler
$x \oplus y$	0.10476×10^1	$22/21$	0.190×10^{-4}	0.182×10^{-4}
$x \ominus y$	0.38096×10^0	$-8/21$	0.238×10^{-5}	0.625×10^{-5}
$x \otimes y$	0.23809×10^0	$5/21$	0.524×10^{-5}	0.220×10^{-4}
$x \oslash y$	0.21429×10^1	$7/15$	0.571×10^{-4}	0.267×10^{-4}

□

Wie man sieht, ist die Größe der Fehler für fünfstellige Dezimalzahlen ausreichend. Das dies nicht immer so sein muss, sieht man an Spezialfällen.

Beispiel 1.4 Seien zusätzlich zu $y = 5/7$ oben noch $u = 0.714251$ und $v = 0.111111 \times 10^{-4}$. Einige Operationen liefern folgende Ergebnisse und Fehler.

	Resultat	Echtes Resultat	Abs. Fehler	Rel. Fehler
$y \ominus u$	0.30×10^{-4}	0.34714×10^{-4}	0.471×10^{-5}	0.136
$(y \ominus u) \oslash v$	0.270×10^{1}	0.31243×10^{1}	0.424	0.136

Die Subtraktion zweier Werte liefert hier einen kleinen absoluten, dafür aber großen relativen Fehler. Die folgende Division durch eine kleine Zahl vergrößert den absoluten, nicht aber den relativen Fehler. □

Wir werden im Rahmen dieser Vorlesung nicht mehr weiter auf Probleme mit Gleitkommadarstellung und Gleitkommaarithmetik eingehen. Man sollte sich aber bei allen in den folgenden Kapiteln präsentierten Verfahren im Klaren sein, dass diese nur auf endlichen Zahlenrepräsentationen operieren. Weiterführende Informationen sind etwa im Paper "What every computer scientist should know about floating-point arithmetic" von David Goldberg zu finden[1].

1.2 Mathematische Grundlagen

In diesem Abschnitt behandeln wir einige elementare Definitionen und Aussagen der Analysis, die wir im Laufe des Semesters an verschiedenen Stellen benötigen werden. Die meisten der Sätze werden wir nicht beweisen, sondern anhand von Abbildungen veranschaulichen.

Definition 1.1 (Grenzwert einer Funktion)
Sei $D \subseteq \mathbb{R}$. Der *Grenzwert* $\lim_{x \to x_0} f(x)$ einer Funktion $f : D \to \mathbb{R}$ im Punkt x_0 ist definiert durch

$$\lim_{x \to x_0} f(x) = a \;\Leftrightarrow \underset{\epsilon \in \mathbb{R}}{\forall} \underset{\delta \in \mathbb{R}}{\exists} \underset{0<|x-x_0|<\delta}{\forall} |f(x) - a| < \epsilon$$

Definition 1.2 (Stetigkeit einer Funktion)
Sei $D \subseteq \mathbb{R}$ und $x_0 \in D$. Eine Funktion $f : D \to \mathbb{R}$ heißt *stetig in* x_0, wenn der Grenzwert an dieser Stelle mit dem Funktionswert übereinstimmt, wenn also gilt

$$\lim_{x \to x_0} f(x) = f(x_0).$$

Eine Funktion heißt stetig auf einer Menge M, wenn sie in allen Punkten $x_0 \in M$ stetig ist.

[1] erhältlich unter http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.22.6768

Definition 1.3 (Ableitung einer Funktion)
Sei (a, b) ein offenes Intervall, $f : (a, b) \to \mathbb{R}$ eine Funktion, und $x_0 \in (a, b)$. Dann nennt man f *differenzierbar in* x_0, wenn der Grenzwert

$$f'(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}$$

existiert; $f'(x_0)$ nennt man die *Ableitung* von f in x_0. Eine Funktion heißt differenzierbar auf einer Menge M, wenn sie in allen Punkten $x_0 \in M$ differenzierbar ist. Eine Funktion, deren Ableitung stetig ist, heißt *stetig differenzierbar*.

Da wir in den späteren Kapiteln viel mit differenzierbaren Funktionen arbeiten werden, führen wir zur Vereinfachung der Schreibweise folgende Bezeichnung ein.

Definition 1.4 (Schreibweise für Ableitungen)
Für $a, b \in \mathbb{R}$ und $n \in \mathbb{N}$ ist $C^n[a, b]$ die Menge aller n-mal stetig differenzierbaren Funktionen $f : [a, b] \to \mathbb{R}$. Die Ableitungen von f werden mit $f', f'', f^{(3)}, \ldots, f^{(n)}$ bezeichnet.

Wichtiger noch als die zwei obigen Sätze ist der Begriff der *Taylorreihenentwicklung*, die im Folgenden eingeführt wird. Mit etwas anderer Notation als in Definition 1.3 kann man die Ableitung einer Funktion schreiben als

$$f'(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}\,.$$

Für $x_0 \approx x$ fällt das Weglassen des Grenzwerts nicht sehr ins Gewicht; Auflösen nach $f(x)$ liefert

$$f(x) \approx \underbrace{f(x_0) + (x - x_0) f'(x_0)}_{=:\, t(x)}\,.$$

Für gegebenes (konstantes) x_0 ist durch die rechte Seite dieser Approximation eine lineare Funktion $t(x)$ definiert: die Tangente an f in x_0. Wie man sofort nachrechnen kann, gilt

$$t(x_0) = f(x_0) \qquad \text{und} \qquad t'(x_0) = f'(x_0),$$

die Funktionswerte und Ableitungen sind in x_0 für f und t gleich.

Man erhält bessere Approximationen $\tilde{f}$ an f, wenn man zusätzlich fordert, dass auch höhere Ableitungen von $\tilde{f}$ und f in x_0 gleich sind. Wenn wir $\tilde{f}$ auf die Klasse der Polynomfunktionen beschränken, lassen sich die zugehörigen Koeffizienten über ein lineares Gleichungssystem bestimmen. In Analogie zur linearen Situation wählen wir den Ansatz

$$\tilde{f}(x) = a(x - x_0)^2 + b(x - x_0) + c,$$

woraus sich wegen $\tilde{f}(x_0) = f(x_0)$ sofort $c = f(x_0)$ ergibt. Weiters ist

$$\tilde{f}'(x) = 2a(x - x_0) + b,$$

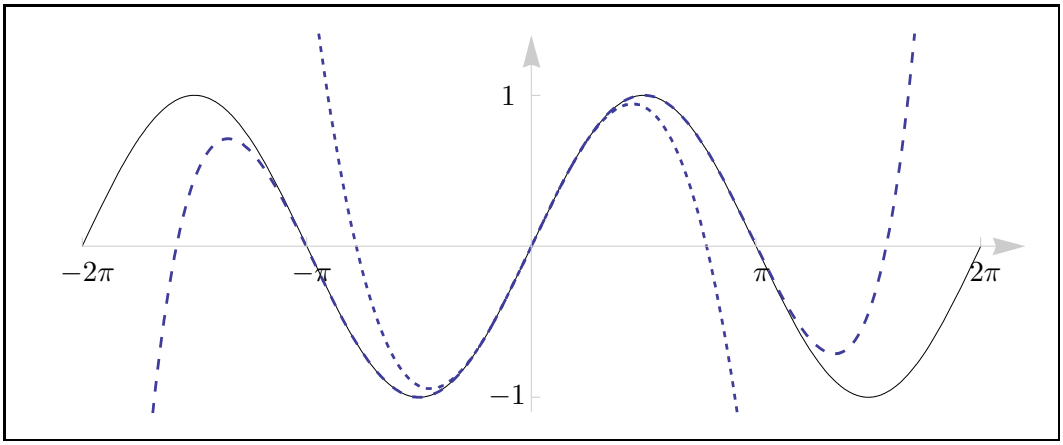

Abbildung 1.1: Die Sinusfunktion sowie die Taylorpolynome dritter (gepunktet) und neunter Ordnung (gestrichelt) mit Entwicklungspunkt $x_0 = 0$.

woraus mit der Bedingung $\tilde{f}'(x_0) = f'(x_0)$ sofort $b = f'(x_0)$ folgt. Mit $\tilde{f}''(x_0) = f''(x_0)$ folgt schließlich $a = \frac{f''(x_0)}{2}$, sodass die beste quadratische Approximation an f in x_0 gegeben ist durch

$$\tilde{f}(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)\,.$$

Die hier gezeigte Herleitung lässt sich beliebig für höhergradige Polynome fortsetzen; dies führt zum Begriff der *Taylorpolynome*.

Definition 1.5 (Taylorpolynome)
Sei $f : D \subseteq \mathbb{R} \to \mathbb{R}$ eine Funktion, die in x_0 mindestens n-mal differenzierbar ist. Dann nennt man

$$T_n(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

das *Taylorpolynom n-ter Ordnung* von f mit *Entwicklungspunkt* x_0.

Beispiel 1.5 Das Taylorpolynom n-ter Ordnung für die Sinus-Funktion mit Entwicklungspunkt $x_0 = 0$ ist

$$T_n(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots + (-1)^{(n-1)/2}\frac{x^n}{n!}\,.$$

Man beachte, dass in diesem Taylorpolynom wegen $\sin(0) = 0$ alle geraden Potenzen wegfallen.

In Abbildung 1.1 sind die Sinusfunktion sowie die Approximationen dritter und neunter Ordnung dargestellt. □

Der interessanteste Aspekt von Taylorpolynomen ist allerdings nicht, dass man damit Funktionen approximieren kann (lokal sogar beliebig genau), sondern dass

- Funktionen sich über unendliche Taylorpolynome definieren lassen, und
- der Fehler in der Approximation der Taylorpolynome abgeschätzt werden kann.

Wir werden diese beiden Punkte nun näher erläutern.

Definition 1.6 (Taylorreihen)
Sei $f : D \subseteq \mathbb{R} \to \mathbb{R}$ eine Funktion, die in x_0 beliebig oft differenzierbar ist. Dann nennt man

$$\begin{aligned} T(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 + \dots \\ &= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \end{aligned}$$

die *Taylorreihe* von f mit *Entwicklungspunkt* x_0.

Beispiel 1.6 Wegen $(e^x)' = e^x$ und $e^0 = 1$ ist die Taylorreihenentwicklung von e^x im Entwicklungspunkt $x_0 = 0$ gegeben durch

$$T(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

Wie wir bereits wissen, gilt hier $T(x) = e^x$. □

Die obige Definition und das obige Beispiel werfen die Frage auf, unter welchen Umständen $T(x) = f(x)$ gilt, die Taylorreihenentwicklung einer Funktion also mit der Funktion identisch ist. Dazu benötigen wir den Begriff des *Konvergenzradius*. Die Taylorreihe $T(x)$ mit Entwicklungspunkt x_0 hat den Konvergenzradius r, wenn gilt

$$\underset{x \in (x_0 - r, x_0 + r)}{\forall} \quad T(x) \text{ konvergiert.}$$

Zur Berechnung des Konvergenzradius verwenden wir ohne Herleitung die Formeln

$$r = \lim_{n \to \infty} (n+1) \left| \frac{f^n(x_0)}{f^{n+1}(x_0)} \right| \qquad \text{bzw.} \qquad r = \lim_{n \to \infty} \sqrt[n]{\frac{n!}{|f^n(x_0)|}}.$$

Innerhalb des Konvergenzradius sind eine Funktion und ihre Taylorreihenentwicklung identisch, wie der folgende Satz besagt.

Satz 1.2 Sei $T(x)$ die Taylorreihenentwicklung von f im Punkt x_0, und r der Konvergenzradius von $T(x)$. Dann gilt

$$\underset{x \in (x_0 - r, x_0 + r)}{\forall} \quad T(x) = f(x).$$

Beispiel 1.7 Wegen $(e^x)^{(n)} = e^x$ ist für $x_0 = 0$ der Konvergenzradius der Exponentialfunktion

$$r = \lim_{n\to\infty}(n+1)\frac{1}{1} = \infty\,.$$

Für die Taylorreihenentwicklung der Funktion $\sqrt{x}$ im Entwicklungspunkt $a > 0$ gilt

$$\begin{aligned} T(x) =& \sqrt{a} + \frac{1}{2\sqrt{a}}(x-a) - \frac{1}{8\,a^{3/2}}(x-a)^2 + \frac{1}{16a^{5/2}}(x-a)^3 + \cdots + \\ & (-1)^{(n+1)}\frac{1\cdot 3\cdots(2n-3)}{n!\,2^n\,a^{(2n-1)/2}}(x-a)^n\,, \end{aligned}$$

sodass für den Konvergenzradius gilt

$$\begin{aligned} r &= \lim_{n\to\infty}(n+1)\frac{1\cdot 3\cdots(2n-3)}{2^n\,a^{(2n-1)/2}}\Bigg/\frac{1\cdot 3\cdots(2n-1)}{2^{n+1}\,a^{(2n+1)/2}} \\ &= \lim_{n\to\infty}(n+1)\frac{2a}{2n-1} = 2a\lim_{n\to\infty}\frac{n+1}{2n-1} = a\,. \end{aligned}$$

Dieses Ergebnis ist auch anschaulich klar: Da $\sqrt{x}$ nur für $x \geq 0$ definiert ist, kann für Entwicklungspunkt a der Konvergenzradius gar nicht größer als a sein. □

Wie vorher bereits angedeutet ist ein großer Vorteil der Taylorreihenentwicklung der, dass man den Fehler in einer endlichen Approximation (über ein Taylorpolynom) abschätzen kann. Formal ist dies im folgenden Satz ausgedrückt.

Satz 1.3 Sei $f : D \subseteq \mathbb{R} \to \mathbb{R}$ eine Funktion, die in x_0 beliebig oft differenzierbar ist. Dann gilt

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \cdots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n + R_n(x),$$

wobei das *Restglied*

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

den Fehler zwischen endlicher und unendlicher Approximation angibt. Dabei ist ξ eine Zahl zwischen x_0 und x.

Aus diesem Satz kann man erkennen, dass die Differenz $T(x) - T_n(x)$ durch einen Term $R_n(x)$ angegeben werden kann, der die gleiche Struktur wie die Terme von Taylorpolynomen hat. Den *exakten* Fehler kann man nicht angeben, da man den Wert von ξ nicht kennt. Man kann den Fehler aber *beschränken*, indem man für ξ denjenigen Wert einsetzt, der das Restglied maximiert. Dann ist der echte Fehler sicher nicht größer als der so berechnete Wert. Wir illustrieren dies anhand von Beispielen.

Beispiel 1.8 In Abbildung 1.1 sind zwei Approximationen der Sinusfunktion durch Taylorpolynome zu sehen. Für die Approximation dritter Ordung gilt

$$\sin(x) = x - \frac{x^3}{3!} + R_3(x)\,,$$

für die Approximation neunter Ordnung

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} + R_9(x)\,.$$

Da n immer ungerade ist, ist die $(n+1)$-te Ableitung von $\sin(x)$ wiederum $\pm\sin(x)$, und das Restglied ist

$$R_n(x) = (-1)^{\frac{n+1}{2}} \frac{\sin(\xi)}{(n+1)!} x^{(n+1)}\,.$$

Wir verwenden Taylorpolynome unterschiedlicher Ordung, um die Sinusfunktion an der Stelle $x = 3$ zu approximieren. Das Restglied gibt an, wie groß der Fehler dieser Approximation maximal ist. Da ein Fehler immer positiv ist, sind wir nur am Absolutbetrag von $R_n(x)$ interessiert. Wegen $|\sin(\xi)| \leq 1$ gilt

$$|R_n(3)| \leq \frac{1}{(n+1)!} 3^{n+1}\,.$$

Die Abschätzungen und die tatsächlichen Fehler (als Absolutbetrag) sind für $n = 3$, 5, 7 und 9 in folgender Tabelle zusammengefasst.

	$n=3$	$n=5$	$n=7$	$n=9$
Abschätzung	3.375	1.0125	0.1627	0.01627
tatsächlicher Fehler	1.641	0.3839	0.05005	0.004192

Die Abschätzung ist in diesem Beispiel für ungerade n etwa um den Faktor 3 zu hoch; dies liegt daran, dass bei der Taylorreihenentwicklung des Sinus jeder zweite Term wegfällt. Da der Konvergenzradius des Sinus unendlich ist, wird die Approximation für höhere n immer besser. Dies gilt unabhängig von dem in diesem Beispiel gewählten Wert von $x = 3$. □

Das folgende Beispiel zeigt, welchen Einfluss der Konvergenzradius auf die Approximationen durch Taylorpolynome hat.

Beispiel 1.9 Wir gehen wie im letzten Beispiel für den natürlichen Logarithmus vor, den wir um den Punkt $x_0 = 1$ entwickeln. Allgemein gilt

$$\log^{(n)}(x) = (-1)^{n+1} \frac{(n-1)!}{x^n},$$

und somit ist die Entwicklung des Logarithmus um 1 durch die Reihe

$$\log(x) = (x-1) - \frac{1}{2}(x-1)^2 + \frac{1}{3}(x-1)^3 - \cdots = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{(x-1)^n}{n}$$

gegeben. Die Taylorpolynome fünfter und achter Ordnung sind zusammen mit dem Logarithmus in Abbildung 1.2 zu sehen. Man erkennt, dass im Gegensatz zum Sinus der Konvergenzradius endlich ist (nämlich 1), und die Approximation außerhalb des Konvergenzradius immer schlechter wird.

Numerisch kann man mit dem Restglied wiederum eine Abschätzung des Approximationsfehlers machen. Für den Logarithmus, entwickelt um $x_0 = 1$, ist das Restglied

$$R_n(x) = \frac{(-1)^n}{(n+1)\xi^{(n+1)}} (x-1)^{(n+1)}\,.$$

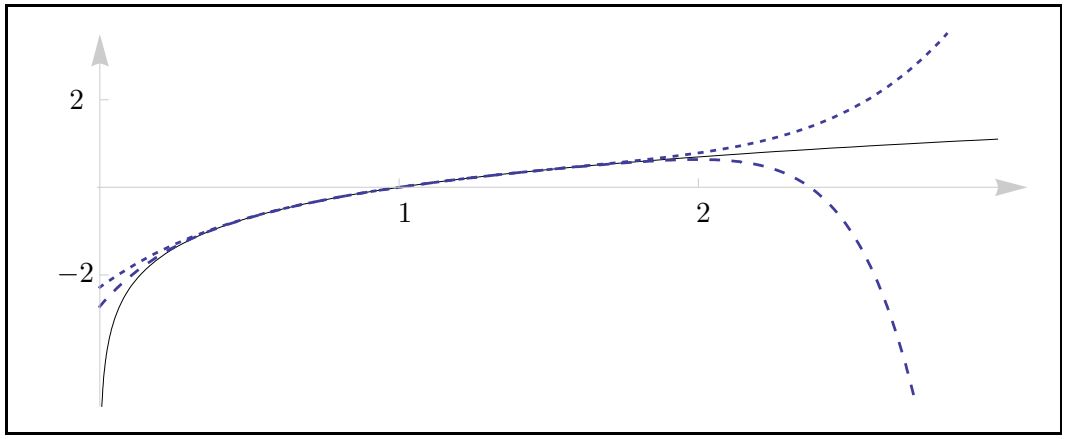

Abbildung 1.2: Der natürliche Logarithmus sowie die Taylorpolynome fünfter (gepunktet) und achter Ordnung (gestrichelt) mit Entwicklungspunkt $x_0 = 1$.

Wir evaluieren die Taylorpolynome $T_n(x)$ für $x = 1.9$, also knapp innerhalb des Konvergenzradius. Da $\xi \in [1, 1.9]$ ist und $|R_n(x)|$ immer kleiner wird, je größere Werte wir für ξ einsetzen, wird das Restglied für $\xi = 1$ maximiert. Man erhält die Abschätzung

$$|R_n(1.9)| \leq \frac{1}{(n+1)1^{(n+1)}}(1.9-1)^{(n+1)}$$

Eine Fehlertabelle für $x = 1.9$ und $n = 5, 6, 7, 8$ ist unten angeführt.

	$n=5$	$n=6$	$n=7$	$n=8$
Abschätzung	0.08857	0.06833	0.05381	0.04305
tatsächlicher Fehler	0.05022	0.03835	0.02997	0.02383

Innerhalb des Konvergenzradius wird die Approximation also mit höhergradigen Taylorpolynomen immer besser. Ausserhalb des Konvergenzradius wird die Approximation allerdings schlechter. Wir schätzen den Fehler für $x = 2.5$ ab. Dann ist $\xi \in [1, 2.5]$, und $|R_n(2.5)|$ wird wiederum durch $\xi = 1$ maximiert. Wir erhalten folgende Werte:

	$n=5$	$n=6$	$n=7$	$n=8$
Abschätzung	1.898	2.4409	3.204	4.271
tatsächlicher Fehler	0.8368	1.062	1.379	1.824

Außerhalb des Konvergenzradius wächst der Fehler somit mit dem Grad des Taylorpolynoms; die Abschätzung durch das Restglied ist aber weiterhin gültig. □

Kapitel 2

Geometrie in $\mathbb{R}^n$

In diesem Kapitel werden wir das Grundgerüst der linearen Algebra um den Begriff des *Skalarprodukts* erweitern. Wir werden sehen, wie man mit wenig mathematischem Aufwand über die Verwendung von *Projektionen* erstaunliche Ergebnisse erreichen kann. Als Einführung betrachten wir im ersten Abschnitt die *Länge* von und den *Winkel* zwischen Vektoren.

2.1 Länge und Winkel

In reellen Vektorräumen macht es oftmals Sinn, die *Länge* eines Vektors bestimmen zu können. Diese ist wie folgt definiert.

Definition 2.1 (Skalarprodukt, Länge)
Seien $v, w \in \mathbb{R}^n$ zwei reelle Vektoren. Dann nennt man die Funktion

$$\begin{aligned} \cdot : \mathbb{R}^n \times \mathbb{R}^n &\to \mathbb{R} \\ (v, w) &\mapsto \sum_{i=1}^{n} v_i w_i \end{aligned}$$

das *Skalarprodukt* von v und w. Oft lässt man das Funktionszeichen weg und schreibt nur vw und v^2 für $v \cdot w$ bzw. $v \cdot v$. Die *Länge* eines Vektors ist mit Hilfe des Skalarprodukts definiert als

$$\|v\| = \sqrt{v^2}.$$

Für das Skalarprodukt macht es keinen Unterschied, ob reelle Vektoren als Zeilen oder Spalten geschrieben werden. Bis zur Notwendigkeit einer Unterscheidung (die bei der Multiplikation mit Matrizen auftritt) werden wir daher beide Schreibweisen als äquivalent betrachten.

Beispiel 2.1 Das Skalarprodukt der beiden Vektoren $v = (1, 2, -1, 4)$ und $w =$

$(-1, 0, 2, 1)$ ist

$$v \cdot w = \begin{pmatrix} 1 \\ 2 \\ -1 \\ 4 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ 0 \\ 2 \\ 1 \end{pmatrix} = 1.$$

Die Längen der beiden Vektoren sind

$$\|v\| = \left\| \begin{pmatrix} 1 \\ 2 \\ -1 \\ 4 \end{pmatrix} \right\| = \sqrt{22} \qquad \text{und} \qquad \|w\| = \left\| \begin{pmatrix} -1 \\ 0 \\ 2 \\ 1 \end{pmatrix} \right\| = \sqrt{6}. \qquad \square$$

Mit dem Skalarprodukt kann auf den Vektoren (fast) wie mit der Multiplikation auf den reellen Zahlen gerechnet werden. So gelten etwa folgende Rechenregeln, die nicht schwer nachzuprüfen sind:

$$\begin{aligned} v \cdot w &= w \cdot v \\ v^2 &= \|v\|^2 \\ u \cdot (v + w) &= u \cdot v + u \cdot w. \end{aligned}$$

Mit der Länge eines Vektors haben wir auch die Möglichkeit, den Abstand zwischen zwei Vektoren (Punkten) zu definieren. Mathematisch fordert man, dass eine solche *Metrik* folgende Axiome erfüllt.

Definition 2.2 (Metrik)
Sei V ein Vektorraum. Dann nennt man eine Funktion $d : V \times V \to \mathbb{R}$, für die die drei Bedingungen

$$\begin{aligned} &d(v, w) \geq 0 \qquad \text{mit } d(v, w) = 0 \text{ genau dann wenn } v = w && \text{(NNG)} \\ &d(v, w) = d(w, v) && \text{(SYM)} \\ &d(v, w) \leq d(v, u) + d(u, w) && \text{(TRI)} \end{aligned}$$

für alle $u, v, w \in V$ gelten, eine *Metrik* auf V.

Geometrisch motiviert definieren wir den Abstand zweier Vektoren v und w im $\mathbb{R}$ durch

$$d(v, w) = \|v - w\|.$$

Für uns ist nun interessant, ob die damit definierte Abstandsfunktion auch eine Metrik im Sinne obiger Definition ist. Dies lässt sich nachprüfen: So gilt für die Bedingung (NNG)

$$\|v - w\| = \sqrt{(v_1 - w_1)^2 + \cdots + (v_n - w_n)^2} \geq 0.$$

Diese Länge ist genau dann null, wenn alle Summanden null sind—und dies ist nur bei $v = w$ der Fall. Die Symmetriebedingung (SYM) folgt ebenso direkt aus der Definition der Abstandsfunktion.

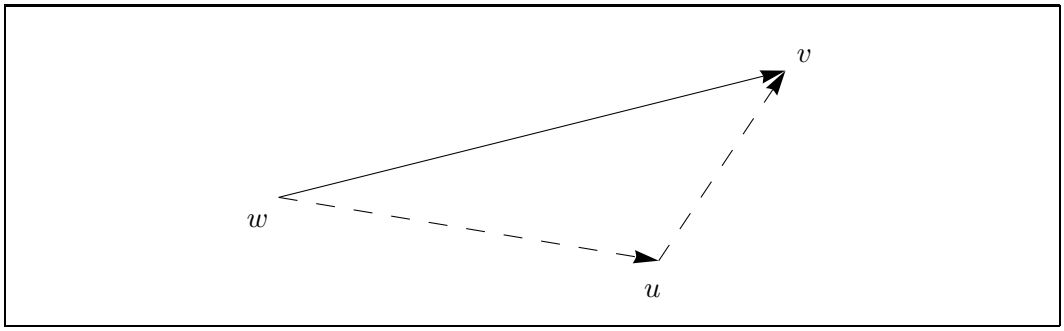

Abbildung 2.1: Illustration der Dreiecksungleichung: Der kürzeste Weg zwischen w und v ist die gerade Linie $v - w$.

Die letzte Bedingung (TRI) ist etwas schwieriger zu beweisen. Diese Bedingung, auch *Dreiecksungleichung* genannt besagt, dass der kürzeste Abstand zwischen zwei Punkten eine Gerade ist: Der Umweg über einen weiteren Punkt kann die Distanz nur länger machen (siehe auch Abbildung 2.1). Zum Beweis rechnen wir mit den Quadraten der Abstände. Dies ist korrekt, da die Ungleichung (TRI) genau dann gilt, wenn sie auch für die Quadrate der beiden Seiten gilt. Wir formen folgendermaßen um:

$$\begin{aligned}
\|v-w\|^2 &= \|v-u+u-w\|^2 \\
&= \big((v-u)+(u-w)\big)\cdot\big((v-u)+(u-w)\big) \\
&= (v-u)^2 + 2(v-u)\cdot(u-w) + (u-w)^2 \\
&\le (v-u)^2 + 2\|v-u\|\|u-w\| + (u-w)^2 \qquad \text{(CS)} \\
&\le \|v-u\|^2 + 2\|v-u\|\|u-w\| + \|u-w\|^2 \\
&\le (\|v-u\| + \|u-w\|)^2
\end{aligned}$$

und somit auch für die Wurzel

$$\|v-w\| \le \|v-u\| + \|u-w\|.$$

In dieser Herleitung ist an der mit (CS) gekennzeichneten Zeile noch ein Umformungsschritt, von dessen Richtigkeit wir uns erst genauer überzeugen müssen. Dieser Schritt folgt unmittelbar aus dem nächsten Satz, den wir nicht beweisen werden.

Satz 2.1 (Cauchy-Schwartz'sche Ungleichung) Für zwei Vektoren v und w in $\mathbb{R}^n$ gilt

$$|v\cdot w| \le \|v\|\|w\|.$$

Gleichheit tritt genau dann ein, wenn v und w linear abhängig sind.

Da natürlich $v\cdot w \le |v\cdot w|$ ist, gilt also auch

$$v\cdot w \le \|v\|\|w\|,$$

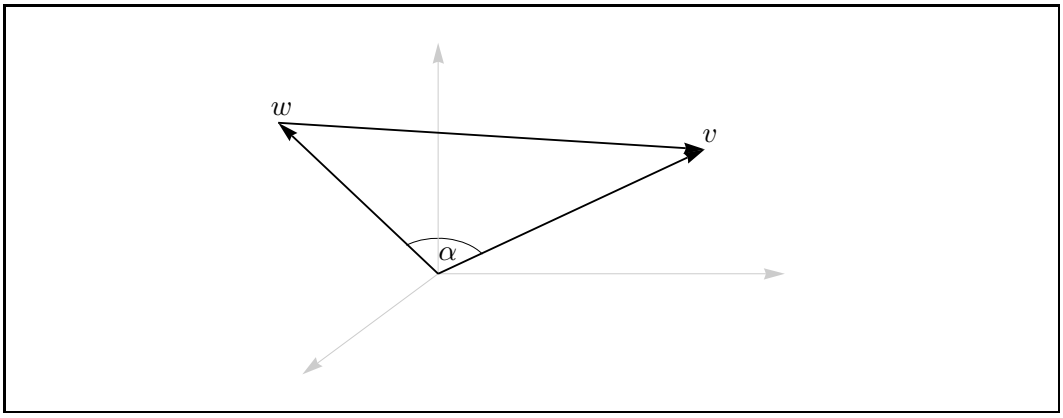

Abbildung 2.2: Illustration zur Herleitung des Winkels zwischen zwei Vektoren im $\mathbb{R}^n$.

wie wir für Umformung (CS) benötigen.

Mit Satz 2.1 gilt somit auch die Dreiecksungleichung, und damit insgesamt folgender Satz.

Satz 2.2 Die durch das Skalarprodukt auf $\mathbb{R}^n$ gegebene Länge ist eine Metrik auf $\mathbb{R}$.

Mit den Definitionen über Skalarprodukt und Länge von Vektoren kann man auch den *Winkel* zwischen zwei Vektoren im $\mathbb{R}^n$ definieren. Wir verallgemeinern dazu unsere geometrische Vorstellung aus dem $\mathbb{R}^3$. Mit dem Cosinussatz gilt für das in Abbildung 2.2 dargestellte Dreieck die Gleichung

$$\|v - w\|^2 = \|v\|^2 + \|w\|^2 - 2\|v\|\|w\| \cos \alpha,$$

wobei α der Winkel zwischen v und w ist. Diese Gleichung ergibt ausmultipliziert unter Verwendung von $\|u\|^2 = u^2$

$$v^2 - 2v \cdot w + w^2 = v^2 + w^2 - 2\|v\|\|w\| \cos \alpha\,.$$

Wir *definieren* nun den Winkel zwischen zwischen zwei Vektoren im $\mathbb{R}^n$ folgendermaßen.

Definition 2.3 (Winkel zwischen Vektoren)
Seien v und w zwei Vektoren im $\mathbb{R}^n$. Dann ist der Kosinus des Winkels α zwischen v und w definiert als

$$\cos \alpha = \frac{v \cdot w}{\|v\|\|w\|}.$$

Hier muss man nun nachprüfen, ob diese Definition überhaupt mathematisch korrekt ist: Der Kosinus eines Winkels muss immer zwischen -1 und 1 liegen. Mit Satz 2.1 gilt das auch für obige Definition.

Beispiel 2.2 Der Winkel α zwischen der x-Achse und der Diagonalen im $\mathbb{R}^2$ ist durch die Gleichung

$$\cos(\alpha) = \frac{\begin{pmatrix}1\\0\end{pmatrix} \cdot \begin{pmatrix}1\\1\end{pmatrix}}{\left\|\begin{pmatrix}1\\0\end{pmatrix}\right\| \left\|\begin{pmatrix}1\\1\end{pmatrix}\right\|} = \frac{1}{\sqrt{2}}$$

als $\alpha = \arccos\left(\frac{\sqrt{2}}{2}\right) = \frac{\pi}{4} = 45°$ zu berechnen. □

Beispiel 2.3 Der Winkel α zwischen den Vektoren $(1, 2, 0)$ und $(0, -1, 3)$ im $\mathbb{R}^3$ ist wegen

$$\cos(\alpha) = \frac{\begin{pmatrix}1\\2\\0\end{pmatrix} \cdot \begin{pmatrix}0\\1\\3\end{pmatrix}}{\left\|\begin{pmatrix}1\\2\\0\end{pmatrix}\right\| \left\|\begin{pmatrix}0\\1\\3\end{pmatrix}\right\|} = \frac{2}{\sqrt{5}\sqrt{10}}$$

gleich $\alpha = \arccos\left(\frac{\sqrt{2}}{5}\right) = 1.284 = 73.57°$. □

2.2 Projektionen

Wir werden in diesem Abschnitt sehen, wie man Vektoren aufeinander projizieren kann, und welche theoretischen Schlussfolgerungen sich daraus ziehen lassen. Dafür müssen wir uns zuerst überlegen, wie man sinnvoll definieren kann, dass zwei Vektoren aufeinander senkrecht stehen. Dafür ist der im letzten Abschnitt eingeführte Begriff des Winkels zwischen Vektoren hilfreich. Da der Kosinus bei einem Winkel von 90° null wird, definiert man den Begriff *senkrecht* für Vektoren wie folgt.

Definition 2.4 (Normalvektor)
Sei $v \in \mathbb{R}^n$. Einen Vektor $w \in \mathbb{R}^n$ mit

$$v \cdot w = 0$$

nennt man *Normalvektor* von v. Man sagt auch, dass v und w *senkrecht*, *normal* oder *orthogonal* aufeinander stehen und schreibt dafür $v \perp w$.

Man kann im folgenden Beispiel sehen, dass die Begriffe *Skalarprodukt* und damit auch *Normalität* in vielen Vektorräumen sinnvoll anwendbar sind.

Beispiel 2.4 Für gegebenes n sein $\mathcal{P}_n$ der Vektorraum aller Polynomfunktionen, deren Grad höchstens n ist. Wir definieren ein Skalarprodukt auf $\mathcal{P}_n$ durch

$$f \cdot g = \int_{-1}^{1} f(x)g(x)dx.$$

Dann stehen die sechs Vektoren

$$P_0 = 1 \qquad P_1 = x \qquad P_2 = x^2 - \frac{1}{3}$$

$$P_3 = x^3 - \frac{3}{5}x \qquad P_4 = x^4 - \frac{6}{7}x^2 + \frac{3}{35} \qquad P_5 = x^5 - \frac{10}{9}x^3 + \frac{5}{21}x$$

(genannt *Legendre-Polynome*) senkrecht aufeinander und bilden eine Basis von $\mathcal{P}_6$. Die zweite Behauptung ist leicht nachzuprüfen, da die Vektoren linear unabhängig sind und $\mathcal{P}_6$ Dimension 6 hat. Für die erste Behauptung überprüfen wir die Normalität nur eines von 15 Paaren von Polynomen, etwa P_2 und P_4:

$$\begin{aligned}\int_{-1}^{1}\Big(x^2 - \frac{1}{3}\Big)\Big(x^4 - \frac{6}{7}x^2 + \frac{3}{35}\Big)dx &= \int_{-1}^{1}\Big(x^6 - \frac{25}{21}x^4 + \frac{13}{35}x^2 - \frac{1}{35}\Big)dx \\ &= \frac{1}{7}x^7 - \frac{5}{21}x^5 + \frac{13}{1-5}x^3 - \frac{1}{35}x\Big|_{-1}^{1} \\ &= 0. \qquad \square\end{aligned}$$

Bei reellen Vektorräumen ist die Überprüfung der Orthogonalität von Vektoren einfacher.

Beispiel 2.5 Von den drei Vektoren $a = (-1, 2, 1, -1)$, $b = (2, 5, 0, 8)$ und $c = (-7, -2, 2, 3)$ stehen sowohl a und b als auch b und c senkrecht aufeinander, nicht aber a und c, wie man aus unterer Rechnung leicht sieht.

$$\begin{pmatrix}-1\\2\\1\\-1\end{pmatrix} \cdot \begin{pmatrix}2\\5\\0\\8\end{pmatrix} = 0, \quad \begin{pmatrix}2\\5\\0\\8\end{pmatrix} \cdot \begin{pmatrix}-7\\-2\\2\\3\end{pmatrix} = 0, \quad \text{aber} \quad \begin{pmatrix}-1\\2\\1\\-1\end{pmatrix} \cdot \begin{pmatrix}-7\\-2\\2\\3\end{pmatrix} = 2. \qquad \square$$

Mit dem Begriff der Normalität zweier Vektoren lässt sich auch einer der bekanntesten Sätze der Geometrie formulieren.

Satz 2.3 (Pythagoras) Seien $v, w \in V$ zwei Vektoren mit $v \perp w$. Dann gilt

$$\|v + w\|^2 = \|v\|^2 + \|w\|^2.$$

Beispiel 2.6 Im letzten Beispiel haben wir gezeigt, dass die beiden Vektoren $(-1, 2, 1, -1)$ und $(2, 5, 0, 8)$ und die beiden Vektoren $(2, 5, 0, 8)$ und $(-7, -2, 2, 3)$ jeweils senkrecht aufeinander stehen. Mit dem Satz von Pythagoras ist das Längenquadrat der Summe der Vektoren gleich der Summe der Längenquadrate der Vektoren, wie man anhand dieses Beispiels nachprüfen kann. Es ist

$$\left\|\begin{pmatrix}-1\\2\\1\\-1\end{pmatrix} + \begin{pmatrix}2\\5\\0\\8\end{pmatrix}\right\|^2 = 100 = \left\|\begin{pmatrix}-1\\2\\1\\-1\end{pmatrix}\right\|^2 + \left\|\begin{pmatrix}2\\5\\0\\8\end{pmatrix}\right\|^2 = 7 + 93$$

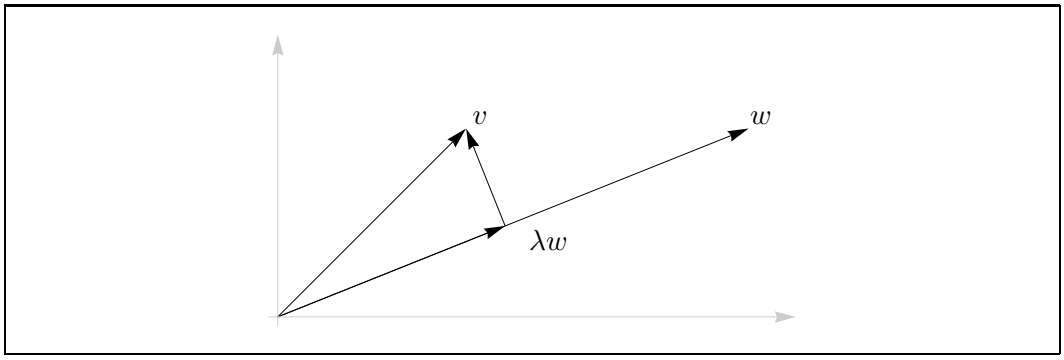

Abbildung 2.3: Illustration zur Projektion des Vektors v auf den Vektor w.

und

$$\left\| \begin{pmatrix} 2 \\ 5 \\ 0 \\ 8 \end{pmatrix} + \begin{pmatrix} -7 \\ -2 \\ 2 \\ 3 \end{pmatrix} \right\|^2 = 159 = \left\| \begin{pmatrix} 2 \\ 5 \\ 0 \\ 8 \end{pmatrix} \right\|^2 + \left\| \begin{pmatrix} -7 \\ -2 \\ 2 \\ 3 \end{pmatrix} \right\|^2 = 93 + 66. \qquad \square$$

Da wir die Länge von Vektoren berechnen können ist es auch möglich, einen Vektor auf einen anderen zu *projizieren*. Für gegebene Vektoren v und w möchte man denjenigen Abschnitt auf w finden, der sich durch senkrechte Projektion von v auf w ergibt. Die Problemstellung ist in Abbildung 2.3 erläutert. Gesucht ist derjenige Skalarfaktor λ, für den der Vektor $v - \lambda w$ senkrecht auf w steht. Dies lässt sich so ausdrücken:

$$(v - \lambda w) \cdot w = 0 \qquad \Leftrightarrow \qquad \lambda = \frac{v \cdot w}{\|w\|^2}.$$

Wir verwenden diese Herleitung für eine allgemeine Definition der orthogonalen Projektion.

Definition 2.5 (Projektion)
Seien v und w zwei Vektoren. Dann bezeichnet man den Vektor

$$\mathrm{proj}_w(v) = \frac{v \cdot w}{\|w\|^2} w$$

als *orthogonale Projektion* von v auf w.

Man kann an einem einfachen Beispiel überprüfen, dass diese Formel mit unserer Intuition von Projektion übereinstimmt.

Beispiel 2.7 Die Projektion eines beliebigen Vektors $v = (v_1, v_2, v_3) \in \mathbb{R}^3$ auf die

y-Achse $\{w \in \mathbb{R}^3 \mid w = \lambda(0,1,0)\}$ ergibt den Vektor

$$\text{proj}_{\text{y-Achse}}(v) = \frac{\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}}{\left\| \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\|^2} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ v_2 \\ 0 \end{pmatrix}.$$

Dies deckt sich mit unseren Erwartung, dass eine Projektion auf die y-Achse dem Abschnitt entspricht, der durch die y-Koordinate des Punkts gegeben ist. □

Beim Arbeiten mit Projektionen lohnt es sich oft, für den umgebenden Vektorraum spezielle Basen zu konstruieren, mit denen sich dann besonders einfach rechnen lässt. Diese Basen, bestehend aus Vektoren, die senkrecht aufeinander stehen, sind wie folgt definiert.

Definition 2.6 (Orthonormalbasen)
Eine Basis $\{b_1, \ldots, b_n\}$ eines Vektorraums V mit den Eigenschaften

$$\begin{aligned} b_i \perp b_j \qquad & \text{für alle } i \neq j \\ \|b_i\| = 1 \qquad & \text{für alle } i \end{aligned}$$

nennt man *Orthonormalbasis (ONB)* von V.

Beispiel 2.8 Das einfachste Beispiel einer Orthogonalbasis ist die Standardbasis des $\mathbb{R}^n$, wie man sofort nachprüfen kann. □

Beispiel 2.9 Die Rotation um den Winkel φ relativ zum Ursprung wird im zweidimensionalen Raum durch die lineare Abbildung

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

ausgedrückt. Wenn wir die Spalten dieser Rotationsmatrix als Vektoren im $\mathbb{R}^2$ auffassen, so formen sie eine Orthonormalbasis des $\mathbb{R}^2$. Dies lässt sich numerisch aus

$$\begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} \cdot \begin{pmatrix} -\sin\varphi \\ \cos\varphi \end{pmatrix} = -\cos\varphi\sin\varphi + \sin\varphi\cos\varphi = 0$$

und

$$\left\| \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} \right\| = \left\| \begin{pmatrix} -\sin\varphi \\ \cos\varphi \end{pmatrix} \right\| = \sqrt{\sin^2\varphi + \cos^2\varphi} = 1$$

für beliebige Winkel φ sofort überprüfen. □

Im letzten Beispiel haben wir nur die beiden Bedingungen einer Orthonormalbasis aus Definition 2.6 nachgeprüft und nicht, ob die Vektoren überhaupt eine Basis bilden. Die Eigenschaften der linearen Unabhängigkeit und Erzeugendensystems

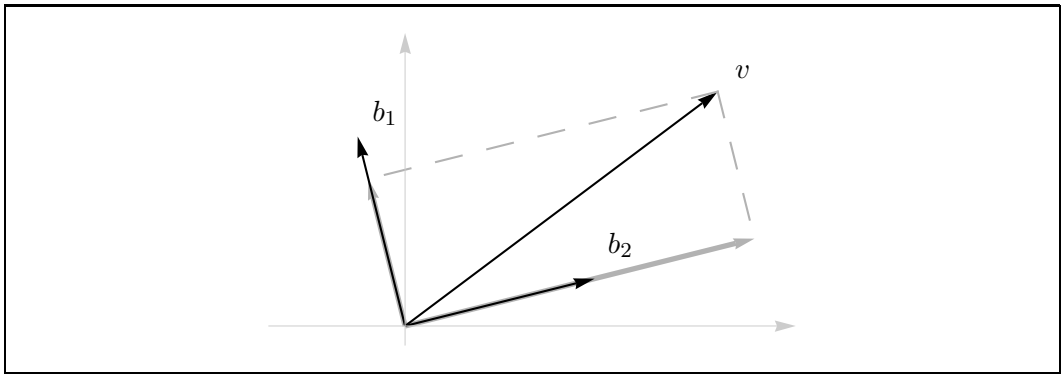

Abbildung 2.4: Der Vektor v lässt sich als Summe der beiden grauen Vektoren schreiben, die Projektionen auf die Orthonormalbasis $\{b_1, b_2\}$ sind.

folgen aber aus den Orthonormalitätsbedingungen, wie man unschwer nachrechnen kann.

Orthonormalbasen haben gegenüber "herkömmlichen" Basen den großen Vorteil, dass man mit ihnen leichter rechnen kann. So muss man zur Berechnung der Koordinaten eines Vektors bezüglich einer Orthonormalbasis keine aufwändigen Berechnungen durchführen, wie dies bei anderen Koordinatentransformationen der Fall ist. Zur Erinnerung: Ein Koordinatenvektor $(\lambda_1, \ldots, \lambda_n)$, der Koordinaten bezüglich einer Basis repräsentiert, deren Spalten die Matrix A bilden, wird durch folgende Formel in einer Koordinatenvektor $(\mu_1, \ldots, \mu_n)$ bezüglich der Basis umgerechnet, deren Spalten durch die Matrix B gebildet werden:

$$\begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix} = B^{-1} \cdot A \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix}.$$

Eine Vereinfachung ergibt sich, wenn der zu tranformierende Vektor bezüglich der Standardbasis gegeben ist, da dann die Matrix A in obiger Gleichung die Einheitmatrix ist.

Koordinaten bezüglich Orthonormalbasen können leichter berechnet werden. Eine einfache Überlegung zeigt, warum das so ist. Sei dazu $(b_1, \ldots, b_n)$ eine geordnete Orthonormalbasis, bezüglich derer der Vektor v die Koordinaten $\lambda_1, \ldots, \lambda_n$ hat. Somit ist

$$v = \lambda_1 b_1 + \cdots + \lambda_n b_n,$$

und wir wollen die λ_i bestimmen. Zuerst müssen wir uns überlegen, dass für beliebige Vektoren u, v und w gilt: aus $u = v$ folgt $u \cdot w = v \cdot w$ (sofort einsichtig). Wir multiplizieren dann die Koordinatendarstellung oben auf beiden Seiten mit beliebigem b_i und erhalten

$$b_i \cdot v = \lambda_1 \underbrace{b_i \cdot b_1}_{=0} + \cdots + \lambda_i \underbrace{b_i \cdot b_i}_{=1} + \cdots + \lambda_n \underbrace{b_i \cdot b_n}_{=0}.$$

Somit ergibt sich folgendes einfache Resultat.

Satz 2.4 Sei $B = (b_1, \ldots, b_n)$ eine Orthonormalbasis. Dann ist die i-te Koordinate λ_i eines Vektors v bezüglich B gegeben durch

$$\lambda_i = b_i \cdot v.$$

Beispiel 2.10 Die drei Vektoren $\frac{1}{\sqrt{3}}(1,1,1)$, $\frac{1}{\sqrt{6}}(-1,2,-1)$ und $\frac{1}{\sqrt{2}}(-1,0,1)$ bilden eine Orthonormalbasis des $\mathbb{R}^3$, wie man leicht nachprüfen kann. Die Koordinaten des Punktes $(-1,0,2)$ bezüglich dieser Basis sind

$$\frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix} \cdot \begin{pmatrix}-1\\0\\2\end{pmatrix} = \frac{1}{\sqrt{3}}, \qquad \frac{1}{\sqrt{6}}\begin{pmatrix}-1\\2\\-1\end{pmatrix} \cdot \begin{pmatrix}-1\\0\\2\end{pmatrix} = -\frac{1}{\sqrt{6}},$$

$$\frac{1}{\sqrt{2}}\begin{pmatrix}-1\\0\\1\end{pmatrix} \cdot \begin{pmatrix}-1\\0\\2\end{pmatrix} = \frac{3}{\sqrt{2}}.$$

Die Berechnungen in diesem Beispiel können mittels Matrizenmultiplikation kürzer geschrieben werden. Wenn wir die drei Basiselemente nämlich in den *Zeilen* einer Matrix anordnen (und nicht wie sonst üblich in den *Spalten*), kann man obige Rechnungen schreiben als

$$\begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix}-1\\0\\2\end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{6}} \\ \frac{3}{\sqrt{2}} \end{pmatrix}. \qquad \square$$

Orthonormalbasen haben also den großen Vorteil, dass Koordinatenberechnungen und Koordinatentransformationen leichter durchzuführen sind als mit normalen Basen. Die wenigsten Mengen von linear unabhängigen Vektoren bestehen aber aus paarweise orthogonalen Vektoren; um die Vorteile von Orthonormalbasen nützen zu können, müssen diese Basen erst orthogonalisiert und dann normiert werden. Es ist durch eine einfache geometrische Überlegung einsichtig, dass dies für alle Mengen von Vektoren möglich ist.

Wie man sich anhand einer Skizze veranschaulichen kann (etwa Abbildung 2.3 auf Seite 20), steht die Verbindungslinie zwischen dem zu projizierenden Vektor v und dem projizierten Vektor $\text{proj}_w(v)$ immer senkrecht auf den Vektor w, auf den projiziert wird. Es gilt also allgemein

$$v - \text{proj}_w(v) \perp w.$$

Dabei ist wichtig, dass sich die lineare Hülle von v und w nicht ändert, wenn man v durch $v - \text{proj}_w(v)$ ersetzt.

Beispiel 2.11 Mit dieser Einsicht können wir die beiden Vektoren $(1,2,3)$ und $(1,0,1)$ orthogonalisieren, indem wir etwa den Vektor $(1,2,3)$ so umformen, dass er orthogonal zu $(1,0,1)$ steht. Es ist

$$\begin{pmatrix}1\\2\\3\end{pmatrix} - \operatorname{proj}_{\left(\begin{smallmatrix}1\\0\\1\end{smallmatrix}\right)}\begin{pmatrix}1\\2\\3\end{pmatrix} = \begin{pmatrix}1\\2\\3\end{pmatrix} - \frac{\begin{pmatrix}1\\2\\3\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix}}{\left\|\begin{pmatrix}1\\0\\1\end{pmatrix}\right\|^2}\begin{pmatrix}1\\0\\1\end{pmatrix} = \begin{pmatrix}1\\2\\3\end{pmatrix} - 2\begin{pmatrix}1\\0\\1\end{pmatrix} = \begin{pmatrix}-1\\2\\1\end{pmatrix}.$$

Es ist sofort ersichtlich, dass die Vektoren $(1,0,1)$ und $(-1,2,1)$ senkrecht aufeinander stehen. □

Diese Methode kann man auf mehrere Vektoren verallgemeinern: Man iteriert durch die Menge der Vektoren und zieht von jeden Vektor v die Projektionen auf alle vorherig orthogonalisierten Vektoren ab, um v auf diese Vektoren senkrecht zu stellen. Dieser Verfahren wird *Gram-Schmidt'sches Orthogonalisierungsverfahren* genannt. Die so erzeugte Menge von Vektoren spannt dabei den gleichen Vektorraum wie die ursprüngliche Menge auf.

Satz 2.5 (Gram-Schmidt) Seien $b_1, \ldots, b_n$ eine Basis von V. Dann ist die Menge von Vektoren

$$\begin{aligned}
v_1 &= b_1\\
v_2 &= b_2 - \operatorname{proj}_{v_1}(b_2)\\
v_3 &= b_3 - \operatorname{proj}_{v_1}(b_3) - \operatorname{proj}_{v_2}(b_3)\\
&\vdots\\
v_n &= b_n - \operatorname{proj}_{v_1}(b_n) - \operatorname{proj}_{v_2}(b_n) - \cdots - \operatorname{proj}_{v_{n-1}}(b_n)
\end{aligned}$$

eine orthogonale Basis von V.

Um aus der so erzeugten Basis eine Orthonormalbasis zu machen, muss man die einzelnen Vektoren nur mehr normalisieren. Wir illustrieren dies an einem Beispiel.

Beispiel 2.12 Die drei Vektoren $(1,0,-1), (2,0,1)$ und $(1,1,1)$ bilden eine Basis des $\mathbb{R}^3$. Um daraus eine Orthonormalbasis zu erzeugen, wenden wir das Gram-Schmidt'sche Verfahren an und erhalten so die Vektoren

$$v_1 = \begin{pmatrix}1\\0\\-1\end{pmatrix}$$

$$v_2 = \begin{pmatrix}2\\0\\1\end{pmatrix} - \operatorname{proj}_{\left(\begin{smallmatrix}1\\0\\-1\end{smallmatrix}\right)}\begin{pmatrix}2\\0\\1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}3\\0\\3\end{pmatrix}$$

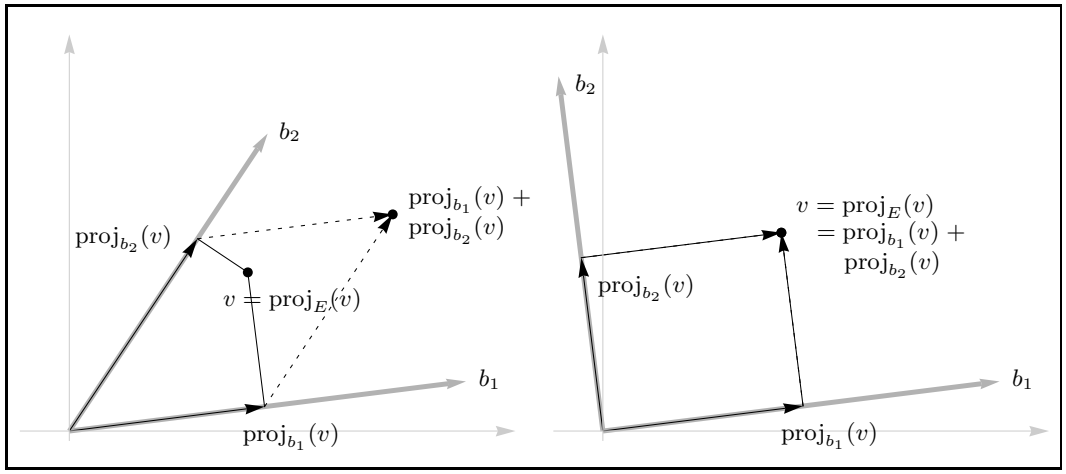

Abbildung 2.5: Illustration der Projektion von v auf die Ebene E, die von b_1 und b_2 aufgespannt wird. Die Ansichten sind von oben entlang der Projektion von v auf E, sodass v und $\text{proj}_E(v)$ in den Abbildungen identisch sind. Links sind b_1 und b_2 nicht orthogonal, sodass $\text{proj}_E(v) \neq \text{proj}_{b_1}(v) + \text{proj}_{b_2}(v)$ ist. Wenn b_1 und b_2 orthogonal sind, dann gilt die Aussage von Satz 2.6.

$$v_3 = \begin{pmatrix}1\\1\\1\end{pmatrix} - \text{proj}_{\begin{pmatrix}1\\0\\-1\end{pmatrix}} \begin{pmatrix}1\\1\\1\end{pmatrix} - \text{proj}_{\frac{1}{2}\begin{pmatrix}3\\0\\3\end{pmatrix}} \begin{pmatrix}1\\1\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\end{pmatrix}.$$

Durch Nachrechnen kann man sich überzeugen, dass diese Vektoren senkrecht aufeinander stehen. Mit Normalisieren erhalten wir die Vektoren $\frac{1}{\sqrt{2}}(1,0,-1)$, $\frac{1}{\sqrt{2}}(1,0,1)$ und $(0,1,0)$, die eine Orthonormalbasis des $\mathbb{R}^3$ bilden. □

Ein weitere wichtiger Zusammenhang zwischen v, w, und $\text{proj}_w(v)$ ist die Tatsache, dass $\text{proj}_w(v)$ derjenige Punkt von w ist, der v am nächsten ist; auch dies ist aus einer Skizze wie in Abbildung 2.3 auf Seite 20 zu erkennen. Man kann diesen Zusammenhang verwenden, um mittels Projektion denjenigen Punkt eines Unterraums zu bestimmen, der einem gegebenen Punkt am nächsten liegt. Dazu muss man zuerst überlegen, wie man auf einen Unterraum projizieren kann.

Im einfachsten Fall ist $E = L(\{b_1, b_2\})$ eine Ebene, die von den beiden Vektoren b_1 und b_2 aufgespannt wird. Wir bezeichnen mit $\text{proj}_E(v)$ den Projektionspunkt von v auf E. Wie man aus Abbildung 2.5 (links) erkennen kann genügt es nicht, v auf b_1 und b_2 zu projizieren, und die Ergebnisse zu addieren: Das Resultat ist im Allgemeinen nicht der gewünschte Projektionspunkt. Erst wenn die beiden Vektoren b_1 und b_2 senkrecht aufeinander stehen (wie in Abbildung 2.5 rechts zu sehen), erhält man den Projektionspunkt durch Addition der Projektionen auf b_1 und b_2.

Die Verallgemeinerung dieser Beobachtung auf beliebige Unterräume ist im nächsten Satz zusammengefasst.

Satz 2.6 Sei U ein Unterraum eines Vektorraums V, $v \in$ V, sowie $\{b_1, \ldots, b_n\}$ eine orthogonale Basis von U. Dann ist derjenige Punkt von U, der v am nächsten liegt, gegeben durch

$$\text{proj}_U(v) = \text{proj}_{b_1}(v) + \cdots + \text{proj}_{b_n}(v).$$

Wenn die Basis $\{b_1, \ldots, b_n\}$ von U im letzten Satz sogar eine ONB ist, vereinfacht sich die Projektion auf U zu $\text{proj}_U(v) = \sum_{k=1}^{n} b_k \cdot v\, b_k$. Aus diesen Formeln sieht man, dass die Projektion eines Punktes v ausserhalb eines Unterraums U auf U der Koordinatenberechnung von v bezüglich einer orthogonalen Basis von U entspricht (obwohl dieser Punkt gar nicht in U ist).

Beispiel 2.13 Sei E die Ebene, die von $b_1 = (-1, 0, 1)$ und $b_2 = (2, 1, 0)$ aufgespannt wird, sowie $v = (2, 1, -2)$ ein Punkt, der nicht auf der Ebene liegt. Da b_1 und b_2 nicht senkrecht stehen, liefert

$$p' = \text{proj}_{b_1}(v) + \text{proj}_{b_2}(v) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ -2 \end{pmatrix}$$

noch nicht das richtige Ergebnis. Das erkennt man etwa daran, dass der Abstand $\|v - p'\| = 2$ noch nicht der kürzeste Abstand von v zu E ist. Wenn man nämlich b_1 und b_2 zuerst orthogonalisiert, erhält man $v_1 = b_1 = (-1, 0, 1)$ und $v_2 = b_2 - \text{proj}_{b_1}(b_2) = (1, 1, 1)$, und damit

$$v' = \text{proj}_E(v) = \text{proj}_{v_1}(v) + \text{proj}_{v_2}(v) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix} + \frac{1}{3}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 7 \\ 1 \\ -5 \end{pmatrix}.$$

Der Abstand von v zu v' ist $\|v - v'\| = \sqrt{\frac{2}{3}}$ und damit kürzer als der Abstand von v zu p'. □

Matrizen, deren Spalten (oder Zeilen) eine Orthonormalbasis bilden, nennt man *orthogonal*. Die Definition lautet anders, ist aber mit dieser Aussage äquivalent, wie wir weiter unten sehen werden.

Definition 2.7 (Orthogonale Matrix)
Eine reelle reguäre $n \times n$ Matrix A heißt *orthogonal*, wenn gilt

$$A^{-1} = A^T.$$

Beispiel 2.14 Wie zu vermuten ist, ist die Matrix A einer Rotation im $\mathbb{R}^2$ aus Beispiel 2.9 eine orthogonale Matrix. Wir rechnen nach, dass für A die (etwa umformulierte) Orthogonalitätsbedingung $A \cdot A^T = I_2$ gilt:

$$\begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \cdot \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} = \begin{pmatrix} \cos^2\varphi + \sin^2\varphi & 0 \\ 0 & \cos^2\varphi + \sin^2\varphi \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$ □

Für orthogonale Matrizen gilt eine spezielle Eigenschaft, die im nächsten Satz ausgedrückt wird. Damit kann man für Matrizen leicht erkennen, ob sie orthogonal sind oder nicht.

Satz 2.7 Sei A eine reelle $n \times n$ Matrix. Dann sind die folgenden drei Bedingungen äquivalent:

(i) A ist orthogonal.

(ii) Die Spalten von A bilden eine Orthonormalbasis von $\mathbb{R}^n$.

(iii) Die Zeilen von A bilden eine Orthonormalbasis von $\mathbb{R}^n$.

Äquivalenz von Aussagen bedeutet, dass jede der Aussagen aus jeder anderen folgt. Wir werden diese Äquivalenzen nicht beweisen.

Beispiel 2.15 Die Matrix

$$A = \frac{1}{3}\begin{pmatrix} 1 & 2 & -2 \\ 2 & 1 & 2 \\ -2 & 2 & 1 \end{pmatrix}$$

ist orthogonal, da die Spalten (und Zeilen) aufeinander senkrecht stehen und Länge 1 haben. Aus Satz 2.7 folgt dann, dass $A^{-1} = A^T$ ist. □

2.3 Geometrie von Funktionen

Wie wir schon in Beispiel 2.4 anhand von Polynomen gesehen haben ist es möglich, auf stetigen Funktionen ein Skalarprodukt festzulegen. Wir geben hier eine etwas allgemeinere Definition, die auch komplexe Funktionen zulässt; dies ist in vielen technischen Anwendung nötig.

Definition 2.8 (Skalarprodukt auf Funktionen)
Seien $f, g : [a, b] \to \mathbb{C}$ zwei komplexwertige, auf dem Intervall $[a, b]$ stetige Funktionen. Das *Skalarprodukt* von f und g ist definiert als

$$f \cdot g = \int_a^b f(x)\overline{g(x)}dx.$$

Hierbei bezeichnet $\overline{z}$ die zu $z = a + ib \in \mathbb{C}$ *konjugiert komplexe Zahl* $\overline{z} = a - ib$.

Mit Hilfe dieses Skalarprodukts können nun Längen und Winkel von Funktionen berechnet werden; so ist etwa die Länge von f (die wir in diesem Kontext meist als *Norm* bezeichnen) durch

$$\|f\| = \sqrt{\int_a^b f(x)\overline{f(x)}dx}$$

gegeben. Fast alle Resultate als den letzten Abschnitten (wie etwa Cauchy-Schwartz'sche Ungleichung oder Projektionen) sind allgemein genug formuliert, um auch für das Funktionenskalarprodukt zu gelten. Der einzige Unterschied zum Skalarprodukt auf reellen Vektoren liegt darin, dass bei reellen Vektoren v aus $v \cdot v = 0$ folgen muss,

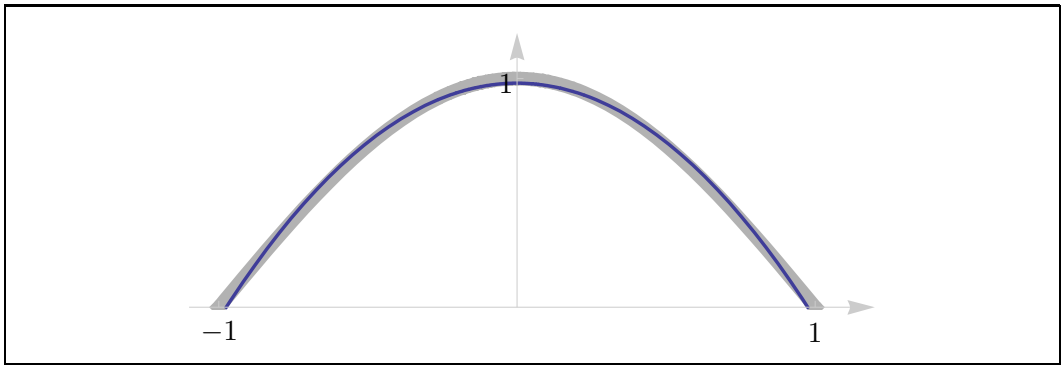

Abbildung 2.6: Approximation von $\cos\left(\frac{\pi}{2}x\right)$ (grau) durch das Polynom 2. Grades (dünn) aus Beispiel 2.16.

dass $v = 0$ ist. Dies ist durch die Verwendung des Integrals beim Skalarprodukt auf Funktionenräumen nicht der Fall.

Da Projektionen nur vom Skalarprodukt abhängen, können wir auch Funktionen aufeinander projizieren; auch hier sind Orthonormalbasen von speziellem Interesse. Wenn wir uns überlegen, dass wir durch Projektion eines Vektors v auf einen Unterraum U (gegeben durch eine Menge von Vektoren, die ihn aufspannen) denjenigen Vektor in U bestimmen, der v am nächsten ist, so macht es durchaus Sinn, Funktionen auf einen Funktionsunterraum zu projizieren. Wir erhalten so diejenigen Funktionen im Unterraum, die die gegebene Funktion am besten approximieren. Wir illustrieren dies an einem einfachen Beispiel.

Beispiel 2.16 Die drei Vektoren 1, x und x^2 spannen den Vektorraum $\mathcal{P}_2$ der Polynomfunktionen mit Grad maximal 2 auf. Wir suchen nun die beste Approximation an die Funktion $\cos(\frac{\pi}{2}x)$ im Intervall $[-1, 1]$. Die Vektoren 1, x und x^2 stehen aber noch nicht senkrecht aufeinander. Mit dem Gram-Schmidt'schen Orthogonalisierungsverfahren erhalten wir die Basis 1, x und $x^2 - \frac{1}{3}$. Die Projektionen von $\cos(\frac{\pi}{2}x)$ auf diese drei Vektoren ergeben

$$\operatorname{proj}_1\Big(\cos\big(\frac{\pi}{2}x\big)\Big) = \frac{\int_{-1}^{1}\cos\big(\frac{\pi}{2}x\big)dx}{\int_{-1}^{1}1^2\,dx}\cdot 1 = \frac{\frac{4}{\pi}}{2}\cdot 1 = \frac{2}{\pi}\cdot 1$$

$$\operatorname{proj}_x\Big(\cos\big(\frac{\pi}{2}x\big)\Big) = \frac{\int_{-1}^{1}\cos\big(\frac{\pi}{2}x\big)x dx}{\int_{-1}^{1}x^2\,dx}\cdot x = \frac{0}{\frac{2}{3}}\cdot x = 0\cdot x$$

$$\begin{aligned}\operatorname{proj}_{x^2-\frac{1}{3}}\Big(\cos\big(\frac{\pi}{2}x\big)\Big) &= \frac{\int_{-1}^{1}\cos\big(\frac{\pi}{2}x\big)\big(x^2-\frac{1}{3}\big)dx}{\int_{-1}^{1}\big(x^2-\frac{1}{3}\big)^2\,dx}\cdot\Big(x^2-\frac{1}{3}\Big)\\ &= \frac{\frac{8\pi^2-96}{3\pi^3}}{\frac{8}{45}}\cdot\Big(x^2-\frac{1}{3}\Big) = \frac{15(\pi^2-12)}{\pi^3}\cdot\Big(x^2-\frac{1}{3}\Big)\end{aligned}$$

Somit ist die beste Approximation an $\cos(\frac{\pi}{2}x)$ im Intervall $[-1, 1]$ durch die Linearkombination

$$\frac{15(\pi^2-12)}{\pi^3}\Big(x^2-\frac{1}{3}\Big)+\frac{2}{\pi}$$

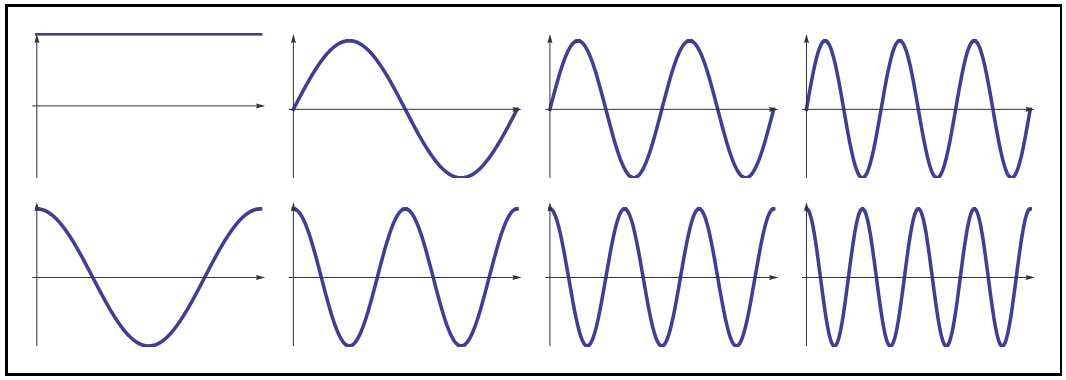

Abbildung 2.7: Die Fourier-Funktionen 1 und $\sin(kx)$ für $k = 1, 2, 3$ (erste Reihe) und $\cos(kx)$ für $k = 1, \ldots, 4$ (zweite Reihe).

gegeben. Eine graphische Darstellung dieser Approximation ist in Abbildung 2.6 zu sehen. □

Von speziellen Interesse werden für uns unendliche Mengen von Funktionen sein, bei denen alle Elemente zueinander orthogonal sind. Wir betrachten einige Beispiele von trigonometrischen Funktionen. Dafür fixieren wir das Intervall $[a, b] = [0, 2\pi]$, alle Resultate in diesem Abschnitt lassen sich mit etwas mehr notationellem Aufwand auf beliebige Intervalle anwenden.

Beispiel 2.17 Eine unendliche Menge von Funktionen ist $\{\sin(kx) \,|\, k \geq 1\}$. Die Elemente dieser Menge stehen senkrecht aufeinander: So gilt für beliebige $m \neq n \in \mathbb{N}$ (nachrechnen mit einigen Integrationstricks)

$$\int_0^{2\pi} \sin(mx)\sin(nx)dx = \frac{1}{m-n}\sin((m-n)\pi) - \frac{1}{m+n}\sin((m+n)\pi) = 0,$$

und für das Quadrat der Norm

$$\int_0^{2\pi} \sin^2(nx)dx = \pi - \frac{1}{2n}\sin(2n\pi) = \pi.$$

Somit ist $\left\{\frac{1}{\sqrt{\pi}}\sin(kx) \,|\, k \geq 1\right\}$ eine Menge von paarweise orthonormalen Funktionen auf dem Intervall $[0, 2\pi]$.

Ebenso kann man nachrechnen, dass die Menge $\{\frac{1}{\sqrt{\pi}}\cos(kx) \,|\, k \geq 1\}$ ebenfalls aus paarweise orthonormalen Elementen besteht. Mehr noch: Es gilt sogar, dass $\sin(mx)$ auf $\cos(nx)$ und auf die konstante Funktion 1 senkrecht steht, sodass die Menge der sogenannten *Fourier-Funktionen*

$$F = \{1\} \cup \{\sin(kx) \,|\, k \geq 1\} \cup \{\cos(kx) \,|\, k \geq 1\} \tag{2.1}$$

orthogonal und mit den geeigneten Skalierungen (um $\frac{1}{\sqrt{2\pi}}$ bzw. $\frac{1}{\sqrt{\pi}}$) orthonormal ist. Die ersten paar Fourier-Funktionen sind in Abbildung 2.7 zu sehen. □

Man kann zeigen, dass die Menge F der Fourier-Funktionen im Sinn des folgenden Satzes eine Basis des ziemlich allgemeinen Vektorraums der *quadratisch integrierbaren* Funktionen ist. Dies sind alle Funktionen f, für die $\int_a^b |f(x)|^2 dx$ endlich ist.

Satz 2.8 Sei g auf $[0, 2\pi]$ quadratisch integrierbar. Dann gibt es für jedes $\epsilon > 0$ eine endliche Linearkombination $\sum_{k=1}^{n} \lambda_k f_k$ mit Fourier-Funktionen f_k und reellen Koeffizienten λ_k, sodass gilt

$$\left\| g - \sum_{k=1}^{n} \lambda_k f_k \right\| < \epsilon.$$

Dieser Satz bedeutet, dass sich g (beliebig genau) als endliche Linearkombination der Basisfunktionen f_k darstellen lässt. Für die Approximation durch Fourier-Funktionen bezeichnet man die Koeffizienten traditionellerweise mit a_k bzw. b_k, sodass man

$$g(x) \approx a_0 + \sum_{k=1}^{m} a_k \cos(kx) + \sum_{k=1}^{n} b_k \sin(kx)$$

schreibt. Diese Reihe wird *Fourier-Reihe* genannt. Da die Fourier-Funktionen bis auf bekannte Normalisierungsfaktoren s_k eine Orthonormalbasis bilden, wissen wir mit Satz 2.4, dass die Koordinaten durch die Skalarprodukte $g \cdot f_k$ (genauer $s_k^2\, g \cdot f_k$)) gegeben sind. Wir erhalten somit folgende Werte, wobei der Skalierungsfaktor für die Konstante 1 den Wert $\frac{1}{\sqrt{2\pi}}$, und für die anderen Fourier-Funktionen den Wert $\frac{1}{\sqrt{\pi}}$ hat:

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} g(x) dx$$

$$a_k = \frac{1}{\pi} \int_0^{2\pi} g(x) \cos(kx) dx$$

$$b_k = \frac{1}{\pi} \int_0^{2\pi} g(x) \sin(kx) dx.$$

Beispiel 2.18 Wir approximieren das Polynom

$$p(x) = -\frac{2}{5}x^3 + \frac{7}{2}x^2 - 6x + 4$$

durch Fourier-Reihen. Mit den Formeln oben ergeben sich für die Koeffizienten der Basisfunktionen die Werte

$$(a_0, a_1, \ldots, a_5) = (6.404, -1.08, -0.27, -0.12, -0.067, -0.043)$$
$$(b_1, b_2, \ldots, b_5) = (-5.2, -0.8, -0.311, -0.175, -0.118).$$

Die Approximationen durch die ersten zwei bzw. fünf Sinus- und Kosinusbasisfunktionen sind in Abbildung 2.8 dargestellt. □

2.4 Hauptkomponentenanalyse

Wenn man einen Datensatz als Menge von Vektoren in einem (hochdimensionalen) reellen Vektorraum auffasst, dann können die Werkzeuge der linearen Algebra

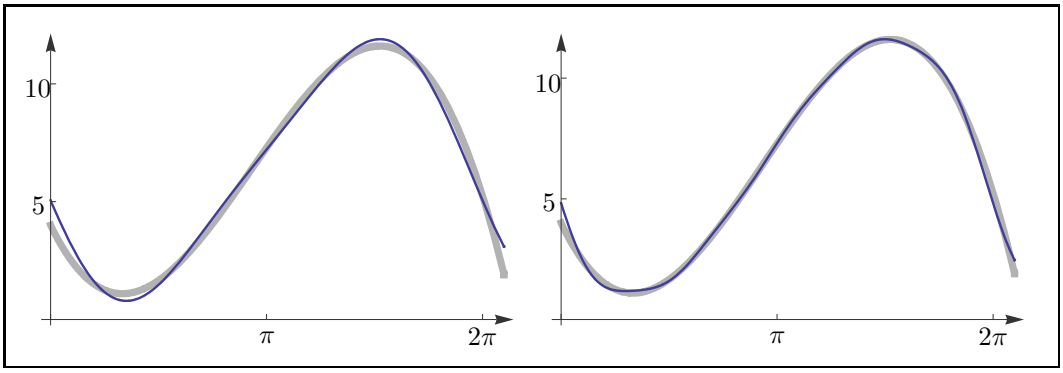

Abbildung 2.8: Approximation des kubischen Polynoms (grau) aus Beispiel 2.18 durch Fourier-Reihen (dunkel). Links die Approximation durch die ersten zwei, rechts durch die ersten fünf Reihenglieder.

verwendet werden, für die Datenanalyse relevante Informationen zu extrahieren. In diesem Abschnitt widmen wir uns einem solchen Verfahren, der *Hauptkomponentenanalyse (principal component analysis)*. Die dazu benötigten mathematischen Grundlagen werden zuerst kurz erarbeitet.

Bei der Hauptkomponentenanalyse handelt es sich um ein Verfahren, aus einer Punktwolke diejenigen Richtungen zu bestimmen, in denen diese Wolke besonders ausgeprägt ist. Da sich die Lage einer Punktwolke als spezielle Matrix repräsentieren lässt ist es von Interesse, spezielle "Richtungen" von Matrizen bestimmen zu können. Dies führt zu den Begriffen der *Eigenwerte* und *Eigenvektoren* von Matrizen.

Definition 2.9 (Eigenwert, Eigenvektor)
Sei A eine quadratische Matrix. Dann bezeichnet man einen Skalar λ, für den es einen Vektor $v \neq 0$ gibt mit

$$A \cdot v = \lambda v$$

als *Eigenwert*, und v als den zum Eigenwert λ gehörenden *Eigenvektor* von A.

Beispiel 2.19 Aus der Gleichung

$$\begin{pmatrix} 1 & 2 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 4 \\ -2 \end{pmatrix} = -2 \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$$

kann man erkennen, dass $\lambda = -2$ ein Eigenwert der gegebenen Matrix ist; ein zugehöriger Eigenvektor ist $(1, -2, 1)$. □

Da die Matrizenmultiplikation linear ist, ist auch der Vektor μv ein Eigenvektor zum Eigenwert λ, wenn v bereits einer ist. Dies folgt aus

$$A \cdot (\mu v) = \mu A \cdot v = \mu \lambda v = \lambda \mu v.$$

Somit spielt die Länge eines Eigenvektors keine Rolle; Eigenvektoren werden daher oft als Vektoren der Länge 1 angegeben (sogenannte *Einheitsvektoren*). Diese Normierung erreicht man, indem man den Vektor mit dem Inversen seiner Länge multipliziert; so ist etwa der Vektor $\frac{1}{\sqrt{6}}(1,-2,1)$ ein Einheitsvektor.

Zur Berechnung der Eigenwerte einer Matrix ist folgende Überlegung hilfreich, die wir anhand eines Beispiels illustrieren.

Beispiel 2.20 Gegeben sei die Matrix

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ -1 & 2 & 3 \end{pmatrix},$$

deren Eigenwerte und Eigenvektoren zu bestimmen sind. Eigenwerte λ und Eigenvektoren v müssen die Gleichung

$$A \cdot v = \lambda\, v$$

erfüllen. Diese Gleichung können wir durch Umformen auf die Gestalt

$$\begin{aligned} A \cdot v - \lambda\, v &= 0 \qquad \text{und weiter auf} \\ A \cdot v - \lambda\, I_n \cdot v &= 0 \qquad (I_n \text{ ist die Einheitsmatrix}), \text{ mit Herausheben} \\ (A - \lambda I_n) \cdot v &= 0 \end{aligned}$$

bringen. Wenn $(A - \lambda I_n)$ regulär ist, wird durch Multiplikation mit dieser Matrix nur der Nullpunkt auf den Nullpunkt abgebildet, und das Gleichungssystem

$$(A - \lambda I_n) \cdot v = 0$$

hat *keine* Lösung $v \neq 0$; und somit auch keinen Eigenvektor. Es kann also nur dann Eigenvektoren (und Eigenwerte) geben, wenn die Matrix $(A - \lambda I_n)$ *nicht* invertierbar ist, und dies ist genau dann der Fall, wenn ihre Determinante null ist.

Dies ergibt mit der Matrix A dieses Beispiels die Bedingung

$$\det \begin{pmatrix} 1-\lambda & 2 & 1 \\ 2 & 0-\lambda & -2 \\ -1 & 2 & 3-\lambda \end{pmatrix} = 0.$$

Nach einigem Umformen erhält man daraus die polynomiale Gleichung

$$\lambda^3 - 4\lambda^2 + 4\lambda = 0,$$

die sich nach Herausheben von λ schreiben lässt als

$$\lambda(\lambda - 2)^2 = 0.$$

Diese Gleichung besitzt die beiden Lösungen $\lambda = 0$ und $\lambda = 2$, die zugehörigen Eigenvektoren erhält man durch Einsetzen der Eigenwerte. Für $\lambda = 0$ ergibt sich

$$\begin{pmatrix} 1-0 & 2 & 1 \\ 2 & 0-0 & -2 \\ -1 & 2 & 3-0 \end{pmatrix} \cdot v = 0$$

Mit Anwendung des Gaußschen Eliminationsverfahren formt man die erweiterte Matrix des Gleichungssystems um auf

$$\left(\begin{array}{ccc|c} 1 & 2 & 1 & 0 \\ 2 & 0 & -2 & 0 \\ -1 & 2 & 3 & 0 \end{array}\right) = \left(\begin{array}{ccc|c} 1 & 2 & 1 & 0 \\ 0 & -4 & -4 & 0 \\ 0 & 4 & 4 & 0 \end{array}\right) = \left(\begin{array}{ccc|c} 1 & 2 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right).$$

Diese Matrix hat Rang 2, mit der Wahl der freien Variable $x_3 = \mu$ ergibt sich $x_2 = -\mu$ und $x_1 = -2x_2 - x_1 = \mu$. Somit ist der zum Eigenwert $\lambda = 0$ gehörige Eigenvektor durch $(1, -1, 1)$ gegeben.

Ähnliche Umformungen zeigen, dass der Eigenvektor zum Eigenwert 2 die Form $(1, 0, 1)$ hat. □

Die prinzipielle Vorgehensweise zum Bestimmen von Eigenwerten ist an obigem Beispiel ablesbar: Für gegebene Matrix A sind die Werte x zu bestimmen, für die $\det(A - xI_n) = 0$ gilt.

Definition 2.10 (Charakteristisches Polynom)
Für eine quadratische Matrix A nennt man das durch $\det(A - xI_n)$ gegebene Polynom das *charakteristische Polynom* von A.

Somit berechnet man die Eigenwerte eine Matrix A durch Nullsetzen des charakteristischen Polynoms von A. Da ja bekanntlich jedes Polynom über den komplexen Zahlen zumindest eine Nullstelle hat, hat auch jede quadratische Matrix einen (möglicherweise komplexen) Eigenwert.

Die Eigenvektoren einer Matrix geben somit die Richtungen an, in denen die Matrix (bei Multiplikation) die Richtung eines Vektors unverändert lässt. Um damit Ausdehnungsrichtungen von Punktwolken bestimmen zu können müssen diese Punktwolken aber zuerst als quadratische Matrix repräsentiert werden. Dabei ist der Begriff der *Kovarianzmatrix* wichtig, deren Einträge aus den Kovarianzen der einzelnen Datenkomponenten bestehen. Die *Kovarianz* zweier Zahlenfolgen ist wie folgt definiert.

Definition 2.11 (Kovarianz)
Seien $x = x_1, \ldots, x_n$ und $y = y_1, \ldots, y_n$ zwei reelle Zahlenfolgen. Dann bezeichnet man mit

$$\mathrm{Cov}(x, y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

die *Kovarianz* von x und y. Dabei geben $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ und $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$ die Mittelwerte von x bzw. y an.

Als *Varianz* $\mathrm{Var}(x) = \mathrm{Cov}(x, x)$ bezeichnet man das mittlere quadratische Abweichen einer Datenmenge von ihrem Mittelpunkt; und als *Standardabweichung* $\sigma = \sqrt{\mathrm{Var}(x)}$ die Wurzel der Varianz. Diese Kennzahlen geben ein Maß für die Streuung einer Datenmenge an.

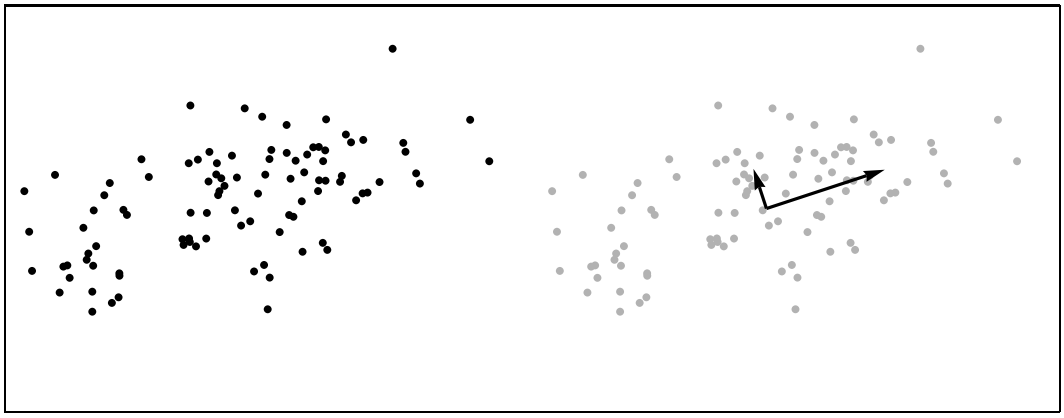

Abbildung 2.9: Die Datenmenge aus Beispiel 2.21. Links sind die Daten, rechts die Daten mit den Richtungen der größten Ausdehnung dargestellt. Die Länge der Pfeile entspricht den Standardabweichungen der Daten in diese Richtungen.

In obiger Definition der Kovarianz kann der Faktor $\frac{1}{n}$, je nach theoretischer Notwendigkeit, durch $\frac{1}{n-1}$ ersetzt werden; wir werden auf die Unterscheidung hier nicht eingehen, da sie für große n irrelevant ist.

Die Kovarianz von x und y gibt somit an, ob (über alle Werte gemittelt) größere Werte von x gleichzeitig mit größeren Werten von y bzw. kleinere Werte von x gleichzeitig mit kleineren Werten von y auftreten. Ist dies überwiegend der Fall, ist die Kovarianz positiv; wenn nicht, so ist sie negativ.

Definition 2.12 (Kovarianzmatrix)
Gegeben sei eine Folge $x = x_1, \ldots, x_n \in \mathbb{R}^m$ von m-dimensionalen Datenpunkten. Dann bezeichnet die $m \times m$-Matrix

$$\mathrm{CovMat}(x) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x}) \cdot (x_i - \bar{x})^T$$

die *Kovarianzmatrix* von x. Der Vektor $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ ist der Vektor der Mittelwerte der einzelnen Komponenten von $x_1, \ldots, x_n$.

Die Einträge der so definierten Kovarianzmatrix sind die Kovarianzen der Komponenten der Datenpunkte. Man beachte, dass die Kovarianzmatrix aufgrund der Konstruktion eine symmetrische Matrix ist.

Zur leichteren graphischen Darstellung verwenden wir im Folgenden zweidimensionale Datenpunkte; obige Definition ist aber auf beliebige Dimensionen anwendbar.

Beispiel 2.21 Gegeben seinen die 100 zweidimensionalen Datenpunkte x, die in Abbildung 2.9 (links) zu sehen sind. Die Kovarianzmatrix dieser Datenpunkte ist

$$\mathrm{CovMat}(x) = \begin{pmatrix} 3.65 & 1.09 \\ 1.09 & 0.96 \end{pmatrix}.$$

Aus dieser Matrix kann man erkennen, dass die Varianz von x in der ersten Dimension größer ist als die in der zweiten Dimension, und dass die Kovarianz der

beiden Dimensionen positiv ist. Die genauen Richtungen, in denen sich diese Punktmenge ausdehnt, sind aus der Kovarianzmatrix ohne weitere Berechnungen nicht abzulesen. □

Wenn wir uns für die Richtungen interessieren, entlang derer eine Datenmenge ihre größten Ausdehnungen hat, so sind diese Richtungen durch die Eigenvektoren der Kovarianzmatrix gegeben. Die Eigenwerte sind ein Maß für die Streuung der Daten in Richtung der Eigenvektoren. Durch die spezielle Struktur der Kovarianzmatrix ist garantiert, dass diese Eigenwerte nicht negativ sind.

Satz 2.9 Sei $x = x_1, \ldots, x_n$ eine Menge von Datenpunkten. Dann geben die Eigenvektoren der Matrix $\mathrm{CovMat}(x)$ mit den größten Eigenwerten die Richtungen an, entlang derer die Varianzen von x am größten sind. Die Varianzen entlang dieser Richtungen sind durch die Eigenwerte gegeben.

Die Begründung dieses Satzes ist uns leider mit dem jetzigen Wissenstand nicht möglich, wird aber in Abschnitt 7.9 nachgereicht.

Beispiel 2.22 (Fortsetzung von Beispiel 2.21) Die normalisierten Eigenvektoren der Kovarianzmatrix

$$\begin{pmatrix} 3.65 & 1.09 \\ 1.09 & 0.96 \end{pmatrix}$$

sind $(0.942, 0.335)$ und $(-0.335, 0.942)$ mit den Eigenwerten 4.04 bzw. 0.569. Die mit den Wurzeln der Eigenwerte skalierten Eigenvektoren sind graphisch rechts in Abbildung 2.9 zu sehen. □

Die Technik der Hauptkomponentenanalyse wird hauptsächlich zur Dimensionsreduktion eingesetzt. Dabei werden hochdimensionale Datenmengen in niedrigere Dimensionen transferiert, indem nur die Koordinaten bezüglich der größten Eigenwerte beibehalten werden. Zu beachten ist dabei, dass für die Koordinatenberechnungen die Originaldaten zuerst in den Ursprung verschoben werden müssen.

Satz 2.10 Sei $x = x_1, \ldots, x_n$ eine Menge von Datenpunkten mit Mittelwert 0 und E die Matrix, deren Spalten aus den Eigenvektoren der k größten Eigenwerte besteht. Dann werden durch die Koordinatentransformation

$$x_i' = E^T \cdot x_i$$

die Datenpunkte x_i auf einen k-dimensionalen Raum reduziert, dessen Koordinatenachsen den Hauptrichtungen der Daten entsprechen.

Beispiel 2.23 Die Kovarianzmatrix einer fünfdimensionalen Datenmenge mit 400 Punkten sei

$$C = \begin{pmatrix} 10.2 & 2.07 & 2.19 & 5.21 & 15.53 \\ 2.07 & 10.37 & 3.08 & 1.96 & 3.53 \\ 2.19 & 3.08 & 6.33 & 4.21 & 3.41 \\ 5.21 & 1.96 & 4.21 & 5.17 & 10.4 \\ 15.54 & 3.53 & 3.41 & 10.4 & 108.74 \end{pmatrix}$$

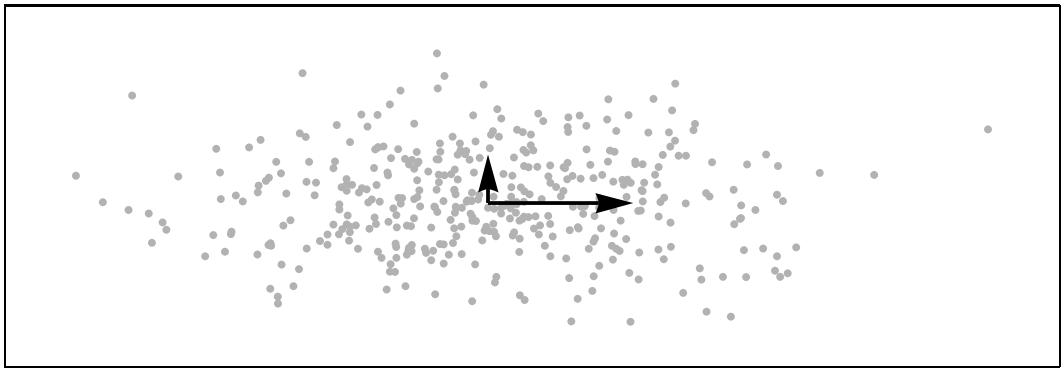

Abbildung 2.10: Die zwei Hauptachsenkoordinaten der 400 Datenpunkte aus Beispiel 2.23. Die Länge der eingezeichneten Achsen entspricht der Standardabweichung der Daten in diese Richtungen.

Die (nach Größe sortierten) Eigenwerte von C sind 112.6, 14.7, 8.0, 4.9 und 0.5. Durch Projektion auf die den beiden größten Eigenwerten entsprechenden Eigenvektoren kann eine Reduktion auf zwei Dimensionen erreicht werden. Mit dem Anordnen der normierten Eigenvektoren in den Zeilen einer Matrix kann die Koordinatentransformation durchgeführt werden. In diesem Beispiel ergeben sich die zweidimensionalen Koordinaten (x'_1, x'_2) durch

$$\begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} = \begin{pmatrix} -0.156 & -0.04 & -0.04 & -0.105 & -0.98 \\ -0.443 & -0.614 & -0.484 & -0.409 & 0.159 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}.$$

Die Hauptrichtungen der transformierten Punktmenge liegen dann achsparallel; die Kovarianzmatrix der reduzierten Daten ist (wie zu erwarten war)

$$C' = \begin{pmatrix} 112.6 & 0 \\ 0 & 14.7 \end{pmatrix}.$$

Man erkennt an den 0-Einträgen, dass die Streuung der reduzierten Daten nur mehr achsparallel ist. Die Standardabweichung in die beiden Richtungen sind $\sqrt{112.6} = 10.6$ und $\sqrt{14.7} = 3.8$. Die dimensionsreduzierten Daten aus diesem Beispiel sind in Abbildung 2.10 zu sehen. □

2.5 Lineare Diskriminanzanalyse

Wenn hochdimensionale Daten auf weniger Dimensionen reduziert werden müssen, dann ist die Hauptkomponentenanalyse aus dem letzten Abschnitt ein guter Ansatz. In vielen Anwendungen gilt es aber, hochdimensionale Daten zu *klassifizieren*, also zu entscheiden, zu welcher Klasse (aus einer vorgegebenen Menge) ein Datenpunkt gehört. Dabei wird vorausgesetzt, dass bei einem Teil der Daten die Klassenzugehörigkeit bekannt ist. Auf Basis dieser Daten soll dann ein Entscheidungskriterium

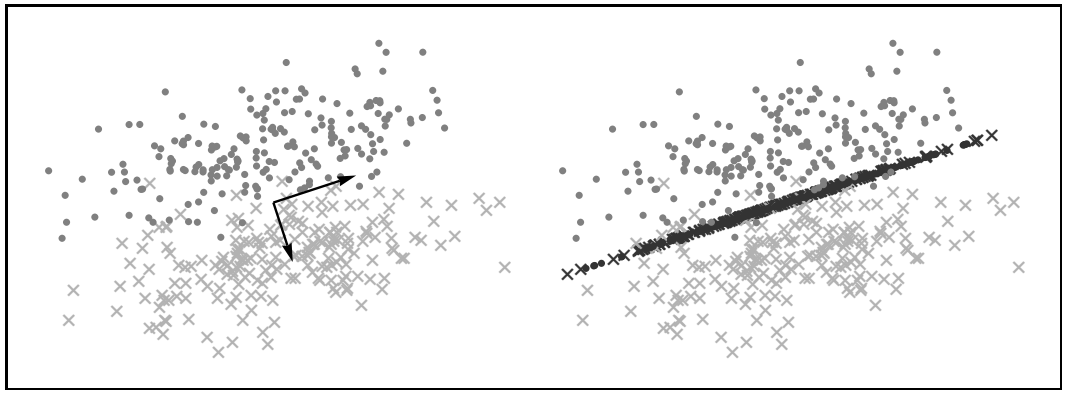

Abbildung 2.11: Die Datenmenge aus Beispiel 2.24. Links sind die zwei Klassen und die Hauptkomponenten der gesamten Daten zu sehen; rechts die Projektionen der Daten auf die erste Hauptkomponente.

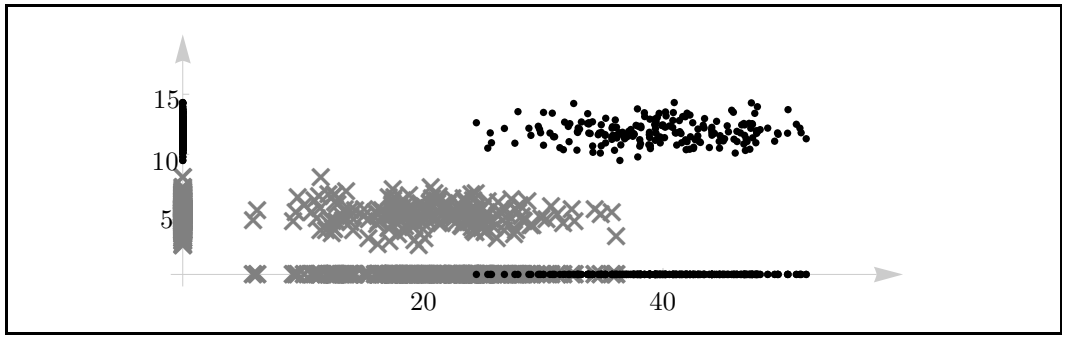

Abbildung 2.12: Zwei achsparallele Klassen zu jeweils 200 Datenpunkten, und ihre Projektionen auf die beiden Koordinatenachsen.

entwickelt werden, das für die Klassifizierung der anderen Datenpunkte eingesetzt werden kann. Im Folgenden werden wir uns auf den Fall zweier Klassen beschränken.

Man kann anhand eines einfachen Beispiels sehen, dass die Hauptkomponentenanalyse auf Daten mit Klasseninformation ungeeignete Resultate liefern kann. Wir werden anschließend einen anderen Ansatz wählen, der in diesem Fall bessere Ergebnisse liefert.

Beispiel 2.24 Gegeben seien die zwei Klassen von zweidimensionalen Daten, die in Abbildung 2.11 zusammen mit ihren Hauptkomponenten zu sehen sind. Diese Daten bestehen aus zweimal 200 Datenpunkten, die (bis auf wenige Ausnahmen) durch eine gerade Trennlinie korrekt der einen bzw. anderen Klasse zugeordnet werden können.

Wenn wir diese Daten zur Dimensionsreduktion nun auf nur eine (die erste) Hauptkomponente projizieren wollen, so geht dabei die leichte Separierbarkeit verloren: Die Koordinaten der beiden Klassen bezüglich dieser Richtung überlappen sich stark. □

Eine Verbesserung dieser Situation kann dadurch erreicht werden, dass nicht auf die erste Hauptkomponente, sondern auf eine andere Richtung projiziert wird. Eine erste Idee ist es diejenige Richtung zu wählen, die die Distanz der Mittelpunkte der

beiden Klassen maximiert. Die Graphik in Abbildung 2.12 illustriert, warum dieser Ansatz noch verbesserungswürdig ist: Obwohl die Separation der Mittelpunkte in x-Richtung größer ist als in y-Richtung, liefert eine Projektion auf die y-Achse eine eindeutige Trennung der Klassen, während sich bei Projektion auf die x-Achse die Klassen überlappen. Somit ist eine Projektion auf die Richtung der größten Distanz zwischen den Datenmittelpunkten nicht optimal.

Aus Abbildung 2.12 kann man ablesen, dass die Separation der Mittelpunkte erst *relativ zur Streuung der Klassen* relevant ist. Da die Streuung in x-Richtung um einiges größer ist als in y-Richtung, gibt es in x-Richtung eine Überlappung der Klassen, obwohl die Distanz der Klassenmittelpunkte in dieser Richtung größer ist als in y-Richtung. Diese Einsicht führt dazu diejenige Richtung auszuwählen, die das Verhältnis der Streuung *zwischen* Klassen zur Streuung *innerhalb* der Klassen maximiert.

Zur mathematischen Herleitung dieser Richtung benötigen wir folgende Notationen. Wie wir bereits aus Satz 2.4 in Abschnitt 2.2 wissen, ist die Koordinate λ eines Vektors v bezüglich eines Einheitsvektors w durch das Skalarprodukt

$$\lambda = w^T \cdot v$$

gegeben. Die Mittelpunkte m_1 und m_2 von zwei Klassen von Datenpunkten $c = c_1, \ldots, c_{n_1}$ und $d = d_1, \ldots, d_{n_2}$ sind gegeben durch

$$m_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} c_i \qquad \text{und} \qquad m_2 = \frac{1}{n_2} \sum_{i=1}^{n_2} d_i,$$

und damit die Distanz der auf w projizierten Mittelpunkte durch

$$w^T \cdot m_1 - w^T \cdot m_2$$

Die quadratischen Abstände von den Mittelpunkten der auf w projizierten Datenpunkte c und d lassen sich schreiben als

$$q_1 = \sum_{i=1}^{n_1} (w^T \cdot c_i - w^T \cdot m_1)^2 \qquad \text{bzw.} \qquad q_2 = \sum_{i=1}^{n_2} (w^T \cdot d_i - w^T \cdot m_2)^2.$$

Da die Varianz quadratische Abstände angibt, muss beim zu maximierenden Verhältnis von Mittelpunktvarianz zu Interklassenvarianz der Abstand der Mittelpunkte noch quadriert werden. Gesucht wird somit diejenige Richtung w, für die das Verhältnis

$$R(w) = \frac{\left(w^T \cdot m_1 - w^T \cdot m_2\right)^2}{q_1 + q_2}$$

maximal wird. Dieser Ausdruck lässt sich in Matrixnotation schreiben als

$$R(w) = \frac{w^T \cdot C_{\mathrm{Z}} \cdot w}{w^T \cdot C_{\mathrm{I}} \cdot w},$$

mit der *Zwischenklassen-Kovarianzmatrix*

$$C_{\mathrm{Z}} = (m_1 - m_2) \cdot (m_1 - m_2)^T$$

und der Summe der *Intraklassen-Kovarianzmatrizen*

$$C_{\mathrm{I}} = \sum_{i=1}^{n_1} (c_i - m_1) \cdot (c_i - m_1)^T + \sum_{i=1}^{n_2} (d_i - m_2) \cdot (d_i - m_2)^T.$$

Der Ausdruck $R(w)$ wird für denjenigen Wert von w maximal, an dem die Ableitung null wird. Man kann in dieser Hinsicht mit Matrizen fast wie mit "normalen" Variablen rechnen; der einzige Unterschied ergibt sich aus der Tatsache, dass Matrizenmultiplikation nicht kommutativ ist, und es deswegen auf die Reihenfolge der Operanden ankommt. Mit der Differentiationsregel

$$\left(\frac{f}{g}\right)' = \frac{f'g - g'f}{g^2}$$

erhält man

$$R'(w) = \frac{2(w^T \cdot C_{\mathrm{I}} \cdot w)\, C_{\mathrm{Z}} \cdot w - 2(w^T \cdot C_{\mathrm{Z}} \cdot w)\, C_{\mathrm{I}} \cdot w}{(w^T \cdot C_{\mathrm{I}} \cdot w)^2}.$$

Die Gleichung $R'(x) = 0$ liefert dann

$$(w^T \cdot C_{\mathrm{I}} \cdot w)\, C_{\mathrm{Z}} \cdot w = (w^T \cdot C_{\mathrm{Z}} \cdot w)\, C_{\mathrm{I}} \cdot w.$$

An dieser Stelle ist zu beachten, dass wir nur an der *Richtung* von w interessiert sind, nicht aber an seiner Länge, da er zur Koordinatenberechnung auf Länge 1 normiert wird. In obiger Gleichung sind die Ausdrücke $(w^T \cdot C_{\mathrm{I}} \cdot w)$ und $(w^T \cdot C_{\mathrm{Z}} \cdot w)$ reelle Zahlen und können somit weggelassen werden. Es bleibt die Gleichung

$$C_{\mathrm{Z}} \cdot w = C_{\mathrm{I}} \cdot w.$$

Mit der Beobachtung, dass

$$C_{\mathrm{Z}} \cdot w = (m_1 - m_2) \cdot (m_1 - m_2)^T \cdot w = (m_1 - m_2)\, \alpha$$

ist, und damit in Richtung von $m_1 - m_2$ zeigt, erhält man schließlich nach Weglassen des skalaren Faktors α die Lösung

$$w = C_{\mathrm{I}}^{-1}(m_1 - m_2).$$

Damit ist die Richtung vorgegeben, entlang derer projiziert werden muss, um eine optimale Separation der beiden Datenklassen zu erhalten. Wir fassen zusammen.

Satz 2.11 Für zwei Datenklassen c und d ist durch

$$w = C_{\mathrm{I}}^{-1}(m_1 - m_2)$$

diejenige Richtung gegeben, entlang derer die Projektionen von c und d geringste Überlappung aufweisen. Die Terme C_{I}, m_1 und m_2 sind wie oben definiert. Die Richtung w wird auch *Fisher's lineare Diskriminante* genannt.

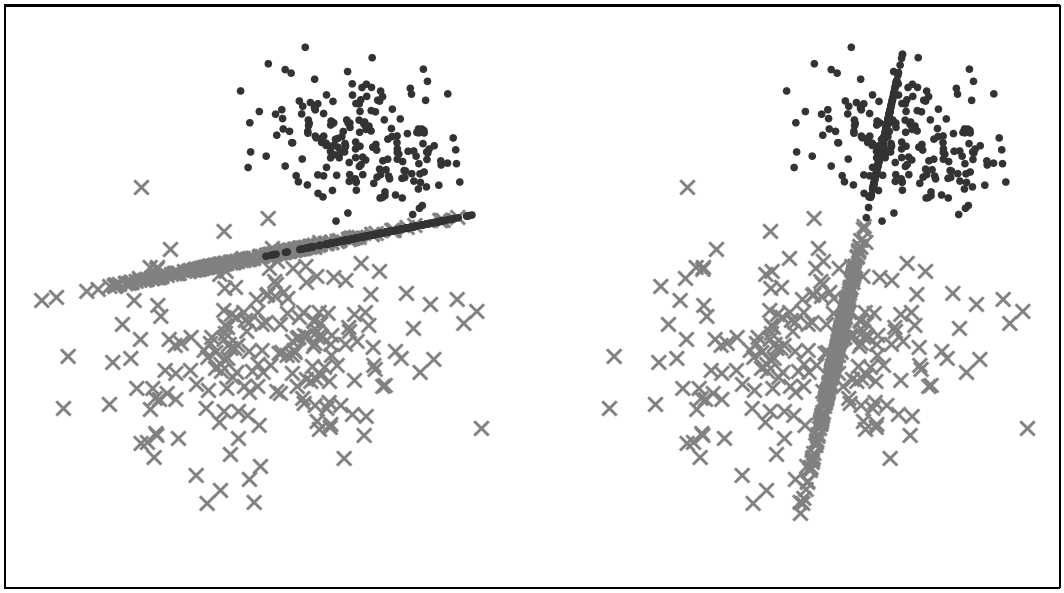

Abbildung 2.13: Die Datenpunkte aus Beispiel 2.25, links mit Projektionen auf einen beliebigen Vektor, recht mit Projektionen auf Fisher's lineare Diskrimante.

Beispiel 2.25 Wir betrachten zwei Klassen von zweidimensionalen Datenpunkten. In Abbildung 2.13 links sind die Datenpunkte dieser Klassen mit den Projektionen auf einen beliebigen Vektor zu sehen. Man kann erkennen, dass die Trennung der Klassen auf der Projektionslinie schlecht ist. In Abbildung 2.13 rechts sind die gleichen Datenpunkte mit den Projektionen auf dem Vektor w aus Satz 2.11 zu sehen. Es ist deutlich zu erkennen, dass sich die projizierten Datenpunkte kaum mehr überlappen. □

Kapitel 3

Interpolation

In technischen Bereichen benötigt man des öfteren Verfahren, um durch eine gegebene Menge von Punkten eine Funktion zu legen. Dabei unterscheidet man zwischen *Approximation* und *Interpolation*. In beiden Fällen wählt man eine Funktionenmenge, um die Daten zu repräsentieren.

Bei der Approximation ist diese Funktionenmenge bewusst restriktiv gewählt (etwa lineare Funktionen oder quadratische Polynome), um eine zugrundeliegende Struktur besser identifizieren zu können. Man nimmt also an, dass die Daten eigentlich dieser Struktur entsprechen, und nur aufgrund von Messfehlern oder vereinfachenden Annahmen nicht genau auf der Approximationsfunktion liegen.

Bei der Interpolation von Datenpunkten trifft man diese vereinfachende Annahme nicht. Stattdessen will man eine Funktion finden, auf deren Graph dann auch alle Datenpunkte liegen. Ohne strukturvereinfachende Annahmen gibt es eine Vielzahl von Funktionen, die für die Interpolation von Datenpunkten verwendet werden.

Wie wir sehen werden, spielen Polynomfunktionen bei der Interpolation eine wichtige Rolle. Dies ist auf folgendes theoretische Resultat zurückzuführen.

Satz 3.1 (Approximationssatz von Weierstraß) Sei $f : \mathbb{R} \to \mathbb{R}$ auf dem Intervall $[a, b]$ stetig. Dann gibt es für jedes $\epsilon > 0$ eine Polynomfunktion $P(x)$ auf dem Intervall $[a, b]$ mit der Eigenschaft

$$\bigvee_{x \in [a,b]} \left| f(x) - P(x) \right| < \epsilon \,.$$

Damit kann theoretisch bewiesen werden, dass jede stetige Funktion auf einem begrenzten Bereich beliebig genau von einer Polynomfunktion approximiert werden kann. Da man nicht überprüfen kann, dass die Polynomfunktion an unendlich vielen Stellen mit der zu approximierenden Funktion übereinstimmt, wählt man eine Menge von *Stützstellen* auf der Funktion und verlangt, dass die Polynomfunktion durch diese Stützstellen verläuft.

Eine kurze Überlegung macht klar, dass es nur *eine* Polynomfunktion vom Grad n durch $(n + 1)$ Datenpunkte geben kann—dies folgt unmittelbar aus dem Fundamentalsatz der Algebra, dass jede Polynomfunktion vom Grad n höchstens n Nullstellen haben kann. Somit sind die beiden Interpolationspolynome, die in den

Abschnitten 3.1 und 3.2 behandelt werden, identisch. Ihr Unterschied liegt nur in der Berechnungsvorschrift. Neben diesen beiden Methoden ist es natürlich auch möglich, die Koeffizienten des Interpolationspolynoms durch Lösen eines linearen Gleichungssystems zu bestimmen. Auf diese naheliegende Variante wird hier nicht näher eingegangen.

3.1 Lagrange Polynome

Zur Motivation der Lagrange Polynome betrachten wir folgendes einfache Beispiel: Eine Gerade sei durch die beiden Punkte (x_0, y_0) und (x_1, y_1) zu legen. Es ist leicht zu überprüfen, dass die lineare Polynomfunktion

$$P(x) = \frac{(x - x_1)}{(x_0 - x_1)} y_0 + \frac{(x - x_0)}{(x_1 - x_0)} y_1$$

diese Bedingung erfüllt und somit die eindeutige Lösung darstellt. Die Einsicht bei der Konstruktion dieses Polynoms ist es so aufzuteilen, dass für jeden der beiden Punkte nur ein Summand etwas beiträgt, während der andere Null wird. Es ist nicht schwer, diese Argumentation auf drei Punkte $(x_0, y_0), (x_1, y_1)$ und (x_2, y_2) auszudehnen: Mit

$$\begin{aligned}
L_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \qquad \text{ist } L_0(x_0) = 1, L_0(x_1) = L_0(x_2) = 0 \\
L_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \qquad \text{ist } L_1(x_1) = 1, L_1(x_0) = L_1(x_2) = 0 \\
L_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \qquad \text{ist } L_2(x_2) = 1, L_2(x_0) = L_2(x_1) = 0
\end{aligned}$$

und man erhält die Interpolationsfunktion als

$$P(x) = L_0(x)y_0 + L_1(x)y_1 + L_2(x)y_2.$$

Allgemein definiert man für $n+1$ Datenpunkte $(x_0, y_0), \ldots, (x_n, y_n)$ die n Funktionen

$$L_i(x) = \frac{(x - x_0) \ldots (x - x_{i-1})(x - x_{i+1}) \ldots (x - x_n)}{(x_i - x_0) \ldots (x_i - x_{i-1})(x_i - x_{i+1}) \ldots (x_i - x_n)}$$

für die wiederum gilt

$$L_i(x_j) = \begin{cases} 1 & \text{wenn } i = j \\ 0 & \text{sonst.} \end{cases}$$

Dann ist die eindeutig bestimmte Polynomfunktion, die diese Punkte interpoliert, gegeben durch

$$P(x) = L_0(x)y_0 + L_1(x)y_1 \cdots + L_n(x)y_n.$$

Beispiel 3.1 Wir illustrieren die Form der zur Konstruktion von P verwendeten Funktionen L_i anhand eines einfachen Beispiels. Seien dafür die fünf x-Werte $0, 0.25, 0.5, 0.75$ und 1 gegeben. Die fünf Polynomfunktionen $L_0, \ldots, L_4$, die jeweils an nur einer der Stützstellen der Wert 1 und an den anderen den Wert 0 annehmen sind in Abbildung 3.1 zu sehen. □

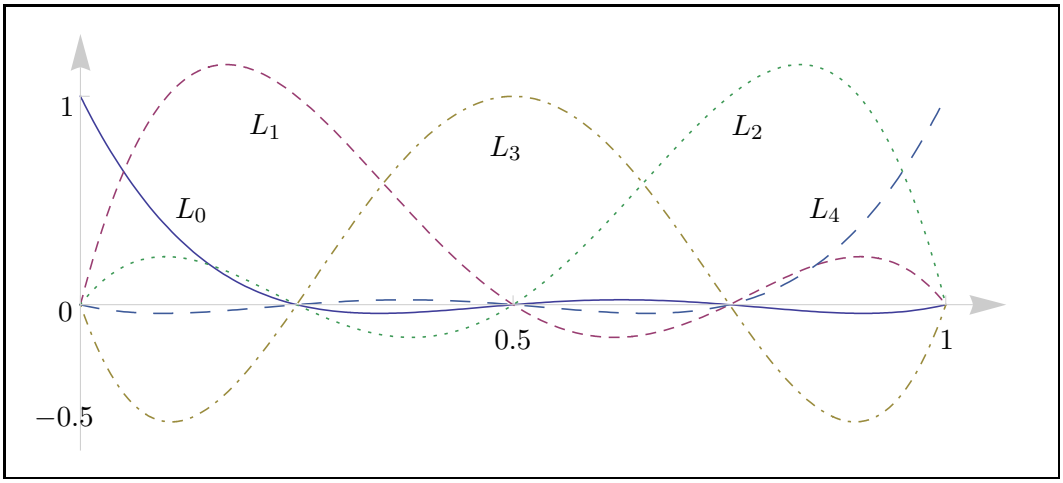

Abbildung 3.1: Die fünf Interpolationspolynome $L_0, \ldots, L_4$ aus Beispiel 3.1.

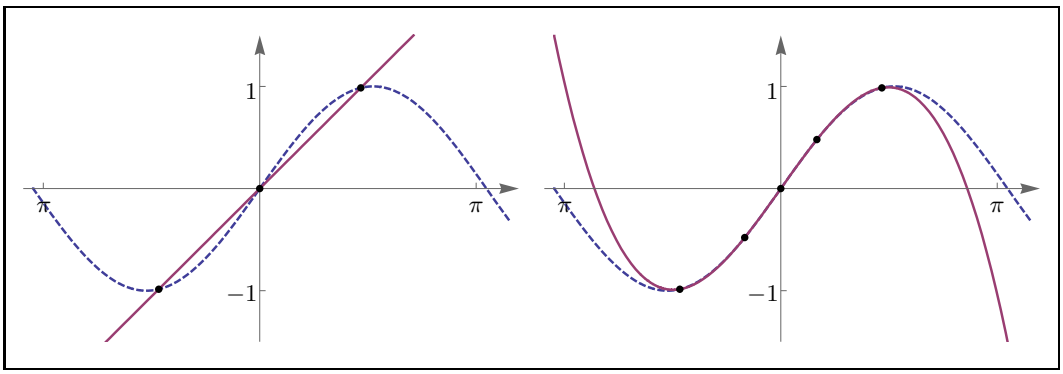

Abbildung 3.2: Approximation des Sinus (gestrichelt dargestellt) durch Interpolation von drei (links) bzw. fünf (rechts) Stützstellen aus Beispiel 3.2.

Je mehr Stützstellen für die Interpolation verwendet werden, desto genauer approximiert die Interpolationsfunktion die zugrundeliegenden Daten. Im folgenden Beispiel kann man sehen, wie die unterschiedliche Anzahl von Stützstellen das Aussehen der Interpolationsfunktion beeinflusst.

Beispiel 3.2 Die Funktion $\sin(x)$ im Intervall $[-\frac{\pi}{2}, \frac{\pi}{2}]$ approximiert werden. Wir wählen zuerst die drei Stützpunkte $-1.4, 0, 1.4$, die symmetrisch um den Nullpunkt sind, und erhalten so die Interpolationspunkte $(-1.4, -0.98545), (0, 0), (1.4, 0.98545)$. Damit ergibt sich die Lagrangepolynomfunktion

$$P(x) = -0.98545L_0(x) + 0.98545L_2(x) = 0.703893x,$$

bei dem sich aufgrund der Symmetrie der Datenpunkte die quadratische Terme in $L_1(x)$ und $L_3(x)$ aufheben. Damit ist die Interpolation nicht befriedigend, wie man in Abbildung 3.2 links sehen kann. Wenn man weitere Stützstellen bei -0.5 und 0.5 (und damit Datenpunkte $(-0.5, -0.479426)$, $(0.5, 0.479426)$) wählt, erhält man als

interpolierende Funktion

$$\begin{aligned} P(x) &= -0.98545L_0(x) - 0.479426L_1(x) + 0.479426L_3(x) + 0.98545L_4(x) \\ &= -0.149098x^3 + 0.996126x. \end{aligned}$$

Wieder fallen durch die Symmetrie einige Terme weg; dennoch ist die Approximation an die Sinusfunktion, wie in Abbildung 3.2 rechts zu sehen, um einiges besser. □

Man sieht anhand dieses Beispiels, dass durch die spezielle symmetrische Wahl der Stützstellen einige Ausdruckskraft verloren wurde, da sich einige Potenzen wegheben. Wenn man also eine Funktion durch Interpolation an Stützstellen approximieren will, wird man bei symmetrischen Funktionen nicht auch symmetrische Stützstellen wählen. Zudem eignen sich Lagrangepolynome nicht zur inkrementellen Interpolation, da man keine der Zwischenergebnisse bei einer Vergrößerung der Stützstellenanzahl weiterverwenden kann. Durch genauere Betrachtung der Struktur der Lagrangepolynome kann man allerdings einen Algorithmus angeben, bei dem Interpolationswerte an einer bestimmenten Stelle inkrementell entwickelt werden. Wir werden darauf aber nicht näher eingehen.

Durch den Ansatz von Newton kann man auch das Problem lösen, ganze Polynome durch sukzessives Hinzufügen von Stützstellen zu entwickeln.

3.2 Newton Polynome

Wir nehmen wiederum an, dass die Datenpunkte $(x_0, y_0), \ldots, (x_n, y_n)$ durch ein Polynom interpoliert werden sollen. Das eindeutige Polynom P, das durch diese $n+1$ Datenpunkte verläuft, ist vom Grad n. Wir nehmen jetzt an, dass sich die zugehörige Polynomfunktion folgendermaßen schreiben lässt:

$$\begin{aligned} P(x) = a_0 &+ a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots \\ &+ a_n(x - x_0)\ldots(x - x_{n-1}) \end{aligned} \tag{3.1}$$

Wir können nun inkrementell folgende Gleichungen lösen, um die Koeffizienten a_0 bis a_n zu bestimmen. Damit sehen wir auch, dass das Polynom mit obiger Struktur tatsächlich die $n+1$ Datenpunkte interpoliert. Durch die spezielle Struktur des Polynoms ist

$$\begin{aligned} P(x_0) &= a_0 + a_1(x_0 - x_0) + \cdots + a_n(x_0 - x_0)\ldots(x_0 - x_{n-1}) \\ &= a_0. \end{aligned}$$

Da P die Datenpunkte interpolieren soll, muss natürlich $P(x_0) = y_0$ gelten, somit erhält man für den ersten Koeffizienten $a_0 = y_0$. Durch Einsetzen der weiteren Datenpunkte fallen immer die höchsten Summanden weg und man erhält

$$P(x_1) = y_1 = a_0 + a_1(x_1 - x_0)$$

also mit $a_0 = y_0$

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0}.$$

Wenn man so weitermacht, ergeben sich für $a_2, \ldots, a_n$ recht komplizierte Ausdrücke. Durch eine rekursive Definition ergibt sich eine äquivalente aber elegante Formulierung. Wir definieren als Weiterentwicklung von obiger Herleitung

$$f[x_i] = f(x_i) \qquad \text{und} \quad f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}.$$

Höhere Terme werden analog dazu rekursiv konstruiert:

$$f[x_i, \ldots, x_{i+k}] = \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}.$$

Die Koeffizienten a_i des Interpolationspolynoms sind dann

$$a_i = f[x_0, \ldots, x_i].$$

Somit kann man Gleichung (3.1) auch schreiben als

$$\begin{aligned} P(x) = {} & f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \ldots \\ & + f[x_0, \ldots, x_n](x - x_0) \ldots (x - x_{n-1}) \end{aligned} \tag{3.2}$$

Man könnte durch einigermaßen kompliziertes Nachrechnen überprüfen, dass das durch diese Koeffizienten bestimmte Polynom tatsächlich die geforderten Eigenschaften hat. Wir benützen stattdessen diese Form des Interpolationspolynoms, um in Analogie zum Taylorpolynom eine Fehlerabschätzung anzugeben.

Satz 3.2 Seien $x_0, \ldots, x_n$ $n+1$ verschiedene Zahlen im Intervall $[a, b]$, $f \in C^{n+1}[a, b]$ und $P(x)$ das in (3.2) definierte Interpolationspolynom. Dann gibt es für jedes $x \in [a, b]$ ein $\xi \in (a, b)$ mit

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1) \ldots (x - x_n).$$

Der Fehler bei der Polynominterpolation lässt sich also ähnlich wie das Restglied bei der Taylorreihenentwicklung anschreiben. Dort ist allerdings die gesamte Information im Punkt x_0 konzentriert, während sie hier auf die Stützstellen $x_0, \ldots, x_n$ verteilt ist.

Wir betrachten nun ein Beispiel zum Bestimmen des Interpolationspolynoms mit der Methode von Newton.

Beispiel 3.3 Gegeben seien die Punkte

$$(0.2, -1.60944), (1.2, 0.182322), (2.2, 0.788457), (3.2, 1.16315)$$

zur Approximation des natürlichen Logarithmus im Bereich $[0.2, 3.2]$. Zur Bestimmung der Koeffizienten in (3.2) benötigt man folgende Terme:

$f[x_0]$

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$$

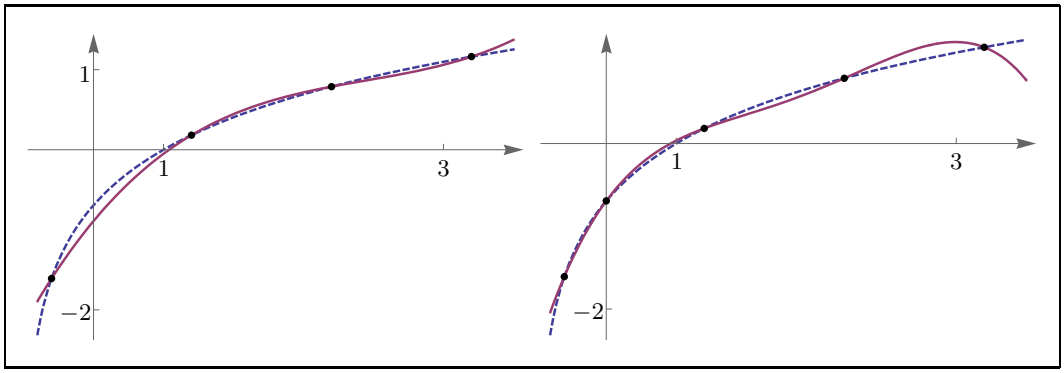

Abbildung 3.3: Approximation des Logarithmus (gestrichelt dargestellt) duch Interpolation von vier (links) bzw. fünf (rechts) Stützstellen aus Beispiel 3.3 bzw. Beispiel 3.4.

$$
\begin{aligned}
&f[x_1] && && f[x_0,x_1,x_2] = \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}\\
& && f[x_1,x_2] = \frac{f[x_2]-f[x_1]}{x_2-x_1} &&\\
&f[x_2] && && f[x_1,x_2,x_3] = \frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}\\
& && f[x_2,x_3] = \frac{f[x_3]-f[x_2]}{x_3-x_2} &&\\
&f[x_3] && &&
\end{aligned}
$$

und abschließend noch

$$f[x_0,x_1,x_2,x_3] = \frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0}$$

Für unser Datenmaterial ergibt sich die Tabelle

$$
\begin{aligned}
&f[x_0] = -1.609 && && \\
& && f[x_0,x_1] = 1.792 && \\
&f[x_1] = 0.182 && && f[x_0,x_1,x_2] = -0.593 && \\
& && f[x_1,x_2] = 0.606 && && f[x_0,x_1,x_2,x_3] = 0.159\\
&f[x_2] = 0.788 && && f[x_1,x_2,x_3] = -0.116 && \\
& && f[x_2,x_3] = 0.375 && \\
&f[x_3] = 1.163 && &&
\end{aligned}
$$

und damit die Polynomkoeffizienten $a_0 = -1.609, a_1 = 1.792$, $a_2 = -0.593$, $a_3 = 0.159$. Die Interpolationsfunktion ist somit

$$
\begin{aligned}
P(x) = \; & -1.609 + 1.792(x-0.2) - 0.593(x-0.2)(x-1.2)\\
& + 0.159(x-0.2)(x-1.2)(x-2.2)\\
= \; & 0.159x^3 - 1.1654x^2 + 3.15x - 2.19367
\end{aligned}
\tag{3.3}
$$

Diese Interpolationsfunktion ist in Abbildung 3.3 links dargestellt. Eine Fehlerabschätzung ist für diese vier Datenpunkte gegeben durch

$$f(x) - P(x) = \frac{1}{4\xi^4}(x - 0.2)(x - 1.2)(x - 2.2)(x - 3.2)$$

In diesem Intervall ist $\frac{1}{4\xi^4}$ bei $\xi = 0.2$ am größten. Eine Abschätzung für den Punkt $x = 0.5$ ist somit

$$\begin{aligned}\big|f(x) - P(x)\big| &< \left|\frac{1}{4 \times 0.2^4}(0.5 - 0.2)(0.5 - 1.2)(0.5 - 2.2)(0.5 - 3.2)\right| \\ &< 150.609\end{aligned}$$

Die Abschätzung ist in diesem Fall sehr schlecht, da der tatsächliche Fehler nur 0.197499 beträgt. □

Wie schon erwähnt, erhält man mit Lagrangepolynomen das gleiche Interpolationspolynom wie mit der Methode von Newton. Letztere hat aber den Vorteil, dass mit relativ geringem Aufwand noch weitere Stützstellen eingefügt werden können.

Beispiel 3.4 (Fortsetzung von Beispiel 3.3) Die durch (3.3) definierte Approximation an den Logarithmus im Bereich $[0.2, 1]$ sei für bestimmte Anwendungen nicht genau genug. Man wählt daher noch eine Stützstelle $(0.5, -0.693147)$, um die Approximation in diesem Bereich zu verbessern. Die obige Tabelle zum Berechnen der Polynomkoeffizienten muss also noch um folgende Einträge ergänzt werden:

$$\begin{aligned} f[x_4] &= -0.693147 & f[x_3, x_4] &= 0.687518 \\ f[x_2, x_3, x_4] &= -0.184014 & f[x_1, x_2, x_3, x_4] &= 0.0975616 \\ f[x_0, x_1, x_2, x_3, x_4] &= -0.204895 \end{aligned}$$

Zum Interpolationspolynom kommt nur der Term $-0.204895(x - 0.2)(x - 1.2)(x - 2.2)(x - 3.2)$ hinzu; somit ergibt sich

$$P(x) = -0.205x^4 + 1.552x^3 - 4.206x^2 + 5.435x - 2.540.$$

Diese Interpolationsfunktion ist in Abbildung 3.3 rechts dargestellt. □

Bei den bisher betrachteten Beispielen haben wir nur wenige Stützstellen verwendet, sodass das Interpolationspolynom geringen Grad hatte. Höhergradige Polynome oszillieren aber viel stärker als Polynome niedrigen Grades; damit sind sie zur Interpolation einer großen Datenmenge nicht geeignet.

Beispiel 3.5 Ein Interpolationspolynom soll durch die 21 Datenpunkte gelegt werden, die in unterer Tabelle angegeben sind:

x	0.9	1.3	1.9	2.1	2.6	3.0	3.9	4.4	4.7	5.0	6.0
y	1.3	1.5	1.85	2.1	2.6	2.7	2.4	2.15	2.05	2.1	2.25
x	7.0	8.0	9.2	10.5	11.3	11.6	12.0	12.6	13.0	13.3	
y	2.3	2.25	1.95	1.4	0.9	0.7	0.6	0.5	0.4	0.25	

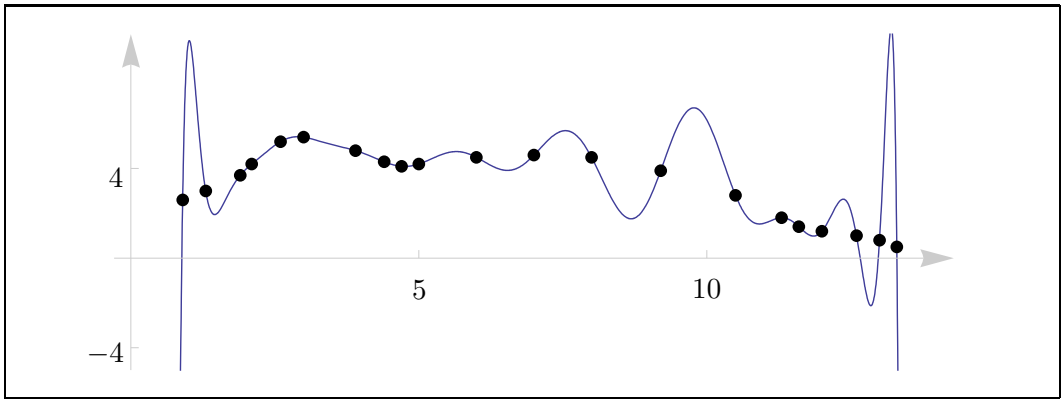

Abbildung 3.4: Die Interpolationsfunktion aus Beispiel 3.5.

Das Ergebnis der Interpolation ist das Polynom 20. Grades, das in Abbildung 3.4 zu sehen ist. Man kann deutlich erkennen, dass Polynomfunktionen aus folgendem Grund für Interpolationen größerer Datenmengen ungeeignet sind: Für jeden Datenpunkt benötigt ein Modell der Daten (in diesem Fall ist das Modell die Menge aller Polynomfunktionen) einen Freiheitsgrad. Diese Freiheitsgrade (freie Parameter) geben dem Modell genügend Flexibilität, um die Datenpunkte zu interpolieren. Das Modell der Polynomfunktionen hat aber den Nachteil, dass jeder zusätzliche Freiheitsgrad automatisch eine Erhöhung des Polynomgrads mit sich bringt. Flexibilität ist in diesem Fall untrennbar mit starker Oszillation verbunden, wie man gut in Abbildung 3.4 sehen kann. □

Im nächsten Abschnitt werden wir ein Modell kennenlernen, bei dem die Freiheitsgrade in anderer Weise als bei Polynomen eingehen, und die damit für Aufgaben obiger Art besser geeignet sind.

3.3 Kubische Splines

Um das Problem der Oszillation, das in Beispiel 3.5 aufgetreten ist, in den Griff zu bekommen, kann man die Interpolationsaufgabe in mehrere Teilaufgaben zerlegen, die dann wieder wenigere Datenpunkte umfassen und damit auch weniger stark oszillieren.

Der einfachste Ansatz dazu ist die *lineare Interpolation*, wobei durch jeweils benachbarte Datenpunkte eine Gerade gelegt wird. Obwohl eine stückweise lineare Funktion in den meisten Anwendungsfällen einem stark oszillierenden Polynom vorzuziehen ist, so hat dieser Ansatz doch einen gravierenden Nachteil: Stückweise lineare Funktionen sind an den Übergangsstellen nicht differenzierbar; gerade diese Bedingung muss aber in vielen Bereichen erfüllt sein.

Damit muss man auf quadratische Polynome ausweichen, mit denen man diesen Nachteil vermeiden kann. Wenn man durch jeweils zwei benachbarte Punkte (x_i, y_i) und (x_{i+1}, y_{i+1}) ein quadratisches Polynom legen will, so hat man für $n+1$ Datenpunkte $3n$ freie Parameter in den n Interpolationspolynomen. Von diesen Parametern werden $2n$ durch die Bedingungen gebunden, dass jedes Polynom durch seinen Anfangs- und Endpunkt verlaufen muss. Für Differenzierbarkeit an den $n-1$ "inneren" Punkten hat man damit noch n Freiheitsgrade zur Verfügung. Die Schwierigkeit

ergibt sich allerdings daraus, dass man meist auch Ableitungen an den "äußeren" Endpunkten x_0 und x_n angeben will, und dafür nur mehr einen freien Parameter hat. Um alle diese Bedingungen im Modell ausdrücken zu können benötigt man damit kubische Polynome.

Kubische Polynome haben pro Intervall 4 Freiheitsgrade, von denen man wiederum 2 zur Festlegung der Start- und Endpunkte benötigt. An den $n-1$ "inneren Punkten" kann man sowohl erste als auch zweite Ableitung festlegen, und hat dann immer noch $4n - 2n - 2(n-1) = 2$ Freiheitsgrade, um Bedingungen am Start- und Endpunkt festzulegen. Formell definiert man diese Objekte wie folgt.

Definition 3.1 (Kubische Splines)
Für eine gegebene Datenmenge $(x_0, y_0), \ldots, (x_n, y_n)$ ist eine *kubische Spline* S ein Interpolationspolynom, das folgende Bedingungen erfüllt:

(1) S ist auf jedem Teilintervall $[x_j, x_{j+1}]$ $(j = 0, \ldots, n-1)$ ein kubisches Polynom S_j,

(2) $S(x_j) = y_j$ für $j = 0, \ldots, n$,

(3) $S_j(x_{j+1}) = S_{j+1}(x_{j+1})$ für $j = 0, \ldots, n-2$,

(4) $S'_j(x_{j+1}) = S'_{j+1}(x_{j+1})$ für $j = 0, \ldots, n-2$,

(5) $S''_j(x_{j+1}) = S''_{j+1}(x_{j+1})$ für $j = 0, \ldots, n-2$,

(6) eine der folgenden Randbedingungen ist erfüllt:

   (a) $S''(x_0) = S''(x_n) = 0$

   (b) $S'(x_0) = s_1$ und $S'(x_n) = s_2$

Wenn die Splinefunktion die Randbedingung $S''(x_0) = S''(x_n) = 0$ erfüllt, so spricht man von einer *natürlichen Spline*.

Die Berechnung von kubischen Splines erfolgt durch das Lösen des Gleichungssystems, das sich aus den Bedingungen der obigen Definition ergibt. Durch die spezielle Form der Bedingungen ergeben sich noch einige Vereinfachungen, auf die wir hier nicht näher eingehen werden. Mit diesen Vereinfachungen lässt sich dann auch zeigen, dass das Gleichungssystem zur Bestimmung der Koeffizienten immer genau eine Lösung hat.

Satz 3.3 Gegeben seien $n+1$ Stützstellen $(x_0, y_0), \ldots, (x_n, y_n)$. Dann gibt es für die beiden Bedingungen

(1) $S''(x_0) = S''(x_n) = 0$

(2) $S'(x_0) = s_1$ und $S'(x_n) = s_2$

jeweils eine kubische Spline, die diese Stützstellen interpoliert.

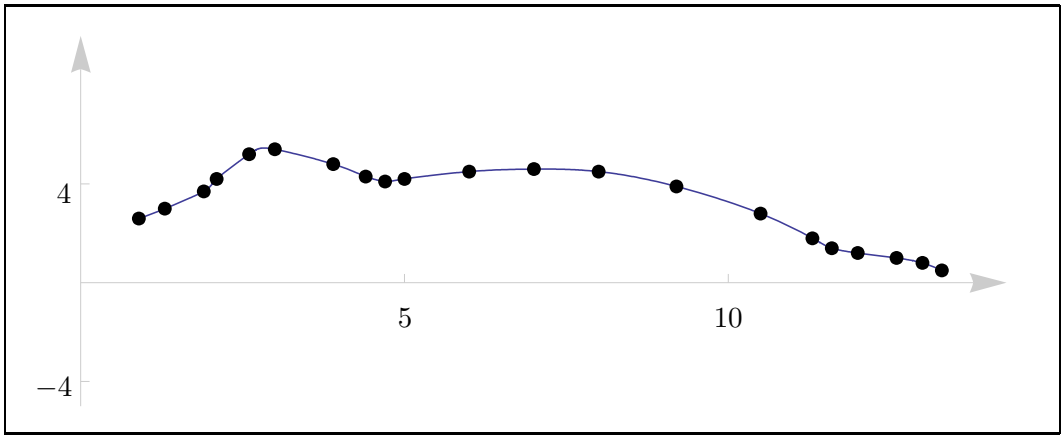

Abbildung 3.5: Natürliche kubische Spline durch die 21 Datenpunkte aus Beispiel 3.5.

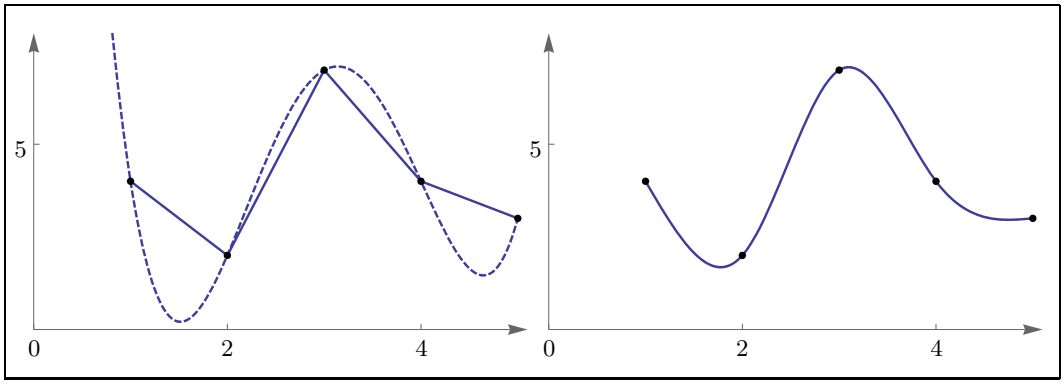

Abbildung 3.6: Links lineare (durchgezogen) und polynomiale (gestrichelt) Interpolation der Datenpunkte aus Beispiel 3.7. Die natürliche kubische Spline rechts hat die Nachteile dieser beiden Interpolationsformen nicht.

Wir betrachten dazu einige Beispiele.

Beispiel 3.6 Eine natürliche kubische Spline durch die Datenpunkte aus Beispiel 3.5 ist in Abbildung 3.5 zu sehen. Die Approximation an die Datenpunkte ist hier viel besser als durch das Polynom in Abbildung 3.4. Auf den ersten Blick ist dieses Ergebnis allerdings kaum von linearer Interpolation zu unterscheiden, da die Datenpunkte nahe aneinander liegen. □

Den Unterschied zwischen linearer und kubischer Interpolation kann man durch geeignetere Wahl von Stützpunkten besser illustrieren.

Beispiel 3.7 Wir konstruieren eine natürliche kubische Spline durch die Datenpunkte

$$(1, 4), (2, 2), (3, 7), (4, 4), (5, 3),$$

und vergleichen dies mit linearer Interpolation und einem Interpolationspolynom 4. Grades. Das Resultat ist in Abbildung 3.6 zu sehen. Man kann erkennen, dass

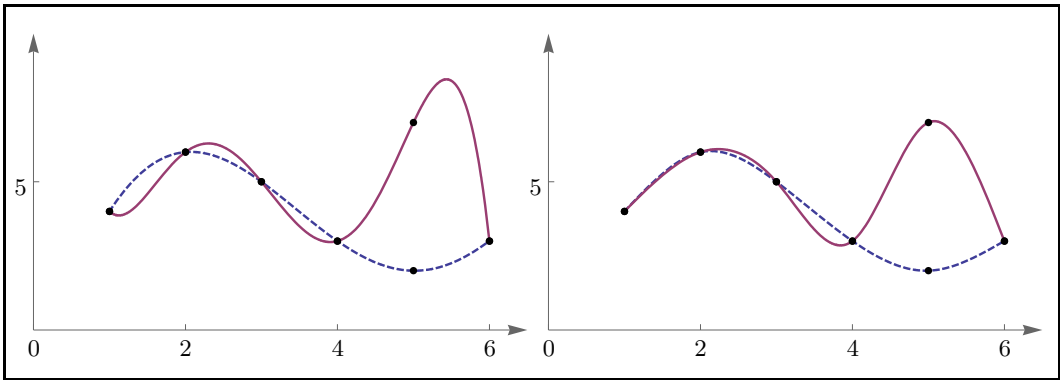

Abbildung 3.7: Interpolation der Datenpunkte aus Beispiel 3.8. Links das Interpolationspolynom 5. Grades vor (gestrichelt) und nach (durchgezogen) dem Verschieben des 5. Datenpunkts, rechts die kubische Spline bei gleicher Veränderung des Datenpunkts.

die kubische Spline den Nachteil der linearen Interpolation wettmacht, nicht differenzierbar zu sein. Im Gegensatz zum Interpolationspolynom oszilliert die kubische Spline auch nicht. □

Ein weiterer Vorteil von kubischen Splines gegenüber globaler polynomialer Interpolation ist, dass lokale Änderungen an den Daten auch nur lokale Änderungen an der Interpolationsfunktion bedingen; bei Interpolationspolynomen ist dies nicht der Fall. Folgendes Beispiel dient dazu als Illustration.

Beispiel 3.8 Wir legen sowohl eine natürliche kubische Spline als auch ein Interpolationspolynom 5. Grades durch die Datenpunkte

$$(1,4),(2,6),(3,5),(4,3),(5,2),(6,3).$$

Wenn wir dann den fünften Datenpunkt auf $(5,7)$ ändern, so verändern sich natürlich auch die Interpolationskurven. In Abbildung 3.7 links ist zu sehen, wie sich dadurch das Interpolationspolynom 5. Grades auch abseits des Punktes $(5,7)$ verändert. Bei der kubischen Spline rechts ist dies nicht in diesem Maß der Fall. □

3.4 Bezierkurven

Als Abschluss dieses Kapitels über Interpolation wenden wir uns nun einem Teilbereich zu, der nicht mehr direkt mit Interpolation zu tun hat. Da dieses Thema aber in der Computergraphik von Bedeutung ist, werden wir es in diesem Kontext behandeln.

Bis jetzt wurden graphische Linienverläufe als Graphen von Funktionen repräsentiert. Es wurde stillschweigend akzeptiert, dass man durch die Vereinfachungen, die sich durch den Umgang mit Funktionen ergeben, auch einen Nachteil in Ausdruckskraft hinnehmen muss: Durch die Verwendung von Funktionen ist man auf einen y-Wert je x-Wert limitiert, sodass solche geometrische Objekte wie etwa Kreise nicht als Graphen einer Funktion dargestellt werden können.

Durch eine einfache Erweiterung kann man allgemeine geometrische Objekte durch Funktionen repräsentieren. Die wesentliche Änderung dabei ist, sowohl die x- als auch die y-Koordinate als Funktion eines weiteren Parameters t aufzufassen. Damit sind $x(t)$ und $y(t)$ unabhängige Funktionen, die für jeden Wert von t ein Wertepaar $(x(t), y(t))$ liefern. Die Menge dieser Wertepaare kann wiederum graphisch dargestellt werden. Wir legen folgende Definition fest.

Definition 3.2 (Parametrische Kurven)
Sei $[a, b] \subseteq \mathbb{R}$ ein Intervall und $x, y : [a, b] \to \mathbb{R}$ zwei reelle Funktionen. Dann bezeichnet man die vektorwertige Funktion $[a, b] \to \mathbb{R}^2$ mit

$$t \mapsto \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

als *parametrische Kurve*.

Obige Definition kann auch auf dreidimensionale Kurven erweitert werden, wenn man noch eine dritte Funktion $z(t)$ zulässt. Wir werden uns hier aber auf den zweidimensionalen Fall beschränken.

Beispiel 3.9 Der Einheitskreis lässt sich mit $x(t) = \cos(t)$, $y(t) = \sin(t)$ für $t \in [0, 2\pi]$ als parametrische Kurve darstellen. Eine Ellipse erhält man etwa, indem man die x-Koordinate um den Faktor a, die y-Koordinate um den Faktor b streckt: $t \mapsto (a\cos(t), b\sin(t))$. □

Man kann parametrische Kurven auch zur Interpolation von Datenpunkten verwenden, wenn diese Datenpunkte so "übereinander" liegen, dass man keine Funktion durchlegen kann.

Beispiel 3.10 Gegeben seien die sechs Datenpunkte

$$(1, 2), (3, 4), (4, 3), (3, 2), (2, 2), (2, 1).$$

Durch diese Datenpunkte lässt sich keine Funktion legen, da es für manche x-Werte mehr als einen y-Wert gibt.

Man kann aber die x- und y-Werte *getrennt* interpolieren, und diese Interpolationspolynome dann als Komponenten einer parametrischen Kurve durch die Datenpunkte auffassen. Dabei hat man für das Auswählen der ersten (t-) Komponenten freie Wahl und kann diese auch für x- und y-Koordinaten getrennt wählen. Eine Möglichkeit ist die äquidistante Wahl $t = (0, 0.2, 0.4, 0.6, 0.8, 1)$, andere wären z.B. $t = (0, 0.1, 0.3, 0.5, 0.7, 1)$ oder $t = (0, 0.15, 0.35, 0.7, 0.9, 1)$. Für jede Wahl der t-Punkte erhält man sowohl für x als auch für y ein Interpolationspolynom 5. Grades. Die drei parametrischen Kurven, die man durch diese Wahl der Stützstellenplazierung von $(t, x(t))$ und $(t, y(t))$ erhält, sind in Abbildung 3.8 dargestellt. Man kann erkennen, dass unterschiedliche Stützpunkte, und seien es nur im Parameter t, sichtbare Auswirkungen auf die Interpolationskurve haben. □

Wir wollen im Folgenden parametrische Kurven behandeln, die nicht zur Interpolation von Datenpunkten verwendet werden. Vielmehr legen die Datenpunkte dabei

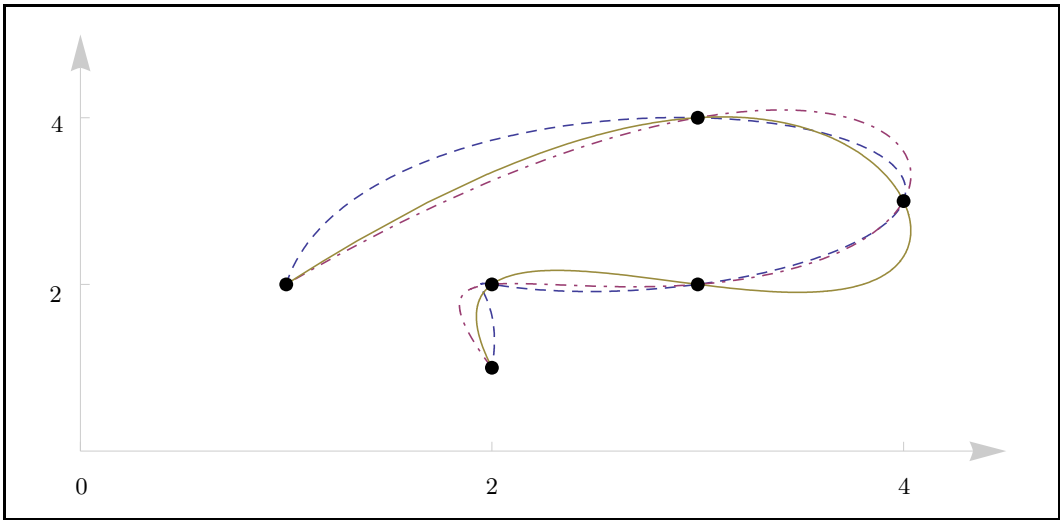

Abbildung 3.8: Parametrische Kurven durch die Datenpunkte aus Beispiel 3.10. Die Unterschiede ergeben sich durch verschiedene Wahl der Parameter t an den Stützstellen.

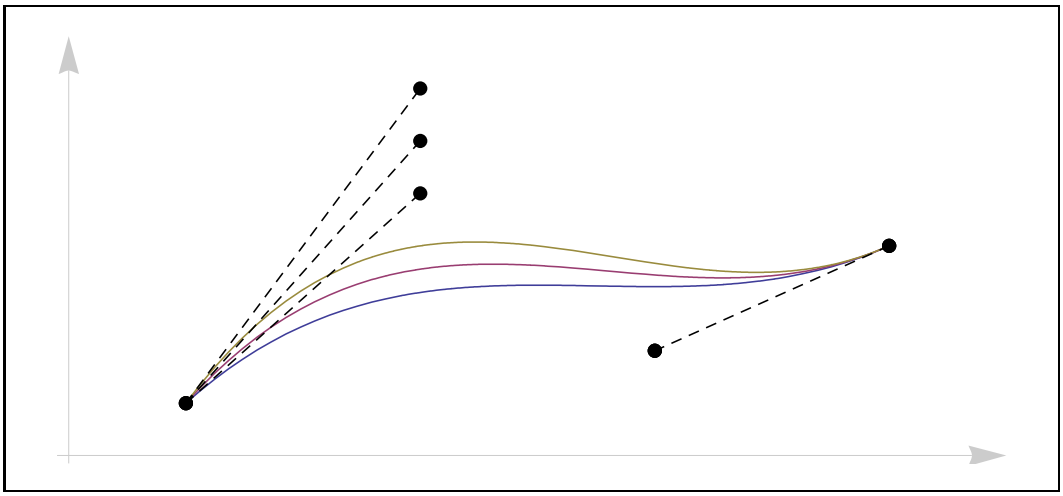

Abbildung 3.9: Kubische Bezier-Kurven mit zwei Stützstellen und zwei Kontrollpunkten. Gestrichelt dargestellt sind die Verbindungen zwischen Stützstellen und Kontrollpunkten.

die Form der Kurve fest; die Kurve verläuft nur durch den ersten und letzten Datenpunkt. Diese Art von Kurven ist in der Computergraphik und im CAD-Bereich wichtig, da der Kurvenverlauf rein von der Lage der Punkte abhängt und nicht noch Bedingungen an erste oder zweite Ableitungen in den Randpunkten zu stellen sind.

Wir betrachten zunächst den einfachsten Fall von vier Datenpunkten $P_i = (x_i, y_i), i = 0, \ldots, 3$, bei denen die äußeren zwei Stützstellen und die inneren zwei Kontrollpunkte der Kurve sind. Wir definieren folgendermaßen.

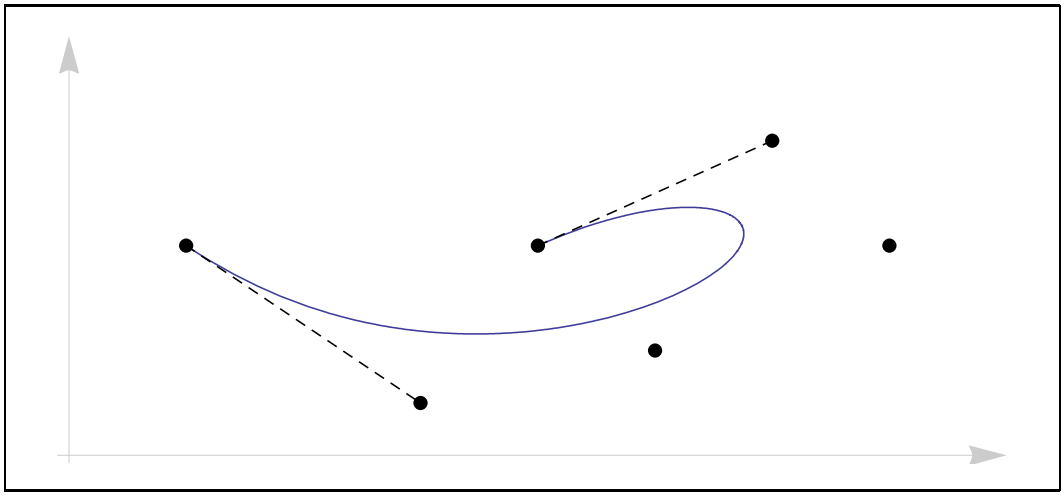

Abbildung 3.10: Durch sechs Datenpunkte definierte Bezier-Kurve 5. Grades. Gestrichelt dargestellt sind die Verbindungen zwischen Stützstellen und benachbarten Kontrollpunkten.

Definition 3.3 (Kubische Bezier-Kurven)
Gegeben seinen vier Punkte $P_0, \ldots, P_3 \in \mathbb{R}^2$. Dann nennt man die durch

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3.$$

für $t \in [0,1]$ definierte Funktion eine *kubische Bezier-Kurve*. Die dabei auftretenden Faktoren $(1-t)^3$, $3t(1-t)^2$, $3t^2(1-t)$ und t^3 werden *Bernstein-Polynome 3. Grades* genannt.

Einige kubische Bezier-Kurven für gleichbleibende Stützstellen und einen verschobenen Kontrollpunkt sind in Abbildung 3.9 dargestellt. Man kann daraus einige Eigenschaften von kubischen Bezier-Kurven ablesen, die man auch mathematisch überprüfen kann:

- Die Kurve verläuft durch die beiden Stützstellen.
- Die Tangentialvektoren an die Kurven in den Stützstellen zeigen in Richtung der Kontrollpunkte.
- Je weiter die Kontrollpunkte von den Stützstellen entfernt sind, desto länger schmiegt sich die Kurve an die Tangenten in den Stützstellen an.
- Die Kurve verläuft gänzlich innerhalb der von den vier Punkten aufgespannten *konvexen Hülle* (dem von den Punkten aufgespannten Polygon).

Die gleichen Eigenschaften gelten auch für höhergradige Bezier-Kurven, die sich aus obiger Definition leicht verallgemeinern lassen. Für $n+1$ Datenpunke $P_0, \ldots, P_n$

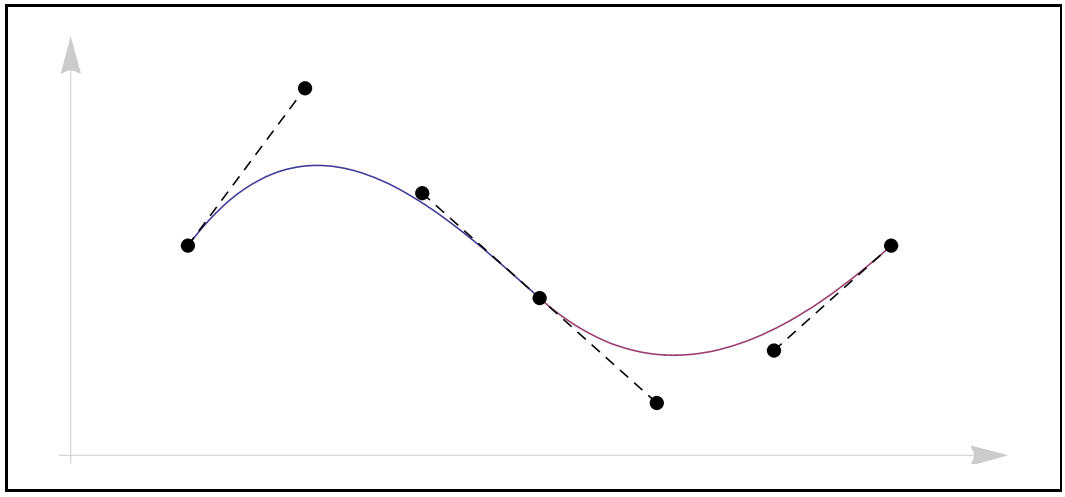

Abbildung 3.11: Stückweise kubische Bezier-Kurven. Gestrichelt dargestellt sind die Verbindungen zwischen Stützstellen und Kontrollpunkten.

und $t \in [0, 1]$ ist die Bezier-Kurve n-ten Grades definiert durch

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} P_i,$$

also wiederum eine durch Bernstein-Polynome höheren Grades gewichtete Summe der Datenpunkte. Ein Beispiel einer Bezier-Kurve 5. Grades ist in Abbildung 3.10 zu sehen.

Obwohl es aus dieser Abbildungen nicht ersichtlich ist, ergibt sich bei Bezier-Kurven höheren Grades wiederum ein Nachteil, der schon bei Interpolationspolynomen aufgetreten ist: die zusätzlichen Freiheitsgrade werden durch zusätzliche Polynomgrade realisiert, was wiederum zu unerwünschten Effekten führen kann. Es werden also auch bei Bezier-Kurven stückweise kubische Lösungen bevorzugt. Diese Kurven verlaufen dann durch jeden vierten Datenpunkt; die Tangenten in diesen Stützstellen werden durch die Lage der restlichen Datenpunkte festgelegt.

Bei stückweise kubischen Bezier-Kurven möchte man "glatte" Übergänge zwischen den Kurventeilen: die Ableitungen an den Stützstellen müssen also für beide Kurvenstücke S_i und S_{i+1}, die an diesem Punkt zusammentreffen übereinstimmen. Die bedeutet, dass der erste Kontrollpunkt in S_{i+1} auf der gleichen Gerade wie der letzte Kontrollpunkt in S_i und der dazwischenliegende Stützpunkt liegen muss. Eine stückweise kubische Bezier-Kurve durch sieben Datenpunkte ist in Abbildung 3.11 dargestellt.

Kapitel 4

Numerische Differentiation und Integration

In diesem Kapitel werden Verfahren präsentiert und angewandt, die Grenzübergänge $h \to 0$ durch "sehr kleine" Werte h ersetzen. Wir werden sehen, wie Restglieder von Taylorreihenentwickungen für Fehlerabschätzungen verwendet werden, und wie die Genauigkeit von Methoden in Abhängigkeit von Funktionsableitungen und Potenzen von h ausgedrückt werden kann.

Wir behandeln numerische Differentiation in Abschnitt 4.1 und numerische Integration in Abschnitt 4.2. In Abschnitt 4.3 wird ein Verfahren präsentiert, mit dem der Grenzübergang $h \to 0$ simuliert und damit noch größere Genauigkeit erreicht werden kann.

4.1 Numerisches Differenzieren

In diesem Abschnitt werden wir Methoden kennenlernen, mit Hilfe derer man die Ableitung

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

einer Funktion f an der Stelle x numerisch berechnen kann. Diese Methoden und die damit verbundenen Fehlerabschätzungen basieren auf der Taylorreihenentwicklung von f um x. Diese liefert die unendliche Reihe

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(x)}{6}h^3 + \ldots, \tag{4.1}$$

woraus man eine Legitimation für das Weglassen des Grenzwerts in der Definition der Ableitung erkennen kann:

$$\begin{aligned} f'(x) &= \frac{f(x+h) - f(x)}{h} - \frac{f''(x)}{2}h - \frac{f'''(x)}{6}h^2 - \ldots \\ &= \frac{f(x+h) - f(x)}{h} - \frac{f''(\xi)}{2}h \\ &\approx \frac{f(x+h) - f(x)}{h}. \end{aligned} \tag{4.2}$$

Hier ersetzen wir in der zweiten Zeile die unendliche Reihe durch das Restglied der Reihenentwicklung; ξ ist also ein Wert zwischen x und $x+h$. Man beachte hier, dass

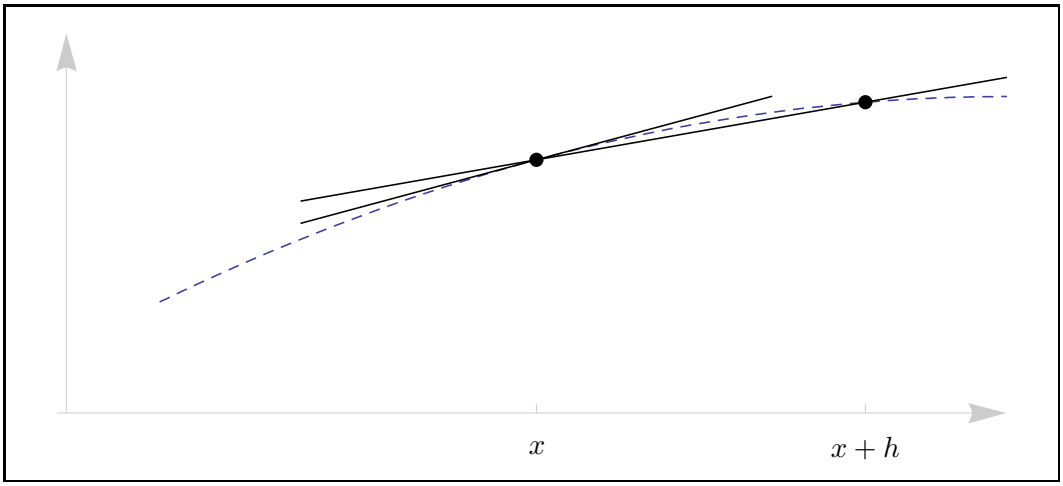

Abbildung 4.1: Approximation der Tangente an f in der Stelle x durch die Sekante in x und $x + h$.

der Fehler in dieser Approximation linear mit h wächst. Wir werden später sehen, wie bei besseren Approximationen der Fehler mit Potenzen von h wächst—dies ist für $|h| < 1$ natürlich vorzuziehen.

Die Approximation an die Ableitung durch diese einfache Formel ist graphisch in Abbildung 4.1 zu sehen.

Beispiel 4.1 Wir betrachten die Funktion $f(x) = \log(x)$ und den Punkt $x = 2$. Die Ableitung ist somit annähernd

$$f'(2) = \frac{f(2+h) - f(2)}{h},$$

die Fehlerabschätzung ist wegen $\log''(x) = -1/x^2$ für $h > 0$

$$\frac{|f''(\xi)|}{2} h = \frac{h}{2\xi^2} \leq \frac{h}{2 \times 2^2},$$

da ja $\xi \in [2, 2 + h]$ sein muss. Die echte Ableitung, die Approximation sowie die Differenz und die Fehlerabschätzung sind für einige Werte von h unten angegeben.

h	$f'(2)$	$\frac{f(2+h)-f(2)}{h}$	$f'(2)$ - $\frac{f(2+h)-f(2)}{h}$	$\frac{h}{2\times 2^2}$
0.1	0.5	0.487902	0.0120984	0.0125
0.01	0.5	0.498754	0.00124585	0.00125
0.001	0.5	0.499875	0.000124958	0.000125

Die hier angegebene Schranke ist somit sehr gut. □

Weitere (auch genauere) Differentiationsformeln erhält man durch Variationen der obigen Herleitung. Ersetzt man etwa in der Taylorreihenentwicklung (4.1) h durch $-h$, so erhält man die Approximation

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2}h^2 - \frac{f'''(x)}{6}h^3 + \dots. \tag{4.3}$$

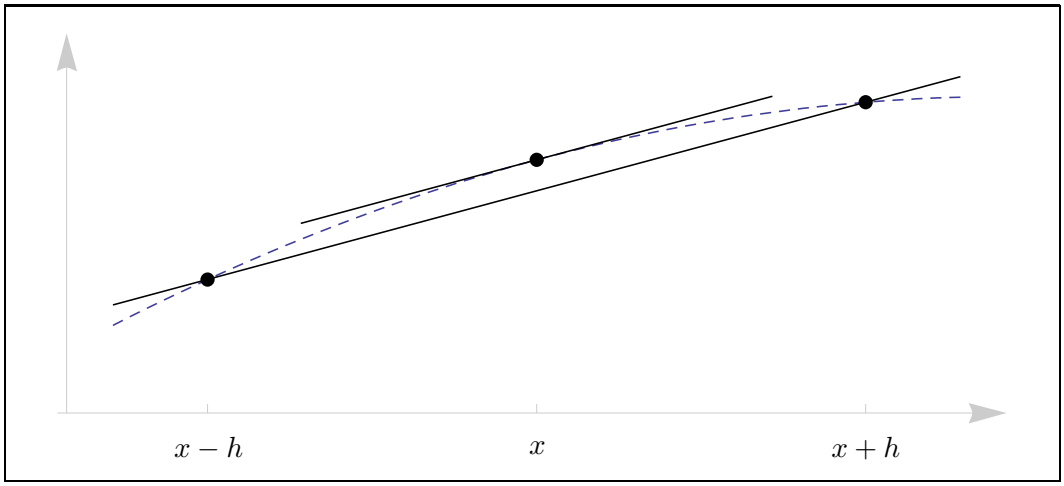

Abbildung 4.2: Approximation der Tangente an f in der Stelle x durch die Sekante in $x-h$ und $x+h$. Der Fehler in dieser Approximation wächst (im Gegensatz zu (4.2) und Abbildung 4.1) nur mit h^2.

Auflösen nach $f'(x)$ liefert hier das zu (4.2) analoge Ergebnis

$$\begin{aligned} f'(x) &= \frac{f(x)-f(x-h)}{h} + \frac{f''(x)}{2}h - \frac{f'''(x)}{6}h^2 - \dots \\ &\approx \frac{f(x)-f(x-h)}{h}. \end{aligned} \tag{4.4}$$

Wenn man die Taylorentwicklung (4.3) von (4.1) abzieht und dann nach $f'(x)$ auflöst, erhält man eine genauere Formel:

$$\begin{aligned} f'(x) &= \frac{f(x+h)-f(x-h)}{2h} - \frac{f'''(x)}{6}h^2 - \dots \\ &\approx \frac{f(x+h)-f(x-h)}{2h} \end{aligned} \tag{4.5}$$

Somit wächst der Fehler bei dieser Approximation nur mehr mit h^2. Diese Verbesserung ist graphisch in Abbildung 4.2 dargestellt. Approximationen wie in (4.5) werden *Dreipunkt-Formeln* genannt, obwohl f nicht explizit am Punkt x ausgewertet wird.

Wenn man noch genauere Approximationen benötigt, kann man die Punkte $x \pm 2h$, $x \pm 3h, \dots$ in die Berechnung einfließen lassen. Damit ergeben sich dann Formeln, deren Fehler mit höheren Potenzen von h wachsen, und die damit noch genauer sind. Wir werden hier aber nicht näher darauf eingehen, sondern auf die Verbesserung von Ergebnissen durch Extrapolation (siehe Abschnitt 4.3) verweisen.

Beispiel 4.2 Wir berechnen die Ableitung der Funktion $f(x) = xe^x$ an der Stelle $x = 1$ für verschiedene Werte von h mit den Formeln (4.2) und (4.5), und vergleichen den dabei gemachten Fehler mit der theoretischen Fehlerschranke. Mit (4.2) gilt

$$f'(x) = \frac{(x+h)e^{x+h} - xe^x}{h},$$

mit (4.5)

$$f'(x) = \frac{(x+h)e^{x+h} - (x-h)e^{x-h}}{2h}.$$

Der genaue Wert der Ableitung (auf 10 Hinterkommastellen) ist $f'(1) = 2e = 5.4365636569$. Die Fehlerabschätzungen für diese Approximationen sind

$$\frac{f''(\xi)}{2}h = \frac{1}{2}h\big(2e^{\xi} + \xi e^{\xi}\big)$$

bzw.

$$\frac{f'''(\xi)}{6}h^2 = \frac{1}{6}h^2\big(3e^{\xi} + \xi e^{\xi}\big).$$

Dabei liegt ξ in der ersten Schranke zwischen x und $x+h$, bei der zweiten Schranke zwischen $x-h$ und $x+h$. In beiden Fällen werden die Schranken für $\xi = x+h$ maximiert.

Wir erhalten somit für konkrete Werte von h folgende Approximationen, Fehlerschranken und tatsächliche Fehler.

h	$f'(x) = \frac{f(x+h)-f(x)}{h}$	Fehler	Schranke
0.1	5.8630079788	0.4264443219	0.4656457337
0.01	5.4775196708	0.0409560139	0.0413212953
0.001	5.4406428924	0.0040792355	0.0040828627
h	$f'(x) = \frac{f(x+h)-f(x-h)}{2h}$	Fehler	Schranke
0.1	5.4546991315	0.0181354750	0.0205284678
0.01	5.4367448771	0.0001812201	0.0001834977
0.001	5.4365654691	0.0000018122	0.0000018145

Man erkennt im ersten Teil dieser Tabellen deutlich das lineare Abnehmen des Fehlers in Abhängigkeit von h, beim zweiten Teil das quadratische Abnehmen. Außerdem sieht man, dass in allen Fällen die Schranken sehr enge Abschätzungen liefern.□

4.2 Numerisches Integrieren

Der Einsatzbereich numerischer Differentiation, die im letzten Abschnitt behandelt wurde, beschränkt sich auf Funktionen, deren symbolische Repräsentation nicht oder nur ungenau angegeben werden kann. Wenn eine Funktion durch einen Term ausgedrückt werden kann, so kann man diesen Term differenzieren, um die Ableitung zu berechnen.

Bei der Umkehrung der Differentiation, der Integration, ist dies nicht der Fall. Zunächst aber eine Erinnerung an den Zusammenhang zwischen Integral und Ableitung.

Satz 4.1 (Fundamentalsatz der Analysis) Sei $F \in C^1[a,b]$. Dann gilt

$$\int_a^b F'(x)dx = F(b) - F(a).$$

Zu einer Funktion F' nennt man F eine *Stammfunktion* von F'. Im Sinne obiger Formel spricht man also von der Integration als Umkehrung der Differentiation. Der Wert des Integrals entspricht dem Flächeninhalt unter der Kurve $y = F'(x)$ zwischen den Grenzen a und b.

Im Gegensatz zur Differentiation ist es bei der Integration aber nicht immer möglich, für gegebene Funktion eine Stammfunktion zu finden. So ist es etwa nicht möglich, das Integral $\int_a^b e^{-x^2}dx$ symbolisch auszuwerten. Auch wenn von einer Funktion nur bestimmte Werte bekannt sind, kann man den Flächeninhalt unter dieser Kurve nicht symbolisch berechnen. In diesen Fällen muss man auf numerische Methoden zurückgreifen, um das Integral evaluieren zu können.

Die meisten dieser numerischen Methoden beruhen darauf, zuerst die zu integrierende Funktion an Stützstellen zu interpolieren, und dann diese Interpolationsfunktion zu integrieren. Dies funktioniert natürlich nur, wenn sich die Interpolationsfunktionen leicht integrieren lassen; dies ist bei den uns bekannten Interpolationsfunktionen (Polynome, Splines) der Fall. Da wir also nur Stützstellen und ihre Funktionswerte zur Integration verwenden, ist der allgemeinste Ansatz durch die Formel

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} w_i f(x_i)$$

gegeben. Dabei nennt man die Parameter w_i *Gewichte*, die den Anteil einzelner Stützstellen am Ergebnis bestimmen. Wenn die Stützstellen äquidistant in $[a,b]$ gewählt sind, nennt man die sich daraus ergebende Methode *Newton-Cotes-Integration* (Abschnitt 4.2.1). Eine Erweiterung dazu ist der Ansatz von Gauß (Abschnitt 4.2.2); dabei wird zusätzlich zu den Gewichten noch die optimale Lage der Stützstellen bestimmt.

4.2.1 Newton-Cotes Integration

Bei der numerischen Integrationsmethode, die auf Newton und Cotes zurückgeht, wird der Wertebereich $[a,b]$ der zu integrierenden Funktion in gleichgroße Bereiche $[x_0,x_1],\ldots,[x_{n-1},x_n]$ mit $x_0 = a, x_n = b$ aufgeteilt, und auf jeden dieser Bereiche dann eine einfache Integrationsregel angewandt. Im folgenden werden wir für die Grenzen dieser Unterbereiche wieder a und b verwenden.

Die drei einfachsten Integrationsformeln für $\int_a^b f(x)dx$ können leicht folgendermaßen bestimmt werden:

- Die zu integrierende Funktion f wird durch einen konstanten Wert approximiert; meist ist dies der Funktionswert im Mittelpunkt des Intervalls:

$$\int_a^b f(x)dx \approx (b-a) f\Big(\frac{a+b}{2}\Big). \tag{4.6}$$

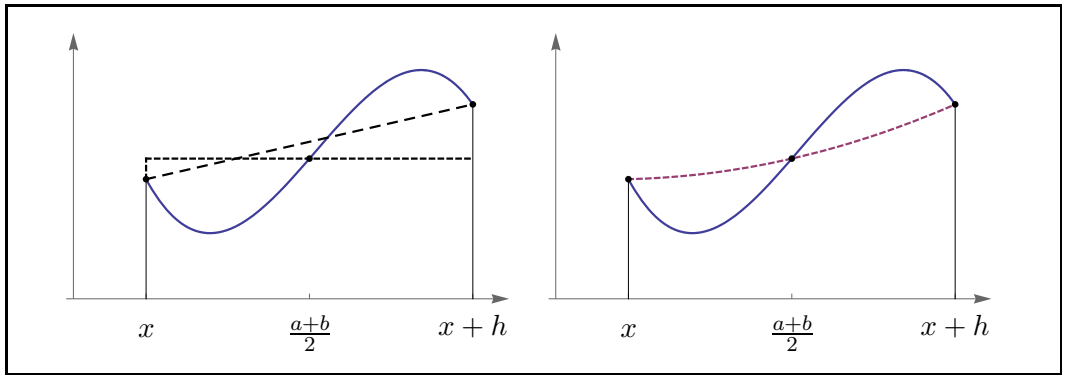

Abbildung 4.3: Integrationsregeln zur Approximation des Integrals unter einer Funktion durch Rechteck- und Trapezregel (gestrichelt, links) bzw. Simpson'sche Regel (gestrichelt, rechts).

Diese Integrationsformel wird als *Rechteckregel* oder speziell in obiger Form als *Mittelpunktregel* bezeichnet.

- Das Integral von f wird durch das Viereck durch die Graphenpunkte $(a, f(a))$ und $(b, f(b))$ angenähert. Somit ergibt sich die *Trapezregel*

$$\int_a^b f(x)dx \approx (b-a)\frac{f(a)+f(b)}{2}. \tag{4.7}$$

- Die zu integrierende Funktion wird zuerst durch ein quadratisches Polynom durch die drei Punkte $(a, f(a))$, $((a+b)/2, f((a+b)/2))$, $(b, f(b))$ interpoliert. Dieses Interpolationspolynom wird dann integriert. Es ergibt sich die *Simpsonsche Regel*

$$\int_a^b f(x)dx \approx \frac{b-a}{6}\left(f(a)+4f\Big(\frac{a+b}{2}\Big)+f(b)\right). \tag{4.8}$$

Veranschaulichungen dieser drei Integrationsregeln sind in Abbildung 4.3 zu sehen.

Die genaue Form der Approximation mittels Simpson'scher Regel ist nicht unmittelbar einsichtig, kann aber leicht hergeleitet werden: Wir schreiben zur Vereinfachung der Ausdrücke $2h = (b-a)$ und damit $(a+b)/2 = a+h$ und $b = a+2h$. Dann ist das Interpolationspolynom durch die drei äquidistanten Stützstellen in der Formulierung als Newton-Polynom gegeben durch

$$P(x) = a_0 + a_1(x-a) + a_2(x-a)(x-(a+h))$$

mit den Koeffizienten

$$a_0 = f(a), a_1 = \frac{f(a+h)-f(a)}{h}, a_2 = \frac{f[a+h, a+2h]-f[a, a+h]}{2h}$$

(siehe auch Abschnitt 3.2). Das Integral $\int_a^{a+2h} P(x)dx$ ergibt dann nach einigen Zwischenschritten die Simpson'sche Regel:

$$\int_a^{a+2h}\left(f(a)+\frac{f(a+h)-f(a)}{h}(x-a)\right.$$

$$+ \frac{f(a) - 2f(a+h) + f(a+2h)}{2h^2}(x-a)(x-(a+h))\Big)\, dx$$
$$= \frac{h}{3}\big(f(a) + 4f(a+h) + f(a+2\,h)\big).$$

Weitere Integrationsregeln, die auf der Interpolation durch mehr als drei Stützstellen beruhen, können völlig analog zur Simpson'schen Regel hergeleitet werden.

Fehlerschranken für Integrationsregeln können ähnlich den Fehlerschranken für Interpolationspolynome bestimmt werden. Wir betrachten zuerst die Mittelpunktregel (4.6). Wenn wir hier eine Taylorreihenentwicklung um den Mittelpunkt $m = (a+b)/2$ vornehmen, erhalten wir

$$f(x) = f(m) + f'(m)(x-m) + \frac{f''(m)}{2}(x-m)^2 + \frac{f^{(3)}(m)}{6}(x-m)^3 + \dots \quad (4.9)$$

Man kann dann nachrechnen, dass bei der Integration der rechten Seite zwischen a und b alle Ausdrücke mit ungeraden Ableitungen wegfallen: Es gilt nämlich

$$\int_a^b (x-m)^k dx = \frac{1}{k+1}(x-m)^{k+1}\Big|_a^b = \frac{1}{k+1}\left(\Big(\frac{b-a}{2}\Big)^{k+1} - \Big(\frac{a-b}{2}\Big)^{k+1}\right)$$
$$= \begin{cases} 0 & \text{falls } k \text{ ungerade} \\ \dfrac{1}{(k+1)2^k}(b-a)^{k+1} & \text{falls } k \text{ gerade} \end{cases}$$

Somit ist das Integral über die Taylorreihenentwicklung

$$\int_a^b f(x)dx = f(m)(b-a) + \frac{f''(m)}{24}(b-a)^3 + \frac{f^{(4)}(m)}{1920}(b-a)^5 + \dots, \quad (4.10)$$

woraus man die Fehlerabschätzung für die Mittelpunktregel ablesen kann.

Satz 4.2 Die Mittelpunktregel und die zugehörige Fehlerabschätzung sind gegeben durch

$$\int_a^b f(x)dx = f\Big(\frac{a+b}{2}\Big)(b-a) + \frac{f''(\xi)}{24}(b-a)^3.$$

für $\xi \in [a,b]$. Der Fehler wächst also mit der dritten Potenz der Intervallänge.

Beispiel 4.3 Wir bestimmen das Integral der Funktion $f(x) = e^{-x^2}$ zwischen $a = 0$ und $b = 1$ mit der Mittelpunktregel. Diese liefert den Wert

$$f\Big(\frac{a+b}{2}\Big)(b-a) = e^{-\frac{1}{4}} = 0.778801.$$

Das auf 10 Dezimalstellen genaue Resultat von $\int_0^1 e^{-x^2}dx$ ist 0.7468241328, der Fehler von 0.0319767 wird durch die theoretische Schranke von

$$\left|\frac{f''(\xi)}{24}(b-a)^3\right| = \left|\frac{1}{12}e^{-\xi^2}(2\xi^2-1)\right| = \frac{1}{12} = 0.0833333 \quad (\text{bei } \xi = 0)$$

abgeschätzt.

Ein genaueres Resultat erhält man, wenn man die zwei Intervalle $[0, \frac{1}{2}]$ und $[\frac{1}{2}, 1]$ getrennt betrachtet. Dann ergibt sich für die Summe beider Mittelpunktregeln

$$\frac{1}{2}e^{-\left(\frac{1}{4}\right)^2} + \frac{1}{2}e^{-\left(\frac{3}{4}\right)^2} = 0.754598,$$

welches einen Fehler von 0.00777381 bedeutet. Die Fehlerschranke ist jetzt

$$\left|\frac{f''(\xi_1)}{24}\left(\frac{1}{2}\right)^3\right| + \left|\frac{f''(\xi_2)}{24}\left(\frac{1}{2}\right)^3\right| = \frac{1}{96}\left(\left|e^{-\xi_1^2}(2\xi_1^2-1)\right| + \left|e^{-\xi_2^2}(2\xi_2^2-1)\right|\right)$$

für $\xi_1 \in [0, \frac{1}{2}]$ und $\xi_2 \in [\frac{1}{2}, 1]$. Diese Schranke wird für $\xi_1 = 0$ und $\xi_2 = \frac{1}{2}$ maximiert und liefert dann den Wert

$$\frac{1}{96}\left(\left|e^{-\xi_1^2}(2\xi_1^2-1)\right| + \left|e^{-\xi_2^2}(2\xi_2^2-1)\right|\right) = \frac{1}{96}(1+0.3894) = 0.0144729. \qquad \square$$

Eine Fehlerschranke für die Trapezregel (4.7) lässt sich elegant aus der Taylorreihenentwicklung für die Mittelpunktregel ableiten. Dafür setzen wir a und b für x in (4.9) ein und erhalten

$$f(a) = f(m) + f'(m)\Big(\frac{a-b}{2}\Big) + \frac{f''(m)}{2}\Big(\frac{a-b}{2}\Big)^2 + \frac{f^{(3)}(m)}{6}\Big(\frac{a-b}{2}\Big)^3 + \dots$$
$$f(b) = f(m) + f'(m)\Big(\frac{b-a}{2}\Big) + \frac{f''(m)}{2}\Big(\frac{b-a}{2}\Big)^2 + \frac{f^{(3)}(m)}{6}\Big(\frac{b-a}{2}\Big)^3 + \dots$$

Wenn wir diese zwei Gleichungen zusammenzählen, fallen wiederum die ungeraden Ableitungen weg und es bleibt

$$f(a) + f(b) = 2f(m) + f''(m)\Big(\frac{a-b}{2}\Big)^2 + \frac{f^{(4)}(m)}{12}\Big(\frac{a-b}{2}\Big)^4 + \dots$$

Diese Gleichung formen wir nun so um, dass wir die schon bekannte Mittelpunktregel verwenden können:

$$f(m)(b-a) = \frac{f(a)+f(b)}{2}(b-a) - \frac{f''(m)}{2}\frac{(b-a)^3}{2^2} - \frac{f^{(4)}(m)}{24}\frac{(b-a)^5}{2^4} - \dots$$

Wenn wir dies mit Gleichung (4.10) verbinden, die umgeformt

$$f(m)(b-a) = \int_a^b f(x)dx - \frac{f''(m)}{24}(b-a)^3 - \frac{f^{(4)}(m)}{1920}(b-a)^5 - \dots$$

lautet, so erhalten wir nach Gleichsetzen der rechten Seiten den Ausdruck

$$\int_a^b f(x)dx = \frac{f(a)+f(b)}{2}(b-a) - \frac{f''(m)}{12}(b-a)^3 - \frac{f^{(4)}(m)}{480}(b-a)^5 - \dots \quad (4.11)$$

Satz 4.3 Die Trapezregel mit Fehlerschranke lautet

$$\int_a^b f(x)dx = \frac{f(a)+f(b)}{2}(b-a) - \frac{f''(\xi)}{12}(b-a)^3.$$

mit $\xi \in [a, b]$. Der Fehler wächst wie bei der Mittelpunktregel mit der dritten Potenz der Intervallänge.

Zur Herleitung der Fehlerschranke für die Simpson'sche Regel geht man ähnlich vor wie bei der Herleitung der Fehlerschranke für die Trapezregel, nämlich durch Kombinieren schon bekannter Ergebnisse. So haben wir aus (4.10) und (4.11)

$$\int_a^b f(x)dx = f\Big(\frac{a+b}{2}\Big)(b-a) + \frac{f''(m)}{24}(b-a)^3 + \frac{f^{(4)}(m)}{1920}(b-a)^5 + \dots$$
$$\int_a^b f(x)dx = \frac{f(a)+f(b)}{2}(b-a) - \frac{f''(m)}{12}(b-a)^3 - \frac{f^{(4)}(m)}{480}(b-a)^5 - \dots$$

Um eine noch genauere Formel zu erhalten, möchte man die Terme mit den zweiten Ableitungen eliminieren. Das erreicht man dadurch, dass man die erste Gleichung mit $\frac{2}{3}$, die zweite mit $\frac{1}{3}$ multipliziert und dann zusammenzählt. Man erhält folgendes Ergebnis.

Satz 4.4 Die Simpson'sche Regel mit Fehlerabschätzung (für $\xi \in [a,b]$) ist

$$\int_a^b f(x)dx = \frac{(b-a)}{6}\left(f(a) + 4f\Big(\frac{a+b}{2}\Big) + f(b)\right) - \frac{f^{(4)}(\xi)}{2880}(b-a)^5.$$

Bei dieser Regel wächst der Fehler somit nur mit der fünften Potenz der Intervallänge.

Beispiel 4.4 Wir integrieren die Funktion $f(x) = e^{-x^2}$ aus Beispiel 4.3 mit Hilfe der Trapezregel und der Simpson'schen Regel für verschiedene Intervallängen. Die Resultate dieser Berechnungen sind, zusammen mit den Resultaten von Beispiel 4.3, in folgender Tabelle zu sehen. Dabei wurden die meisten Zwischenrechnungen in Beispiel 4.3 durchgeführt. Für die Simpon'sche Regel benötigt man noch die vierte Ableitung von f, diese ist

$$f^{(4)}(x) = 4e^{-x^2}\big(4x^4 - 12x^2 + 3\big).$$

Das Maximum von $|f^{(4)}|$ wird im Intervall $[0,1]$ in $\xi = 0$, in den Intervallen $[0,\frac{1}{2}]$ und $[\frac{1}{2},1]$ an den Stellen $\xi_1 = 0$ bzw. $\xi_2 = 1$ angenommen.

	Mittelpunktregel		Trapezregel		Simpson'sche Regel	
	Fehler	Schranke	Fehler	Schranke	Fehler	Schranke
$[0,1]$	0.031977	0.083333	0.062884	0.166667	0.000356	0.004167
$[0,\frac{1}{2}] \cup [\frac{1}{2},1]$	0.007774	0.014473	0.015454	0.028946	0.000031	0.000210

□

Aus dem Ergebnis des letzten Beispiels, und aus den vorhergehenden theoretischen Überlegungen kann man folgende Schlüsse ziehen:

- Da die Fehlerabschätzungen für die drei Integrationsregeln von f'' bzw. $f^{(4)}$ abhängen, sind diese Regeln für Polynome ersten bzw. dritten Grades exakt.

- Da die Fehlerabschätzungen für die drei Integrationsregeln von $(b-a)^3$ bzw. $(b-a)^5$ abhängen, verringert sich der Fehler durch Halbieren des Integrationsintervalls um den Faktor $\frac{1}{4}$ bzw. $\frac{1}{16}$ (ein Faktor 2 fällt durch das Verdoppeln der Intervallanzahl weg).
- Die Mittelpunktregel ist um den Faktor 2 genauer als die Trapezregel, obwohl sie auf der Approximation durch eine Konstante beruht.
- Obwohl man theoretisch immer genauere Integrationsregeln durch immer höhergradigere Interpolation entwerfen könnte, verwendet man in der Praxis Intervallunterteilungen und Extrapolation (siehe Abschnitt 4.3), um Rundungsfehler zu vermeiden und trotzdem genauere Ergebnisse zu erhalten.

Trotz dieser Vorteile gibt es Situationen, in denen Newton-Cotes Regeln nicht optimale Ergebnisse liefern. Wir werden hier nicht im Detail auf diese Bedingungen einzugehen, sondern mit der Methode von Gauß einen weiteren numerischen Integrationsansatz kennenlernen, der diese Schwierigkeiten vermeidet.

4.2.2 Gauß'sche Integration

Der Ansatz von Gauß unterscheidet sich von den Newton-Cotes Regeln in der Lage der Stützstellen der Polynominterpolation, die der Integration zugrundeliegt. Im allgemeinen Ansatz, ein Integral durch eine gewichtete Summe von Funktionswerten

$$\int_a^b f(x)dx = \sum_{i=0}^{n} w_i f(x_i) \tag{4.12}$$

zu approximieren, sind bei Newton-Cotes die Stützstellen äquidistant in $[a, b]$ verteilt. Bei Gauß werden die Stützstellen so gewählt, dass die Approximation an das Intervall optimal für die gegebene Anzahl der Stützstellen ist. Man nennt eine Integrationsregel dann optimal, wenn sie für Polynome mit höchstmöglichem Grad exakte Resultate liefert. In obiger Formel sind $2n + 2$ freie Parameter; ebensoviele freie Parameter gibt es in einem Polynom vom Grad $2n + 1$. Eine Intergrationsregel der Art (4.12) ist somit optimal, wenn sie auf Polynomen von Grad $2n + 1$ exakt ist.

Wir illustrieren die optimale Wahl von Stützstellen x_i und Gewichtungen w_i anhand eines einfachen Beispiels.

Beispiel 4.5 Sei $n = 1$ und das Integrationsintervall $[-1, 1]$. Wir müssen somit die vier Parameter w_0, w_1, x_0 und x_1 in (4.12) so bestimmen, dass Polynome dritten Grades exakt integriert werden. Somit muss für die Polynomfunktion $f(x) = a_3x^3 + a_2x^2 + a_1x + a_0$ die Gleichung

$$\int_{-1}^{1} f(x)dx = w_0 f(x_0) + w_1 f(x_1) \tag{4.13}$$

gelten. Wir benötigen aber vier Gleichungen, um die vier Parameter auszurechnen. Diese Gleichungen erhalten wir, da aus Approximation (4.13) folgt, dass die Approximation auf den Polynomen $1, x, x^2$ und x^3 exakt sein muss. Einsetzen dieser vier Funktionen für $f(x)$ in (4.13) liefert die vier Gleichungen

$$\int_{-1}^{1} 1dx = 2 = w_0 + w_1 \qquad \int_{-1}^{1} xdx = 0 = w_0x_0 + w_1x_1$$

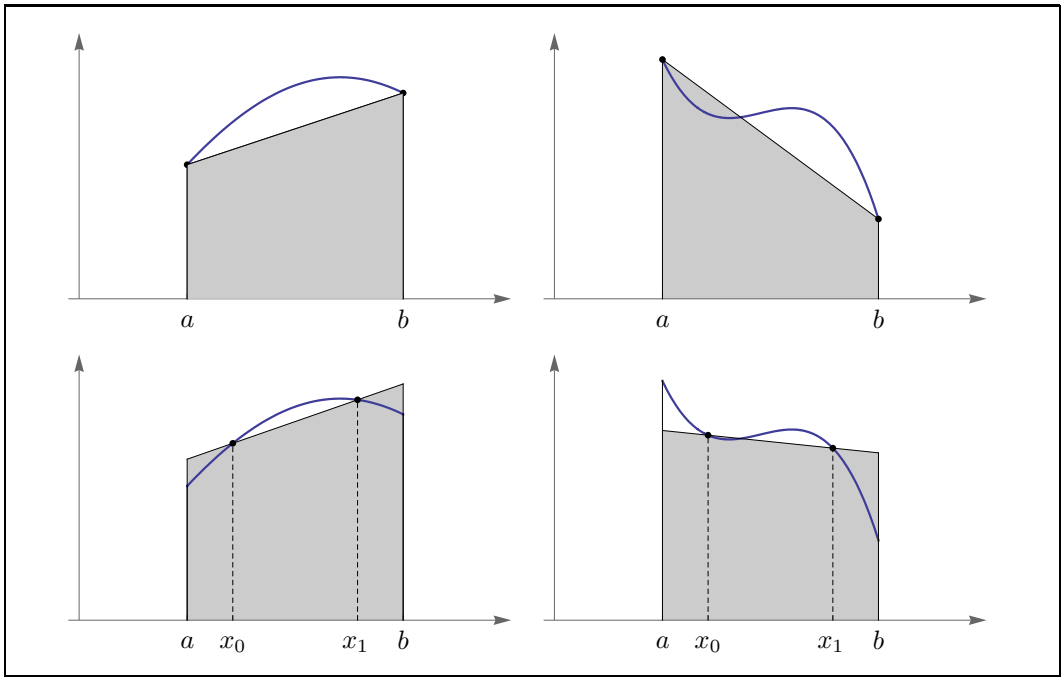

Abbildung 4.4: Vergleich von Newton-Cotes und Gauß'schen Integrationsregeln. Oben zwei Beispiele für die Trapezregel (Interpolationspunkte an den Intervallenden), unten zwei Beispiele für Gauß'sche Integration mit zwei Stützstellen. Hier liegen die Interpolationspunkte im Intervallinneren.

$$\int_{-1}^{1} x^2 dx = \frac{2}{3} = w_0 x_0^2 + w_1 x_1^2 \qquad \int_{-1}^{1} x^3 dx = 0 = w_0 x_0^3 + w_1 x_1^3$$

Dieses Gleichungssystem hat die eindeutige Lösung

$$w_0 = 1, \qquad w_1 = 1, \qquad x_0 = -\frac{1}{\sqrt{3}} = -0.57735, \qquad x_1 = \frac{1}{\sqrt{3}} = 0.57735.$$

Somit wird das Integral über f angenähert durch

$$\int_{-1}^{1} f(x)dx = f\Big(-\frac{1}{\sqrt{3}}\Big) + f\Big(\frac{1}{\sqrt{3}}\Big) \qquad \square$$

Der Unterschied zwischen Newton-Cotes und Gauß'schem Ansatz ist für zwei Beispielfunktionen in Abbildung 4.4 graphisch dargestellt. Man erkennt, dass durch die Wahl der Stützstellen im Inneren des Intervalls der Fehler bei der Integralsapproximation verringert wird. Die oben abgeleitete Intergrationsregel mit nur zwei Stützstellen ist für Polynome bis zum dritten Grad exakt.

Wenn man obigen Ansatz verallgemeinert, so kann man für gegebenen Polynomgrad diejenige Regel (also Gewichte w_i und Stützstellen x_i) herleiten, die für Polynome diesen Grades genau ist. Da aber die Wahl der Gewichte und Stützstellen nicht von der zu integrierenden Funktion abhängt, kann man sich auch allgemein die optimale Lage der Stützstellen und Gewichte berechnen. Die Stützstellen hängen aber sehr wohl vom Intervall ab, über das eine Funktion integriert werden soll. Dieses

Problem löst man damit, dass man die Stützstellen für ein Intervall (etwa $[-1, 1]$) bestimmt, da die *relative Lage* der Stützstellen im Intervall ja konstant ist. Die Stützstellenpositionen für andere Intervalle können dann aus den schon bekannten mit linearer Transformation bestimmt werden.

Es kann nachgeprüft werden, dass die Nullstellen einer speziellen Menge von Polynomen gerade die von uns gesuchten Stützstellen sind. Wir werden folgenden Satz nicht beweisen.

Satz 4.5 Sei $\{P_0, P_1, \dots\}$ eine Menge von Polynomen mit $\text{grad}(P_n) = n$ mit der Orthogonalitätseigenschaft

$$\int_a^b P_n(x)Q(x)dx = 0 \qquad \text{für alle Polynome } Q \text{ mit } \text{grad}(Q) < n.$$

Dann gilt:

- Die $n+1$ Nullstellen von P_{n+1} sind alle einfach, reell, und liegen im offenen Intervall (a, b).
- Die Gauß'sche Integrationsregel basierend auf den Nullstellen von P_{n+1} als Stützstellen ist optimal, also genau für alle Polynome bis zum Grad $2n + 1$.

Durch diesen Satz werden also die optimalen Stützstellen für Gauß'sche Integration als Nullstellen von speziellen Polynomen charakterisiert. Für das Intervall $[-1, 1]$ nennt man Polynome, die die Bedingungen des obigen Satzes erfüllen, *Legendre-Polynome*. Die ersten sechs Legendre-Polynome sind

$$P_0 = 1 \qquad P_1 = x \qquad P_2 = x^2 - \frac{1}{3}$$

$$P_3 = x^3 - \frac{3}{5}x \qquad P_4 = x^4 - \frac{6}{7}x^2 + \frac{3}{35} \qquad P_5 = x^5 - \frac{10}{9}x^3 + \frac{5}{21}x$$

Somit sind die Hälfte der Parameter in (4.12) bestimmt. Die andere Hälfte (die Gewichte) bestimmt man folgendermaßen.

Satz 4.6 Seien $x_0, \dots, x_n$ die Nullstellen des $(n + 1)$-ten Legendre-Polynoms. Dann ist die Gauß'sche Integrationsregel mit Gewichten

$$w_i = \int_{-1}^{1} \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} dx$$

exakt für Polynome bis zum Grad $2n + 1$.

Wir verzichten auf einen Beweis dieses Satzes.

Da Stützstellen und Gewichte unabhängig von der zu integrierenden Funktion und vom Integrationsintervall sind, kann man sich durch einmaliges Berechnen und

Tabellieren dieser Werte Arbeit ersparen. Für ein konkretes Integrationsproblem müssen dann nur noch die Stützstellen in das Integrationsintervall transformiert werden.

Beispiel 4.6 Wir greifen nochmals die Integration der Funktion $f(x) = e^{-x^2}$ aus Beispielen 4.3 und 4.4 im Intervall $[0, 1]$ auf.

Für $n = 1$ benötigt man zwei Stützstellen x_0, x_1 und zwei Gewichte w_0, w_1. Diese sind schon aus Beispiel 4.5 bekannt, die Stützstellen müssen nur mehr in das Intervall $[0, 1]$ transformiert werden. Wie man sich leicht überlegen kann, entspricht die Lage des Punktes $x \in [a, b]$ der Lage des Punktes $\xi \in [\alpha, \beta]$, wenn man ξ als der Beziehung

$$\frac{\xi - \alpha}{\beta - \alpha} = \frac{x - a}{b - a}$$

ausrechnet. Damit ergibt sich aus $x_0 = -\frac{1}{\sqrt{3}} = -0.57735$ in $[-1, 1]$ der Punkt $\xi_0 = -\frac{1}{2\sqrt{3}} + \frac{1}{2} = 0.211325$, aus $x_1 = \frac{1}{\sqrt{3}} = 0.57735$ der Punkt $\xi_1 = \frac{1}{2\sqrt{3}} + \frac{1}{2} = 0.788675$.

Für die Gewichte w_0 und w_1 muss man noch beachten, dass diese in Beispiel 4.5 für ein Intervall der Länge 2 berechnet wurden. Für ein Intervall der Länge 1 müssen diese Werte dann noch mit dem Faktor $\frac{1}{2}$ multipliziert werden. Allgemein gilt, dass man Gewichte w_i, die für ein Integrationsintervall $[a, b]$ bestimmt wurden, bei Transformation auf ein Intervall $[\alpha, \beta]$ durch Gewichte

$$\omega_i = \frac{\beta - \alpha}{b - a} w_i$$

ersetzen muss.

Die Approximation an das Integral ist somit

$$\int_0^1 e^{-x^2} dx \approx \frac{1}{2} f(\xi_0) + \frac{1}{2} f(\xi_1) = 0.74659469.$$

Dieser Wert weist einen Fehler von 0.000229 auf; dieser Fehler ist um zwei Zehnerpotenzen kleiner als der beste Newton-Cotes Fehler mit zwei Stützstellen (siehe Beispiel 4.4).

Für $n = 2$ hat man drei Stützstellen zu transformieren. Diese liegen als Nullstellen von $P_3 = x^3 - \frac{3}{5}x$ bei $x_0 = -\sqrt{\frac{3}{5}}, x_1 = 0$ und $x_2 = \sqrt{\frac{3}{5}}$, und nach Transformation ins Intervall $[0, 1]$ bei

$$\xi_0 = -\frac{\sqrt{3}}{2\sqrt{5}} + \frac{1}{2} = 0.112702, \qquad \xi_1 = 0.5, \qquad \xi_2 = \frac{\sqrt{3}}{2\sqrt{5}} + \frac{1}{2} = 0.887298.$$

Die Gewichte der Dreipunkt-Gauß'schen Regel sind $w_0 = w_2 = \frac{5}{9}$ und $w_1 = \frac{8}{9}$ für das Intervall $[-1, 1]$; die transformierten Gewichte für das Intervall $[0, 1]$ sind somit $\omega_0 = \omega_2 = \frac{5}{18}$ und $\omega_1 = \frac{4}{9}$.

Die Integrationsregel liefert mit diesen Parameterwerten die Approximation

$$\int_0^1 e^{-x^2} dx \approx \frac{5}{18} f(\xi_0) + \frac{4}{9} f(\xi_1) + \frac{5}{18} f(\xi_2) = 0.74681458.$$

Dieser Wert unterscheidet sich vom exakten Resultat nur mehr um 0.000009 und ist somit wieder um zwei Zehnerpotenzen genauer als die Newton-Cotes Regel mit drei Stützstellen. □

Beispiel 4.7 Eine weitere Funktion, für die keine Stammfunktion bestimmt werden kann, ist

$$f(x) = \frac{\sin(x)}{x}.$$

Wir berechnen einen numerischen Wert für den Flächeninhalt unter dieser Funktion zwischen 0 und π. Mit dem Gauß'schen Ansatz und den zwei Stützstellen

$$x_0 = -\frac{\pi}{2\sqrt{3}} + \frac{\pi}{2} = 0.6638966 \quad \text{und} \quad x_1 = \frac{\pi}{2\sqrt{3}} + \frac{\pi}{2} = 2.4776960$$

erhält man das Resultat

$$\int_0^\pi \frac{\sin(x)}{x} dx \approx \frac{\pi}{2}\big(f(x_0) + f(x_1)\big) = 1.84857153,$$

welches vom exakten Resultat 1.85193705 um 0.00336552 abweicht.

Mit den drei Stützstellen

$$x_0 = -\frac{\pi\sqrt{3}}{2\sqrt{5}} + \frac{\pi}{2} = 0.35406, x_1 = \frac{\pi}{2} = 1.57080, x_2 = -\frac{\pi\sqrt{3}}{2\sqrt{5}} + \frac{\pi}{2} = 2.78753$$

und Gewichten w_0, w_1 und w_2 wie in Beispiel 4.6 erhält man das verbesserte Resultat

$$\int_0^\pi \frac{\sin(x)}{x} dx \approx \frac{\pi}{2}\Big(\frac{5}{9}f(x_0) + \frac{8}{9}f(x_1) + \frac{5}{9}f(x_1)\Big) = 1.85197605,$$

welches mit einem Fehler von 0.00003900 um zwei Zehnerpotenzen genauer ist als die Gaußregel mit nur zwei Stützstellen. □

Wie wir aus der theoretischen Herleitung und jetzt anhand konkreter Beispiele gesehen haben, sind die Gauß'schen Integrationsregeln um einiges genauer als Newton-Cotes Regeln. Die Herleitung von Fehlerschranken für Gauß'sche Integrationsregeln ist aber komplizierter als für Newton-Cotes Regeln, da bei zweiteren relativ leicht Fehlerschranken durch Addition von Taylorreihenentwicklungen hergeleitet werden können. Dies ist aber nur dann möglich, wenn die Stützstellen bei beiden Expansionen gleich sind (damit sich entsprechende Terme wegheben können). Die Nullstellen von Legendre-Polynomen sind aber für verschiedene Grade bis auf 0 immer verschieden, sodass die Herleitung von Fehlerabschätzungen für Gauß'sche Integrationsregeln außerhalb des hier behandelten Themenbereichs liegt.

Eine weitere Möglichkeit, die Genauigkeit von sowohl numerischer Differentiation als auch Integration zu erhöhen bietet die im nächsten Abschnitt besprochene *Extrapolation*.

4.3 Richardson Extrapolation

Extrapolation kann überall dort eingesetzt werden, wo ein zu berechnender Wert (Ableitung, Fächeninhalt) in Abhängigkeit eines Parameters (Distanz h zweier Sekantenpunkte beim Differenzieren, Intervallänge $b - a$ beim Integrieren) erst bei "unendlich kleinem" Parameterwert das exakte Resultat liefert. Bei der Differentiation ist dieser Sachverhalt durch den Grenzwert $h \to 0$ in der Definition direkt enthalten. Bei der Integration kann man das Integral als Flächeninhalt unter einer

Kurve auch als Grenzwert der Summe immer kleinerer (in x-Richtung) rechteckiger Approximationen an die Fläche definieren. In beiden Fällen wird das exakte Ergebnis somit erst durch einen Grenzübergang erreicht, der am Computer natürlich nicht zu realisieren ist.

Im Folgenden sei $F(h)$ ein in Abhängigkeit des Parameters h zu berechnender Wert; wir wollen damit eine unbekannte Größe $A = F(0)$ annähern. Den Grenzübergang $\lim_{h\to 0} F(h)$ kann man vernünftig annähern, wenn man die Entwicklung von $F(h)$ für immer kleinere Werte von h betrachtet und daraus auf den gesuchten Wert $F(0)$ schließt.

Mathematisch können wir obige Überlegungen folgendermaßen formalisieren: Sei A die durch die Formel F zu approximierende Größe, wobei die Formel von einem Parameter h abhängt. Der Fehler in dieser Approximation lasse sich durch Potenzen von h ausdrücken, wie dies bei numerischer Differentiation und Integration der Fall ist:

$$F(h) - A = a_1 h + a_2 h^2 + a_3 h^3 + \dots \tag{4.14}$$

wobei die Parameter a_i meist von den Ableitungen der zu integrierenden bzw. differenzierenden Funktionen abhängen. Dabei sind viele der $a_i = 0$; je mehr der ersten a_i null sind, desto genauer ist die Approximation. Die Summe auf der rechten Seite von (4.14) kann durch das Restglied ersetzt werden, wenn die Approximationen auf der Taylorreihenentwicklung basieren. Um die Notation zu vereinfachen, sei k der kleinste Index mit $a_k \neq 0$, und wir ersetzen die unendliche Reihe durch das Restglied mit dem nächsten Koeffizienten ungleich null, um damit (4.14) als

$$F(h) - A = a_k h^k + a_m h^m \tag{4.15}$$

mit $m > k$ schreiben zu können.

Wenn man nun in (4.15) eine zweite Schrittweite qh mit $q > 1$ wählt, so erhält man

$$F(qh) - A = a_k (qh)^k + a_m (qh)^m.$$

Die Näherung wird besser, wenn man die k-te Potenz aus den letzten beiden Gleichungen eliminieren kann. Nach einigem Umformen erhält man dann

$$\begin{aligned} A &= \frac{F(h)q^k - F(qh) + a_m h^m (q^m - q^k)}{q^k - 1} \\ &= F(h) + \frac{F(h) - F(qh)}{q^k - 1} + ch^m \end{aligned} \tag{4.16}$$

mit $c = \frac{1}{q^k - 1}\big(a_m(q^m - q^k)\big)$. Somit hängt die Approximation nur mehr von h^m und nicht mehr von h^k ab; dies ist für $m > k$ natürlich eine Verbesserung.

Eine graphische Veranschaulichung dieser Überlegung ist in Abbildung 4.5 zu sehen. Dabei seien $F(h)$ und $F(qh)$ zwei (mehr oder weniger genaue) Approximationen an die unbekannte Größe $A = F(0)$. Wenn bekannt ist, wie sich der Fehler in der Approximation F entwickelt (dies geht durch den Parameter k in die Formel ein), kann man durch Extrapolation einen genaueren Wert von $F(0)$ schätzen.

Wir illustrieren diese Herleitung anhand von zwei Beispielen.

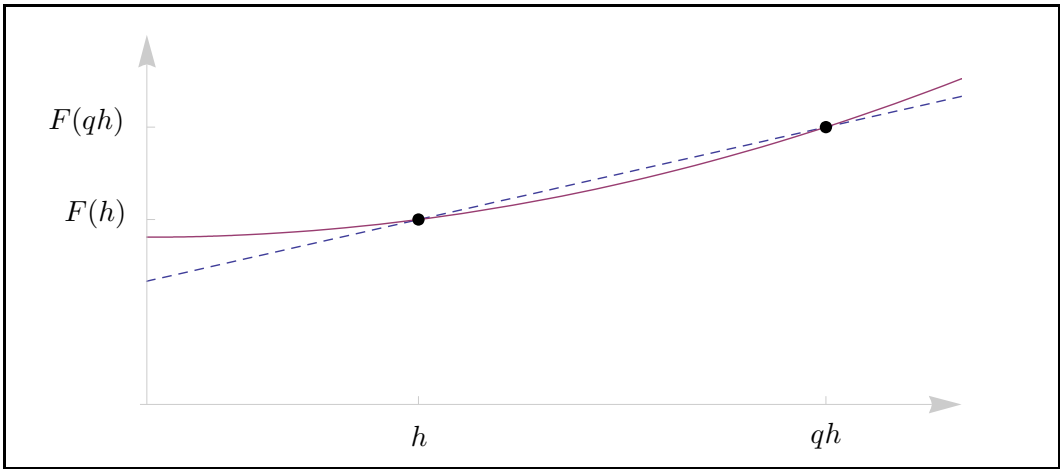

Abbildung 4.5: Illustration zur Richardson Extrapolation. Die Extrapolation bei linearem Fehler ist gestrichelt, die bei quadratischem Fehler durchgezogen gezeichnet. Die genauere Approximation ist die Stelle, an der die richtige Kurve die y-Achse schneidet.

Beispiel 4.8 Zu berechnen sei die Ableitung der Funktion $f(x) = \sin(x)/x$ an der Stelle $x = 1$. Wir verwenden dazu die einfache Approximation (4.2), die wir hier etwa genauer schreiben, um den Vergleich mit Gleichung (4.15) zu erleichtern:

$$\frac{f(x+h) - f(x)}{h} - f'(x) = \frac{f''(x)}{2}h + \frac{f'''(\xi)}{6}h^2.$$

Somit erhalten wir die Parameter $k = 1$ und $m = 2$ in Gleichung (4.15).

Wenn wir nun einmal $h = 0.25$ und einmal $h = 0.5$ (und somit einen Faktor $q = 2$) verwenden, so erhalten wir folgende Approximationen $F(h)$ an $f'(1)$:

$$F(0.25) = \frac{f(1.25) - f(1)}{0.25} = -0.329133$$

und

$$F(0.5) = \frac{f(1.5) - f(1)}{0.5} = -0.352949$$

Die durch Extrapolation dieser Werte gewonnene Approximation an $A = F(0) = f'(1)$ ist dann

$$A = F(0.25) + \frac{F(0.25) - F(0.5)}{2^1 - 1} = 2F(0.25) - F(0.5) = -0.305318.$$

Da das exakte Ergebnis $f'(1) = \cos(1) - \sin(1) = -0.301169$ ist, gewinnt man mit der Extrapolation eine weitere Stelle Genauigkeit (im Vergleich zu den Approximationen $F(0.25)$ und $F(0.5)$). Der Fehler in dieser Approximationsmethode wächst nur mehr mit h^2, und nicht mehr mit h. □

Beispiel 4.9 Zu bestimmen sei das Integral der Funktion $f(x) = x\log(x)$ zwischen 0 und 1. Wir verwenden zuerst zweimal die Trapezregel

$$\int_a^b f(x)dx = \frac{f(a)+f(b)}{2}(b-a) - \frac{f''(\xi)}{12}(b-a)^3.$$

Dabei muss beachtet werden, dass man allgemein bei der Integration das Intervall $[a,b]$ teilen muss, um die Schrittweite des Verfahrens zu ändern. Bei Teilung in n Teilintervalle ist die Schrittweite dann $h = (b-a)/n$, und das Gesamtintegral besteht aus n Teilergebnissen. Dies bedeutet allerdings auch, dass durch das Aufsummieren von $n = (b-a)/h$ Fehlern eine Potenz im Fehlerterm verlorengeht, sodass die zusammengesetzte Trapezregel nur mehr Genauigkeit h^2 hat.

Wie man aus der Herleitung der Trapezregel in Gleichung 4.11 erkennen kann, ist die nächsthöhere Potenz in der Reihenentwicklung $(b-a)^5$, welches durch Aufsummieren auch wieder auf die Form h^4 reduziert wird. Somit sind in diesem Fall die Extrapolationsparameter aus (4.16) $k = 2$ und $m = 4$. Mit einer Schrittweite von $h = 0.5$ teilen wir das Integrationsintervall in zwei Bereiche und erhalten mit der zusammengesetzten Trapezregel die Approximation

$$F(0.5) = \frac{f(0)+f(0.5)}{2}\cdot\frac{1}{2} + \frac{f(0.5)+f(1)}{2}\cdot\frac{1}{2} = -0.173287$$

Mit $q = 2$ ergibt sich die "normale" Trapezregel auf $[0,1]$:

$$F(1) = \frac{f(0)+f(1)}{2} = 0$$

Mit Extrapolation erhält man

$$A = F(0) = F(0.5) + \frac{F(0.5)-F(1)}{2^2-1} = \frac{4}{3}F(0.5) = -0.231049.$$

Obwohl dieses Ergebnis noch keine sehr gute Approximation zum exakten Resultat -0.25 ist, so ist es doch um einiges besser als die beiden Näherungen der Trapezregeln. Der Fehler in der Extrapolation wächst hier mit h^4, sodass für kleinere Werte von h schnell bessere Ergebnisse zu erwarten sind. □

Kapitel 5

Numerische Lösungen von Differentialgleichungen

In vielen Bereichen der Naturwissenschaften und der Technik treten Phänomene auf, bei denen sich eine oder mehrere Größen in Abhängigkeit von Zeit ändern. Beispiele dafür sind viele physikale Vorgänge wie etwa Pendelschwingungen, biologische Vorgänge wie Bevölkerungsentwicklungen, oder auch so scheinbar triviale Abläufe wie Verkehrsfluss und Verkehrsdichte.

Zur Beschreibung dieser Vorgänge werden meist *Differentialgleichungen* verwendet. Die allgemeine Form einer einfachen Differentialgleichung lautet

$$y' = f(y, t),$$

wobei y eine Funktion von t und f eine Funktion von y und t ist. Die Lösung einer Differentialgleichung ist somit eine *Funktion* und nicht so wie bisher ein Punkt oder eine Menge von Punkten. Da y meist eine Größe beschreibt, die sich in Abhängigkeit der Zeit ändert, hat sich die Verwendung von t als unabhängige Variable durchgesetzt.

Wir werden im Folgenden auf einige theoretische Grundlagen von Differentialgleichungen eingehen, bevor wir in Abschnitten 5.3, 5.5 und 5.6 numerische Lösungsmethoden vorstellen.

5.1 Grundlagen

Differentialgleichungen beschreiben die Veränderung einer oder mehrerer Größen in Abhängigkeit einer unabhängigen Variablen (meist der Zeit). Mathematisch lässt sich dieser Sachverhalt folgendermaßen ausdrücken: Sei dazu $[a, b]$ ein Intervall und $y : [a, b] \to \mathbb{R}^n$ eine unbekannte vektorwertige Funktion. Eine Differentialgleichung wird durch eine Funktion

$$\begin{aligned} f : \mathbb{R}^n \times [a, b] &\to \mathbb{R}^n \\ (y(t), t) &\mapsto f(y(t), t) \end{aligned}$$

definiert, die über die Gleichung

$$y'(t) = f(y(t), t) \tag{5.1}$$

die Ableitung der Funktion y an einer Stelle t festlegt. Meist schreiben wir statt (5.1) kurz $y' = f(y,t)$. Wenn die Funktion f linear in ihrem ersten Argument ist, spricht man von einer *linearen* Differentialgleichung; da nur die erste Ableitung von y auftritt, ist sie eine Differentialgleichung *erster Ordnung*. Eine *Lösung* von (5.1) ist eine Funktion y^*, deren Ableitung diese Gleichung erfüllt. Für spezielle Funktionen f ist es möglich, die Gleichung direkt zu lösen, sodass man eine Repräsentation der Lösung als Term erhält. Meist muss man sich aber damit begnügen, die Lösung nur als angenäherte Funktionswerte angeben zu können. Wir werden uns in den nächsten Abschnitten mit Verfahren beschäftigen, die möglichst gute Näherungen an die Lösung ergeben.

Wir können Gleichung (5.1) auch in ihre Komponenten aufteilen. Man erkennt dann besser, dass es sich bei dieser Formulierung eigentlich um ein *System gekoppelter Differentialgleichungen* handelt:

$$\begin{aligned} y_1' &= f_1(y_1, \ldots, y_n, t) \\ y_2' &= f_2(y_1, \ldots, y_n, t) \\ &\vdots \\ y_n' &= f_n(y_1, \ldots, y_n, t) \end{aligned}$$

Im Folgenden werden wir aber zur Vereinfachung meist nur eine dieser Komponenten behandeln.

Differentialgleichungen *höherer Ordnung* haben die allgemeine Gestalt

$$u^{(n)} = f(u, u', u'', \ldots, u^{(n-1)}, t);$$

die gesuchte Funktion u ist also über ihre höheren Ableitungen gegeben. Solch eine Differentialgleichung lässt sich aber mit den Definitionen $y_1 = u$, $y_2 = u'$, $\ldots$, $y_n = u^{(n-1)}$ auch wiederum als Differentialgleichungssystem erster Ordnung schreiben:

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ &\vdots \\ y_n' &= f(y_1, y_2, \ldots, y_n, t) \end{aligned}$$

Somit stellen Differentialgleichungen höherer Ordnungen eigentlich einen Spezialfall eines Differentialgleichungssystems erster Ordnung dar und werden von uns nicht gesondert behandelt.

Differentialgleichungen und Systeme der obigen Form werden als *gewöhnliche* Differentialgleichungen bezeichnet, da die unbekannte Funktion y eine Funktion nur einer Variablen ist. Wenn y auf mehrdimensionalen Mengen definiert ist, kann man y nach den verschiedenen Koordinaten partiell ableiten. Differentialgleichungen dieser Art werden somit als *partielle Differentialgleichungen* bezeichnet. Wir werden auf diesen Themenbereich nicht näher eingehen.

Beispiel 5.1 Als Beispiel einer Differentialgleichung betrachten wir

$$y'(t) = y(t) - t^2 + 1$$

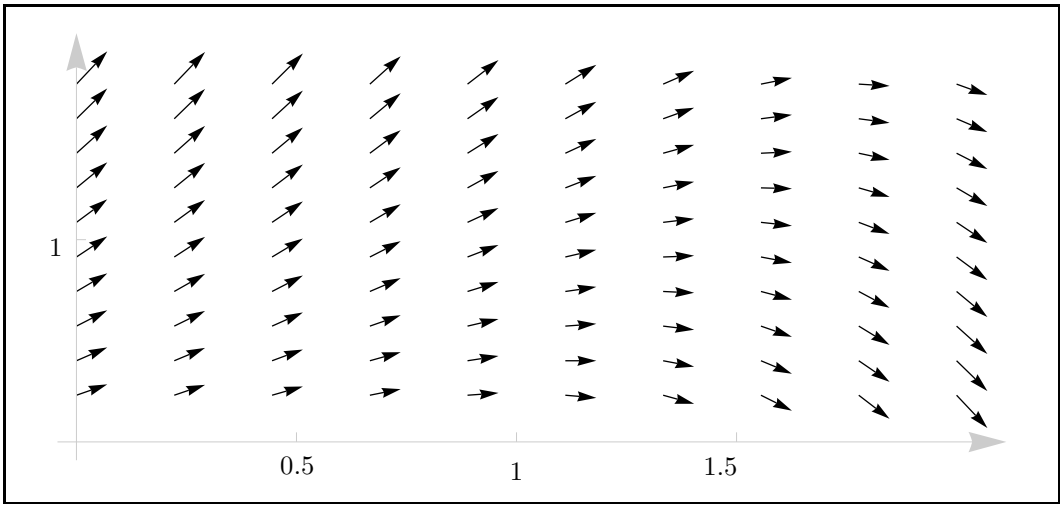

Abbildung 5.1: Visualisierung der Ableitungen, die durch die Differentialgleichung in Beispiel 5.1 gegeben sind.

Wir suchen also eine Funktion $y : t \mapsto y(t)$, deren Ableitung obige Gleichung erfüllt. Die Lösung dieser Differentialgleichung kann man sich folgendermaßen veranschaulichen: Für gegebenes t ist nur die *Ableitung* y' von y gegeben, nicht aber der Funktionswert selber. Diese Ableitung kann man nun an vielen *möglichen* Punkten $(t, y(t))$ durch einen Pfeil $(1, y'(t))$ visualisieren. An jedem möglichen Funktionswert ist damit dargestellt, in welche Richtung sich der Graph von y entwickelt. Dies ist für obige Gleichung in Abbildung 5.1 dargestellt.

Man sieht, dass es unendlich viele Lösungen für die Differentialgleichung gibt; man kann eine von diesen auswählen, indem man einen Startwert $y(t_0)$ für einen Anfangsparameter t_0 festlegt. In diesem konkreten Beispiel ist die Differentialgleichung symbolisch lösbar; man kann nachrechnen, dass

$$y(t) = (1 + t)^2 + ce^t$$

für beliebiges c eine Lösung der gegebenen Differentialgleichung ist. Wenn man nun eine dieser Lösungen auswählen will, legt man eine Anfangsbedingung fest (etwa $y(0) = 0$) und kann daraus direkt c ausrechnen (hier also $c = -1$). □

Aufgabenstellungen wie im letzten Beispiel, die durch

$$y' = f(y, t) \qquad \text{und} \qquad y(t_0) = y_0 \tag{5.2}$$

gegeben sind, nennt man *Anfangswertprobleme*. Wir werden uns im Folgenden nur mit solchen Problemen beschäftigen, da man für die numerische Lösung von Differentialgleichungen immer einen Startwert für die iterativen Verfahren benötigt. Wenn f stetig differenzierbar ist, dann hat das Anfangswertproblem (5.2) immer eine eindeutige Lösung.

Wenn Anfangswertprobleme nicht in geschlossener Form lösbar sind, so muss man sich mit Approximationen an die Funktionswerte $y(t)$ begnügen. Dabei beginnt man mit der Information $y(t_0) = y_0$ aus der Problemstellung, und kann dann $y'(t_0) = f(y(t_0), t_0)$ berechnen. Damit weiß man, in welche Richtung sich y entwickelt, und kann so eine Approximation an $y(t_0 + h)$ berechnen.

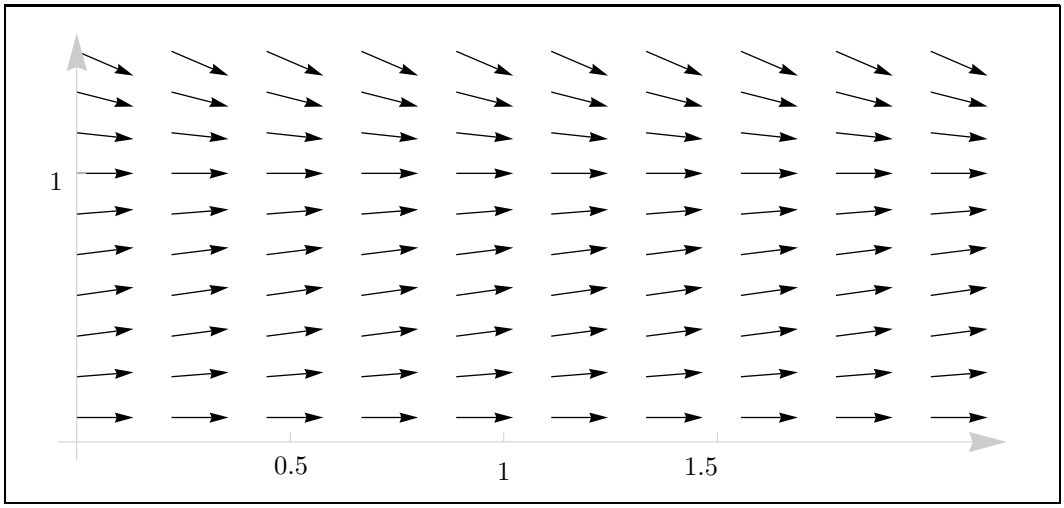

Abbildung 5.2: Visualisierung der Ableitungen, die durch die Differentialgleichung $y' = y(1-y)$ in Beispiel 5.2 gegeben sind.

Wir werden im Folgenden verschiedene Möglichkeiten kennenlernen, mit denen genaue Approximationen an $y(t_0 + h)$ möglich ist. Die einfachste von ihnen ist das in Abschnitt 5.3 behandelte *Eulerverfahren*, das als guter Einstieg in kompliziertere Verfahren dienen kann. Zuvor wenden wir uns einigen Überlegungen zu, mit denen sich einfache Aussagen über die Lösungen von Differentialgleichungen machen lassen.

5.2 Qualitative Analyse

Wie schon im vorigen Abschnitt beschrieben gibt eine Differentialgleichung

$$y' = f(y, t),$$

für jede Kombination von y und t die Steigung am Punkt (t, y) an. Ein Beispiel einer Visualisierung dieser Steigung ist in Abbildung 5.1 zu sehen. In diesem Abschnitt werden wir über solche Visualisierungen ein tieferes Verständnis von Differentialgleichungen erarbeiten. Dabei behandeln wir hier meist *autonome Differentialgleichungen*, also solche, bei denen die Zeit t nicht explizit eingeht. Von Interesse wird hauptsächlich die Entwicklung der Lösungen über die Zeit sein.

Beispiel 5.2 Die autonome Differentialgleichung

$$y' = y(1-y)$$

beschreibt ein *logistisches Modell* der Bevölkerungsentwicklung. Eine Veranschaulichung des durch diese Differentialgleichung gegebenen Vektorfelds ist in Abbildung 5.2 zu sehen. Man kann erkennen, dass sich die Lösungen beliebiger Anfangswerte im dargestellen Bereich (mit einer Ausnahme) auf den Wert $y = 1$ zubewegen. Das logistische Modell beschreibt somit ein Wachstum, das durch eine obere Schranke limitiert ist.

Dieses Verhalten kann man aus der Definition der Differentialgleichung ablesen. Als *Gleichgewichtslage* einer Differentialgleichung bezeichnet man jene Werte y, die

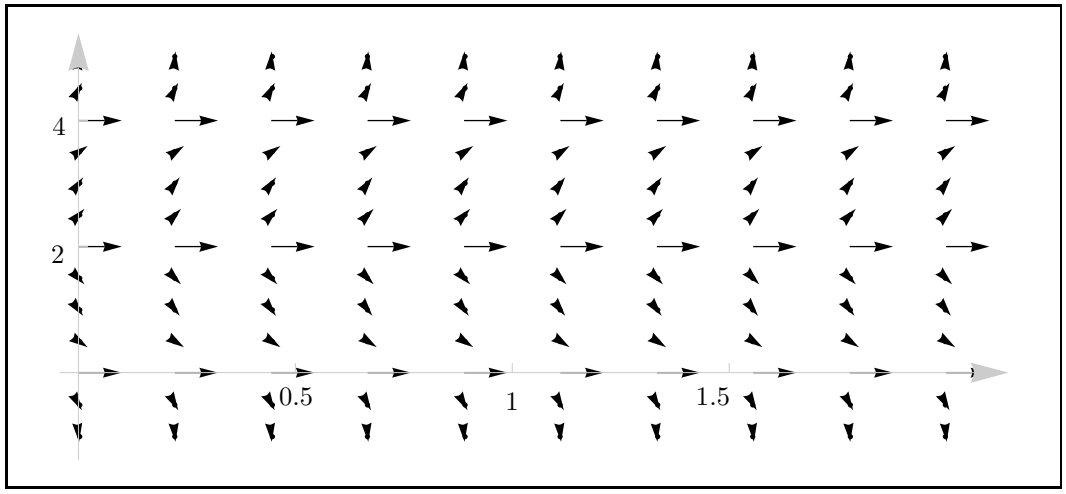

Abbildung 5.3: Visualisierung der Ableitungen, die durch die Differentialgleichung in Beispiel 5.3 gegeben sind.

sich über die Zeit nicht ändern, bei denen also $y' = 0$ gilt. Das logistische Modell hat somit Gleichgewichtslagen bei $y = 0$ und $y = 1$, wie sich aus $y(1 - y) = 0$ und auch graphisch leicht feststellen lässt. Die beiden Gleichgewichte scheinen sich aber qualitativ zu unterscheiden, da die Lösungen für verschiedene Anfangswerte zu $y = 1$ hin, von $y = 0$ aber wegstreben. Im ersten Fall spricht man von einem *stabilen Gleichgewicht*, im zweiten von einem *instabilen Gleichgewicht*. Dies lässt sich durch eine einfache Überlegung auch numerisch begründen. Wir unterteilen dazu die Menge aller möglichen Anfangswerte in drei Intervalle: $(1, \infty)$, $(0, 1)$, und $(-\infty, 0)$. In diesen Bereichen entwickeln sich beliebige Anfangswerte wie folgt:

- In $(1, \infty)$ ist $y' < 0$, da $y > 0$ und $(1 - y) < 0$ sind. Damit bewegen sich die Lösungen nach unten (gegen 1).
- In $(0, 1)$ ist $y' > 0$, da sowohl $y > 0$ und $(1 - y) > 0$ sind. Damit bewegen sich die Lösungen nach oben (gegen 1).
- In $(-\infty, 0)$ ist $y' < 0$, da $y < 0$ ist, aber $(1 - y) > 0$ ist. Damit bewegen sich die Lösungen nach unten (weg von 0). □

Ähnliche Überlegungen können angestellt werden, um das Verhalten der folgenden Differentialgleichung zu untersuchen.

Beispiel 5.3 Gegeben sei die autonome Differentialgleichung

$$y' = y^2(y - 4)(y - 2)^2,$$

deren Vektorfeld in Abbildung 5.3 zu sehen ist. Man kann erkennen, dass $y = 2$ ein instabiles Gleichgewicht ist, die beiden Punkte $y = 0$ und $y = 4$ aber weder stabil noch instabil sind, weil sich Lösungen von einer Seite zu diesen Punkten hin, von der anderen Seite aber von diesen Punkten wegentwickeln. Dies lässt sich auch numerisch nachrechnen. □

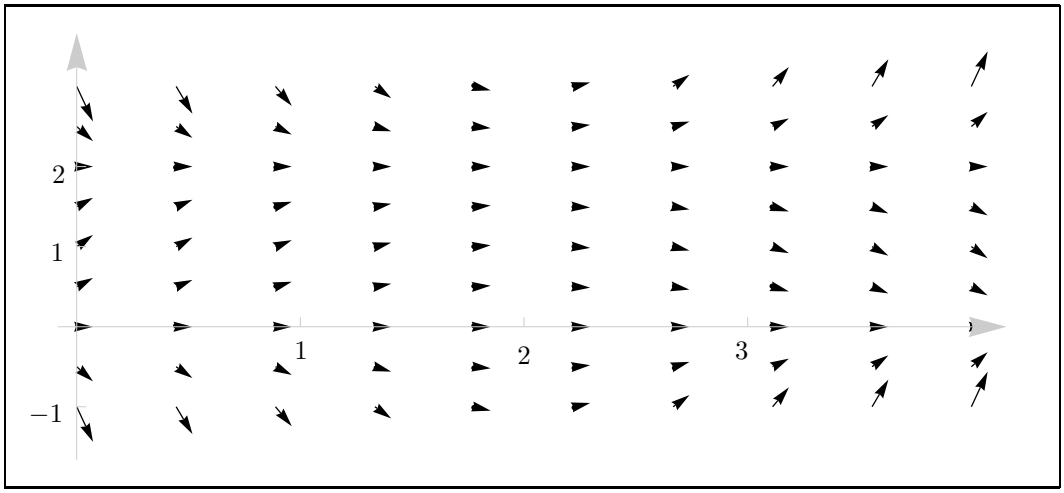

Abbildung 5.4: Visualisierung der Ableitungen, die durch die Differentialgleichung in Beispiel 5.4 gegeben sind.

Die Überlegungen der letzten beiden Beispiele sind nicht auf autonome Differentialgleichungen beschränkt. Durch Einbeziehung der Zeitachse ergeben sich weitere interessante Aspekte.

Beispiel 5.4 Gegeben sei die Differentialgleichung

$$y' = (t-2)y(y-2)$$

mit der Visualisierung als Vektorfeld, die in Abbildung 5.4 dargestellt ist. Man erkennt, dass im Zeitintervall $(0, 2)$ der Punkt $y = 2$ ein stabiles, und der Punkt $y = 0$ ein instabiles Gleichgewicht ist. Im Intervall $(2, \infty)$ ändert sich dieses Verhalten durch den Vorzeichenwechsel des Terms $(t-2)$ in der Definition der Differentialgleichung: Von $t = 2$ an ist $y = 2$ ein instabiles, $y = 0$ aber ein stabiles Gleichgewicht.□

Die Analyse von Gleichgewichtszuständen wird interessanter, wenn wir *Systeme* von Differentialgleichungen untersuchen. Dabei beschränken wir uns auf Systeme autonomer linearer Differentialgleichungen mit zwei Komponenten y_1 und y_2, da sich diese Systeme leicht visualisieren lassen. Die allgemeine Form eines solchen Systems lautet

$$\begin{aligned} y_1' &= a_{11}y_1 + a_{12}y_2 \\ y_2' &= a_{21}y_1 + a_{22}y_2 \end{aligned}$$

oder kürzer $y' = A \cdot y$, mit $y = (y_1, y_2)$ und $A = (a_{ij})$. Wie aus der linearen Algebra bekannt ist, haben Systeme mit regulärer Matrix A den einzigen Gleichgewichtszustand $(0, 0)$, da dies die einzige Lösung des Gleichungssystems $A \cdot x = 0$ ist. Wir betrachten zunächst ein Beispiel.

Beispiel 5.5 Gegeben sei das autonome lineare Differentialgleichungssystem

$$\begin{aligned} y_1' &= y_1 + 3y_2 \\ y_2' &= 3y_1 + y_2 \end{aligned}$$

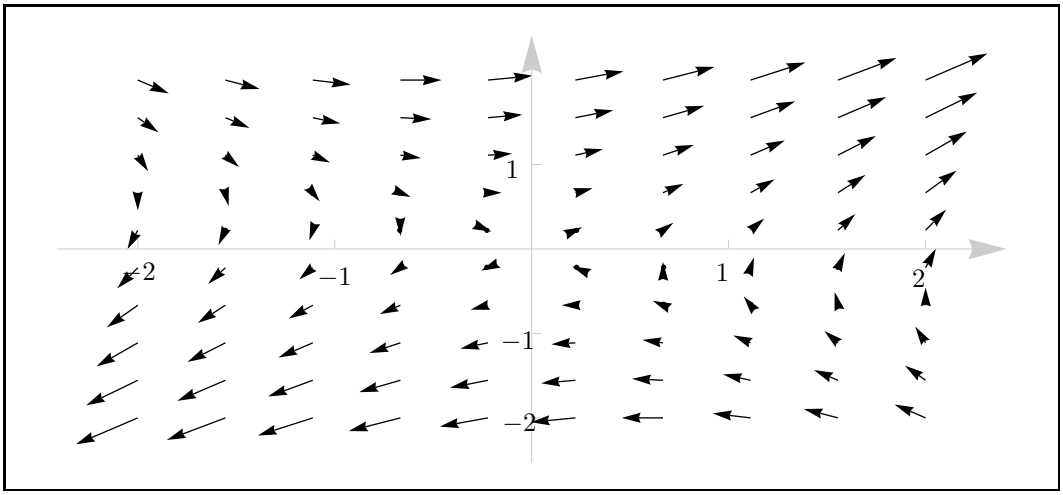

Abbildung 5.5: Phasenportrait des Differentialgleichungssystems in Beispiel 5.5

mit der Visualisierung in Abbildung 5.5. Eine Darstellung eines Differentialgleichungssystems, bei denen die beiden Komponenten die beiden Achsen eines Koordinatensystems darstellen, wird *Phasenportrait* eines Differentialgleichungssystems genannt. Da die Matrix des Systems regulär ist, ist der Ursprung der einzige Gleichgewichtszustand. Dieser Gleichgewichtszustand ist Beispiel eines *Sattelpunktes*, da sich die Lösungen entlang einer Linie dem Ursprung nähern, sich aber entlang einer anderen Linie von diesem entfernen.

Die Matrix

$$\begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}$$

des hier untersuchten Differentialgleichungssystems hat das charakteristische Polynom $(1-x)^2-9 = x^2-2x-8$, mit den Nullstellen (und damit Eigenwerten) -2 und 4. Man kann nachweisen, dass der Gleichgewichtspunkt *jedes* Systems mit Eigenwerten λ_1, λ_2 mit $\lambda_1 > 0 > \lambda_2$ ein Sattelpunkt ist. Die beiden Richtungen, entlang derer sich die Lösungen direkt zum bzw. vom Sattelpunkt hin- bzw. wegbewegen, sind durch die Eigenvektoren der Matrix gegeben. Diese sind für obige Matrix die beiden Vektoren

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Jede Lösung, die auf einer der durch diese Vektoren definierten Linien liegt, bleibt auf dieser Linie: So bewegt sich die Lösung im Punkt $(2, 2)$ wegen

$$\begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 8 \\ 8 \end{pmatrix}$$

weiter auf dieser Linie. □

Die Analyse der Eigenwerte eines Systems erlaubt es uns, verschiedene qualitativ unterschiedliche Klassen von linearen Differentialgleichungssystemen zu unterscheiden. Wir verzichten allerdings auf eine vollständige Katalogisierung aller möglichen Situationen, sondern betrachten zwei weitere Beispiele.

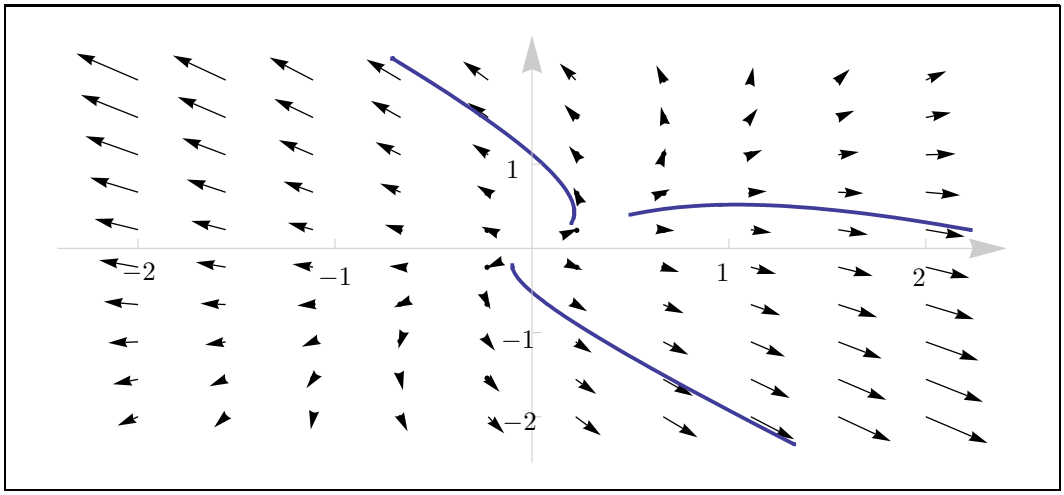

Abbildung 5.6: Phasenportrait mit drei Lösungskurven des Differentialgleichungssystems in Beispiel 5.6.

Beispiel 5.6 Das System

$$\begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

hat die Eigenwerte $\lambda_1 = 1$ und $\lambda_2 = 3$, mit den Eigenvektoren $(1, 1)$ und $(-1, 1)$. Obwohl die Eigenvektoren gleich sind wie in Beispiel 5.5, ergibt sich durch die beiden positiven Eigenwerte ein unterschiedliches Phasenportrait: In diesem Fall ist der Ursprung ein sogenannter *Knoten 2. Art*, von dem sich die Lösungen wegentwickeln. Die Lösungen von drei Anfangsbedingungen sind im Phasenportrait in Abbildung 5.6 eingetragen; die zeitliche Entwicklung der Lösungen verläuft dabei von innen nach außen. □

Beispiel 5.7 Eine völlig andere Situation ergibt sich beim System

$$\begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} -1 & -1 \\ 4 & -1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \end{pmatrix},$$

das die zwei komplexen Eigenwerte $\lambda_{1,2} = -1 \pm 2i$ hat (und damit keine reellen Eigenwerte). Es gibt somit keine Richtungen, entlang derer sich Lösungen dem Ursprung nähern bzw. sich von diesem entfernen. Dies ist auch aus dem Phasenportrait in Abbildung 5.7 ersichtlich, das eine spiralförmige Entwicklung zweier Lösungen hin zum Ursprung zeigt. □

5.3 Eulerverfahren

Das Eulerverfahren implementiert die einfachste Möglichkeit, von einem gegebenen Punkt auf einer Lösung eines Anfangswertproblems zum nächten (approximierten) Punkt zu gelangen. Gegeben sei ein Anfangswertproblem der Art (5.2). Ausgehend

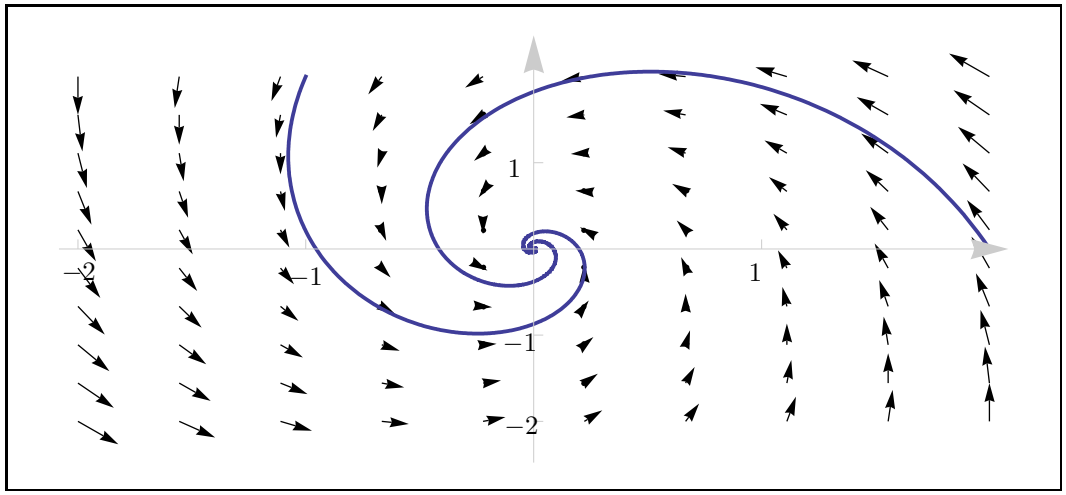

Abbildung 5.7: Phasenportrait mit zwei Lösungskurven des Differentialgleichungssystems in Beispiel 5.7.

von einem bereits erreichten Punkt $(t_n, y(t_n))$ wird aus der Definition der Differentialgleichung eine Ableitung $y'(t_n)$ berechnet, die dann wiederum zur Berechnung eines neuen Punktes $(t_{n+1}, y(t_{n+1}))$ verwendet wird.

Mathematisch lässt sich diese Vorgehensweise aus der Taylorreihenentwicklung von y um den Punkt t ablesen:

$$\begin{aligned} y(t+h) &= y(t) + y'(t)h + \frac{y''(t)}{2}h^2 + \dots \\ &= y(t) + f(y,t)h + \frac{y''(t)}{2}h^2 + \dots \\ &= y(t) + f(y,t)h + \frac{y''(\xi)}{2}h^2 \end{aligned} \tag{5.3}$$

mit $\xi \in [t, t+h]$. Durch Weglassen des Fehlerterms $\frac{1}{2}y''(\xi)h^2$ kann man mit Gleichung (5.3) Funktionswerte an den Stellen $t_0, t_0 + h, t_0 + 2h, \dots$ generieren; diese Vorgehensweise wird *Eulerverfahren*, und der Parameter h *Schrittweite* des Verfahrens genannt. Eine graphische Illustration des Eulerverfahrens ist in Abbildung 5.8 zu sehen.

Satz 5.1 (Eulerverfahren) Die Iterationsvorschrift zur Berechnung der Approximationsschritte des Eulerverfahrens für das Anfangswertproblems $y'(t) = f(y(t), t)$, $y(t_0) = y_0$ mit Schrittweite h lautet

$$\begin{aligned} y_{i+1} &:= y_i + hf(y_i, t_i) \\ t_{i+1} &:= t_i + h\,. \end{aligned}$$

Aus obiger Herleitung kann man erkennen, dass der Fehler des Eulerverfahrens in jedem Schritt proportional zu h^2 ist. Genauere Abschätzungen unter Verwendung

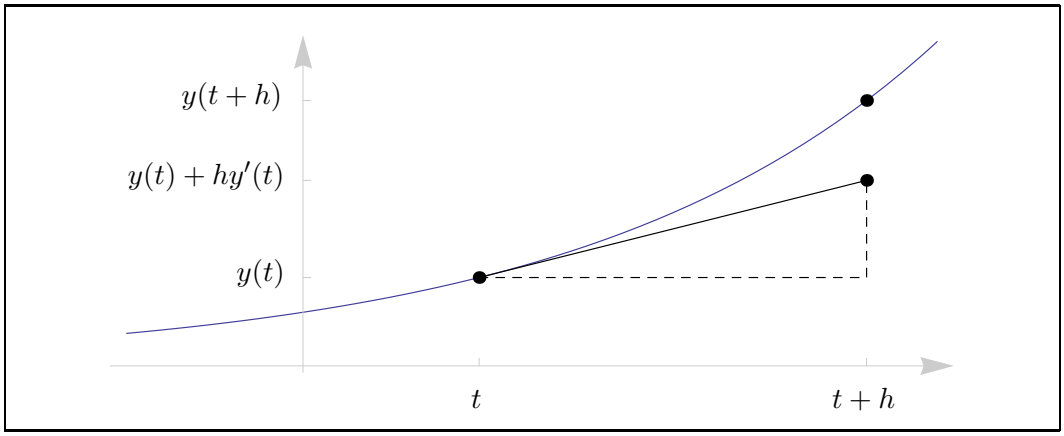

Abbildung 5.8: Illustration des Eulerverfahrens: Der echte, unbekannte Funktionswert $y(t+h)$ wird durch $y(t)+hy'(t)$ approximiert.

des Fehlerglieds sind meist nicht möglich, da wir die Lösungsfunktion y ja nicht kennen, die wir für die Berechnung der Grenze $y''(\xi)$ benötigen würden. Da die Schrittweite h frei wählbar ist, kann man durch Schrittweiten, die klein genug gewählt sind, beliebig genaue Ergebnisse *im ersten Iterationsschritt* erreichen. Alle nachfolgenden Iterationsschritte verwenden dann aber zur Berechnung von $y(t_{i+1}) = y(t_i + h)$ nicht mehr den genauen Wert $y(t_i)$ auf der rechten Seite von (5.3), sondern die im letzten Schritt generierte Approximation daran. Mit den uns zur Verfügung stehenden Methoden ist es aber leider nicht erkennbar, ob sich Fehler in den nächsten Iterationsschritten vergrößern oder verkleinern.

Wir untersuchen diese Situation anhand eines Beispiels.

Beispiel 5.8 Wir betrachten nochmals das Anfangswertproblem

$$y'(t) = y(t) - t^2 + 1 \qquad \text{mit} \qquad y(0) = 2$$

mit der exakten Lösung $y(t) = (1+t)^2 + e^t$ aus Beispiel 5.1; hierbei wurde nur die Anfangsbedingung geändert. Bei einer Schrittweite von $h = 0.1$ ergibt sich eine Approximation $y(0.1) = 2.3$ mit einem Fehler von 0.0151709 (verglichen mit dem exakten Ergebnis). Die ersten Approximationen $y(t)$ sind, zusammen mit den dabei gemachten Fehlern, in folgender Tabelle zusammengefasst:

	t					
	0	0.1	0.2	0.3	0.4	0.5
exakt	2.0	2.31517	2.6614	3.03986	3.45182	3.89872
Euler	2.0	2.3	2.629	2.9879	3.37769	3.79946
Fehler	0.0	0.015171	0.032403	0.051959	0.074135	0.099262

Der Fehler scheint linear mit der Entfernung der Stützstellen zum Anfangspunkt 0 zu wachsen. Man kann zusätzlich noch nachweisen, dass der Fehler linear mit h wächst (siehe Ende dieses Abschnitts); man kann somit genauere Ergebnisse durch Herabsetzen der Schrittweite erreichen. Zum Vergleich die Fehler an den gleichen Stellen wie in obiger Tabelle, aber diesmal mit einer Schrittweite von $h = 0.05$ berechnet (zu sehen ist also nur jeder zweite Wert):

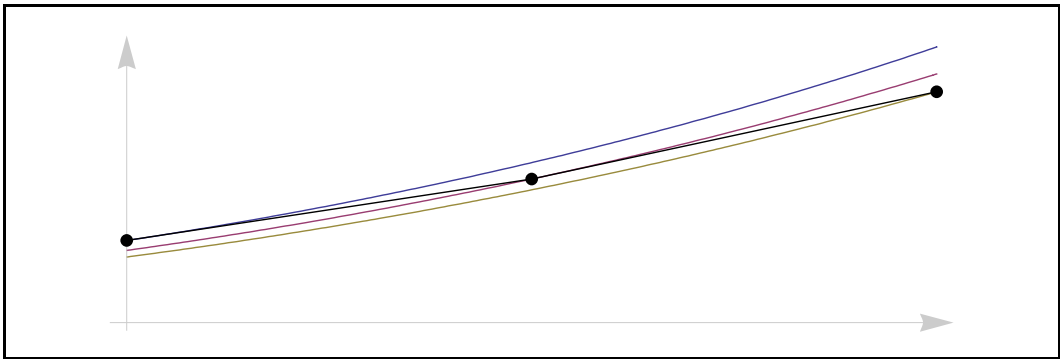

Abbildung 5.9: Lösungen des Anfangswertproblems aus Beispiel 5.8 entfernen sich mit jedem Iterationsschritt von der exakten Lösung (oberste Kurve).

	t					
	0	0.1	0.2	0.3	0.4	0.5
exakt	2.0	2.31517	2.6614	3.03986	3.45182	3.89872
Euler	2.0	2.30738	2.64473	3.01309	3.41358	3.84745
Fehler bei $h = 0.05$	0.0	0.007796	0.016672	0.026768	0.038242	0.051271
Fehler bei $h = 0.1$	0.0	0.015171	0.032403	0.051959	0.074135	0.099262

Man kann deutlich die lineare Abhängigkeit des Fehlers von der Schrittweite erkennen.

Anhand dieses Beispiels lässt sich noch folgender wichtiger Sachverhalt über Fehlerquellen beim numerischen Lösen von Differentialgleichungen illustrieren: Mit einer Schrittweite $h = 0.1$ ist die erste Approximation an die Lösung gegeben durch $y(0.1) = 2.3$, die exakte Lösung ist jedoch 2.31517. Der Fehler an der nächsten Stützstelle 0.2 kommt jetzt aus zwei Quellen: einerseits ist der Ausgangswert $y(0.1)$ schon fehlerhaft, andererseits ist das Eulerverfahren nicht exakt, sodass dadurch nochmals ein Fehler gemacht wird.

Wie wir bereits wissen, gibt es zu jeder Anfangsbedingung genau eine Lösung des Anfangswertproblems (5.2); durch den inheränten Fehler des Eulerverfahrens springt man so von einer exakten Lösung zur nächsten. Dies ist in Abbildung 5.9 für obige Differentialgleichung zu sehen. □

Das im obigen Beispiel gezeigte Verhalten der numerischen Approximation an die Differentialgleichungslösung ist nicht für alle Differentialgleichungen charakteristisch: Bei anderen Anfangswertproblemen können die Fehler in der Approximation mit Fortschreiten der Lösungsentwicklung *abnehmen*. Man spricht in diesem Zusammenhang von der *Stabilität* einer Differentialgleichung: Wenn kleine Änderungen der Anfangsbedingungen zu großen Änderungen der Lösung führen, so spricht man von *instabilen* Gleichungen; wenn kleine Änderungen sich aber nicht sehr auf die Lösungen auswirken, so nennt man solche Gleichungen *stabil*.

Wir betrachten nun ein Beispiel einer stabilen Differentialgleichung.

Beispiel 5.9 Ein Anfangswertproblem sei gegeben durch

$$y'(t) = -y(t) \qquad \text{mit} \qquad y(0) = 2,$$

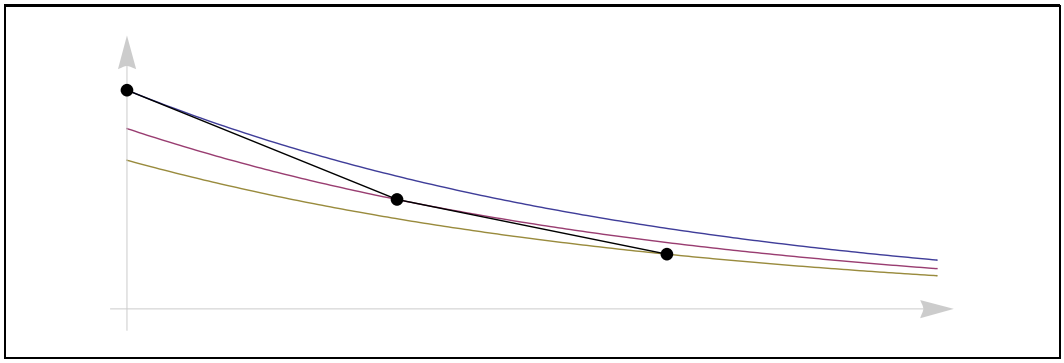

Abbildung 5.10: Approximationen an die Lösungen des Anfangswertproblems aus Beispiel 5.9 liegen auf Lösungskurven, die sich sich immer mehr der exakten Lösung (oberste Kurve) annähern.

dieses Problem hat die exakte Lösung $y(t) = 2e^{-t}$. Die Approximationen an diese Lösung durch das Eulerverfahren mit Schrittweite $h = 0.5$ sind, zusammen mit den dabei gemachten Fehlern, in untenstehender Tabelle zusammengefasst.

	t					
	0	0.5	1.0	1.5	2.0	2.5
Euler	2.0	1.0	0.5	0.25	0.125	0.0625
exakt	2.0	1.21306	0.735759	0.44626	0.270671	0.16417
Fehler	0.0	0.213061	0.235759	0.19626	0.145671	0.10167

Wie man in Abbildung 5.10 erkennen kann, laufen die unterschiedlichen Lösungen dieses Anfangswertproblems immer näher zusammen. Damit ist der Fehler, wie aus obiger Tabelle ersichtlich, auch gering und nur durch den Approximationsfehler des Eulerverfahrens bedingt (da die exakten Lösungen ja immer näher zusammenrücken). □

Wie aus den letzten beiden Beispielen ersichtlich wurde, muss man beim numerischen Lösen von Differentialgleichungen zwei Arten von Fehlern unterscheiden: einerseits die *globalen* Fehler (also der absolute Fehler zwischen Approximation und echter Lösung an einer Stelle t_i), andererseits der in einem Schritt gemachte *lokale* Fehler. Der globale Fehler ist aber mehr als nur die Summe der lokalen Fehler. Dies ergibt sich daraus, dass bei einer instabilen Differentialgleichung lokale Fehler in jedem Schritt amplifiziert werden, während bei stabilen Gleichungen das Gegenteil eintritt. Dabei spielt nicht nur die Charakteristik einer Differentialgleichung (stabil oder instabil), sondern auch die (relative) Stabilität des Lösungsverfahrens eine Rolle.

Wir werden auf die Fragestellung der Charakterisierung von Lösungsverfahren nach ihrer Stabilität nicht eingehen, sondern zum Abschluss dieses Abschnitts einen Blick auf die Qualität des Eulerverfahrens werfen. Wie schon in Gleichung (5.3) hergeleitet, ist der nächste Schritt (mit Restglied) im Eulerverfahren gegeben durch

$$y(t+h) = y(t) + f(y,t)h + \frac{y''(\xi)}{2}h^2\,.$$

Der *lokale Fehler* in jedem Schritt wächst somit nur mit h^2. Da eine Näherungslösung an ein Anfangswertproblem im Intervall $[a,b]$ aber $n = (b-a)/h$ Schritte benötigt,

fällt durch das Aufsummieren der n lokalen Fehler eine Potenz von h^2 weg, wodurch sich auch theoretisch die beobachtete lineare Abhängigkeit zwischen Schrittweite h und Fehler erklären lässt.

Größere Genauigkeit bei der numerischen Lösung von Differentialgleichungen kann man dann damit erreichen, dass man Verfahren mit quadratischen, kubischen oder höheren lokalen Fehlern entwickelt. Diesen Gedanken formalisieren wir in Abschnitt 5.5. Davor betrachten wir noch eine einfache Möglichkeit, auch ohne höhere Ableitungen das Eulerverfahren viel genauer zu machen.

5.4 Heunverfahren

Die Überlegungen im letzten Abschnitt haben gezeigt, dass der globale Fehler beim Eulerverfahren linear mit der Schrittweite h wächst. Wir haben den Trick, zwei unterschiedliche Taylorreihenentwicklungen geschickt zu kombinieren, bereits in Kapitel 4 kennengelernt. Das Ziel dort und auch hier ist es, dass sich höhere Potenzen der Reihenentwicklung wegheben, und damit der Fehlerterm eine noch höhere Potenz in h aufweist.

Wir erinnern uns nochmals an die Taylorreihenentwicklung zur numerischen Lösung der Differentialgleichung $y' = f(y,t)$:

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2}y''(t) + \frac{h^3}{3!}y'''(t) + \dots \tag{5.4}$$

Wenn wir stattdessen den Entwicklungpunkt $t+h$ wählen, und die Reihenentwicklung an der Stelle t auswerten, so erhalten wir

$$y(t) = y(t+h) - hy'(t+h) + \frac{h^2}{2}y''(t+h) - \frac{h^3}{3!}y'''(t+h) + \dots \tag{5.5}$$

Wir wollen nun diese beiden Gleichungen so miteinander kombinieren, dass die Terme mit h^2 wegfallen, und damit der lokale Fehler mit h^3 wächst. Dies ist aber dadurch erschwert, dass die entsprechenden Terme nicht identisch sind, sondern einmal den Faktor $y''(t)$, und einmal den Faktor $y''(t+h)$ enthalten.

Man kann aber kennen, dass der Unterschied zwischen $y''(t)$ und $y''(t+h)$ nur von y''' abhängt, indem man etwa die Taylorreihenentwicklung von y'' betrachtet:

$$y''(t+h) = y''(t) + hy'''(t) + \dots$$

Wenn wir dies in Gleichung (5.5) einsetzen, so erhalten wir

$$y(t) = y(t+h) - hy'(t+h) + \frac{h^2}{2}y''(t) + \frac{h^3}{2}y'''(t) + \dots + \frac{h^3}{3!}y'''(t+h) + \dots \tag{5.6}$$

Sowohl die ursprüngliche Taylorreihenentwicklung (5.4) als auch die letzte Gleichung (5.6) enthalten somit Terme, die proportional zu h^3 sind, und die wir nur für die Fehlerentwicklung benötigen.

Das Addieren der umgeformten Gleichung (5.6), bei der wir alle höheren Potenzen in ein Restglied verpacken,

$$y(t+h) = y(t) + hy'(t+h) - \frac{h^2}{2}y''(t) - \alpha\frac{h^3}{3!}y'''(\xi)$$

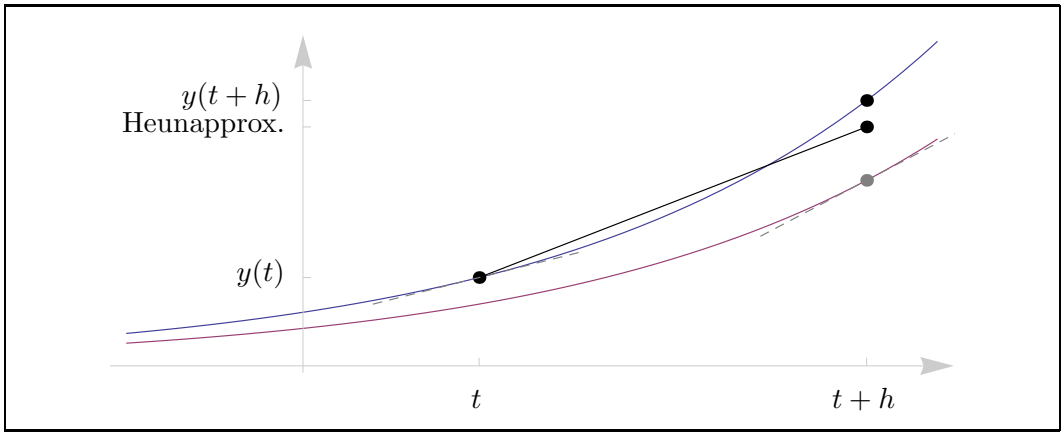

Abbildung 5.11: Illustration des Heunverfahrens: Der echte, unbekannte Funktionswert $y(t+h)$ wird mit einer Steigung approximiert, die der Durchschnitt der Steigungen am Anfang und am Ende eines Eulerschrittes sind.

zu Gleichung (5.4) liefert dann

$$y(t+h) = y(t) + \frac{h}{2}\big(y'(t) + y'(t+h)\big),$$

bei dem zur leichteren Lesbarkeit der Fehlerterm in h^3 weggelassen wurde.

Diese Gleichung kann noch nicht direkt als Iterationsvorschrift umgesetzt werden, da $y'(t+h)$ ja nicht bekannt ist. Man behilft sich damit, die Ableitung in dem Punkt zu bestimmen, den das Eulerverfahren liefern würde – dies entspricht einer lineare Approximation des gesuchten Werts. Sauber ausformuliert liefert dies die Iterationsvorschrift des *Heunverfahrens*.

Satz 5.2 (Heunverfahren) Die Iterationsvorschrift zur Berechnung der Approximationsschritte des Heunverfahrens für das Anfangswertproblems $y'(t) = f(y(t), t)$, $y(t_0) = y_0$ mit Schrittweite h lautet

$$\begin{aligned} \tilde{y} &:= y_i + h f(y_i, t_i) \qquad \text{(Hilfsvariable, Eulerschritt)} \\ t_{i+1} &:= t_i + h \\ y_{i+1} &:= y_i + \frac{h}{2}\big(f(y_i, t_i) + f(\tilde{y}, t_{i+1})\big)\,. \end{aligned}$$

Das Heunverfahren hat eine einfache und elegante geometrische Interpretation, die in Abbildung 5.11 zu sehen ist: Während beim Eulerverfahren die Steigung $y'(t)$ im Punkt t zur Berechnung des nächsten Punktes $y(t+h)$ verwendet wird, wird beim Heunverfahren der Durchschnitt aus der Steigung in t und der (mit dem Eulerverfahren geschätzten) Steigung in $t+h$ verwendet.

Wir werden in Abschnitt 5.6 sehen, dass die Heun-Methode identisch mit der Runge-Kutta-Methode 2. Ordnung ist. Bei der dortigen Herleitung wird auch deutlich, warum man als Approximation an $y'(t+h)$ den Wert verwendet, den der Eulerschritt liefert.

Beispiel 5.10 Zum Vergleich der Genauigkeit des Heunverfahrens und des Eulerverfahrens wenden wir die beiden Methoden auf das Anfangswertproblem in Beispiel 5.8 an:

$$y'(t) = y(t) - t^2 + 1 \qquad \text{mit} \qquad y(0) = 2$$

Die exakte Lösung ist $y(t) = (1+t)^2 + e^t$. In folgender Tabelle sind für eine Schrittweite von $h = 0.1$ die Entwicklung der Fehler der beiden Verfahren aufgelistet.

	t					
	0	0.1	0.2	0.3	0.4	0.5
exakt	2.0	2.31517	2.6614	3.03986	3.45182	3.89872
Fehler Euler	0.0	0.015171	0.032403	0.051959	0.074135	0.099262
Fehler Heun	0.0	0.000671	0.001430	0.002289	0.003260	0.004358

Man kann erkennen, dass der Fehler des Heunverfahrens deutlich unter dem des Eulerverfahrens liegt. Wie in Beispiel 5.8 untersuchen wir auch hier exemplarisch den Einfluss der Schrittweite h auf den Fehler der Approximation des Heunverfahrens, indem wir h halbieren:

	t					
	0	0.1	0.2	0.3	0.4	0.5
exakt	2.0	2.31517	2.6614	3.03986	3.45182	3.89872
Fehler bei $h = 0.05$	0.0	0.000173	0.000368	0.000589	0.000839	0.001122
Fehler bei $h = 0.01$	0.0	0.000671	0.001430	0.002289	0.003260	0.004358

An diesem Beispiel kann die (auch theoretisch begründbare) Fehlerentwicklung des Heunverfahrens in Abhängigkeit von h^2 abgelesen werden. Wir werden an dieser Stelle auf die Herleitung dieser Fehlerentwicklung verzichten; in Abschnitt 5.6 ist dies bei der Herleitung des Runge-Kutta-Verfahrens 2. Ordnung (identisch zum Heunverfahren) besser ersichtlich. □

5.5 Taylorapproximationen höherer Ordnung

Bei der Herleitung des Eulerverfahrens im letzten Abschnitt haben wir in Gleichung (5.3) die Taylorreihenentwicklung der Lösung y nach dem linearen Term abgebrochen, und so einen linearen lokalen Fehler erhalten. Eine höhere Genauigkeit der Lösung erhält man, wenn man mehr Terme der Reihenentwicklung verwendet. So ergibt sich mit den ersten n Potenzen der Reihenentwicklung der Ausdruck

$$y(t+h) = y(t) + y'(t)h + \frac{y''(t)}{2}h^2 + \cdots + \frac{y^{(n)}(t)}{n!}h^n + \frac{y^{(n+1)}(\xi)}{(n+1)!}h^{n+1} \qquad (5.7)$$

mit $\xi \in [t, t+h]$. Hier treten nun aber höhere Ableitungen von y auf. Diese Ableitungen kann man aus dem Anfangswertproblem

$$y' = f(y,t) \qquad \text{und} \qquad y(t_0) = y_0$$

ausrechnen. Hierzu fehlen uns aber einige elementare Voraussetzungen, die wir in einem kurzen Einschub nachholen. Wir beginnen mit der Ableitung in höheren Dimensionen.

Definition 5.1 (Richtungsableitung)
Sei $D \subseteq \mathbb{R}^n$, $f : D \to R$ eine Funktion, $x, v \in D$ mit $\|v\| = 1$. Dann nennt man den Grenzwert

$$\frac{\partial f(x)}{\partial v} = \lim_{h \to 0} \frac{f(x + hv) - f(x)}{h}$$

die *Richtungsableitung* von f in Richtung v im Punkt x.

Richtungsableitungen in beliebige Richtungen werden uns nicht weiter beschäftigen. Wir benötigen nur den Spezialfall, dass die Ableitung in eine der Koordinatenrichtungen erfolgt.

Definition 5.2 (Partielle Ableitungen)
Sei $D \subseteq \mathbb{R}^n$ und f eine Funktion

$$f : D \to \mathbb{R}$$
$$(x_1, \ldots, x_n) \mapsto f(x_1, \ldots, x_n).$$

Dann bezeichnet man den Grenzwert

$$\frac{\partial f(x_1, \ldots, x_n)}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_i + h, \ldots, x_n) - f(x_1, \ldots, x_n)}{h}$$

als *partielle Ableitung* von f nach x_i.

Eine partielle Ableitung ist somit nichts anderes als eine Richtungsableitung in Richtung einer der Koordinatenachsen. Mit partiellen Ableitungen kann man wie mit eindimensionalen Ableitungen rechnen.

Beispiel 5.11 Sei $f : \mathbb{R}^3 \to \mathbb{R}$ gegeben durch $f(x, y, z) = \sin(xy) + e^{yz}$. Dann ist

$$\begin{aligned}
\frac{\partial f(x, y, z)}{\partial x} &= y \cos(xy) \\
\frac{\partial f(x, y, z)}{\partial y} &= x \cos(xy) + z e^{yz} \\
\frac{\partial f(x, y, z)}{\partial z} &= y e^{yz}.
\end{aligned}$$

□

Weiters werden wir im Folgenden für die Ableitung einer Funktion f einer Funktion g die *Kettenregel* benötigen. Dies lautet für den eindimensionalen Fall

$$\frac{d}{dt} f(g) = \frac{df}{dg} \cdot \frac{dg}{dt}$$

Beispiel 5.12 Sei $f(x) = x^2$ und $g(x) = \sin(x)$. Dann ist $f(g(x)) = \sin^2(x)$, und $f'(g(x)) = 2\sin(x)\cos(x)$. □

Bei höherdimensionalen Funktionen verallgemeinert die *mehrdimensionale Kettenregel* die Kettenregel. Seien $x, y : \mathbb{R} \to \mathbb{R}$ und $f : \mathbb{R}^2 \to \mathbb{R}$ reellwertige Funktionen einer bzw. zweier Variablen. Dann gilt

$$\frac{d}{dt} f(x,y) = \frac{\partial f}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial f}{\partial y} \cdot \frac{dy}{dt}.$$

Die mehrdimensionale Kettenregel werden wir in Abschnitt 5.6 benötigen. Zunächst aber ein einfaches Beispiel.

Beispiel 5.13 Seien $x(t) = t^2$ und $y(t) = \sin(t)$ sowie $f(x,y) = xy + \cos(x)$. Dann gilt

$$\begin{aligned}\frac{d}{dt} f(x,y) &= \frac{\partial f}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial f}{\partial y} \cdot \frac{dy}{dt} \\ &= (y - \sin(x)).2t + x\cos(t) \\ &= 2t\sin(t) - 2t\sin(t^2) + t^2\cos(t).\end{aligned}$$ □

Nach dieser kurzen Zusammenfassung wieder zurück zur Anwendung von Ableitungen bei höherdimensionalen Taylorreihenverfahren.

Mit der Schreibweise $f'(y,t) = \frac{d}{dt} f(y,t)$ ergibt sich

$$\begin{aligned}y' &= \frac{dy}{dt} = f(y,t) \\ y'' &= \frac{d^2 y}{dt^2} = f'(y,t) \\ y''' &= \frac{d^3 y}{dt^3} = f''(y,t) \\ &\vdots\end{aligned}$$

Damit können wir die Iterationsschritte eines *Taylorverfahrens höherer Ordnung* folgendermaßen angeben.

Satz 5.3 (Taylorverfahren n-ter Ordnung) Die Iterationsvorschrift zur Berechnung der Approximationsschritte eines Taylorverfahrens n-ter Ordnung für das Anfangswertproblems $y'(t) = f(y(t),t)$, $y(t_0) = y_0$ mit Schrittweite h lautet

$$\begin{aligned}y_{i+1} &:= y_i + hf(y_i,t_i) + \frac{f'(y_i,t_i)}{2}h^2 + \cdots + \frac{f^{(n-1)}(y_i,t_i)}{n!}h^n \qquad (5.8)\\ t_{i+1} &:= t_i + h\end{aligned}$$

Ein Verfahren n-ter Ordnung verwendet auf der rechten Seite Terme bis zur n-ten Ableitung von y, also bis zur $(n-1)$-ten Ableitung von f nach t. Das Restglied

(und damit auch der lokale Ein-Schritt-Fehler) dieser Entwicklung hängt dann von der n-ten Ableitung von f und von h^{n+1} ab, der globale (aufsummierte) Fehler von h^n. Wir können also auf Basis der Approximation (5.8) genauere numerische Lösungen erhalten.

Wir illustrieren diese Vorgangsweise an einem Beispiel.

Beispiel 5.14 Wir betrachten nochmals das schon in Beispiel 5.8 behandelte Anfangswertproblem

$$y'(t) = y(t) - t^2 + 1 \qquad \text{mit} \qquad y(0) = 2$$

mit der exakten Lösung $y(t) = (1+t)^2 + e^t$. Für Lösungen mit quadratischem bzw. kubischem lokalen Fehler benötigen wir die Ableitungen von $f(y,t) = y - t^2 + 1$. Diese sind

$$f'(y,t) = y' - 2t = y - t^2 - 2t + 1$$

und

$$f''(y,t) = y' - 2t - 2 = y - t^2 - 2t - 1$$

Wenn wir diese Ableitungen in (5.8) einsetzen, erhalten wir für die Approximation zweiter Ordnung

$$\begin{aligned} y_{i+1} &= y_i + hf(y_i,t_i) + \frac{f'(y_i,t_i)}{2}h^2 \\ &= y_i + h\big(y_i - t_i^2 + 1\big) + \frac{h^2}{2}\big(y_i - t_i^2 - 2t_i + 1\big), \end{aligned}$$

und für die Approximation dritter Ordnung

$$\begin{aligned} y_{i+1} &= y_i + hf(y_i,t_i) + \frac{f'(y_i,t_i)}{2}h^2 + \frac{f''(y_i,t_i)}{6}h^3 \\ &= y_i + h\big(y_i - t_i^2 + 1\big) + \frac{h^2}{2}\big(y_i - t_i^2 - 2t_i + 1\big) + \frac{h^3}{6}\big(y_i - t_i^2 - 2t - 1\big). \end{aligned}$$

Mit der Anfangsbedingung $w_0 = 2$ und einer Schrittweite von $h = 0.1$ ergeben sich die in folgender Tabelle zusammengefassten numerischen Approximationen. Dabei ist zum Vergleich das Ergebnis des Eulerverfahrens inkludiert.

t	exakt	Fehler Euler	Fehler 2. Taylor	Fehler 3. Taylor
0.0	2.0	0.0	0.0	0.0
0.1	2.315171	0.015171	0.000171	0.00000425
0.2	2.66140	0.032403	0.000378	0.00000940
0.3	3.039859	0.051959	0.000626	0.00001558
0.4	3.451825	0.074134	0.000922	0.00002296
0.5	3.898721	0.099262	0.001274	0.00003171

□

Wie man aus diesem Beispiel erkennen kann, liefern Taylormethoden höherer Ordnung um vieles bessere Ergebnisse als das Eulerverfahren (das ja die Taylermethode erster Ordnung ist). Man benötigt zu ihrer Anwendung aber die Ableitungen von

$f(y,t)$; da diese in vielen Fällen aufwendig zu berechnen sind, werden höhergradige Taylormethoden nur selten eingesetzt.

Im nächsten Abschnitt werden wir ein Verfahren kennenlernen, das die hohe Genauigkeit von Taylormethoden aufweist, ohne zusätzliche Ableitungen berechnen zu müssen.

5.6 Runge-Kutta Methoden

Die Berechnung der höheren Ableitungen von y bzw. f bei Taylorverfahren höherer Ordnung kann vermieden werden, wenn man diese Ableitung ebenfalls wieder approximiert. Dabei muss man allerdings auf *zweidimensionale Taylorreihenentwicklung* zurückgreifen, die wie folgt definiert ist.

Satz 5.4 (Zweidimensionale Taylorreihenentwicklung) Gegeben sei eine Funktion $f \in C^{n+1}\big([a,b]\times[c,d]\big)$. Dann gilt für $(x+h, y+k) \in [a,b]\times[c,d]$:

$$\begin{aligned} f(x+h,y+k) = f(x,y) &+ \left(h\frac{\partial f(x,y)}{\partial x} + k\frac{\partial f(x,y)}{\partial y}\right) \\ &+ \left(\frac{h^2}{2}\frac{\partial^2 f(x,y)}{\partial x^2} + hk\frac{\partial^2 f(x,y)}{\partial y \partial x} + \frac{k^2}{2}\frac{\partial^2 f(x,y)}{\partial y^2}\right) \\ &+ \dots \\ &+ \frac{1}{n!}\sum_{i=0}^{n}\binom{n}{i} h^{n-i}k^i \frac{\partial^n f(x,y)}{\partial x^{n-i}\partial y^i} + R_n(\xi,\zeta) \end{aligned}$$

mit dem Restglied

$$R_n(\xi,\zeta) = \frac{1}{(n+1)!}\sum_{i=0}^{n+1}\binom{n+1}{i} h^{n+1-i}k^i \frac{\partial^{n+1} f(\xi,\zeta)}{\partial x^{n+1-i}\partial y^i},$$

wobei $\xi \in [x, x+h]$ und $\zeta \in [y, y+h]$ liegt.

Dieser Satz besagt somit, dass sich eine zweidimensionale Funktion in der Nähe eines Punktes (x,y) durch spezielle Kombinationen ihre partiellen Ableitungen approximieren lässt. Der dabei gemachte Fehler kann wiederum durch ein Restglied abgeschätzt werden.

Beispiel 5.15 Die Funktion $f(x,y) = \sin(\cos(x))e^{\sin(y)}$ hat um den Punkt $(0,0)$ die zweidimensionale Taylorreihenentwicklung

$$f(h,k) = \sin(1) + k\sin(1) + \frac{1}{2}\big(-h^2\cos(1) + k^2\sin(1)\big) + R_2(\xi,\zeta)$$

mit $\xi \in [0,h]$ und $\zeta \in [0,k]$. Diese Approximation ist in der Nähe von $(0,0)$ erwartungsgemäß gut, weiter entfernt schon nicht mehr so gut: Es ist $f(0.1,-0.1) = 0.759069$ und $g(0.1,-0.1) = 0.75883$; andererseits ist $f(-1,1) = 1.19328$, aber

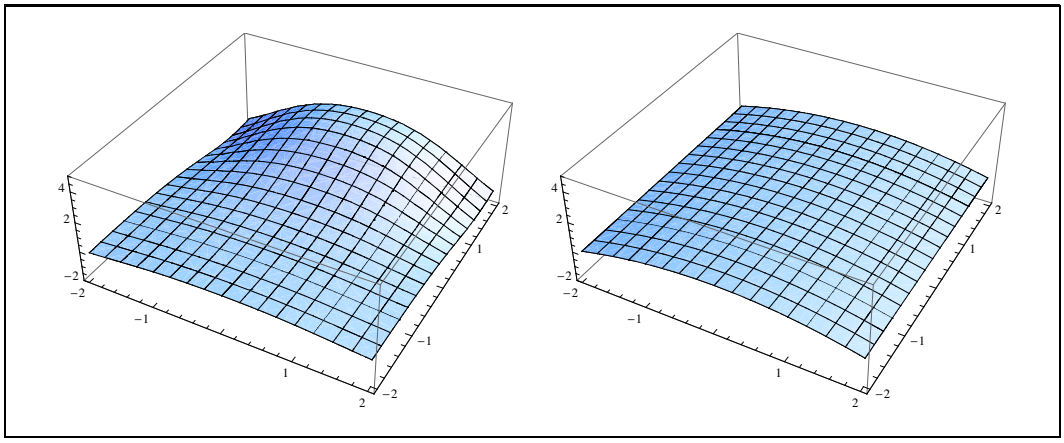

Abbildung 5.12: Plot der Funktion $f(x,y) = \sin(\cos(x))e^{\sin(y)}$ aus Beispiel 5.15 (links) und ihrer Taylorreihenentwicklung um den Punkt $(0,0)$ (rechts).

$g(-1,1) = 1.83353$. Die beiden Funktionen sind graphisch in Abbildung 5.12 zu sehen. Man erkennt, dass die Approximation mit zunehmender Entfernung von $(0,0)$ immer schlechter wird. □

Mit der zweidimensionalen Taylorreihenentwicklung ist es nun möglich, die im letzten Abschnitt aufgetretenen Terme $f'(y,t), f''(y,t), \ldots$ zu approximieren. Mit der höherdimensionalen Kettenregel und der Tatsache $y' = f(y,t)$ gilt

$$\begin{aligned} f'(y,t) = \frac{d}{dt}f(y,t) &= \frac{\partial f(y,t)}{\partial y} \cdot \frac{dy(t)}{dt} + \frac{\partial f(y,t)}{\partial t} \cdot \frac{dt}{dt} \\ &= \frac{\partial f(y,t)}{\partial y} \cdot f(y,t) + \frac{\partial f(y,t)}{\partial t}. \end{aligned}$$

Wenn man nun $f(y + hf(y+t), t+h)$ als Reihenexpansion um (y,t) anschreibt, erhält man als erste Glieder

$$f(y + hf(y,t), t+h) = f(y,t) + hf(y,t)\frac{\partial f(y,t)}{\partial y} + h\frac{\partial f(y,t)}{\partial t} + R_1(\xi,\zeta)$$

mit $\xi \in [y, y + hf(y,t)]$ und $\zeta \in [t, t+h]$. Daraus kann man folgende Approximation an $f'(y,t)$ ablesen:

$$f'(y,t) = \frac{\partial f(y,t)}{\partial y} \cdot f(y,t) + \frac{\partial f(y,t)}{\partial t} \approx \frac{f(y + hf(y,t), t+h) - f(y,t)}{h}.$$

Somit kann man die Lösung der Differentialgleichung $y' = f(y,t)$ als Taylorentwicklung zweiter Ordnung anschreiben als

$$\begin{aligned} y(t+h) &= y(t) + hy'(t) + \frac{h^2}{2}y''(t) + \ldots \\ &\approx y(t) + hf(y,t) + \frac{h^2}{2}\frac{f(y + hf(y,t), t+h) - f(y,t)}{h} \\ &= y(t) + \frac{h}{2}\big(f(y,t) + f(y + hf(y,t), t+h)\big). \end{aligned}$$

Daraus ergibt eine Iterationsvorschrift, die im folgenden Satz zusammengefasst wird.

Satz 5.5 (Runge-Kutta-Verfahren 2. Ordnung) Die Iterationsvorschrift zur Berechnung der Approximationsschritte des Runge-Kutta-Verfahrens 2. Ordnung für das Anfangswertproblems $y'(t) = f(y(t), t)$, $y(t_0) = y_0$ mit Schrittweite h lautet

$$\begin{aligned} t_{i+1} &:= t_i + h \\ k_1 &:= hf(y_i, t_i) \qquad \text{(Hilfsvariable 1)} \\ k_2 &:= hf(y_i + k_1, t_{i+1}) \qquad \text{(Hilfsvariable 2)} \\ y_{i+1} &:= y_i + \frac{1}{2}(k_1 + k_2) \end{aligned} \tag{5.9}$$

Beispiel 5.16 Gegeben sei das Anfangswertproblem

$$y'(t) = -2ty^2(t) \qquad \text{mit} \qquad y(0) = 1.$$

Dieses Problem hat die exakte Lösung $y(t) = \frac{1}{1+t^2}$. Für eine Schrittweite $h = 0.1$ ergeben sich mit der Runge-Kutta Methode (5.9) im ersten Iterationsschritt die Parameter

$$k_1 = 0.1f(1, 0) = 0 \qquad \text{und} \qquad k_2 = 0.1f(1 + k_1, 0.1) = -0.02$$

und somit $w_1 = 1 + \frac{1}{2}(k_1 + k_2) = 0.99$. Die exakte Lösung ist $y(0.1) = 0.990099$. Die weiteren Ergebnisse sind für die ersten fünf Schritte in folgender Tabelle zusammengefasst.

	t					
	0	0.1	0.2	0.3	0.4	0.5
exakt	1.0	0.990099	0.961538	0.917431	0.862069	0.8
Fehler Runge-Kutta	0.0	0.000099	0.000172	0.000185	0.000114	0.000034
Fehler Euler	0.0	0.009900	0.018462	0.024152	0.026320	0.025250

□

Meist wird zur numerischen Lösung von Differentialgleichungen eine Runge-Kutta Methode vierter Ordnung verwendet, deren Herleitung aus der zweidimensionalen Taylorreihenentwicklung bis zum vierten Term allerdings recht kompliziert ist. Wir beschränken uns daher auf ihre Formulierung, und geben direkt die Iterationsvorschrift an.

Satz 5.6 (Runge-Kutta-Verfahren 4. Ordnung) Die Iterationsvorschrift zur Berechnung der Approximationsschritte des Runge-Kutta-Verfahrens 4. Ordnung für das Anfangswertproblems $y'(t) = f(y(t), t)$, $y(t_0) = y_0$ mit Schrittweite h lautet

$$\begin{aligned} t_{i+1} &:= t_i + h \\ k_1 &:= hf(y_i, t_i) && \text{(Hilfsvariable 1)} \\ k_2 &:= hf\Big(y_i + \frac{1}{2}k_1, t_i + \frac{1}{2}h\Big) && \text{(Hilfsvariable 2)} \\ k_3 &:= hf\Big(y_i + \frac{1}{2}k_2, t_i + \frac{1}{2}h\Big) && \text{(Hilfsvariable 3)} \\ k_4 &:= hf(y_i + k_3, t_{i+1}) && \text{(Hilfsvariable 4)} \\ y_{i+1} &:= y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) && (5.10) \end{aligned}$$

Beispiel 5.17 Wir betrachten das Anfangswertproblem

$$y'(t) = \frac{1}{t^2}\big(ty(t) - y^2(t)\big) \qquad \text{mit} \qquad y(1) = 2$$

mit der exakten Lösung $y(t) = \frac{2t}{1+2\log(t)}$. Mit der Runge-Kutta Methode vierter Ordnung (5.10) und einer Schrittweite $h = 0.1$ erhalten wir folgende Approximation an die Lösung, wobei wir zum Vergleich auch die Fehler des Eulerverfahrens und des Runge-Kutta Verfahrens zweiter Ordnung auflisten.

t	exakt	Fehler RK 4. Ord.	Fehler RK 2. Ord.	Fehler Euler
0.0	2.0	0.0	0.0	0.0
0.1	1.847776	0.000009	0.000158	0.047776
0.2	1.758702	0.00001	0.000009	0.062834
0.3	1.705222	0.000009	0.000174	0.067751
0.4	1.673696	0.000009	0.000296	0.068924
0.5	1.656607	0.000008	0.000381	0.068601
0.6	1.649478	0.000008	0.000439	0.067684
0.7	1.649479	0.000007	0.000478	0.06656
0.8	1.654736	0.000007	0.000505	0.065404
0.9	1.663961	0.000007	0.000523	0.064294
1.0	1.676239	0.000006	0.000536	0.063264

□

Weitere Lösungsmöglichkeiten für Differentialgleichungen, mit denen wir uns aber nicht weiter beschäftigen wollen, sind Extrapolationsmethoden (vergleiche dazu Abschnitt 4.3), Multischrittmethoden (bei denen nicht nur t_i und w_i in die Berechnung eingehen, sondern auch Werte mit Indizes $i-1, i-2, \dots$), oder Verfahren, bei denen die Schrittweite h der Differentialgleichung angepasst wird.

Zum Abschluss betrachten wir noch zwei interessante Beispiele von nichtlinearen Differentialgleichungssystemen.

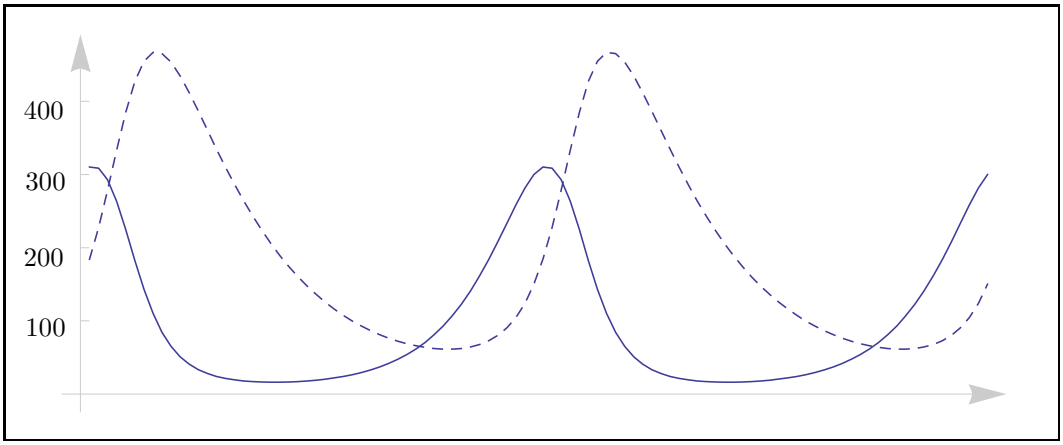

Abbildung 5.13: Zeitliche Entwicklung der Räuberpopulation (gestrichelt) und Beutepopulation (durchgezogen) aus Beispiel 5.18.

Beispiel 5.18 Ein Modell für die Entwicklung zweier Populationen, von denen eine die Jäger und die anderen die Beute sind, ist durch die *Lotka-Volterra* Differentialgleichungen gegeben. Diese Gleichungen lauten für Räuber r, Beute b und drei positive Parameter α, γ und δ

$$r'(t) = -\alpha r(t) + \gamma r(t)b(t)$$
$$b'(t) = \delta b(t) - \gamma r(t)b(t).$$

Dieses Modell beruht auf folgenden theoretischen Überlegungen: Eine Beutepopulation b nimmt ohne Räubereinfluss um den Faktor δ zu; ebenso nimmt die Räuberpopulation r ohne Zugang zu Beute um den Faktor α ab. Die Anzahl des Aufeinandertreffens von Räubern und Beute ist proportional zum Produkt der Populationen; der Faktor γ gibt an, bei welchem Anteil dieser Aufeinandertreffen der Ausgang für die Beute tödlich ist. Vereinfachend wird angenommen, dass jedes tote Beutetier das Entstehen eines neuen Räubers ermöglicht; kompliziertere Modelle können dafür noch einen weiteren Parameter verwenden.

Obiges System ist nur numerisch lösbar und nicht explizit von der Zeit t abhängig, wir wählen daher als Startpunkt $t_0 = 0$. Für die numerische Lösung mittels Runge-Kutta Verfahren vierter Ordnung und den Parameterwerten $\alpha = -1, \gamma = 0.01$ und $\delta = 2$ ergibt sich mit den Anfangsbedingungen $r(0) = 150$ und $b(0) = 300$ das in Abbildung 5.13 dargestellte zyklische Verhalten. □

Beispiel 5.19 Eine mögliche Formulierung des dritten Kepler'schen Gesetzes unter normalisierter Masse und Gravitation (Einheiten so gewählt, dass Masse mal Gravitation eins ist) ist durch folgendes Differentialgleichungssystem zweiter Ordnung gegeben:

$$x''(t) = -\frac{x(t)}{r^3}$$
$$y''(t) = -\frac{y(t)}{r^3},$$

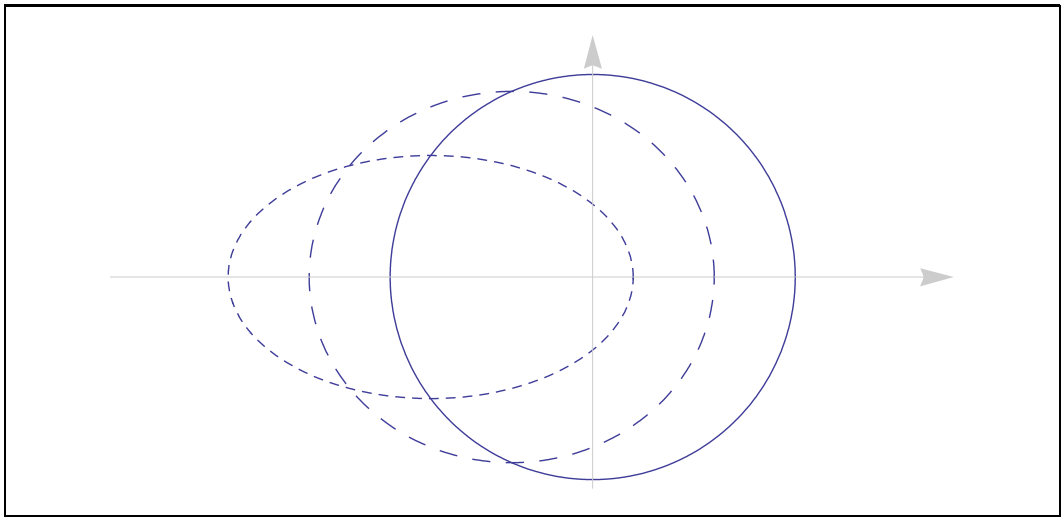

Abbildung 5.14: Die Lösungsmenge $\{x(t), y(t) \mid 0 \leq t \leq 2\pi\}$ des Differentialgleichungssystems aus Beispiel 5.19 für Werte von $e = 0$ (rechts), $e = 0.3$ (Mitte), und $e = 0.8$ (links).

wobei $r = \sqrt{x^2 + y^2}$ den Abstand des Punktes (x, y) vom Ursprung angibt. Dieses System kann als Differentialgleichungssystem erster Ordnung angegeben werden:

$$\begin{aligned} x'(t) &= x_1(t) \\ x_1'(t) &= -\frac{x(t)}{r^3} \\ y'(t) &= y_1(t) \\ y_1'(t) &= -\frac{y(t)}{r^3} \end{aligned}$$

Dieses System kann mit den uns bekannten Mittel numerisch gelöst werden. Für verschiedene Werte von $e \in [0, 1]$ ergeben sich mit den Anfangsbedingungen

$$x(0) = 1 - e, \quad x_1(0) = 0, \quad y(0) = 0, \quad y_1(0) = \sqrt{\frac{1+e}{1-e}}$$

die in Abbildung 5.14 gezeigten Trajektorien der Punkte (x, y) entlang der Lösung dieses Anfangswertproblems. Der Parameter e gibt somit die Exzentrizität der Umlaufbahn an. □

Kapitel 6

Numerische Methoden zur Nullstellenbestimmung

In diesem Kapitel betrachten wir Algorithmen, die Gleichungen nicht exakt, sondern nur näherungsweise lösen. Im Gegensatz zu exakten Algorithmen für begrenzte Anwendungsgebiete (wie z.B. Nullstellenbestimmung bei Polynomen oder das Gauß'sche Eliminationsverfahren für lineare Gleichungssysteme) können diese Algorithmen auf viele Arten von Gleichungen angewandt werden. Wir werden im folgenden Verfahren kennenlernen, mit denen man Nullstellen von sehr allgemeinen Funktionen bestimmen kann. Da das Lösen von Gleichungen ja als das Nullstellenbestimmen von Funktionen aufgefasst werden kann (wenn man die Funktionsdefinition aus der Gleichung erhält), kann man mit diesen Verfahren auch sehr allgemeine Gleichungen lösen.

Um die Methoden in diesem Kapitel anwenden zu können, müssen die untersuchten Funktionen nur wenige Bedingungen—wie etwa Stetigkeit oder Differenzierbarkeit—erfüllen. Alle hier vorgestellten Methoden sind *iterativ*, nähern sich also Lösungen in mehreren Wiederholungen. Dabei wird angenommen, dass der Algorithmus *konvergiert*, sich also auch wirklich einer Nullstelle nähert. Bei ungünstig gewählten Startpunkten kann der Algorithmus aber auch *divergieren*, muss sich also nicht einer Nullstelle annähern. In manchen Fällen (so wie etwa beim Newtonverfahren in Abschnitt 6.3) kann man Einschränkungen angeben, unter denen die Nullstellenbestimmung garantiert konvergiert. Meist verzichtet man aber auf eine Überprüfung dieser Bedingungen, sondern versucht durch verschiedene (mehr oder weniger zufällig gewählte) Startpunkte eine Nullstelle zu finden.

Wir beginnen mit der einfachsten iterativen Methode zur Nullstellenbestimmung von Gleichungen.

6.1 Bisektionsmethode

Die einfachste Methode zur Nullstellenbestimmung ist nichts anderes als eine binäre Suche: Ein Intervall, in dem sich sicher eine Nullstelle befindet, wird solange geteilt, bis es klein genug ist und die Nullstelle somit ausreichend genau bestimmt ist. Bei der Intervallteilung wird dabei diejenige Intervallhälfte verworfen, in der sich die Nullstelle nicht befindet, und mit der anderen weitergearbeitet.

Mathematisch begründet sich diese Methode auf dem Zwischenwertsatz. Dieser besagt, dass es für jeden Punkt ξ im Bildbereich einer stetigen Funktion $f : [a, b] \to \mathbb{R}$ einen Punkt $c \in [a, b]$ mit $f(c) = \xi$ gibt. Wenn nun a und b unterschiedliche Vorzeichen haben, so garantiert dieser Satz mit $\xi = 0$ die Existenz einer Nullstelle in $[a, b]$.

Beispiel 6.1 Durch graphische Inspektion sieht man, dass die Funktion

$$f(x) = 3x^5 + 4x^2 - 9x + 2$$

neben $\xi = 1$ im Intervall $[0, 1]$ eine weitere Nullstelle hat. Eine Iteration des Bisektionsalgorithmus mit den Startwerten $a = 0, b = 1$ liefert folgendes Ergebnis, wobei $p = (a + b)/2$ den Mittelpunkt des Intervalls $[a, b]$ bezeichnet.

n	a_n	p_n	b_n	$f(a_n)$	$f(p_n)$	$f(b_n)$
1	0	0.5	1	2	-1.40625	0
2	0	0.25	0.5	2	0.00293	-1.40625
3	0.25	0.375	0.5	0.00293	-0.79025	-1.40625
4	0.25	0.3125	0.375	0.00293	-0.41293	-0.79025
5	0.25	0.28125	0.3125	0.00293	-0.20956	-0.41293
6	0.25	0.26563	0.28125	0.00293	-0.10443	-0.20956
7	0.25	0.25781	0.26563	0.00293	-0.05103	-0.10443
8	0.25	0.25391	0.25781	0.00293	-0.02412	-0.05102
9	0.25	0.25195	0.25391	0.00293	-0.01061	-0.02412
10	0.25	0.25097	0.25195	0.00293	-0.00384	-0.01061
11	0.25	0.25049	0.25098	0.00293	-0.00046	-0.00384
12	0.25	0.25024	0.25049	0.00293	0.00124	-0.00046
13	0.25024	0.25037	0.25049	0.00124	0.00039	-0.00046

Somit liegt die Nullstelle bei $\xi \approx 0.2503$. □

Da das Intervall, in dem sich die Nullstelle befindet, in jedem Iterationsschritt halbiert wird, ist die Nullstelle n Schritte nach Start des Algorithmus mit Intervall $[a_1, b_1]$ in einem Intervall der Länge $(b_1 - a_1)/2^n$. Es ergeben sich also mehrere Möglichkeiten für ein Abbruchkriterium des Algorithmus, je nachdem, wie man den Begriff der Genauigkeit einer Nullstelle versteht:

- Man bricht ab, wenn der Funktionswert $f(p_n)$ der Approximation p_n klein genug ist, also bei
$$\big|f(p_n)\big| < \varepsilon$$
für gegebenes ε.

- Man bricht ab, wenn die Nullstelle hinreichend genau bekannt ist, wenn also
$$b_n - a_n = \frac{b_1 - a_1}{2^n} < \delta$$
für gegebenes δ.

Die Bisektionsmethode findet garantiert eine Approximation einer Nullstelle, wenn sie mit einem Intervall gestartet wird, in dem sich diese Nullstelle befindet. Diese Methode konvergiert jedoch recht langsam, da sich der Suchbereich bei jedem

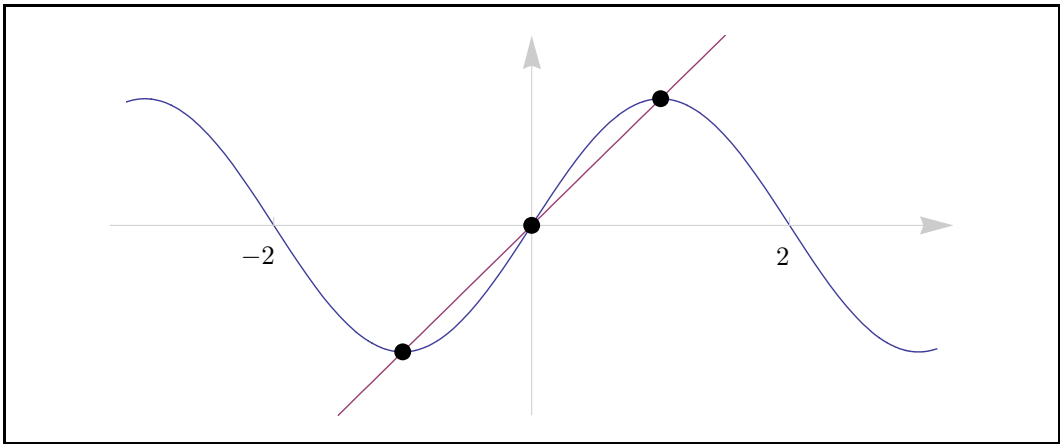

Abbildung 6.1: Die Fixpunkte der Funktion $\sin\left(\frac{\pi}{2}x\right)$.

Iterationsschritt nur um den Faktor 2 verringert. Die folgenden Verfahren sind anspruchsvoller und konvergieren auch schneller.

6.2 Fixpunktiterationen

In diesem Abschnitt wenden wir Verfahren zur Fixpunktbestimmung von Funktionen an, um Nullstellen zu finden. Somit ist zuerst zu klären, was Fixpunkte von Funktionen sind, und wie diese zum Bestimmen von Nullstellen eingesetzt werden können.

Definition 6.1 (Fixpunkt)
Sei M eine beliebige Menge. Ein *Fixpunkt* einer Funktion $f : M \to M$ ist ein Punkt p mit

$$f(p) = p.$$

Fixpunkte einer Funktion bleiben somit unter Anwendung dieser Funktion unverändert.

Im folgenden werden wir nur reelle Funktionen $f : \mathbb{R} \to \mathbb{R}$ betrachten. Eine erste Observation ist, dass die Koordinaten (x, y) eines Fixpunkts wegen $y = f(x) = x$ immer auf der 45°-Diagonalen $y = x$ liegen.

Beispiel 6.2 Die einzige Fixpunkt der Funktion $f(x) = \sin(x)$ liegt bei $x = 0$, da nur im Punkt $(0, 0)$ der Graph der Sinusfunktion die Diagonale schneidet. Wenn man die Periode der Sinusfunktion auf 4 reduziert—also $g(x) = \sin(\frac{\pi}{2}x)$ definiert—so hat diese Funktion die drei Fixpunkte $x = -1$, $x = 0$ und $x = 1$, da an diesen Stellen der Graph von g die Diagonale schneidet. Dies ist in Abbildung 6.1 dargestellt. □

Wenn man nun eine Möglichkeit zur Fixpunktbestimmung von Funktionen zur Verfügung hat (siehe unten), dann kann man auf folgende Weise Nullstellen berechnen: Sei etwa f eine Funktion, für die eine Nullstelle gesucht ist. Dann konstruiert

man eine Funktion g, deren Fixpunkt eine Nullstelle von f ist, etwa durch

$$g(x) = x - f(x) \qquad \text{oder} \qquad g(x) = x + 2f(x).$$

Man sieht leicht, dass für solche Funktionen g aus $g(x) = x$ folgt, dass dann auch $f(x) = 0$ sein muss.

Wir "reduzieren" somit das Problem der Nullstellenbestimmung auf das der Fixpunktbestimmung. Wie wir sehen werden, ist dies in vielen Fällen tatsächlich eine Reduktion der Schwierigkeit. Die theoretische Grundlage der Fixpunktiteration ist folgender Satz.

Satz 6.1 Sei $g : \mathbb{R} \to \mathbb{R}$ eine auf dem Intervall $[a, b]$ stetige Funktion mit $g([a, b]) \subseteq [a, b]$. Dann hat g in $[a, b]$ einen Fixpunkt. Wenn weiters die Ableitung $g'(x)$ in diesem Intervall existiert und es eine positive Konstante $k < 1$ gibt mit

$$\left|g'(x)\right| \leq k \qquad \text{für alle } x \text{ mit } a < x < b,$$

so ist dieser Fixpunkt eindeutig bestimmt.

Der Beweis dieses Satzes ist einfach: Wenn $g(a) = a$ oder $g(b) = b$ ist, dann hat g in diesem Punkt einen Fixpunkt. Wenn nicht, dann ist $g(a) > a$ und $g(b) < b$. Für die Hilfsfunktion $h(x) = g(x) - x$ gilt dann

$$h(a) = g(a) - a > 0 \qquad \text{und} \quad h(b) = g(b) - b < 0.$$

Mit dem Zwischenwertsatz hat dann h in $[a, b]$ eine Nullstelle, und somit g einen Fixpunkt. Der zweite Teil folgt ebenso einfach aus dem Mittelwertsatz.

Beispiel 6.3 Wir betrachten die Funktion $g(x) = \log(x+2)$ (Logarithmus zur Basis e), auf dem Intervall $[0, 2]$. Da der Logarithmus eine monoton wachsende Funktion ist und $g(0) = 0.693147, g(2) = 1.38629$ ist, gilt $g([0, 2]) \subseteq [0, 2]$. Mit Satz 6.1 gibt es somit einen Fixpunkt in diesem Intervall. Da weiters $g'(x) = 1/(x + 2)$ und diese Ableitung monoton fallend ist, haben wir wegen $g'(0) = 0.5$ und $g'(2) = 0.25$ eine Konstante $k < 1$ (etwa $k = 0.75$) mit $|g'(x)| < k$ für alle $x \in [0, 2]$. Somit ist der Fixpunkt in $[0, 2]$ eindeutig bestimmt. □

Nachdem mit den Bedingungen aus Satz 6.1 die Existenz eines Fixpunkts garantiert werden kann ist nur mehr zu klären, wie dieser Fixpunkt gefunden werden kann. Dazu gibt es folgenden Satz.

Satz 6.2 Sei $g : [a, b] \to [a, b]$ eine Funktion, die alle Bedingungen aus Satz 6.1 für die Existenz eines eindeutigen Fixpunkts erfüllt. Dann konvergiert die Folge

$$p_n = g(p_{n-1}) = g^n(p_0)$$

für beliebiges $p_0 \in [a, b]$ gegen den eindeutigen Fixpunkt von g in $[a, b]$.

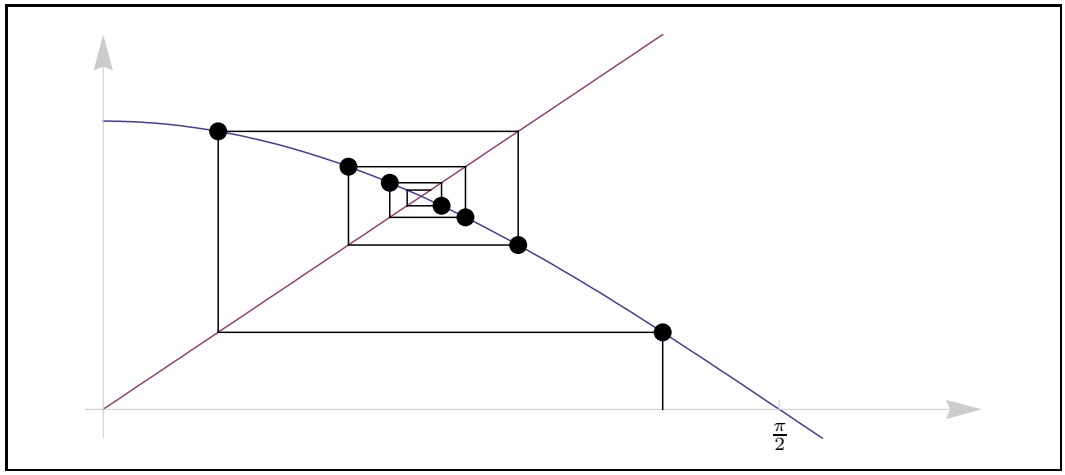

Abbildung 6.2: Iterationspunkte beim Bestimmen des Fixpunktes der Cosinusfunktion für Startwert $p_1 = 1.3$.

Die Erfüllung der Bedingungen dieses Satzes ist *hinreichend*, um die Konverenz des Iterationsverfahrens zu garantieren. Die Erfüllung der Bedingungen ist jedoch nicht *notwendig*; es ist also möglich, mit dem Iterationsverfahren einen Fixpunkt zu finden, obwohl die Funktion die Bedingungen des Satzes nicht erfüllt.

Als Abbruchkriterium für Fixpunktiterationen bietet sich

$$|p_n - p_{n-1}| < \varepsilon$$

für gegebene Konstante $\varepsilon > 0$ an. Allerdings muss man aufpassen, wenn bei den Funktionen sehr kleine Koeffizienten auftreten: Man könnte dann schon beim Startwert das Terminationskriterium erfüllen, falls alle Zahlenwerte kleiner als ε sind. Man verwendet daher oftmals folgendes Kriterium, welchen den relativen Fehler in p_n beschränkt:

$$\left| \frac{p_n - p_{n-1}}{p_n} \right| < \varepsilon.$$

Beispiel 6.4 Mit dem theoretischen Resultat des letzten Satzes können wir nun den eindeutigen Fixpunkt der Funktion $g(x) = \log(x + 2)$ im Intervall $[0, 2]$ bestimmen. Bei einem Startwert von etwa $p_0 = 0.5$ ergibt sich die Approximationfolge

$$\begin{aligned}(p_1, \ldots, p_{12}) = (&0.5, 0.916291, 1.07031, 1.12178, 1.1384, 1.14371, 1.1454,\\ &1.14594, 1.14611, 1.14617, 1.14619, 1.14619, 1.14619)\end{aligned}$$

Für $\varepsilon = 10^{-5}$ ist $|p_{11} - p_{10}| < \varepsilon$ und wir brechen die Berechnung ab. Der Fixpunkt ist somit $p = 1.14619$. □

Beispiel 6.5 Die Funktion $g(x) = \cos(x)$ erfüllt im Intervall $[0, 1.5]$ die Bedingungen von Satz 6.2, da $\cos([0, 1.5]) \subseteq [0, 1]$ ist und die Ableitung $|\cos'| = |-\sin|$ in diesem Intervall immer kleiner als 1 ist. Die Zwischenergebnisse der Fixpunktiteration lassen sich an diesem Beispiel gut illustrieren, wie in Abbildung 6.2 zu sehen ist. Die Konvergenz ist bei diesem Beispiel allerdings sehr langsam; man erhält für den Startwert $p_1 = 1.3$ die Iterationsfolge

$$(p_1, \ldots, p_{31}) = (1.3, 0.267499, 0.964435, 0.569881, 0.841965, 0.665998, 0.7863,$$

$0.706469, 0.760659, 0.724382, 0.748909, 0.732432, 0.74355,$
$0.73607, 0.741113, 0.737718, 0.740006, 0.738465, 0.739503,$
$0.738804, 0.739275, 0.738957, 0.739171, 0.739027, 0.739124,$
$0.739059, 0.739103, 0.739073, 0.739093, 0.73908,$
$0.739089, 0.739083)$

Erst ab diesem Punkt ($n = 31$) ist $|p_n - p_{n-1}| < 10^{-5}$ und ein mögliches Terminationskriterium erfüllt. Der Fixpunkt der Cosinusfunktion in diesem Intervall liegt bei $p = 0.739085$. □

Um jetzt die Methode der Fixpunktiteration zur Nullstellenbestimmung nutzen zu können, muss man sich für gegebene Funktion f, deren Nullstelle man bestimmen möchte, eine Funktion g konstruieren, der Fixpunkt man berechnen kann. Mit Satz 6.1 wissen wir, wie g beschaffen sein muss, um eine Konvergenz zur Nullstelle von f zu garantieren.

Beispiel 6.6 Durch Plotten sieht man, dass die Funktion

$$f(x) = x^3 + 4x^2 - 10$$

im Intervall $[1, 2]$ eine Nullstelle hat. Man kann nun auf verschiedene Arten versuchen, eine Funktion g zu finden, deren Fixpunkt die Nullstelle von f ist. Die einfachste Möglichkeit ist sicher

$$g_1(x) = x - f(x) = -x^3 - 4x^2 + x + 10$$

Somit ergibt sich $g_1'(x) = -3x^2 - 8x + 1$. Man sieht leicht, dass $|g'(x)|$ im Intervall $[1, 2]$ immer größer als 1 ist, somit ist eine Bedingung aus Satz 6.1 nicht erfüllt und die Konvergenz der Fixpunktiteration mit dieser Funktion nicht garantiert.

Eine andere Möglichkeit ergibt sich aus der Umformung der Gleichung $x^3+4x^2-10 = 0$ in $x = \sqrt{10/x - 4x}$. Eine Lösung dieser Gleichung, also ein Fixpunkt von

$$g_2(x) = \sqrt{\frac{10}{x} - 4x},$$

ist dann auch Nullstelle von f. Es ist aber $g_2(1) = 2.44949$ und $g_2(2) = \sqrt{-3}$ nicht einmal eine reelle Zahl; somit ist auch für diese Funktion nicht garantiert, dass die Fixpunktiteration konvergiert.

Man kann die ursprüngliche Nullstellen-Gleichung noch auf viele andere Arten umformen; man erhält z.B. $x = \sqrt{10/(x + 4)}$, also

$$g_3(x) = \sqrt{\frac{10}{x + 4}}.$$

Diese Funktion ist monoton fallend; aus $g_3(1) = 1.41421$ und $g_3(2) = 1.29099$ sieht man, dass die Bedingung $g_3([1, 2]) \subseteq [1, 2]$ erfüllt ist. Weiters erhält man $g_3'(x) = -\sqrt{5/2}(x+4)^{-\frac{3}{2}}$. Diese Ableitung ist monoton wachsend, wegen $g_3'(1) = -0.141421$ und $g_3'(2) = -0.107583$ gibt es ein k (etwa $k = 0.5$), sodass $|g_3'(x)| \leq k$ für alle $x \in [1, 2]$. Somit sind alle Bedingungen von Satz 6.2 erfüllt und die Fixpunktiteration konvergiert garantiert gegen den Fixpunkt von g_3 bzw. die Nullstelle von f.

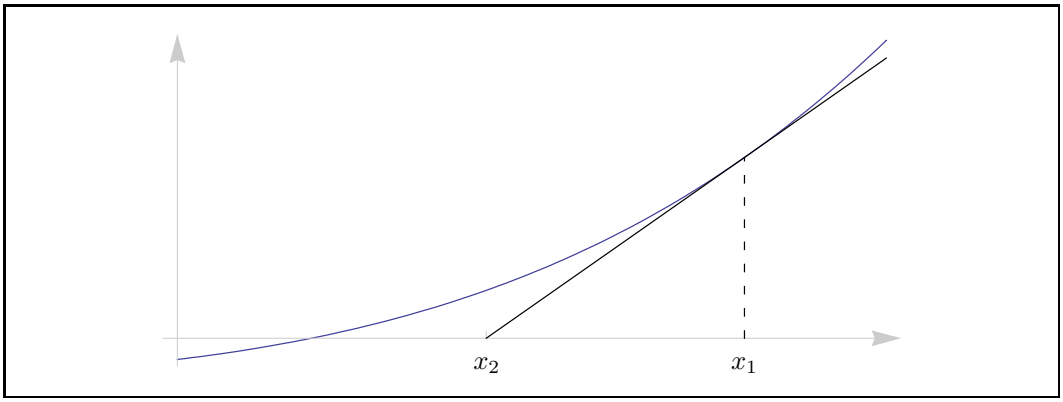

Abbildung 6.3: Illustration des Newtonverfahrens: Der Punkt $x_2 = x_1 - f(x_1)/f'(x_1)$ liegt näher bei der Nullstelle als x_1.

Numerisch ergeben sich für die Fixpunktiterationen mit den Funktionen g_1, g_2 und g_3 von oben folgende Sequenzen p_n, wenn man mit $p_1 = 1.5$ startet:

n	p_n bei g_1	p_n bei g_2	p_n bei g_3
1	1.5	1.5	1.5
2	-0.875	0.81649	1.3484
3	6.73242	2.99691	1.36738
4	-469.72001	$\sqrt{-8.65086}$	1.36496
5	$1.027 \cdot 10^8$		1.36526
6			1.36523
7			1.36523

Man sieht also, dass die Folgen p_n bei Verwendung der Funktionen g_1 und g_2 nicht konvergieren, bei der durch Satz 6.2 garantieren Konvergenz von g_3 natürlich schon. Somit ergibt sich eine Nullstelle von f bei $\xi = 1.36523$. □

6.3 Newtonverfahren

Die bisherigen zwei Verfahren zur Nullstellenbestimmung waren allgemein anwendbar, aber durch diese vielseitige Verwendbarkeit auch nicht besonders schnelle Verfahren: sie konvergieren eher langsam. So benötigt man für die Berechnung des Cosinus-Fixpunkts in Beispiel 6.5 28 Iterationsschritte, um vom Startwert $p_1 = 1.3$ auf ein Ergebnis zu kommen, das weniger als 10^{-5} vom Fixpunkt entfernt liegt.

Das Newtonverfahren konvergiert schneller, ist allerdings nicht auf alle Funktionen anzuwenden. Man benötigt für das Newtonverfahren nämlich die Ableitung der Funktion, deren Nullstelle gesucht ist. Wenn diese Ableitung entweder nicht bekannt ist oder man zur einfachen Evaluierung der Ableitung mehr Aufwand benötigt als zur vielfachen Evaluierung der Funktion selbst, dann muss man (sollte man) auf die anderen Verfahren zurückgreifen. Wenn die Ableitung aber bekannt und einfach zu evaluieren ist, dann ist das Newtonverfahren die beste Methode zur Nullstellenbestimmung.

Das Newtonverfahren kann man aus der Taylorreihenentwicklung sehr einfach herleiten: Wenn wir f um x_1 entwickeln, so erhalten wir als Funktionswert eines

anderen Punktes x_2

$$f(x_2) = f(x_1) + f'(x_1)(x_2 - x_1) + \frac{f''(\xi)}{2}(x_2 - x_1)^2$$

mit ξ zwischen x_1 und x_2. Nach Weglassen des Restglieds können wir obige Gleichung für gegebenes x_1 als lineare Funktion in x_2 betrachten. Zur Berechnung der Nullstelle dieser Funktion setzen wir also nur $f(x_2) = 0$ und erhalten so sofort

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}.$$

Diese Situation ist graphisch in Abbildung 6.3 erklärt.

Diese Gleichung liefert nun eine Iterationsvorschrift für Näherungswerte an die Nullstelle von f:

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$$

Wie bei der Fixpunktiteration ist auch hier die Konvergenz des Verfahrens nicht immer gegeben. Folgender Satz gibt hinreichende Bedingungen zur Konvergenz.

Satz 6.3 Gegeben sei eine zweimal differenzierbare Funktion $f : [a, b] \to \mathbb{R}$. Wenn $\xi \in [a, b]$ eine Nullstelle von f ist mit $f'(\xi) \neq 0$, dann gibt es ein solches $\delta > 0$, dass die von Newtonverfahren erzeugte Folge x_n für jeden Startwert in $[\xi - \delta, \xi + \delta]$ gegen ξ konvergiert.

Dieser Satz sagt also nur aus, dass das Newtonverfahren unter schwachen Bedingungen konvergiert, wenn man nur nahe genug an der Nullstelle beginnt. Wenn man restriktivere Bedingungen an f in einem bestimmten Bereich stellt, kann man sogar zeigen, dass das Newtonverfahren dann für alle Startwerte in diesem Bereich konvergiert.

Beim Newtonverfahren verwendet man meist die gleichen Abbruchsbedingungen wie beim Bisektionsverfahren: Man terminiert den Algorithmus, wenn entweder $|x_n - x_{n-1}| < \delta$ oder $|f(x_n)| < \varepsilon$ ist.

Beispiel 6.7 Wir verwenden das Newtonverfahren, um den numerischen Wert von $\sqrt{2}$ zu berechnen. Gesucht ist also eine Nullstelle der Funktion $f(x) = x^2 - 2$, deren Ableitung $f'(x) = 2x$ ist. Die Iterationsvorschrift des Newtonverfahrens ist also

$$x_n = x_{n-1} - \frac{x_{n-1}^2 - 2}{2x_{n-1}} = \frac{x_{n-1}}{2} + \frac{1}{x_{n-1}}.$$

Angewandt auf den Startwert $x_1 = 2$ ergibt dies die Näherungsfolge

$$(x_1, \ldots, x_5) = 2, 1.5, 1.41667, 1.41422, 1.41421.$$

Ab $n = 4$ gilt bereits $|f(x_n)| < 10^{-5}$, das Verfahren konvergiert also sehr schnell. □

Beispiel 6.8 Wir vergleichen die Geschwindigkeit des Newtonverfahrens mit dem der Fixpunktiteration für die Funktion $f(x) = \cos(x) - x$. Eine Nullstelle dieser Funktion ist somit ein Fixpunkt der Cosinusfunktion; in Beispiel 6.5 haben wir gesehen, dass die Fixpunktiteration nur langsam konvergiert.

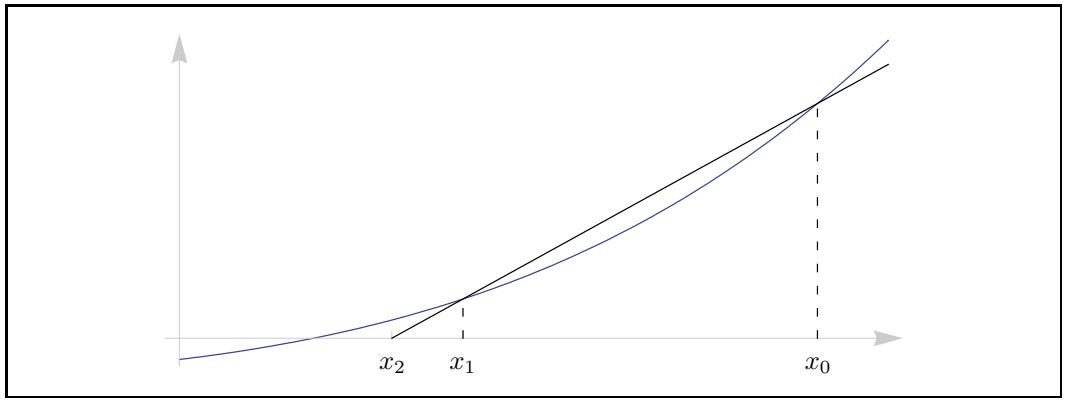

Abbildung 6.4: Illustration des Sekantenverfahrens: Der Punkt x_2, der durch das Sekantenverfahren aus x_0 und x_1 bestimmt wird, liegt näher bei der Nullstelle als x_1.

Für das Newtonverfahren erhalten wir wegen $\cos' = -\sin$ die Iterationsvorschrift

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} = x_{n-1} - \frac{\cos(x_{n-1}) - x_{n-1}}{-\sin(x_{n-1}) - 1}$$

Wir verwenden auch hier wieder den Startwert $x_1 = 1.3$ und erhalten die Sequenz

$$(x_1, \ldots, x_5) = (1.3, 0.774168, 0.739347, 0.739085, 0.739085).$$

Bereits für $n \geq 5$ gilt hier $|x_n - x_{n-1}| < 10^{-5}$. Die Konvergenz ist somit deutlich schneller als bei der Fixpunktiteration. □

Man kann die Vorteile der schnelleren Konvergenz des Newtonverfahrens auch nutzen, wenn die Ableitung der untersuchten Funktion nicht bekannt ist. Wir wissen ja bereits, dass man die Ableitungstangente an einem Punkt x_1 approximieren kann durch die Gerade, die durch die Punkte $(x_1, f(x_1))$ und einen anderen "nahen" Punkt $(x_0, f(x_0))$ verläuft. Wenn man diese Approximation

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

in das Newtonverfahren $x_2 = x_1 - f(x_1)/f'(x_1)$ übernimmt, so erhält man

$$x_2 = x_1 - f(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)}$$

und die allgemeine Iterationsvorschrift

$$x_n = x_{n-1} - f(x_{n-1})\frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})}.$$

Da die Tangente an die Funktion hier durch die Sekante (Gerade durch zwei Punkte einer Funktion) ersetzt wird, nennt man diese Variation des Newtonverfahrens auch *Sekantenverfahren*. Die Funktionsweise dieses Verfahrens ist graphisch in Abbildung 6.4 dargestellt. Es konvergiert zwar etwas langsamer als das Newtonverfahren, und man benötigt auch zwei Startwerte x_1 und x_0, dafür muss man die Ableitung der Funktion aber nicht kennen.

Beispiel 6.9 Wir vergleichen die Konvergenzgeschwindigkeit des Sekantenverfahrens anhand der Funktion $f(x) = \cos(x) - x$; diese Funktion wurde schon in den Beispiel 6.5 und 6.8 behandelt. Mit den Startwerten $x_1 = 1.3, x_0 = 1.7$ erhalten wir die Iterationsfolge

$$(x_0, \ldots, x_6) = (1.7, 1.3, 0.781379, 0.74283, 0.739119, 0.739085, 0.739085).$$

Das Terminationskriterium $|x_n - x_{n-1}| < 10^{-5}$ ist ab $n = 6$ erfüllt; für diese Werte gilt auch $|f(x_n)| < 10^{-5}$. Man sieht, dass bei diesem Beispiel das Sekantenverfahren praktisch gleich schnell wie das Newtonverfahren konvergiert. □

6.4 Mehrdimensionales Newtonverfahren

Das Newtonverfahren aus Abschnitt 6.3 kann verwendet werden, um beliebige nichtlineare Gleichungen der allgemeinen Form $f(x) = g(x)$ zu lösen, indem man diese Aufgabenstellung als Nullstellenbestimmungsproblem für die Funktion $f(x) - g(x)$ betrachtet. In diesem Abschnitt werden wir das Newtonverfahren mit Erkenntnissen aus der linearen Algebra kombinieren, um Lösungen von nichtlinearen Gleichungssystemen zu bestimmen.

Zur Erinnerung: Beim eindimensionalen Newtonverfahren werden Nullstellen einer Funktion $f : \mathbb{R} \to \mathbb{R}$ durch die Iterationsvorschrift

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$$

bestimmt. In der Verallgemeinerung dieser Vorschrift tritt nun eine mehrdimensionale Funktion $f : \mathbb{R}^n \to \mathbb{R}^n$; man benötigt aber auch eine Verallgemeinerung der Ableitung. Da es sich bei f um einen Vektor von n Funktionen mit jeweils n Variablen handelt, kann jede der n Komponenten von f in jede der n Koordinatenrichtungen abgeleitet werden. Die Verallgemeinerung der Ableitung ist somit eine Matrix, in der alle möglichen partiellen Ableitungen zusammengefasst sind. Diese Matrix ist wie folgt definiert.

Definition 6.2 (Jacobimatrix)
Sei $f : \mathbb{R}^m \to \mathbb{R}^n$ eine differenzierbare Funktion. Dann nennt man die Matrix

$$J f = \frac{\partial(f_1, \ldots, f_n)}{\partial(x_1, \ldots, x_m)} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_m} \end{pmatrix}$$

aller partiellen Ableitungen von f die *Jacobi-Matrix* von f.

Für unsere Situation ist $n = m$, und die Jacobi-Matrix ist somit eine quadratische Matrix.

Die Iterationsvorschrift zur Bestimmung der Nullstellen einer mehrdimensionalen Funktion $f : \mathbb{R}^n \to \mathbb{R}^n$, ausgehend von einem Startwert x_0, kann somit über die

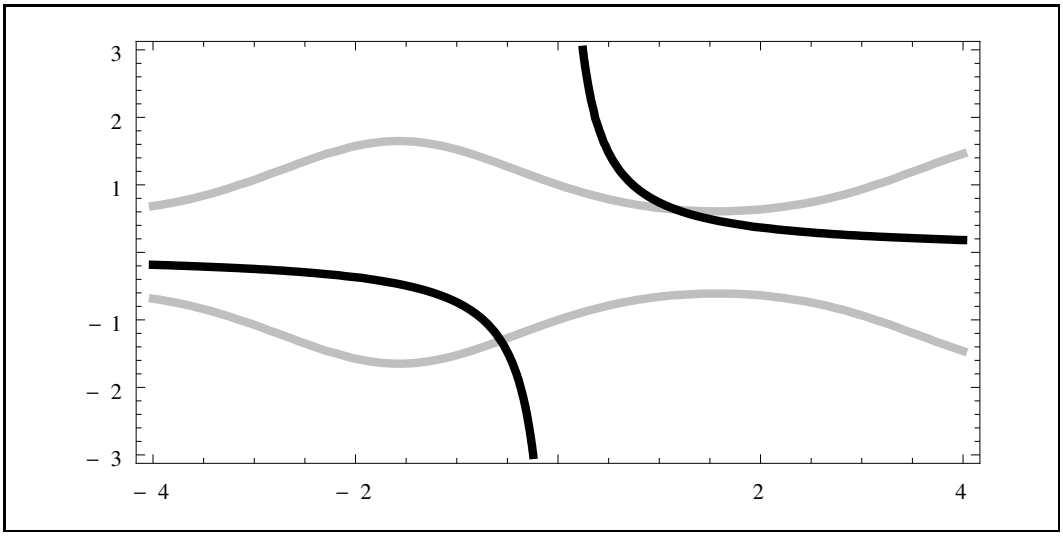

Abbildung 6.5: Illustration zum Beispiel 6.10. Die Lösungen der Gleichung $\log(y^2) = -\sin(x)$ sind grau, die der Gleichung $xy = \cos(xy)$ schwarz dargestellt.

Iterationsvorschrift

$$x_n = x_{n-1} - \big(J\, f(x_{n-1})\big)^{-1}\, f(x_{n-1})$$

berechnet werden. Hierbei ist wie im eindimensionalen Fall zu beachten, dass die Konvergenz nur für Startwerte garantiert werden kann, die ausreichend nahe an der Nullstelle liegen.

Von der Effizienz her sind die Berechnungsschritte beim mehrdimensionalen Newtonverfahren wesentlich aufwändiger als beim eindimensionalen. Dies ist dadurch begründet, dass bei jeder Iteration die Inverse der Jacobi-Matrix zu berechnen ist. Um die Effizienz etwas zu erhöhen, wird das mehrdimensionale Newtonverfahren nicht über Matrixinvertierung berechnet, sondern der Korrekturterm

$$\Delta_{n-1} = \big(J\, f(x_{n-1})\big)^{-1}\, f(x_{n-1})$$

als Lösung des linearen Gleichungssystems

$$J\, f(x_{n-1}) \cdot \Delta_{n-1} = f(x_{n-1})$$

bestimmt (dies ist numerisch günstiger). In technischen Anwendungen, bei denen es auf Geschwindigkeit ankommt, kann man zusätzlich noch auf die genaue Bestimmung der Jacobi-Matrix und anschließendes Gleichungslösen verzichten, indem man Approximationen an die Jacobi-Matrix verwendet, die in jedem Rechenschritt adaptiert werden. Solche Methoden werden als *Quasi-Newton-Methoden* bezeichnet; wir werden hier nicht näher darauf eingehen.

Beispiel 6.10 Zu bestimmen seinen die Lösungen des nichtlinearen Gleichungssystems

$$\begin{aligned} \log(y^2) &= -\sin(x) \\ xy &= \cos(xy). \end{aligned}$$

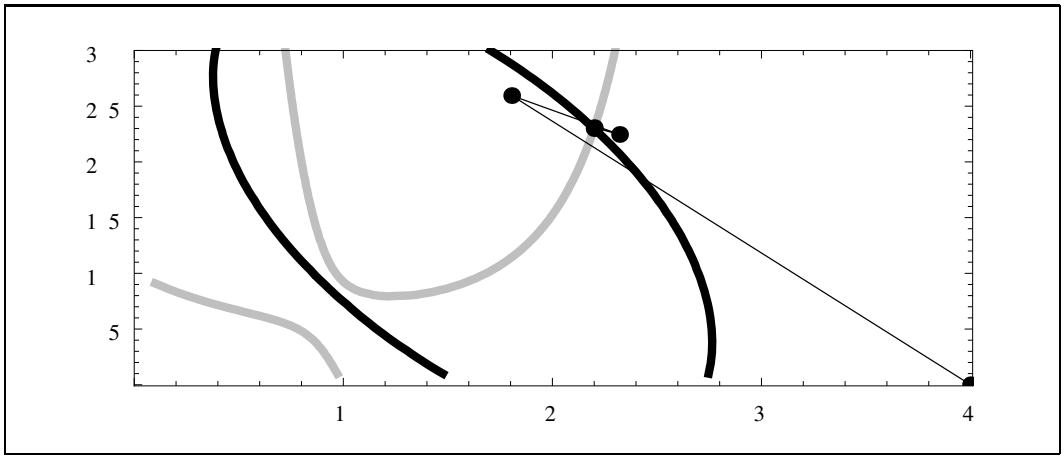

Abbildung 6.6: Illustration zum Beispiel 6.11. Die Lösungen der Gleichung $y\sin^2(x) = \sin(x+y)$ sind grau, die der Gleichung $\exp(\cos(x+y)) = \sin(x)$ schwarz dargestellt.

Die Lösungsmengen der einzelnen Gleichungen sind in Abbildung 6.5 dargestellt. Man kann erkennen, dass diese Lösungsmengen sich in der Nähe der Punkte $(1,1)$ und $(-1,-1)$ schneiden. Diese Punkte können damit als Startwerte für das mehrdimensionale Newtonverfahren dienen.

Wir benötigen für diese Methode noch die Jacobi-Matrix der Funktion, deren Nullstelle zu bestimmen ist. Die Funktion ist durch $f : \mathbb{R}^2 \to \mathbb{R}^2$ mit

$$f\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \log(y^2) + \sin(x) \\ xy - \cos(xy) \end{pmatrix}$$

gegeben; die zugehörige Jacobi-Matrix ist somit

$$J\,f(x,y) = \begin{pmatrix} \cos(x) & \frac{2}{y} \\ y + y\sin(xy) & x + x\sin(xy) \end{pmatrix}.$$

Ausgehend vom Startwert $(1,1)$ ergeben sich die Iterationsschritte

$$(1,1), (1.2344, 0.5159), (1.1951, 0.6172), (1.1715, 0.6306), (1.1715, 0.6309);$$

nach diesen Iterationsschritten ändern sich die ersten vier Hinterkomma-Stellen des Ergebnisses nicht mehr. Ähnliches gilt für den Startwert $(-1,-1)$ mit der Iterationsfolge

$$(-1,-1), (-0.4722, -1.2816), (-0.5699, -1.3107), (-0.5654, -1.3072).$$

Durch Einsetzen kann man verifizieren, dass es sich bei den gefundenen Werten tatsächlich um Lösungen des gegebenen Gleichungssystems handelt. □

Beispiel 6.11 Zu bestimmen seien Lösungen des Gleichungssystems

$$y\sin^2(x) = \log(x+y)$$
$$e^{\cos(x+y)} = \sin(x).$$

Die ersten vier Schritte des mehrdimensionalen Newtonverfahrens mit einem Startwert in der Nähe von $(4,0)$ sind in Abbildung 6.6 zu sehen. Man kann erkennen, dass innerhalb weniger Schritte eine Lösung des Gleichungssystems gefunden wird. □

Kapitel 7

Ausgewählte Kapitel der Optimierung

Im letzten Kapitel haben wir Methoden kennengelernt, mit denen man Nullstellen von Funktionen bestimmen kann. Inhalt dieses Kapitels sind ähnliche iterative Verfahren, mit deren Hilfe man Funktionen *optimieren* kann. Man sucht also nach den Werten, für die eine Funktion einen möglichst großen bzw. kleinen Wert annimmt. Dabei unterscheidet man zwischen *lokalen* und *globalen* Optimierungsverfahren.

Der Zusammenhang zu den bisherigen Themen dieser Vorlesung ergibt sich daraus, dass sich lokale Optimierungsprobleme als Probleme der Nullstellenbestimmung für spezielle Funktionen formulieren lassen können. Wir behandeln auch eine Methode zur Bestimmung von Optima, die zusätzliche *Nebenbedingungen* erfüllen müssen.

Wir beginnen im nächsten Abschnitt mit dem allgemeinen Problem der lokalen Funktionsoptimierung in einer Dimension und erweitern die dabei verwendeten Begriffe in Abschnitt 7.2 auf mehrere Dimensionen. Verfahren zur multidimensionalen Minimierung sind die *Methode des steilsten Abstiegs* (Abschnitt 7.3), die *Quasi-Newton-Verfahren* (Abschnitt 7.4) und die Methode der konjugierten Gradienten (Abschnitt 7.5). Wir betrachten in Abschnitt 7.6 als Anwendung konjugierter Gradienten einen Algorithmus, mit dem lineare Gleichungssysteme numerisch gelöst werden können. Einen weiteren Spezialfall multidimensionaler Optimierungsverfahren stellt die Gauss-Newton Methode dar (Abschnitt 7.7), mit der sich Fehlerquadrat-Problemstellungen lösen lassen. Die Erweiterung dieser Methode zum Verfahren von Levenberg-Marquardt wird in Abschnitt 7.8 präsentiert. Formale Methoden zur Einbeziehung von Nebenbedingungen in das Optimierungsproblem werden in Abschnitt 7.9 behandelt.

Zu guter Letzt werden wir uns in Abschnitt 7.10 dem Gebiet der dynamischen Optimierung zuwenden, die spezielle Probleme rekursiver Natur effizient lösen kann.

7.1 Minimumsuche in einer Dimension

Als Einstieg in die Problematik der Bestimmung von Optima behandeln wir eine einfache Methode in einer Dimension, die ein der Bisektionsmethode zur Nullstellenbestimmung analoges Verfahren ist. Dabei wird, wie bei der Bisektionsmethode, ein Intervall solange verkleinert, bis es das Optimum in gewünschter Genauigkeit enthält. Zur Vereinfachung der Diskussion beschränken wir uns auf die Bestimmung von Minima.

Wie man sich leicht überlegen kann benötigt man drei Punkte (und nicht nur zwei wie beim Bisektionsverfahren), um ein Intervall anzugeben, in dem sich ein Minimum einer Funktion befindet. Wenn es nämlich in einem Intervall $[a, c]$ einen Punkt b gibt mit $f(b) < f(a)$ und $f(b) < f(c)$, dann hat die Funktion f in $[a, c]$ ein Minimum. Die Iterationsvorschrift ist ähnlich dem Bisektionsverfahren: Wir wählen einen neuen Punkt d im *größeren* der beiden Teilintervalle $[a, b]$ und $[b, c]$, damit im nächsten Iterationsschritt das verbleibende Intervall möglichst *klein* wird (es soll also ein möglichst großer Teil verworfen werden). Wir nehmen hier an, dieses Intervall sei $[b, c]$. Es wird später noch genauer darauf eingegangen, wie die Lage des Punktes in $[b, c]$ zu wählen ist. Wir unterscheiden nun zwei Fälle:

- $f(d) < f(b)$: Der Punkt d ist tiefer als der bisherige beste (tiefste) Punkt b, und nimmt dessen Platz ein. Das neue Intervall ist $[b, c]$, der Rest wird verworfen.

- $f(d) > f(b)$: Der Punkt b ist weiterhin der tiefste Punkt, und das Teilintervall zwischen d und c kann verworfen werden. Das neue Intervall ist $[a, d]$.

Durch diese Iterationsvorschrift wird das Intervall solange verkleinert, bis es zum Minimum mit genügender Präzision konvergiert ist.

Die Wahl der optimalen Lage des Punktes d ist etwas subtil. Wir wollen d so wählen, dass das neue Intervall möglichst klein wird, egal ob dies $[b, c]$ oder $[a, d]$ ist. Wir nehmen dazu an, der Punkt b wäre bereits nach dieser Methode im Intervall $[a, c]$ ausgewählt worden. Wir bezeichnen mit f_1 den Anteil des Teilintervalls $[a, b]$ am Gesamtintervall $[a, c]$, also

$$f_1 = \frac{b-a}{c-a}.$$

Der neue Punkt d liegt rechts oder links des Punktes b, je nachdem, ob $[b, c]$ oder $[a, b]$ größer ist, und zwar nocheinmal einen Anteil f_2 neben b:

$$f_2 = \frac{d-b}{c-a}.$$

Man beachte, dass f_2 nicht positiv sein muss. Wir machen eine Fallunterscheidung:

- Fall $f_2 > 0$ (d liegt zwischen b und c): Wie wir oben erläutert haben, ist in diesem Fall das Intervall im nächsten Iterationsschritt entweder $[b, c]$ oder $[a, d]$; diese Intervalle haben (relativ zum aktuellen Intervall $[a, c]$) die Längen $1 - f_1$ bzw. $f_1 + f_2$. Der springende Punkt dieses Verfahrens ist es nun, d möglichst konservativ zu wählen: Da man nicht weiss, ob das nächste Intervall $[b, c]$ oder $[a, d]$ sein wird, sichert man sich ab und wählt d so, dass man in beiden Situationen gleich gut wegkommt. Das bedeutet, dass beide möglichen Intervalle gleich lang sein sollen. Daraus erhält man die Bedingung $f_2 = 1-2f_1$.

- Fall $f_2 < 0$ (d liegt zwischen a und b): Hier sind die möglichen Intervalle im nächsten Iterationsschritt entweder $[a, b]$ oder $[d, c]$; die Längen dieser Intervalle sind f_1 bzw. $1 - (f_1 + f_2)$. Gleichsetzen dieser Längen liefert wiederum $f_2 = 1 - 2f_1$.

Um f_2 ausrechnen zu können benötigt man noch die Überlegung, dass der Punkt b im vorigen Iterationsschritt ebenfalls optimal platziert wurde; die Lage von b in

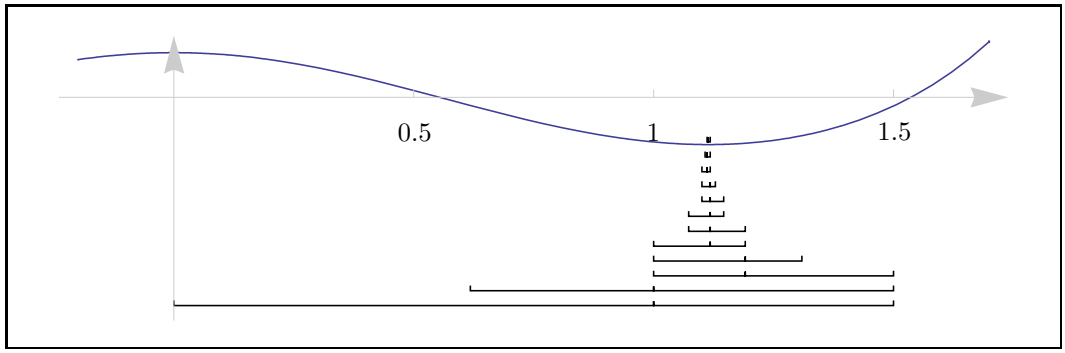

Abbildung 7.1: Graph der Polynomfunktion aus Beispiel 7.1 in der Nähe des Minimums; unter dem Graphen sind die Intervalle der ersten 12 Schritte der goldenen-Schnitt-Regel zur Minimumbestimmung dargestellt.

$[a, c]$ soll also gleich der Lage von d im größeren der Teilintervalle sein. Dies bedeutet

$$\frac{f_2}{1 - f_1} = \frac{f_1}{1} \qquad \text{für } f_2 > 0$$

und zusammen mit der Gleichung $f_2 = 1-2f_1$ erhält man die quadratische Gleichung

$$1 - 2f_1 = f_1(1 - f_1)$$

mit einer Lösung in $[0, 1]$: $f_1 = 0.382$, und $1 - f_1 = 0.618$. Das Verhältnis dieser beiden Zahlen ist der bekannte *goldene Schnitt* $\Phi = \frac{1+\sqrt{5}}{2} = 1.61803$. In diesem Fall ist $f_2 = 0.236$.

Für $f_2 < 0$ erhält man mit der Überlegung

$$\frac{f_1}{1} = \frac{f_1 + f_2}{f_1}$$

die quadratische Gleichung $f_1^2 = 1 - f_1$ mit der Lösung $f_1 = 0.618$ und $f_2 = -0.236$.

Im ersten Fall ($[b, c]$ ist das größere Intervall) liegt d somit einen Anteil $\frac{f_2}{1-f_1} = 0.382$ der Intervalllänge rechts von b; im zweiten Fall ($[a, b]$ ist größer) liegt d einen Anteil $\frac{f_2}{f_1} = -0.382$ links von b.

Wir fassen zusammen: Bei diesem Verfahren zur Minimumsuche ist in jedem Schritt als nächster Punkt derjenige im größeren Teilintervall zu wählen, dessen Distanz zum mittleren Punkt 0.382-mal die Länge dieses Teilintervalls ist. Das Verfahren muss nicht mit drei Punkten gestartet werden, die diese Bedingung erfüllen, sondern liefert auch mit beliebigen Startwerten nach wenigen Iterationsschritten Teilintervalle, deren Längenverhältnisse sich dem goldenen Schnitt nähern.

Durch die Intervallteilung mit der goldenen-Schnitt-Regel hat jedes neue Intervall nur mehr die 0.618-fache Länge des alten Intervalls, und ist damit nicht ganz so schnell wie das Bisektionsverfahren, bei dem jeder Schritt eine Halbierung der Intervalllänge bedeutet.

Beispiel 7.1 Die Polynomfunktion $P(x) = x^7 - 3x^6 + 8x^4 - 12x^2 + 3$ hat in der Nähe von $x = 1$ ein Minimum. Mit den Startwerten $a = 0$, $b = 1$ und $c = 1.5$ und einer Fehlerschranke von 0.01 liefert das eindimensionale Minimierungsverfahren mit der

goldenen-Schnitt-Regel die Iterationsschritte, die in Abbildung 7.1 zu sehen sind. Nach Erreichen des Terminationskriteriums sind $a = 1.11146$, $b = 1.11397$ und $c = 1.11803$ mit einer Intervalllänge von 0.00657. Der korrekte Wert des Minimums ist $x = 1.113997$. □

7.2 Minimumsuche in mehreren Dimensionen

Zur Bestimmung der Optima reellwertiger Funktionen gibt es eine einfache (allerdings nicht hinreichende, sondern nur notwendige!) Bedingung. Dafür muss zuerst natürlich festgelegt werden, wie Optima definiert sind.

Definition 7.1 (Optima)
Sei $f : G \subseteq \mathbb{R}^n \to \mathbb{R}$ eine reellwertige Funktion, die auf einer Menge G definiert ist. Wenn es einen Punkt x_0 gibt und eine Umgebung $U(x_0)$, sodass gilt

$$f(x_0) \geq f(x) \qquad \text{für alle } x \in U(x_0),$$

dann nennt man x_0 *lokales Maximum* von f. Wenn obige Bedingung für alle $x \in G$ (und nicht nur um x_0 herum) gilt, so heißt x_0 *globales Maximum*. Die Definition für lokale und globale Minima ist analog. *Optima* ist ein Sammelbegriff für Minima und Maxima.

Der in obiger Definition verwendete Begriff der *Umgebung um* x_0 ist für gegebenes $\epsilon > 0$ definiert als

$$U(x_0) = \left\{ x \in \mathbb{R}^n \middle| \|x - x_0\| < \epsilon \right\}.$$

Satz 7.1 Sei $f : \mathbb{R} \to \mathbb{R}$ eine stetig differenzierbare Funktion. Wenn f an der Stelle x_0 ein Optimum hat, dann ist $f'(x_0) = 0$.

Die Umkehrung dieses Satzes gilt nicht, wie man etwa am Beispiel $f(x) = x^3$ sehen kann. Punkte, bei denen zwar die erste Ableitung null ist, die aber trotzdem keine Optima sind, nennt man *Sattelpunkte*. Für uns interessant wird die Verallgemeinerung dieses Konzepts auf höherdimensionale Funktionen.

Da wir im Folgenden an manchen Stellen zwischen Zeilen- und Spaltenvektoren unterscheiden müssen, seien von nun an alle Vektoren *Spaltenvektoren*.

Definition 7.2 (Gradient)
Sei $f : \mathbb{R}^n \to \mathbb{R}$ eine stetig differenzierbare Funktion. Dann bezeichnet man den Vektor der partiellen Ableitungen von f als *Gradient* von f und schreibt dafür

$$\text{grad} f(x) = \left(\frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_k}(x) \right)^T.$$

Beispiel 7.2 Für $f : \mathbb{R}^3 \to \mathbb{R}$ mit $f(x, y, z) = \sin(xy) + \cos(xz)$ ist

$$\mathrm{grad} f(x) = \big(y\cos(xy) - z\sin(xz), x\cos(xy), -x\sin(xz)\big)^T. \qquad \square$$

Für höherdimensionale Funktionen gilt eine Verallgemeinerung von Satz 7.1.

Satz 7.2 Sei $f : \mathbb{R}^n \to \mathbb{R}$ eine stetig differenzierbare Funktion, die im Punkt x_0 ein Optimum hat. Dann ist $\mathrm{grad} f(x_0) = 0$.

Um zu überprüfen, ob ein Optimum ein Maximum oder Minimum ist, benötigt man eine Verallgemeinerung der zweiten Ableitung für den mehrdimensionalen Fall.

Definition 7.3 (Hesse-Matrix)
Sei $f : \mathbb{R}^n \to \mathbb{R}$ eine zweimal stetig differenzierbare Funktion. Dann bezeichnet man die quadratische Matrix

$$\mathrm{Hess} f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

der zweifachen partiellen Ableitungen von f als *Hesse-Matrix* von f.

Da die Reihenfolge des Ableitens in die einzelnen Koordinatenrichtungen keine Rolle spielt (also $\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}$ gilt), ist die Hesse-Matrix immer symmetrisch.

Beispiel 7.3 Die Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ mit $f(x, y) = \sin(x\,y)$ hat die Hesse-Matrix

$$\mathrm{Hess} f = \begin{pmatrix} -y^2 \sin(x\,y) & \cos(x\,y) - x\,y\,\sin(x\,y) \\ \cos(x\,y) - x\,y\,\sin(x\,y) & -x^2 \sin(x\,y) \end{pmatrix}. \qquad \square$$

In Analogie zum eindimensionalen Fall benötigt man noch ein Kriterium, ob die zweite Ableitung im Optimum positiv oder negativ ist. Dies ist für Matrizen über die Eigenschaft der positiven Definitheit gegeben, die wie folgt definiert ist.

Definition 7.4 (positiv definit)
Eine symmetrische $n \times n$ Matrix A heißt *positiv definit*, wenn alle Eigenwerte von A positiv sind.

Völlig analog dazu kann man *negativ definit* für symmetrische Matrizen über die Negativität der Eigenwerte definieren.

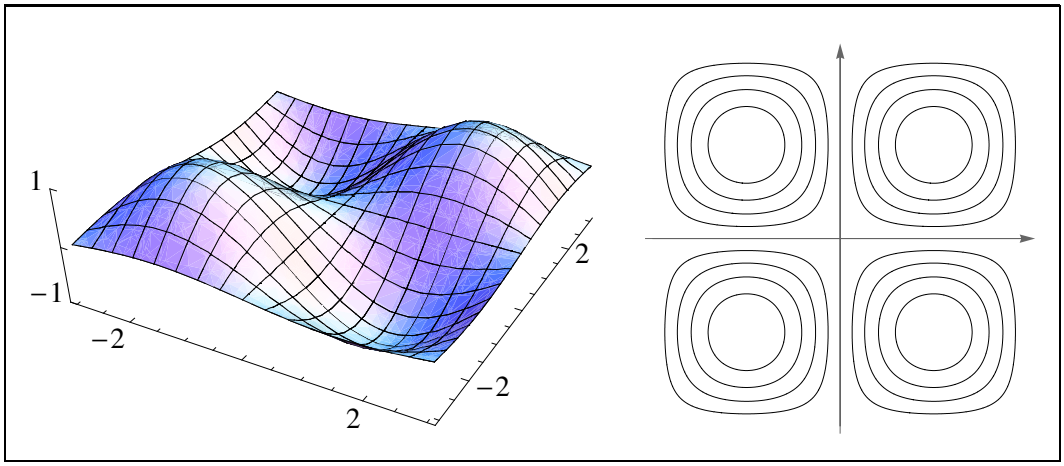

Abbildung 7.2: Die Funktion $f(x,y) = \sin(x)\sin(y)$ aus Beispiel 7.4 als 3D-Plot (links) und von oben mit Isoniveaulinien gleicher z-Werte (rechts). Der Punkt $(0,0)$ ist ein Sattelpunkt, während die Punkte $\left(\pm\frac{\pi}{2}, \pm\frac{\pi}{2}\right)$ Optima sind.

Satz 7.3 Sei $f : \mathbb{R}^n \to \mathbb{R}$ eine zweimal stetig differenzierbare Funktion, und x_0 ein Punkt mit $\mathrm{grad} f(x_0) = 0$. Wenn $\mathrm{Hess} f(x_0)$ positiv definit ist, dann besitzt f in x_0 ein Minimum.

Ebenso gilt, dass f in einem Punkt x_0 mit $\mathrm{grad} f(x_0) = 0$ ein Maximum hat, wenn die Hesse-Matrix an dieser Stelle negativ definit ist. Wenn die Hesse-Matrix in x_0 sowohl positive als auch negative Eigenwerte besitzt, so hat f in in x_0 einen Sattelpunkt.

Beispiel 7.4 Die Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ mit $f(x,y) = \sin(x)\sin(y)$ hat den Gradienten

$$\mathrm{grad} f(x,y) = \big(\cos(x)\sin(y), \sin(x)\cos(y)\big)^T.$$

Zur Bestimmung der Optima kann man das nichtlineare Gleichungssystem

$$\begin{aligned} \cos(x)\sin(y) &= 0 \\ \sin(x)\cos(y) &= 0 \end{aligned}$$

lösen. Wegen $\cos(\frac{\pi}{2} + k\pi) = 0$ und $\sin(k\pi) = 0$ für $k \in \mathbb{Z}$ sind alle Punkte der Form $(k\frac{\pi}{2}, m\frac{\pi}{2})$ mit $k, m \in \mathbb{Z}$ Lösungen dieses Gleichungssystems. Damit ist auch der Ursprung $(0,0)$ eine Lösung des Gleichungssystems, nicht aber ein Optimum: f hat an dieser Stelle "nur" einen Sattelpunkt, wie man in Abbildung 7.2 sehen kann.

Für eine genauere numerische Analyse benötigen wir die Hesse-Matrix von f. Diese ist

$$\mathrm{Hess} f = \begin{pmatrix} -\sin(x)\sin(y) & \cos(x)\cos(y) \\ \cos(x)\cos(y) & -\sin(x)\sin(y) \end{pmatrix}.$$

Im Ursprung ist

$$\mathrm{Hess} f(0,0) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

mit charakteristischem Polynom x^2-1 und Eigenwerten 1 und -1. Damit hat f im Ursprung weder ein Minimum noch ein Maximum.

Laut Abbildung 7.2 hat f im dargestellten Bereich in den Punkten $\left(\frac{\pi}{2},-\frac{\pi}{2}\right)$ und $\left(-\frac{\pi}{2},\frac{\pi}{2}\right)$ Minima, in $\left(-\frac{\pi}{2},-\frac{\pi}{2}\right)$ und $\left(\frac{\pi}{2},\frac{\pi}{2}\right)$ hingegen Maxima. In den ersten beiden Fällen ist die Hesse-Matrix

$$\mathrm{Hess}f\left(\frac{\pi}{2},-\frac{\pi}{2}\right)=\mathrm{Hess}f\left(-\frac{\pi}{2},\frac{\pi}{2}\right)=\begin{pmatrix}1&0\\0&1\end{pmatrix}$$

mit charakteristischem Polynom $(1-x)^2$ und zweifachem Eigenwert 1; damit hat f an diesen Stellen Minima.

In den zweiten beiden Fällen erhalten wir

$$\mathrm{Hess}f\left(-\frac{\pi}{2},-\frac{\pi}{2}\right)=\mathrm{Hess}f\left(\frac{\pi}{2},\frac{\pi}{2}\right)=\begin{pmatrix}-1&0\\0&-1\end{pmatrix}$$

mit zweifachem Eigenwert -1. Diese Punkte sind somit Maxima von f. □

Beispiel 7.5 Zu bestimmen seien alle Minima und Maxima der Funktion $f(x,y)=e^{-(x^2+y^2)}$. Nullsetzen des Gradienten

$$\mathrm{grad}f(x,y)=\left(-2xe^{-(x^2+y^2)},-2ye^{-(x^2+y^2)}\right)^T.$$

liefert sofort $x=0$ und $y=0$, da die Exponentialfunktion nirgends null wird. Die Hesse-Matrix von f ist

$$\mathrm{Hess}f=\begin{pmatrix}2(2x^2-1)e^{-(x^2+y^2)}&4x\,y\,e^{-(x^2+y^2)}\\4x\,y\,e^{-(x^2+y^2)}&2(2y^2-1)e^{-(x^2+y^2)}\end{pmatrix},$$

und im Ursprung ist

$$\mathrm{Hess}f(0,0)=\begin{pmatrix}-2&0\\0&-2\end{pmatrix}$$

mit dem zweifachen Eigenwert -2. Somit besitzt f im Ursprung ein globales Maximum. □

Da das Lösen des nichtlinearen Gleichungssystems $\mathrm{grad}f=0$ für nur wenige interessante Funktionen f exakt durchführbar ist, verwendet man in der Praxis meist iterative Verfahren. Diese Verfahren zur Optimierung mehrdimensionaler Funktionen suchen nach Punkten, an denen der Gradient null wird. Damit ist nicht garantiert, dass das Verfahren in einem Optimum terminiert (und nicht nur an einem Sattelpunkt). Selbst wenn ein Optimum gefunden wird, muss dies noch nicht das globale Optimum sein. Hier muss man sich durch Plotten der Funktion und geeignete Startwerte der Verfahren helfen.

Im Weiteren werden wir nur das Suchen von Minima behandeln. Da das Maximum einer Funktion f das Minimum der Funktion $-f$ ist, stellt dies keine wirkliche Einschränkung dar.

7.3 Methode des steilsten Abstiegs

Die einfachste Suchmethode für Minima lässt sich wie folgt herleiten: Man kann nachrechnen, dass für die Ableitung von f in Richtung eines Vektors v mit $\|v\|=1$

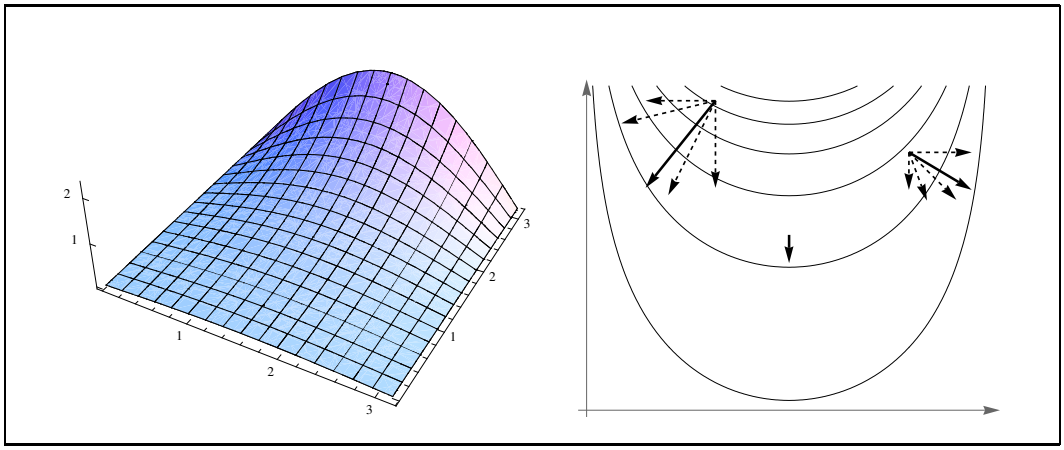

Abbildung 7.3: Die Funktion $f(x, y) = \sin(x)e^y$ als 3D-Plot (links) und als Isoniveaulinien-Plot (rechts). Die dicken Pfeile geben Richtung und Länge der negativen Gradienten an; zum Vergleich dazu sind die dünnen, gestrichelten Pfeile die Ableitungen in andere Richtungen.

gilt (siehe auch Definition 5.1 in Abschnitt 5.5):

$$\frac{\partial f(x)}{\partial v} = v^T \cdot \mathrm{grad} f(x),$$

Dann ist

$$\left|\frac{\partial f(x)}{\partial v}\right| = \left|v^T \cdot \mathrm{grad} f(x)\right| \leq \|v\| \|\mathrm{grad} f(x)\| = \|\mathrm{grad} f(x)\|.$$

Die hier verwendete Ungleichung ist die Cauchy-Schwartz'sche Ungleichung. Somit ist die Ableitung immer kleiner oder gleich der Länge des Gradienten. Man kann in obiger Abschätzung aber Gleichheit erreichen: für den Fall $v_0 = \mathrm{grad} f(x)/\|\mathrm{grad} f(x)\|$ gilt $\|v_0\| = 1$ und

$$\left|\frac{\partial f(x)}{\partial v_0}\right| = \frac{\mathrm{grad} f(x)^T \cdot \mathrm{grad} f(x)}{\|\mathrm{grad} f(x)\|} = \|\mathrm{grad} f(x)\|.$$

Somit ist der größte Anstieg in Richtung des Gradienten gegeben; folgender Satz folgt direkt daraus.

Satz 7.4 Sei $f : \mathbb{R}^n \to \mathbb{R}$ eine reellwertige Funktion und $x_0 \in \mathbb{R}^n$ ein Punkt mit $\mathrm{grad} f(x_0) \neq 0$. Dann zeigt $-\mathrm{grad} f(x_0)$ in Richtung des steilsten Abstiegs von f im Punkt x_0.

Anschaulich gesagt besagt der Satz, dass in einer kleinen Umgebung von x_0 die Funktion in Richtung $-\mathrm{grad} f$ schneller abnimmt als in irgendeiner anderen Richtung. Dies wird auch in Abbildung 7.3 illustriert. Man sieht, wie die negativen Gradienten in die Richtung des steilsten Abstiegs zeigen. Die Gradienten sind länger, wenn der Abstieg steil ist, und kürzer, wenn der Graph der Funktion flach ist.

In dieser Abbildung ist auch zu sehen, dass die Gradienten senkrecht auf die Isoniveaulinien einer Funktion stehen.

Beispiel 7.6 Für die Funktion $f(x,y) = \sin(x)e^y$ aus Abbildung 7.3 ist der Gradient

$$\text{grad} f = \left(\cos(x)e^y, \sin(x)e^y\right)^T.$$

Die Isoniveaulinien sind die "Höhenlinien" der Funktion, sie verbinden also in der xy-Ebene Punkte mit gleichem z-Wert. Somit lautet die allgemeine Gleichung einer Isoniveaulinie $\sin(x)e^y = c$. Wenn wir diese Gleichung nach y auflösen, erhalten wir die Gleichung einer Kurve in der xy-Ebene:

$$y = \log\left(\frac{c}{\sin(x)}\right).$$

Man kann nun nachrechnen, dass der Gradient senkrecht auf diese Kurve steht. Dazu berechnen wir die Ableitung dieser Kurve nach x und erhalten mit den Rechenregeln $\log'(x) = \frac{1}{x}$ und $\sin'(x) = \cos(x)$

$$y' = -\frac{\sin(x)}{c} c \frac{\cos(x)}{\sin^2(x)} = -\frac{\cos(x)}{\sin(x)}.$$

Der Richtungsvektor der Tangente ist $(1, y')$, den Gradienten haben wir oben schon berechnet. Wir erhalten für das Produkt dieser beiden Vektoren

$$(1, y') \cdot \begin{pmatrix}\cos(x)e^y \\ \sin(x)e^y\end{pmatrix} = \cos(x)e^y - \frac{\cos(x)}{\sin(x)}\sin(x)e^y = 0. \qquad \square$$

Diese Tatsache, die wir nun sowohl graphisch als auch anhand eines Beispiels numerisch überprüft haben, gilt auch allgemein.

Satz 7.5 Der Gradient einer Funktion steht immer senkrecht auf die Isoniveaulinien dieser Funktion.

Man kann die Tatsache, dass der Gradient immer in Richtung des steilsten Abstiegs zeigt verwenden, um ein einfaches Suchprogramm zum Finden von Minima zu entwickeln. Dabei geht man von einem Startwert x_1 aus und iteriert folgendermaßen:

$$x_k = x_{k-1} - \eta_k \text{grad} f(x_{k-1}).$$

Verfahren, die auf dieser Iteration basieren, nennt man *Gradientenabstiegsverfahren*. Dabei bestimmen die Parameter η_k (die *Schrittweiten*), wie weit man sich in Richtung des steilsten Abstiegs bewegt.

Als mögliche Kriterien für die Terminierung von Gradientenabstiegsverfahren bieten sich die Überprüfungen

$$\left|f(x_k) - f(x_{k-1})\right| < \varepsilon \quad \text{oder} \quad \left|\frac{f(x_k) - f(x_{k-1})}{f(x_k)}\right| < \varepsilon$$

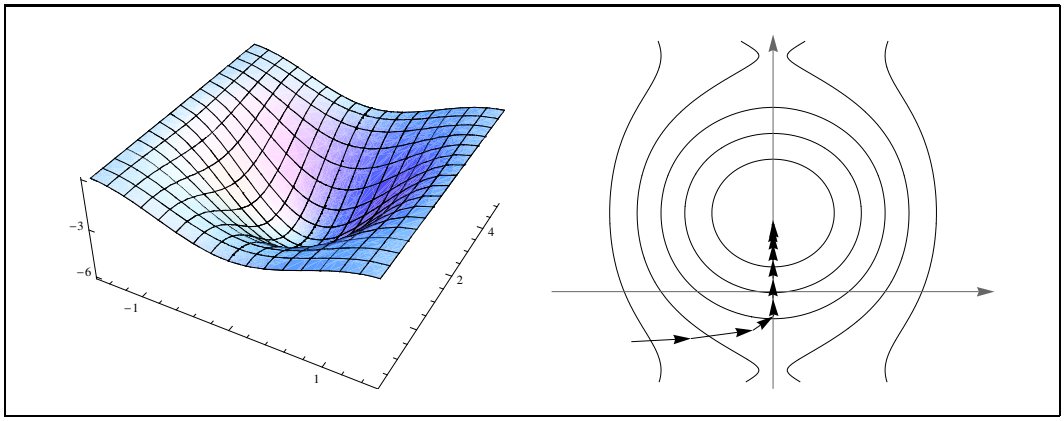

Abbildung 7.4: Die Funktion $f(x,y) = -(2+\sin(y))(1+\cos(2x))$ aus Beispiel 7.7. Rechts sind die ersten 10 Iterationsschritte des Gradientenabstiegsverfahrens mit Schrittweite $\eta = 0.2$ zu sehen.

auf absolute bzw. relative Änderung in den Funktionswerten von f an. Alternativ dazu kann man auch abbrechen, wenn die Länge des Gradienten unter eine gegebene Schranke fällt, also bei

$$\|\mathrm{grad} f(x_k)\| < \varepsilon.$$

Die Konvergenz dieses Verfahrens zu einem Punkt mit $\mathrm{grad} f = 0$ ist allerdings nur asymptotisch für spezielle Folgen η_k garantiert. Einfacher ist es, einen fixen Wert η zu wählen und dann zu überprüfen, ob das Verfahren mit dieser Schrittweite konvergiert.

Beispiel 7.7 Gegeben sei die Funktion $f(x,y) = -(2+\sin(y))(1+\cos(2x))$ mit dem Gradienten

$$\mathrm{grad} f = \big(-2\sin(2x)(-2-\sin(y)), -(1+\cos(2x))\cos(y)\big)^T.$$

Das Gradientenabstiegsverfahrens mit dem Startpunkt $(-1,-1)$ und einer konstanten Schrittweite von $\eta = 0.2$ führt zu folgenden Werten.

k	x_k	y_k	$f(x_k,y_k)$
1	-1.0	-1.0	-0.676411
2	-0.578621	-0.936909	-1.67421
3	-0.141185	-0.770849	-2.5549
⋮			
11	-0.000907401	1.4319	-5.98073
12	0.00246948	1.5407	-5.99906
13	-0.00345636	1.55274	-5.9996

An dieser Stelle ist das Terminationskriterium $|f(x_k,y_k) - f(x_{k-1},y_{k-1})| < 10^{-3}$ erfüllt. Die ersten 10 Iterationsschritte dieser Sequenz sind in Abbildung 7.4 graphisch dargestellt. □

Das letzte Beispiel ist mit dem Gradientenabstiegsverfahren gut zu lösen gewesen; dies ist nicht immer der Fall. Speziell die richtige Wahl der Schrittweite η ist nicht

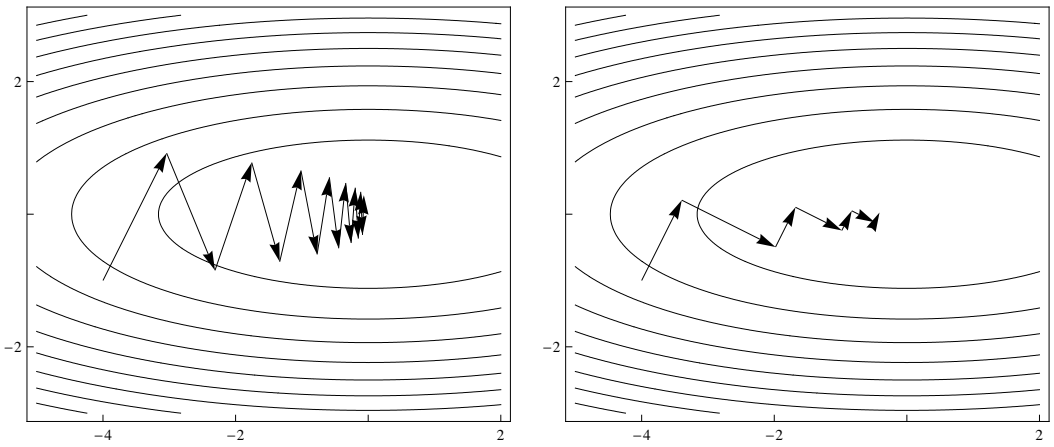

Abbildung 7.5: Die Isoniveaulinien der Funktion $f(x, y) = x^2 + 8y^2$ aus Beispiel 7.8. Zu sehen sind links die ersten Iterationsschritte des Gradientenabstiegsverfahrens mit Startpunkt $(-4, -1)$ mit einer Schrittweite $\eta = 0.12$, rechts die ersten sieben Schritte bei optimaler Schrittweitenanpassung.

immer einfach. Meist müssen mehrere Werte durchprobiert werden, bis das Verfahren konvergiert. Alternativ dazu (aber mit mehr Aufwand) kann man die eindimensionale Minimierung aus Abschnitt 7.1 anwenden, um die optimale Schrittweite in Richtung des negativen Gradienten zu bestimmen.

Beispiel 7.8 Zu minimieren sei die quadratische Funktion

$$f(x, y) = x^2 + 8y^2,$$

die in y-Richtung viel schmäler ist als in x-Richtung. Der Gradient dieser Funktion ist $\text{grad} f(x, y) = (2x, 16y)^T$. In Abbildung 7.5 kann man erkennen, wie das Gradientenabstiegsverfahren zwischen den Seiten der Funktion hin- und herspringt. Mit der gezeigten Schrittweite konvergiert das Verfahren zum globalen Minimum $(0, 0)$; mit einer etwas größeren Schrittweite divergiert es. Mit der optimalen Schrittweitenanpassung durch Minimumsuche in Richtung des negativen Gradienten sind weniger Iterationen notwendig. Man beachte, dass bei der optimalen Schrittweite die negativen Gradienten in aufeinanderfolgenden Schritten senkrecht aufeinander stehen.□

7.4 Quasi-Newton Optimierung

Wenn neben dem Gradienten noch Informationen über die zweite Ableitung einer Funktion zur Verfügung stehen, können effizientere Optimierungsalgorithmen als der im letzten Abschnitt behandelte Gradientenabstieg entwickelt werden. Der Grundgedanke der Quasi-Newton-Methoden ist es, eine Funktion durch ein quadratisches Taylorpolynom zu approximieren; in mehreren Dimensionen (und in Matrixschreibweise) lässt sich diese Funktion als Polynom um einen Punkt x_k folgendermaßen entwickeln (vergleiche auch Satz 1.3 in Abschnitt 1.2 und Satz 5.4 in Abschnitt 5.6):

$f(x) \approx$

$$P_{x_k}(x) = f(x_k) + \mathrm{grad} f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T \mathrm{Hess} f(x_k)(x - x_k). \quad (7.1)$$

Die Idee von Quasi-Newton-Methoden ist nun die folgende: Wenn in der Nähe des Punktes x_k die Funktion f durch eine quadratische Polynomfunktion P_{x_k} approximiert werden kann, dann liegt das eindeutige Minimum von P_{x_k} näher beim Minimum von f als x_k. Im nächsten Iterationsschritt wird somit x_k durch das Minimum von P_{x_k} ersetzt. Da P_{x_k} nur eine Approximation von f ist, wird an dieser Stelle noch nicht das echte Minimum von f zu finden sein, und diese Schritte werden nochmals durchlaufen.

Das eindeutige Minimum von P_{x_k} ist derjenige Punkt, an dem der Gradient von P_{x_k} null wird. Man kann nachrechnen, dass

$$\mathrm{grad} P_{x_k}(x) = \mathrm{grad} f(x_k) + \mathrm{Hess} f(x_k)(x - x_k) \quad (7.2)$$

ist. Nullsetzen der rechten Seite und Auflösen nach x liefert als Minimum den Punkt

$$x = x_k - \mathrm{Hess} f(x_k)^{-1} \mathrm{grad} f(x_k). \quad (7.3)$$

Diese Gleichung kann man auch (etwas wenig formal korrekt) erhalten, indem man die Newton-Methode zur Bestimmung von Nullstellen (siehe Abschnitt 6.3), also die Iterationsvorschrift

$$x = x_k - \frac{f(x_k)}{f'(x_k)},$$

nicht zur Nullstellenbestimmung einer Funktion, sondern zur Nullstellenbestimmung der *Ableitung* einer Funktion verwendet. Ersetzen von $f(x_k)$ durch $\mathrm{grad} f(x_k)$ und $f'(x_k)$ durch $\mathrm{Hess} f(x_k)$ liefert Gleichung (7.3).

Die aus dieser Gleichung ablesbare Iterationsvorschrift lautet also

$$x_{k+1} = x_k - \lambda_k \, \mathrm{Hess} f(x_k)^{-1} \mathrm{grad} f(x_k), \quad (7.4)$$

wobei durch $-\big(\mathrm{Hess} f(x_k)\big)^{-1} \mathrm{grad} f(x_k)$ eine Richtung vorgegeben wird. Wie beim Gradientenabstiegsverfahren muss noch festgelegt werden, *wie weit* der Schritt in diese Richtung ausgeführt werden soll. Zur Bestimmung des Schrittweitenparameters λ_k bietet sich das eindimensionale Minimierungsverfahren aus Abschnitt 7.1 an.

Für die meisten reellen Probleme ist das Bestimmen und Invertieren von $\mathrm{Hess} f(x_k)$ zu aufwändig. Stattdessen approximiert man $\big(\mathrm{Hess} f(x_k)\big)^{-1}$ iterativ; diese Approximationen geben *Quasi-Newton-Methoden* ihren Namen. Wir werden im Folgenden nicht alle Details einer genauen Herleitung präsentieren, sondern uns auf eine Motivation der Formeln beschränken. Das Ziel dieser Formeln wird sein, eine Folge von Matrizen H_k mit

$$\lim_{k \to \infty} H_k = \mathrm{Hess} f(x_k)^{-1}$$

zu konstruieren. Zur Vereinfachung der Berechnungen wird noch gefordert, dass sich die jeweils nächste Matrix H_{k+1} aus H_k durch einen additiven *Korrekturterm* ergibt:

$$H_{k+1} = H_k + \Delta H_k$$

Zur Herleitung von H_{k+1} verwendet man die Bedingung, dass die quadratische Approximation $P_{x_{k+1}}$ aus Gleichung (7.1) im Iterationsschritt $k+1$ an den Iterationspunkten x_k und x_{k+1} denselben Gradienten wie die zu minimierende Funktion haben soll, dass also

$$\mathrm{grad}P_{x_{k+1}}(x_k) = \mathrm{grad}f(x_k) \quad \text{und} \quad \mathrm{grad}P_{x_{k+1}}(x_{k+1}) = \mathrm{grad}f(x_{k+1})$$

gelten sollen. Mit Gleichung (7.2) ist die zweite dieser Bedingungen sofort erfüllt. Die erste Bedingung hingegen führt zur Gleichung

$$\mathrm{grad}f(x_{k+1}) + \mathrm{Hess}f(x_{k+1})(x_k - x_{k+1}) = \mathrm{grad}f(x_k),$$

und mit $H_{k+1} = \big(\mathrm{Hess}f(x_{k+1})\big)^{-1}$ zur sogenannten *Sekanten-Bedingung*

$$H_{k+1}\big(\mathrm{grad}f(x_{k+1}) - \mathrm{grad}f(x_k)\big) = x_{k+1} - x_k.$$

Zur Herleitung des Korrekturterms ΔH_k stehen als Informationen eigentlich "nur" die Terme in oberer Gleichung zur Verfügung; diese können in mehreren Arten kombiniert werden, um die Sekantenbedingung zu erfüllen. Die zwei bekanntesten Verfahren sind der *Davidon-Fletcher-Powell (DFP)*-Algorithmus, und der *Broyden-Fletcher-Goldberg-Shanno (BFGS)*-Algorithmus, wobei der BFGS-Algorithms als der generell bessere angesehen wird. Der Korrekturterm ΔH_k des DFP-Algorithmus lautet mit den Abkürzungen

$$\begin{aligned}
\triangle x_k &= x_{k+1} - x_k \\
\triangle g_k &= \mathrm{grad}f(x_{k+1}) - \mathrm{grad}f(x_k) \\
\Delta H_k &= \frac{1}{\triangle x_k^T \triangle g_k}\Big[\triangle x_k \triangle x_k^T\Big] - \frac{1}{\triangle g_k^T H_k \triangle g_k}\Big[H_k \triangle g_k \triangle g_k^T H_k\Big],
\end{aligned}$$

für den BFGS-Algorithmus ist er mit den gleichen Abkürzungen

$$\Delta H_k = \frac{\triangle x_k^T \triangle g_k + \triangle g_k^T H_k \triangle g_k}{(\triangle x_k^T \triangle g_k)^2}\Big[\triangle x_k \triangle x_k^T\Big] - \frac{1}{\triangle x_k^T \triangle g_k}\Big[H_k \triangle g_k \triangle x_k^T + \triangle x_k \triangle g_k^T H_k\Big].$$

In obigen Formeln wurden zur leichteren Lesbarkeit die Matrizeneinträge mit eckigen Klammern hervorgehoben.

Beispiel 7.9 Für quadratische Funktionen findet der BFGS-Algorithmus das Minimum in zwei Schritten, wie man an folgender Funktion erkennen kann:

$$\begin{aligned}
f(x,y) &= (1,2)\begin{pmatrix} x \\ y \end{pmatrix} + \frac{1}{2}(x,y)\begin{pmatrix} 2 & -4 \\ -4 & 18 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \\
&= x + 2y + x^2 - 4xy + 9y^2.
\end{aligned}$$

Mit einem Startpunkt von $x_1 = (2.5, -1.5)$ und der ersten Matrix $H_1 = I_2$ erhält man $x_2 = (1.861, 0.365)$ und $x_3 = (-1.3, -0.4)$. Dann gilt für die Matrix H_3

$$H_3 = \begin{pmatrix} 0.9 & 0.2 \\ 0.2 & 0.1 \end{pmatrix} = \begin{pmatrix} 2 & -4 \\ -4 & 18 \end{pmatrix}^{-1}.$$

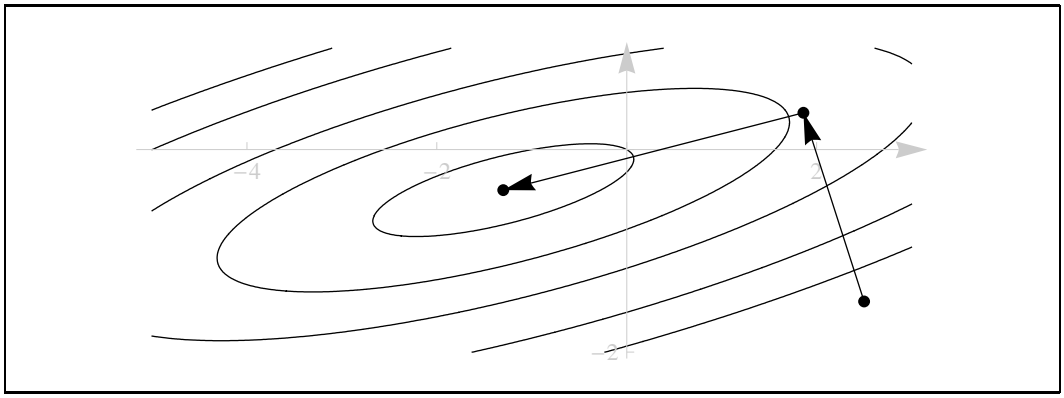

Abbildung 7.6: Die zwei Iterationsschritte, mit denen das BFGS-Verfahren das Minimum der Funktion aus Beispiel 7.9 bestimmt.

Da f eine quadratische Funktion ist, ist die Hesse-Matrix dieser Funktion konstant, und kann damit vom BFGS-Verfahren leicht iterativ ermittelt werden. Die zwei Schritte, die für diese Funktion vom Startwert zum Minimum führen, sind in Abbildung 7.6 zu sehen. □

7.5 Konjugierte Gradienten

Neben der Quasi-Newton Methode, die im letzten Abschnitt behandelt wurde, gibt es mit der *Methode der konjugierten Gradienten* ein weiteres Verfahren, das für quadratische Funktionen $f : \mathbb{R}^n \to R$ mit

$$f(x) = x + b^T x + \frac{1}{2} x^T A\, x$$

und positiv definiter Matrix A in n Schritten das Minimum findet. Ähnlich wie bei der Quasi-Newton Methode kann auch dieses Verfahren zur Minimierung beliebiger Funktionen verwendet werden, da angenommen werden kann, dass jede Funktion (zumindest lokal) durch ein Taylorpolynom zweiten Grades approximiert werden kann.

Eine Motivation für die Methode der konjugierten Gradienten kann aus Beispiel 7.8 abgelesen werden: Bei der Minimierung mit der Methode des steilsten Abstiegs und optimaler Schrittweite stehen aufeinanderfolgende Suchrichtungen immer senkrecht aufeinander; die Minimumsuche verläuft deshalb entlang eines Zickzack-Wegs (siehe Abbildung 7.5). Besser wäre es, wenn jeder Schritt den Abstand zum Minimum entlang jeweils einer Richtung minimieren würde: dann wäre nach n Schritten (in n Dimensionen) das Minimum gefunden. Man kann nachweisen, dass eine Menge von Vektoren, die paarweise *konjugiert* sind (in Bezug auf Matrix A), diese Bedingung erfüllen.

Definition 7.5 (konjugierte Vektoren)
Seien $h_1, \ldots, h_n$ Vektoren in $\mathbb{R}^n$ und A eine positiv definite $n \times n$ Matrix. Dann nennt man $h_1, \ldots, h_n$ *paarweise konjugiert in Bezug auf* A, wenn gilt

$$h_i^T A\, h_j = 0 \qquad \text{für alle } i < j.$$

Das Minimieren entlang der konjugierten Vektoren h_i minimiert eine quadratische Funktion mit Matrix A in n Schritten. Bei der Minimierung einer nichtlinearen Funktion ergibt sich allerdings das Problem, dass die Matrix A (bzw. die Hesse-Matrix im Minimum) nicht bekannt ist. Es gibt nun das bemerkenswerte Ergebnis, dass man die Richtungen h_i, die bezüglich einer Matrix A konjugiert sind, berechnen kann, ohne A zu kennen (wir gehen auf die Herleitung dieses Resultats nicht ein).

Zur Minimierung einer Funktion $f : \mathbb{R}^n \to R$ ergibt sich die Iterationsvorschrift

$$x_{k+1} = x_k + \lambda_k h_k,$$

wobei die Schrittweite λ_k so gewählt wird, dass f entlang der Linie $x_k + \lambda_k h_k$ minimal wird. Die jeweils nächsten Richtungen h_{k+1} sind definiert als

$$h_{k+1} = g_{k+1} + \gamma_k h_k,$$

wobei $g_{k+1} = -\text{grad} f(x_{k+1})$ der negative Gradient von f in x_{k+1} ist, und der Parameter γ_k über

$$\gamma_k = \begin{cases} \dfrac{g_{k+1}^T g_{k+1}}{g_k^T g_k} & \text{Methode von Fletcher-Reeves} \\ \dfrac{(g_{k+1} - g_k)^T g_{k+1}}{g_k^T g_k} & \text{Methode von Polak-Robière.} \end{cases} \tag{7.5}$$

gegeben ist. Geeignete Startwerte sind $g_1 = h_1 = -\text{grad} f(x_1)$. Die beiden Varianten des konjugierten-Gradienten Verfahrens (Fletcher-Reeves und Polak-Robiere) liefern identische Resultate bei der Minimierung quadratischer Funktionen, unterscheiden sich aber bei nichtlinearen Funktionen. Generell ist die Methode von Polak-Robière der von Fletcher-Reeves vorzuziehen.

Da beide Varianten nur n konjugierte Vektoren h_k erzeugen ist es angebracht, nach jeweils n Iterationsschritten das Verfahren neu zu starten. Alternativ dazu kann man bei der Methode von Polak-Robière die nächste Richtung als

$$h_{k+1} = g_{k+1} + \max(\gamma_k, 0) h_k,$$

festlegen; dies entspricht einem Zurücksetzen der Suchrichtung auf den negativen Gradienten, wenn $\gamma_k < 0$ ist.

Beispiel 7.10 Um den Vorteil der Verwendung konjugierter Abstiegsrichtungen zu zeigen minimieren wir nochmals die Funktion

$$f(x, y) = x^2 + 8y^2$$

aus Beispiel 7.8, deren Minimierung mit der Methode des steilsten Abstiegs in Abbildung 7.5 zu sehen ist. Mit der Methode von Polak-Robière führt bei einem Startwert von $(-4, -1)$ der erste Schritt in Richtung des negativen Gradienten zum Punkt $(-3.394, 0.212)$, und von dort aus direkt zum Minimum in $(0, 0)$. Die Richtungen der beiden Schritte sind bezüglich Hess$f$ konjugiert zueinander: Es ist

$$\text{Hess} f = \begin{pmatrix} 2 & 0 \\ 0 & 16 \end{pmatrix}, \quad \text{und} \quad (0.606, 1.212) \begin{pmatrix} 2 & 0 \\ 0 & 16 \end{pmatrix} \begin{pmatrix} 3.394 \\ -0.212 \end{pmatrix} = 0.$$

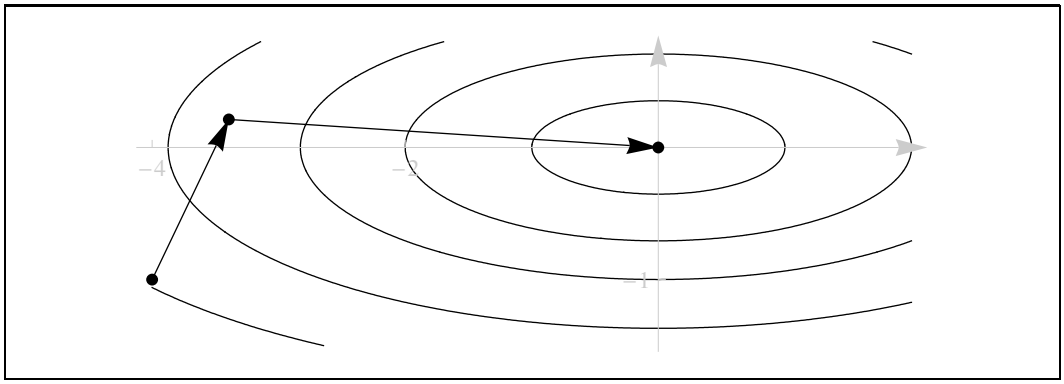

Abbildung 7.7: Die zwei Iterationsschritte, mit denen das konjugierte Gradienten Verfahren das Minimum der Funktion aus Beispiel 7.10 bestimmt.

Die Minimierung dieser Funktion mit der Methode der konjugierten Gradienten ist graphisch in Abbildung 7.7 zu sehen. Da diese Funktion quadratisch ist, liefert auch die Variante von Fletcher-Reeves identische Iterationsschritte. □

7.6 Iteratives Lösen linearer Gleichungssysteme

Wir können die Optimierungsmethode aus dem letzten Abschnitt verwenden, um bestimmte lineare Gleichungssysteme näherungsweise zu lösen. Dies wird speziell bei sehr großen Gleichungssystemen vorteilhaft sein, da das Lösen linearer Gleichungssysteme ja Laufzeitkomplexität $O(n^3)$ hat.

Die spezielle Klasse von linearen Gleichungssytemen $A\,x = b$, die durch konjugierte Gradienten approximativ gelöst werden kann, benötigt eine symmetrische positiv definite Matrix A. Das Lösen einer solchen Gleichung kann als Optimierungsproblem formuliert werden, wenn wir eine Hilfsfunktion

$$f(x) = \frac{1}{2} x^T A\, x - x^T\, b$$

definieren. Das Extremum dieser Funktion liegt an dem Punkt, an dem der Gradient von f null wird. Wegen $\mathrm{grad} f(x) = A\,x - b$ ist eine Lösung von $\mathrm{grad} f(x) = 0$ auch eine Lösung des ursprünglichen Gleichungssystems. Da A laut Voraussetzung positiv definit ist, und $\mathrm{Hess} f(x) = A$ ist, befindet sich an dieser Stelle das einzige Minimum von f.

Komplizierte Varianten des hier präsentierten Algorithmus können verwendet werden, um allgemeine quadratische Gleichungssysteme mit der Methode der konjugierten Gradienten zu lösen. Wir werden im Folgenden die Iterationsvorschriften zur Berechnung von x_k und h_k aus Abschnitt 7.5 nur für die einfachere Situation mit positiv definiter Matrix A diskutieren. Die Iterationspunkte

$$x_{k+1} = x_k + \lambda_k h_k$$

mit Abstiegsrichtungen h_k werden wie vorher berechnet. Allerdings kann, da die zu minimierende Funktion quadratisch ist, die optimale Schrittweite direkt berechnet

werden, muss also nicht auch wieder numerisch bestimmt werden. Man erhält dafür

$$\lambda_k = \frac{g_k^T g_k}{h_k^T A h_k}$$

für den negativen Gradienten g_k. Sowohl dieser negative Gradient, als auch die nächste Abstiegsrichtung h_{k+1} können aus den jeweils letzten Werten berechnet werden:

$$g_{k+1} = g_k - \lambda_k A\, h_k, \qquad h_{k+1} = g_{k+1} + \gamma_k h_k,$$

wobei γ_k wie in Gleichung (7.5) berechnet wird. Man kann nachrechnen, dass g_k und g_{k+1} senkrecht aufeinander stehen; die Methoden von Fletcher-Reeves und Polak-Robière sind hier also identisch.

Initialisiert wird diese Methode mit beliebigem Startwert x_0 (etwa 0), und $g_0 = h_0 = b - A\,x_0$. Die Berechnung von Näherungslösungen x_k wird solange iteriert, bis der Fehler in der Approximation klein genug ist. Theoretisch (bei Verwendung exakter Arithmetik) werden für ein $n \times n$ Gleichungssystem genau n Iterationsschritte benötigt. Bei großen Gleichungsystemen ist eine hinreichend genaue Lösung oft schon nach viel weniger Schritten erreicht.

Man beachte, dass der Fehler $g_k = b - Ax_k$ in der Approximation gleich dem negativen Gradienten ist — der Fehler in x_k selbst kann nicht berechnet werden, da die echte Lösung eben nicht bekannt ist. Ein Abbruchskriterium ist etwa $\|g_k\| < \varepsilon$.

Beispiel 7.11 Gegeben sei das lineare Gleichungssystem

$$\begin{pmatrix} 1 & 0 & -1 \\ 0 & 5 & 2 \\ -1 & 2 & 2 \end{pmatrix} x = \begin{pmatrix} 3 \\ 2 \\ -1 \end{pmatrix}.$$

Die Eigenwerte der Matrix A dieses Gleichungssystems sind alle positiv; A ist somit positiv definit, und die Methode der konjugierten Gradienten ist anwendbar.

Die untenstehende Tabelle gibt in exakter Arithmetik die drei Schritte an, die zum Lösen dieses linearen Gleichungssystems mit Hilfe von konjugierten Gradienten benötigt werden.

k	x_k	g_k	h_k	$\|g_k\|$
0	(0,0,0)	$(3, 2, -1)$	$(3, 2, -1)$	3.742
1	$\left(\frac{42}{29}, \frac{28}{29}, -\frac{14}{29}\right)$	$\left(\frac{31}{29}, -\frac{54}{29}, -\frac{15}{29}\right)$	$\left(\frac{1778}{841}, -\frac{980}{841}, -\frac{728}{841}\right)$	2.209
2	$\left(\frac{4969}{2545}, \frac{350}{509}, -\frac{1754}{2545}\right)$	$\left(\frac{912}{2545}, -\frac{152}{2545}, \frac{2432}{2545}\right)$	$\left(\frac{5255248}{6477025}, -\frac{400824}{1295405}, \frac{4988032}{6477025}\right)$	1.045
3	(9,-2,6)	(0,0,0)	—	0

□

7.7 Gauss-Newton Optimierung

Viele (wenn nicht sogar die meisten) Optimierungs-Problemstellungen in Naturwissenschaft und Technik haben eine spezielle Form, die eine Vereinfachung des

allgemeinen Optimierungsansatzes (wie bisher besprochen) zulässt. Diese Problemstellungen treten überall dort auf, wo die Parameter eines Modells mit Hilfe von gegebenen (gemessenen) Daten bestimmt werden sollen. In diesen Fällen sollen die Parameter so berechnet werden, dass die Fehler zwischen vom Modell vorhergesagten und tatsächlich gemessenen Daten möglichst gering werden. Ein einfaches Beispiel soll diesen Sachverhalt erläutern.

Beispiel 7.12 In einem Industrieprozess sei der Zusammenhang zwischen Eingabegrößen x_i und Ausgabegrößen y_i für verschiedene Eingabegrößen gemessen worden; acht nach den x_i sortierte Paare (x_i, y_i) seien etwa

$$\big((0.82, 6.88), (3.28, 8.49), (3.85, 8.96), (4.72, 14.49), (5.38, 12.27), \\ (8.78, 30.65), (8.9, 32.38), (9.47, 39.76)\big) .$$

Auf Basis technischer Überlegungen vermutet man, dass der Zusammenhang zwischen den Ein- und Ausgabegrößen exponentiell ist und durch

$$y = g(x;\, \alpha) = \alpha_1 + \alpha_2 \exp(\alpha_3 x)$$

ausgedrückt werden kann. Aus den gemessenen Werten (x_i, y_i) soll nun der Parametervektor $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ bestimmt werden, für den der Zusammenhang zwischen echten Werten y_i und vorhergesagten Werten $g(x_i;\, \alpha)$ möglichst gut ist.

Mathematisch gesehen liegt somit eine Optimierungsproblemstellung vor, bei der die Abweichungen $|y_i - g(x_i;\, \alpha)|$ minimiert werden sollen. Aus Gründen, die erst in Abschnitt 8.4 klar werden, wird zur Bestimmung der optimalen α-Werte die Summe der Fehlerquadrate minimiert, α^* ist also definiert als

$$\alpha^* = \arg\min_{\alpha} \sum_{i=1}^{8} \big(y_i - g(x_i;\, \alpha)\big)^2 .$$

Das Modell mit den optimalen Werten α^* sowie den Daten ist in Abbildung 7.8 zu sehen. □

Zu beachten ist, dass der in obigem Beispiel 7.12 durch g gegebene Zusammenhang zwischen den x_i und y_i nichtlinear ist, und auch nicht *linear in den Parametern* ist. Für Modelle, die linear in den Parametern sind, eignet sich zur Parameterbestimmung *lineare Regression* am besten; diese wird bei uns in Abschnitt 8.4 behandelt.

In diesem Abschnitt behandeln wir Probleme, bei denen optimale Parameterwerte α^* für Modelle bestimmt werden sollen. Die Daten (x_i, y_i) sind dabei bereits gegeben und werden in den folgenden Ausführungen zur Vereinfachung der Notation meist weggelassen. Die im Folgenden präsentierte Methode ist aber auch allgemeiner verwendbar; die Problemstellungen müssen aber wie hier angegeben formuliert werden können.

Zu minimieren ist im allgemeinsten Fall eine Funktion $f : \mathbb{R}^n \to \mathbb{R}$

$$f(\alpha) = \frac{1}{2} \sum_{i=1}^{m} r_i(\alpha)^2 = \frac{1}{2} R(\alpha)^T R(\alpha) , \tag{7.6}$$

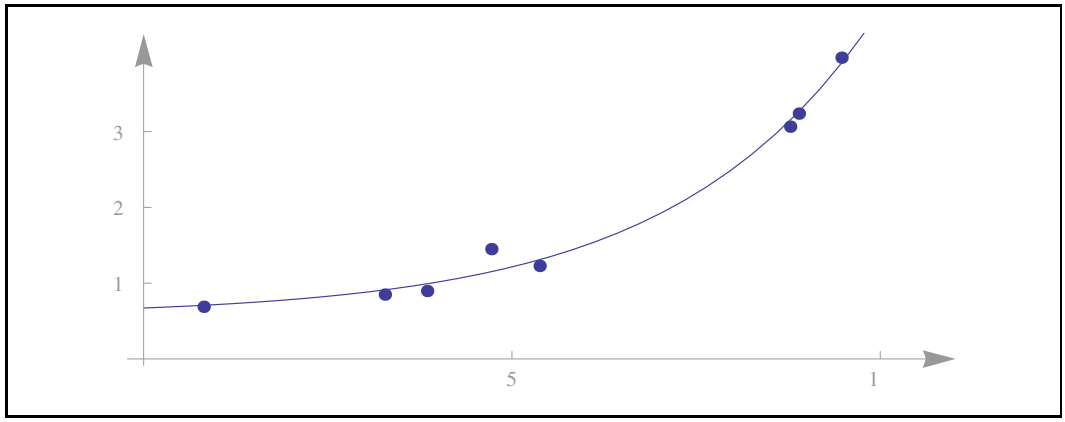

Abbildung 7.8: Graphische Veranschaulichung der Daten und des optimalen Modells aus Beispiel 7.12.

die von m Funktionen $r_i : \mathbb{R}^n \to \mathbb{R}$ abhängt — meist die Fehler (genannt *Residuen*) bei m Datenpunkten — und die in Vektor-Notation in einer Funktion $R : \mathbb{R}^n \to \mathbb{R}^m$

$$R(\alpha) = \big(r_1(\alpha), \ldots, r_m(\alpha)\big)^T$$

zusammengefasst werden können.Der Faktor $\frac{1}{2}$ dient zur Vereinfachung der folgenden Berechnungen.

Obige Problemstellung stellt somit eine Konkretisierung der allgemeiner gehaltenenen Problemstellungen vom Anfang dieses Kapitels dar (auch wenn die unabhängige Größe dort mit x und nicht α bezeichnet wurde). Wie dort benötigen wir auch hier die ersten und zweiten Ableitungen von f, also den Gradienten und die Hesse-Matrix der zu minimierenden Funktion. Dazu ist die Jacobi-Matrix von R hilfreich, also die in Definition 6.2 eingeführte Matrix aller partiellen Ableitungen einer vektorwertigen Funktion. Für R ist dies

$$J\,R = \frac{\partial(r_1, \ldots, r_m)}{\partial(\alpha_1, \ldots, \alpha_n)} = \begin{pmatrix} \frac{\partial r_1}{\partial \alpha_1} & \cdots & \frac{\partial r_1}{\partial \alpha_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial \alpha_1} & \cdots & \frac{\partial r_m}{\partial \alpha_n} \end{pmatrix}$$

Die Jacobi-Matrix besteht also Zeilenweise aus den Gradienten von r_i. Zur weiteren Vereinfachung der Notation werden wir im Folgenden nur mehr J statt $J\,R$ schreiben.

Unter Verwendung von J können wir die Ketten- und Produktregel im mehrdimensionalen Fall verwenden, um die Ableitungen der Funktion $f(\alpha)$ zu berechnen. Die Anwendung dieser Regeln kann mit der Analogie zur eindimensionalen Situation plausibel gemacht werden kann, wird hier aber nicht bewiesen. Dazu beachten wir, dass $f(\alpha)$ wie oben definiert sehr schlampig als $\frac{1}{2}R(\alpha)^2$ behandelt werden kann ("offiziell" nicht, weil sich das mit der Dimensionalität der Objekte nicht ausgeht). Damit erhält man, ebenso schlampig geschrieben, und nur als Eselsbrücke für die Herleitungen weiter unten gedacht:

$$\begin{aligned} f'(\alpha) &\mathrel{\widehat{=}} R(\alpha)R'(\alpha) \\ f''(\alpha) &\mathrel{\widehat{=}} R'(\alpha)R'(\alpha) + R(\alpha)R''(\alpha) \end{aligned}$$

Für die genauen Berechnungen muss man auch auf die Dimensionalität achten: Die erste Ableitung ist der Gradient, und die zweite Ableitung die Hesse-Matrix. Somit ergibt sich:

$$\begin{aligned}
\mathrm{grad} f(\alpha) &= \sum_{i=1}^{m} r_i(\alpha)\mathrm{grad} r_i(\alpha) = J^T R(\alpha) \\
\mathrm{Hess} f(\alpha) &= \sum_{i=1}^{m} \mathrm{grad} r_i(\alpha)\mathrm{grad} r_i(\alpha)^T + \sum_{i=1}^{m} r_i(\alpha)\mathrm{Hess} r_i(\alpha) \\
&= J^T J + \sum_{i=1}^{m} r_i(\alpha)\mathrm{Hess} r_i(\alpha) \qquad (7.7)
\end{aligned}$$

Die soeben hergeleiteten Formeln werden nun in einem Newton-Schritt wie in Abschnitt 7.4 verwendet. Zur Erinnerung: Ohne Approximation der Hesse-Matrix im DFP bzw. BFGS-Verfahren ist der Newton-Schritt wie in Gleichung 7.4 angegeben:

$$x_{k+1} = x_k - \lambda_k \,\mathrm{Hess} f(x_k)^{-1} \mathrm{grad} f(x_k).$$

Das *Gauss-Newton Verfahren* nützt nun die spezielle Struktur von Problemstellungen, die Summen von Fehlerquadraten minimieren. Dabei wird der zweite Summand in der Gleichung 7.7 vernachlässigt, womit sich die Vereinfachung $\mathrm{Hess} f = J^T J$ ergibt. Es gibt mehrere Motivationen für das Weglassen dieses zweiten Summanden: In der Nähe eines Minimums sind die Residuen klein (dann ist der Anteil $r_i(\alpha)$ an der Summe klein), oder lassen sich zumindest linear approximieren (dann ist $\mathrm{Hess} r_i(\alpha) = 0$). Wenn diese Annahmen nicht gelten, konvergiert das Gauss-Newton Verfahren nicht oder nur langsam.

Somit ist der Gauss-Newton Schritt gegeben durch

$$\alpha_{k+1} = \alpha_k - \lambda_k \big(J_k^T J_k\big)^{-1} J_k^T R(\alpha_k),$$

wobei $J_k = J\,R(\alpha_k)$ ist, und λ_k durch Minimumsuche in einer Dimension bestimmt wird. Achtung: Hier bezeichnen die Indizes k die Iterationsschritte, und nicht die Komponenten im Vektor α.

Beispiel 7.13 Gegeben sei eine Menge von 15 Datenpunkten, an die das Modell

$$y = g(x;\, \alpha) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 \exp(\alpha_5 x)$$

angepasst werden soll. Einzelne Schritte des Gauss-Newton Verfahrens zur Minimierung der Fehlerfunktion

$$f(\alpha) = \frac{1}{2} \sum_{i=1}^{15} \big(y_i - g(x_i;\, \alpha)\big)^2$$

sind in Abbildung 7.9 zu sehen. Der Startwert $\alpha_1 = (100, 2, 3, -0.2, 0.5)$ lag dabei (bis auf die erste Komponete) nicht weit von der optimalen Lösung $\alpha_7 = (-5.3, 7.2, 1.8, -0.03, 0.77)$ entfernt. Mit vielen anderen, zufällig gewählten Startwerten konvergiert das Verfahren nicht. □

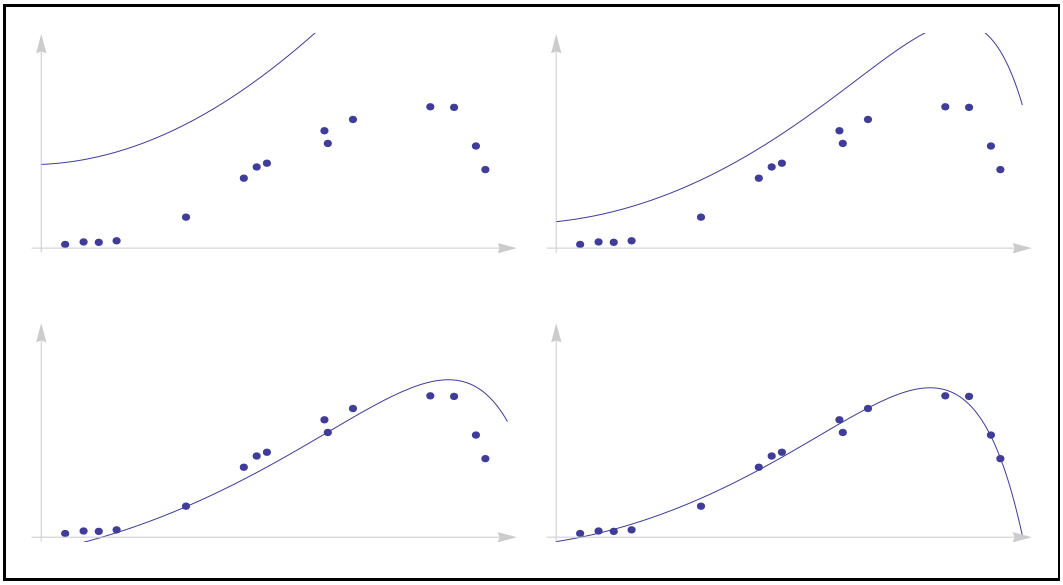

Abbildung 7.9: Illustration der Konvergenz des Gauss-Newton Verfahrens für das Modell in Beispiel 7.13. Zu sehen sind (zeilenweise) die Startkonfiguration, sowie der dritte, vierte, und siebte Schritt des Verfahrens.

7.8 Levenberg-Marquardt Optimierung

Bei Betrachtung der bisherigen Methoden zur Optimierung multivariater Funktionen fällt auf, dass alle Methoden den negativen Gradienten benutzen, um eine Abstiegsrichtung zu bestimmen. Bei der Methode des steilsten Abstiegs (Abschnitt 7.3) wird nur dieser negative Gradient verwendet, also

$$x_{k+1} = x_k - \lambda_k \mathrm{grad} f(x_k)\,.$$

Bei Methoden, die auf dem Newton-Verfahren aufbauen (also Quasi-Newton Verfahren in Abschnitt 7.4 oder die Gauss-Newton Approximation in Abschnitt 7.7), wird die Richtung noch durch die inverse Hesse-Matrix bzw. Approximationen daran verändert:

$$x_{k+1} = x_k - \lambda_k \,\mathrm{Hess} f(x_k)^{-1} \mathrm{grad} f(x_k).$$

Es ist nun möglich, beide Ansätze miteinander zu kombinieren, und je nach lokaler Charakteristik der zu minimierenden Funktion zwischen den jeweiligen Anteilen umzuschalten. Da diese Kombination hauptsächlich zur Minimierung von Fehlerquadrat-Aufgabenstellungen wie Gleichung 7.6 verwendet wird, greift man auf die Gauss-Newton Approximation $\mathrm{Hess} f \approx J^T J$ aus Abschnitt 7.7 zurück.

Die *Methode von Levenberg* implementiert obige Überlegungen, indem der nächste Iterationsschritt über

$$x_{k+1} = x_k - \big(\mathrm{Hess} f(x_k) + \beta_k I\big)^{-1} \mathrm{grad} f(x_k) \tag{7.8}$$

definiert wird; I bezeichnet hier die Einheitsmatrix. Der neue Parameter β_k bestimmt den relativen Anteil von Gradientenabstieg und Gauss-Newton Optimierung:

Für kleine Werte von β_k dominiert $\mathrm{Hess} f(x_k)$ den Klammerausdruck, für große Werte von β_k die Einheitsmatrix I. Damit wird für großes β_k Gradientenabstieg durchgeführt.

Der Wert von β_k wird in jedem Iterationsschritt folgendermaßen adaptiert, nachdem x_{k+1} aus Gleichung 7.8 berechnet und darüber $f(x_{k+1})$ bestimmt wurde:

- Falls $f(x_{k+1}) \geq f(x_k)$ ist, der Schritt nach x_{k+1} also nach *oben* geführt hat, wird der Anteil des Gradientenabstiegs erhöht, da die Gauss-Newton Approximation anscheinend noch nicht gut funktioniert. Meist wird $\beta_{k+1} = 10\,\beta_k$ verwendet. Weiters wird der letzte Schritt zurückgenommen, also $x_{k+1} = x_k$ gesetzt.

- Falls $f(x_{k+1}) < f(x_k)$ ist, der Schritt nach x_{k+1} also nach *unten* geführt hat, wird der Gauss-Newton Anteil weiter erhöht (etwa über $\beta_{k+1} = \frac{1}{10}\,\beta_k$) und x_{k+1} behalten.

Als Startwert β_0 wird in der Literatur ein ziemlich kleiner Wert vorgeschlagen, etwa $\beta_0 = 0.001$.

Diese Idee, die in der Praxis schon gut funktioniert, wurde schließlich von Marquardt verbessert, indem auch bei dominierendem Gradientenabstiegs-Anteil noch Krümmungsinformation verwendet wird, und zwar über den Diagonalanteil der Hesse-Approximation. Wenn die Krümmung in einer Koordinatenrichtung *groß* ist, wird durch die Inversenbildung der Schritt in diese Richtung *klein* (und umgekehrt). Dies ist genau das erwünschte Verhalten: in gerade Richtungen weit zu gehen, in gekrümmte nur kurz. Die *Methode von Levenberg-Marquardt* verwendet somit den Iterationsschritt

$$x_{k+1} = x_k - \big(\mathrm{Hess} f(x_k) + \beta_k \,\mathrm{diag}(\mathrm{Hess} f(x_k))\big)^{-1} \mathrm{grad} f(x_k)$$

mit obigen Anpassungen von β_k. Mit explizit geschriebener Termen für Gradienten und Hesse-Matrix, die sich aus Fehlerquadrat-Aufgabenstellung ergeben, lässt sich der Levenberg-Marquardt Schritt schreiben als

$$x_{k+1} = x_k - \big(J_k^T J_k + \beta_k \,\mathrm{diag}(J_k^T J_k)\big)^{-1} J_k^T R(x_k)\,.$$

Diese Update-Methode der de-facto Standard zum Lösen von Optimierungsproblemstellungen, die sich wie Gleichung 7.6 schreiben lassen.

7.9 Lagrange Multiplikatoren

In den letzten Abschnitten haben wir gesehen, wie man numerisch Extrema von Funktionen finden kann. In diesem Abschnitt behandeln wir nun einen Spezialfall: Wenn das gesuchte Extremum eine weitere Bedingung erfüllen soll. Mit Hilfe von *Lagrange-Multiplikatoren* kann in diesem Fall das Extremum bestimmt werden.

Wir betrachten zuerst ein Beispiel.

Beispiel 7.14 Gesucht ist das Minimum der Funktion

$$f(x, y) = -e^{-\frac{1}{2}(x^2+y^2)}.$$

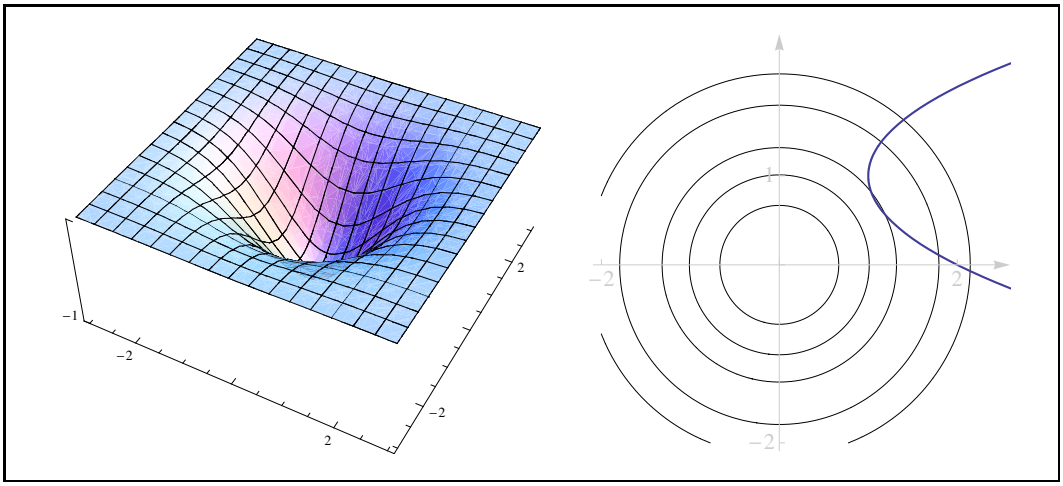

Abbildung 7.10: Die Funktion $f(x,y) = -e^{-\frac{1}{2}(x^2+y^2)}$ aus Beispiel 7.14 als 3D-Plot (links) und als Isoniveaulinien-Plot mit der Nebenbedingung $x = y^2-2y+2$ (rechts).

Diese Funktion hat bei $(0,0)$ ein globales Minimum. Wenn wir zusätzlich fordern, dass das Minimum noch die Bedingung

$$g(x,y) = y^2 - 2y - x + 2 = 0 \qquad (\text{also } x = y^2 - 2y + 2)$$

erfüllen soll, so schließt dies sofort das ursprüngliche Minimum aus. Die Skizze in Abbildung 7.10 mit den Isoniveaulinien von f und der Nebenbedingung $g(x) = 0$ zeigt die eigentliche Problemstellung: Wir suchen diejenigen Punkte x mit $g(x) = 0$, die "am tiefsten" liegen. Optisch sieht man anhand der Abbildung, dass der tiefste Punkt mit $g(x) = 0$ derjenige ist, an dem sich g und die Höhenlinien von f tangieren. Da wir wissen, dass Gradienten immer senkrecht auf Höhenlinien (und auch auf Tangenten!) stehen, können wir diese Observation durch Gradienten ausdrücken: Am niedrigsten Punkt muss der Gradient von f in die gleiche Richtung wie der Gradient von g zeigen, also ein Vielfaches dieses Vektors sein:

$$\operatorname{grad} f = \lambda \operatorname{grad} g.$$

In diesem Beispiel ist

$$\operatorname{grad} f(x,y) = \left(xe^{-\frac{1}{2}(x^2+y^2)}, ye^{-\frac{1}{2}(x^2+y^2)}\right)^T$$

und

$$\operatorname{grad} g(x,y) = \big(-1, 2y-2\big)^T.$$

Wir haben somit drei Unbekannte (x, y und λ) in zwei Gleichungen (den Komponenten der Gradientengleichung). Als dritte Gleichung können wir die Nebenbedingung verwenden und erhalten somit das System

$$\begin{aligned} xe^{-\frac{1}{2}(x^2+y^2)} &= -\lambda \\ ye^{-\frac{1}{2}(x^2+y^2)} &= \lambda(2y-2) \end{aligned}$$

$$x = y^2 - 2y + 2$$

Dieses System kann man sogar noch mit der Hand lösen: Aus den ersten beiden Gleichungen erhält man nach Elimination von $e^{-\frac{1}{2}(x^2+y^2)}$

$$y = -x(2y - 2),$$

und zusammen mit der letzten Gleichung dann eine polynomiale Gleichung dritten Grades in y:

$$-2y^3 + 6y^2 - 9y + 4 = 0.$$

Die einzige reelle Lösung dieser Gleichung liegt bei $y = 0.687092$; somit ergibt sich für die andere Koordinate des Minimums $x = 1.09791$. Der Funktionswert an dieser Stelle ist -0.432249. □

Wenn sich in schwierigeren Beispielen das nichtlineare Gleichungssystem nicht mehr von Hand lösen lässt, muss man auf numerische Verfahren ausweichen (etwa das mehrdimensionale Newtonverfahren aus Abschnitt 6.4). Der Lösungsansatz durch Gleichsetzen der Gradienten ist aber allgemein gültig.

Satz 7.6 Gegeben sei eine zu optimierende Funktion $f : \mathbb{R}^n \to \mathbb{R}$ und eine durch eine Funktion $g : \mathbb{R}^n \to \mathbb{R}$ definierte Nebenbedingung $g(x) = 0$. Man bezeichnet

$$L(x, \lambda) = f(x) - \lambda g(x)$$

als *Lagrange-Funktion* dieses Optimierungsproblems. Den Faktor λ nennt man dabei *Lagrange Multiplikator*. Die Lösungen von

$$\text{grad}_x L(x, \lambda) = 0 \qquad \text{und} \qquad \frac{\partial L(x, \lambda)}{\partial \lambda} = 0$$

sind diejenigen Punkte von $g(x) = 0$, an denen f minimal ist.

Durch die Verwendung der Lagrange-Funktion $L(x, \lambda)$ lassen sich f und g in einer Funktion zusammenfassen. Das Nullsetzen der Ableitungen von L liefert genau wieder die Bedingungen

$$\text{grad} f(x) = \lambda \text{grad} g(x) \quad \text{und} \quad g(x) = 0,$$

die wir für das Lösen von Aufgabe 7.14 benötigt haben.

Es gibt auch für Optimierungsprobleme mit Nebenbedingungen Kriterien, anhand derer man entscheiden kann, ob eine gefundene Lösung ein Minimum oder ein Maximum ist. Wir werden auf diese Kriterien nicht eingehen, da sie nicht so einsichtig sind wie die Bedingungen an die zweiten Ableitungen bei Optimierungsproblemen ohne Nebenbedingungen. Stattdessen werden wir untersuchen, wie man durch einfaches Einsetzen (bei geeigneten Aufgaben) zwischen Minima und Maxima unterscheiden kann.

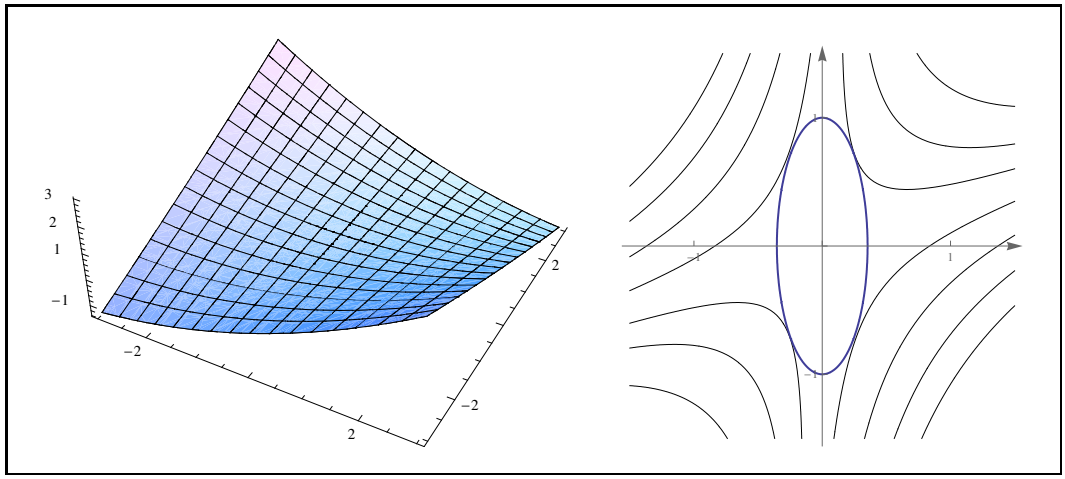

Abbildung 7.11: Die Funktion $f(x,y) = 8x^2 - 24xy + y^2$ aus Beispiel 7.15 als 3D-Plot (links) und als Isoniveaulinien-Plot mit der Nebenbedingung $8x^2 + y^2 = 1$ (rechts).

Beispiel 7.15 Zu optimieren sei die Funktion

$$f(x,y) = 8x^2 - 24xy + y^2$$

unter der Nebenbedingung $g(x,y) = 8x^2 + y^2 = 1$. Wie man in Abbildung 7.11 erkennen kann, ist durch diese Nebenbedingung eine Ellipse definiert. Für die Gradienten erhält man

$$\text{grad} f(x,y) = \big(16x - 24y, -24x + 2y\big)^T \quad \text{und} \quad \text{grad} g(x,y) = \big(16x, 2y\big)^T$$

Das zu lösende nichtlineare Gleichungssystem ist somit

$$\begin{aligned} 16x - 24y &= \lambda 16x \\ -24x + 2y &= \lambda 2y \\ 8x^2 + y^2 &= 1, \end{aligned}$$

das man mit etwas Aufwand sogar händisch lösen kann (etwa indem man die ersten beiden Gleichungen zu einer quadratischen Gleichung für λ umformt, daraus x durch y ausdrückt, und dies in die dritte Gleichung einsetzt). Daraus erhält man die vier Lösungen $\big(\pm\frac{1}{4}, \pm\frac{1}{\sqrt{2}}\big)$. Einsetzen dieser Lösungen in f liefert

$$f\Big(-\frac{1}{4}, -\frac{1}{\sqrt{2}}\Big) = f\Big(\frac{1}{4}, \frac{1}{\sqrt{2}}\Big) = -3.24 \quad \text{und} \quad f\Big(-\frac{1}{4}, \frac{1}{\sqrt{2}}\Big) = f\Big(\frac{1}{4}, -\frac{1}{\sqrt{2}}\Big) = 5.24.$$

Die Lösungen im 2. und 4. Quadranten sind somit Maxima, die Lösungen im 1. und 3. Quadraten Minima von f unter der Nebenbedingung g. □

Mit Lagrange-Multiplikatoren lässt sich auch endlich eine noch offene Fragestellung aus Abschnitt 2.4 klären: Warum ist die Richtung der größten Varianz einer Punktwolke durch den Eigenvektor mit dem größten Eigenwert der entsprechenden Kovarianzmatrix gegeben?

Zur Beantwortung dieser Frage verwenden wir nochmals die Notation von Abschnitt 2.4. Seien also $x_1, \ldots, x_n \in \mathbb{R}^m$ eine Menge von m-dimensionalen Datenpunkten, die wir hier als Spaltenvektoren verwenden. Weiters seien diese Datenpunkte zentriert, ihr Mittelpunkt $\bar{x}$ sei also null. Wir suchen nun eine Richtung v, entlang derer die Varianz der Datenpunkte maximal ist. Wie wir aus Kapitel 2 wissen, ist für einen Einheitsvektor v die Menge der Projektionen der x_i auf v gegeben durch $v^T x_1, \ldots, v^T x_n$. Die Richtung mit der größten Varianz löst somit die Problemstellung

$$\mathrm{Var}\big(\{v^T x_1, \ldots, v^T x_n\}\big) \to \max\,. \tag{7.9}$$

Um eine Lösung dieser Problemstellung ausrechnen zu können, gehen wir wie folgt vor. Mit der Definition der Varianz als mittleres quadratisches Abweichen vom arithmetischen Mittel erhalten wir

$$\mathrm{Var}\big(\{v^T x_1, \ldots, v^T x_n\}\big) = \frac{1}{n}\sum_{i=1}^{n}\big(v^T x_i - v^T \bar{x}\big)^2 = \frac{1}{n} v^T \sum_{i=1}^{n} x_i v^T x_i\,,$$

wobei wir verwendet haben, dass $\bar{x} = 0$ ist und v^T als konstanter Faktor aus der Klammer herausgehoben werden kann. Da weiters für zwei Vektoren x und y immer $x^T y = y^T x$ gilt, erhält man

$$\mathrm{Var}\big(\{v^T x_1, \ldots, v^T x_n\}\big) = \frac{1}{n} v^T \sum_{i=1}^{n} x_i v^T x_i \;= \frac{1}{n} v^T \sum_{i=1}^{n} x_i x_i^T v = v^T C v\,;$$

hier bezeichnet C die aus Definition 2.12 bekannte Kovarianzmatrix der x_i.

Die umgeschriebene Aufgabenstellung 7.9 lässt sich somit schreiben als

$$v^T C v \to \max\,.$$

Diese Aufgabenstellung hat, wie man sich leicht überlegen kann, in dieser Form kein Maximum, das man $v^T C v$ für immer größere Vektoren v (also $\|v\|$ immer größer) ebenfalls immer größer machen kann. Da es aber auf $\|v\|$ aber nicht ankommt, weil nur die Richtung von v wichtig ist, kann man als Bedingung $\|v\|^2 = v^T v = 1$ festsetzen. Wir erhalten somit ein Optimierungsproblem mit Nebenbedingungen:

$$\text{maximiere } v^T C v \;\text{ unter der Nebenbedingung } v^T v = 1\,.$$

Daraus erhält man die Lagrange-Funktion

$$L(v, \lambda) = v^T C v - \lambda(v^T v - 1)$$

und aus Satz 7.6 die beiden Gleichungen

$$\mathrm{grad}_v L(v, \lambda) = \frac{1}{2} C v - \frac{1}{2}\lambda v = 0 \qquad \text{und} \qquad \frac{\partial L(v, \lambda)}{\partial \lambda} = v^T v - 1 = 0\,.$$

Die erste Gleichung liefert das Eigenwertproblem $Cv = \lambda v$, die zweite die Nebenbedingung $v^T v = 1$. Somit ist einleuchtend, warum in Abschnitt 2.4 zur Bestimmung der Richtung der größten Ausdehnung ein Eigenwertproblem gelöst wurde. Es ist aber noch nicht ganz klar, warum man bei der Hauptkomponentenanalyse als erste Richtung den Eigenvektor verwendet, der den größten Eigenwert hat. Dies lässt sich

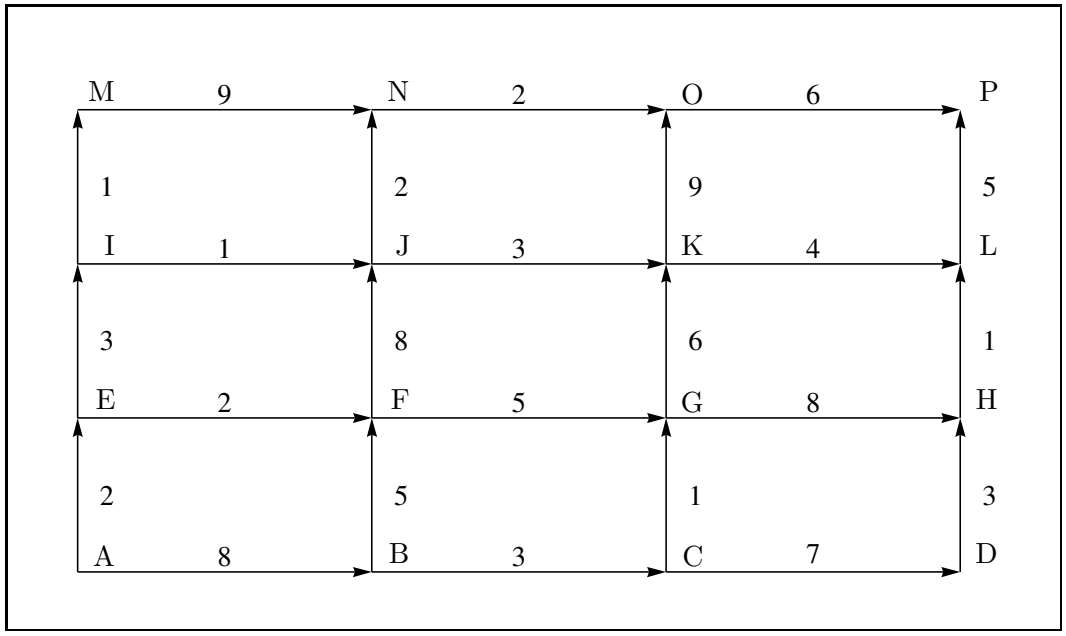

Abbildung 7.12: Illustration zur Problem der kürzesten Wegzeitbestimmung aus Beispiel 7.16.

nachrechnen, indem wir die Varianz in Richtung eines Eigenvektors v mit $\|v\| = 1$ betrachten:

$$\mathrm{Var}\big(\{v^T x_1, \ldots, v^T x_n\}\big) = v^T C v = v^T \lambda v = \lambda\,.$$

Somit ist die Ausdehnung in Richtung der Hauptkomponente v durch den zu v gehörigen Eigenwert λ bestimmt. Die Richtung der größten Ausdehnung ist also der Eigenvektor mit größtem Eigenwert.

7.10 Dynamische Optimierung

Als *dynamische Optimierung* bezeichnet man eine Klasse von Algorithmen, die sequentielle Problemstellungen optimal lösen können. Um im Folgenden die auftauchenden Begriffe sinnvoll verwenden zu können, betrachten wir ein Beispiel für die Art von Problemen, die mit dynamischer Optimierung zu lösen sind.

Beispiel 7.16 Gegeben sei ein rechteckiger Stadtplan aus Einbahnstraßen wie in Abbildung 7.12 und die Aufgabenstellung, mit dem Auto so schnell wie möglich vom Startpunkt A links unten zum Zielpunkt P rechts oben zu gelangen. Dabei geben die Zahlen an den Kanten die Zeit an, die für das Durchfahren der Straße entlang dieses Blocks benötigt wird.

Eine kurze kombinatorische Überlegung zeigt, dass es 20 verschiedene Wege von A nach P gibt; eine vollständige Suche würde somit die Zeiten aller 20 Wege berechnen müssen. Sei dazu t_A die Wegzeit von A nach P, und analog dazu t_B, t_C, ... die kürzesten Zeiten, die von B, C, ... bis zum Zielpunkt P gebraucht werden.

Mit folgender Überlegung kann gezeigt werden, dass das Problem einfacher gelöst werden kann; diese Überlegung beinhaltet bereits den Grundgedanken der dynami-

schen Programmierung. Wenn man nämlich irgendwie wüsste, wie lange man von den benachbarten Punkten von A nach P braucht (also die Werte t_B und t_E kennen würde), dann könnte man das gesuchte t_A einfach als

$$t_A = \min\{8 + t_B, 2 + t_E\}$$

berechnet werden. Die unbekannten Werte t_B und t_E könnten wiederum durch t_F und t_C bzw. t_I und t_F ausgedrückt werden. Dieser Vorgang lässt sich solange fortsetzen, bis wir alle Zeiten auf t_O und t_L (den Nachbarn von P) zurückgeführt haben; diese Zeiten sind aber direkt aus der Graphik ablesbar! Somit kann durch Einsetzen in die rekursiven Definitionen die optimale Lösung bestimmt werden. Wir könnten jetzt weiter rekursiv Werte bestimmen, die nächsten wären etwa

$$t_B = \min\{5 + t_F, 3 + t_C\} \qquad \text{und} \qquad t_E = \min\{3 + t_I, 2 + t_F\}.$$

Sinnvoller ist es jedoch, das Problem von hinten zu lösen, da wir dann keine rekursiven Berechnungen aufbauen müssen. Wir erhalten dann mit den direkt ablesbaren Werten $t_O = 6$ und $t_L = 5$ die Resultate

$$t_N = 2 + t_O = 8, t_M = 9 + t_N = 17, t_H = 1 + t_L = 6, t_D = 3 + t_H = 9.$$

Die restlichen Werte sind über Vergleiche von Alternativrouten zu berechnen. Es ergibt sich

$$\begin{aligned} t_K &= \min\{9 + t_O, 4 + t_L\} = 9, & t_J &= \min\{2 + t_N, 3 + t_K\} = 10, \\ t_I &= \min\{1 + t_M, 1 + t_J\} = 11, & t_G &= \min\{6 + t_K, 8 + t_H\} = 14, \\ t_F &= \min\{8 + t_J, 5 + t_G\} = 18, & t_E &= \min\{3 + t_I, 2 + t_F\} = 14, \\ t_C &= \min\{1 + t_G, 7 + t_D\} = 15, & t_B &= \min\{5 + t_F, 3 + t_C\} = 18. \end{aligned}$$

Damit haben wir alle notwendigen Zwischenresultate, und die Lösung des Problems ist

$$t_A = \min\{2 + t_E, 8 + t_B\} = 16.$$

Natürlich wird man im konkreten Fall nicht nur an der Minimalzeit interessiert sein, sondern auch am Weg, der diese Zeit realisiert. Dies ist etwa in Computerimplementierungen durch die Verwendung von Referenzen an den jeweils nächsten kürzesten Wegpunkt leicht zu realisieren. □

Im letzten Beispiel haben wir bereits die wichtigste Eigenschaft von Problemen gesehen, die durch dynamische Optimierung gelöst werden können. Diese Eigenschaft wird *Optimalitätsprinzip* genannt.

Definition 7.6 (Optimalitätsprinzip)
Ein Problem genügt dem Optimalitätsprinzip, wenn jede Teillösung der optimalen Lösung ebenfalls optimal ist.

Das Problem aus Beispiel 7.16 genügt dem Optimalitätsprinzip, da jede Teilstrecke in der Gesamtlösung optimal ist. Dieses Prinzip ist implizit in unsere Lösungsstrategie

eingegangen, indem wir das Minimum der beiden Teillösungen verwenden, die beide wiederum optimale Lösungen ihrer Teilprobleme sind.

Vom *algorithmischen Ansatz* her unterscheidet sich dynamische Optimierung von anderen Methoden dadurch, dass (durch das Optimalitätsprinzip bedingt) kleine Teillösungen zur globalen Lösung zusammengefasst werden können. Die Struktur der Lösung ist am einfachsten zu verstehen, wenn man sie von hinten betrachtet. Dadurch wird die Rekursion, die sich bei direkter Problemformulierung ergibt, in die Iteration umgewandelt, die der Auflösung der Endrekursion entspricht. Dies ist am nächsten Beispiel gut zu erkennen.

Beispiel 7.17 Eine Informatikerin möchte den Mann fürs Leben finden und beschließt, aus zehn Männern den besten auszuwählen. Da es durch soziale Konventionen bedingt nicht möglich ist, alle Männer durchzuprobieren und dann den besten auszuwählen, muss sie alle nacheinander bewerten und nach jedem Mann eine Entscheidung treffen, ob sie diesen auswählen möchte oder nicht. Wenn sie einen Mann abgelehnt hat, kann sie später nicht mehr auf ihn zurückkommen; wenn sie einen Mann akzeptiert hat, verwirft sie dadurch alle anderen und beendet damit das Auswahlverfahren. Wenn sie weiß, dass sie jeden Mann auf einer Skala von 1 bis 1000 bewerten kann, welche Strategie gibt ihr dann die größte Chance, den besten der zehn Männer auszuwählen?

Dieses Problem ist erst auf den zweiten Blick durch dynamische Optimierung lösbar, da man die Anwendbarkeit des Optimalitätsprinzips nicht direkt aus der Aufgabenstellung herauslesen kann. Dies wird erst durch das Aufrollen des Problems von hinten einsichtig: Seien dafür $K_1, \ldots, K_{10}$ die zehn Kandidaten, die in dieser Reihenfolge evaluiert werden. Wenn die Frau beim letzten Mann K_{10} angelangt ist, muss sie diesen akzeptieren, da sie alle anderen abgelehnt hat. Der Mann K_{10} hat eine auf der Menge $\{1, \ldots, 1000\}$ gleichverteilte Bewertung, zu erwarten ist damit eine Bewertung von 500. Folgende Überlegung beinhaltet die Übertragung der dynamischen Optimierung auf dieses Problem: Wenn die Informatikerin beim vorletzten Mann in der Reihe (also K_9) angelangt ist, wird sie diesen *nur dann* auswählen, wenn seine Bewertung höher als 500 ist, da ihr sonst ein Warten auf den letzten Mann (im Durchschnitt) höhere Bewertung verspricht. Dieses Argument lässt sich auch auf K_8 (und in weiterer Folge auf alle anderen Kandidaten) übertragen: K_8 wird nur dann ausgewählt werden, wenn er eine höhere Bewertung hat, als die Anwendung der optimalen Strategie auf die beiden letzten Kandidaten verspricht. Der Erwartungswert der optimalen Strategie für K_9 und K_{10} kann folgendermaßen berechnet werden: K_9 wird akzeptiert, wenn seine Bewertung höher als 500 ist, dies ist mit Wahrscheinlichkeit $\frac{1}{2}$ der Fall; der dann zu erwartende Wert ist 750. Mit Wahrscheinlichkeit $\frac{1}{2}$ verwirft sie K_9 und entscheidet sich für K_{10} mit Erwartungswert 500. Somit ist der durch diese Strategie zu erwartende Wert für die letzten beiden Kandidaten $\frac{1}{2}750 + \frac{1}{2}500 = 625$. Sie wird sich somit nur dann für Kandidat K_8 entscheiden, wenn seine Bewertung höher als 625 ist. Dieses Argument kann für alle anderen Kandidaten von 7 bis 1 rückwerts angewandt werden, und für jede Position ergeben sich so Schrankenwerte, über denen die Kandidaten an diesen Positionen angenommen, sonst aber verworfen werden. Für unser Beispiel ergeben sich folgende Werte:

k	1	2	3	4	5	6	7	8	9	10
Schranke	850	836	820	800	775	742	695	625	500	0

Wenn sich die Frau an diese Auswahlstrategie hält, hat sie die größtmögliche Chance, den besten Mann auszuwählen.

Die Aufgabenstellung dieses Beispiels genügt dem Optimalitätsprinzip, da jede Teillösung—etwa für zwei, drei, vier oder fünf Kandidaten—wiederum optimal ist und ebenso aus obiger Tabelle abgelesen werden kann. □

Wie man sehen kann, ist dynamische Optimierung auf alle möglichen (und unmöglichen) Probleme anzuwenden. Ein in den letzten Jahren immer bedeutenderer Bereich ist der Vergleich von Zeichenketten geworden. Dabei sollen aus zwei unterschiedlich langen Strings diejenigen Teile herausgesucht werden, die am besten übereinstimmen, wobei Löcher in den Übereinstimmungen zugelassen sind. Diese Problemstellung ist speziell in der Bioinformatik sehr wichtig, da sich die Primärstruktur (eindimensionale Struktur) von DNA, RNA und Proteinen als Zeichenketten darstellen lassen. Da Ähnlichkeiten in der Primärstruktur auf Ähnlichkeiten in der Funktionen schließen lassen, werden unbekannte Nukleotid- und Aminosäuresequenzen mit einer Datenbank von bekannten Sequenzen verglichen, um so Anhaltspunkte für deren Funktion zu erhalten.

Jede Sequenz des menschlichen Genoms besteht aus einer Abfolge der vier Nukleotide *Adenin*, *Guanin*, *Cytosin* und *Thymin*; bei RNA ersetzt *Uracil* das Thymin. Bei Proteinen sind die Möglichkeiten verschiedener Kombinationen größer, da jedes Protein aus bis zu 20 verschiedenen Aminosäuren bestehen kann.

Für das Problem der Bestimmung von Sequenzübereinstimmungen (*sequence alignments*) benötigen wir folgende Definition.

Definition 7.7 (Alignment)
Gegeben sind zwei endliche Sequenzen $x = x_1 \dots x_n$ und $y = y_1 \dots y_m$ über einem endlichen Alphabet Σ und ein Spezialsymbol "$-$" (Leerzeichen). Als *Alignment* von x und y bezeichnet man zwei Strings x' und y' gleicher Länge k mit $n, m \leq k \leq n + m$ über dem Alphabet $\Sigma \cup \{-\}$ mit den Eigenschaften

(i) an keiner Position dürfen sowohl in x' als auch in y' Leerzeichen sein,

(ii) x' und y' ergeben nach Löschen der Leerzeichen wieder x und y.

Für das Übereinstimmung von DNA-Sequenzteilen ist $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, für das Bestimmen von Proteinalignments enthält Σ 20 Symbole. Wir betrachten zuerst einige Beispiele von Alignments, die obiger Definition entsprechen.

Beispiel 7.18 Drei Beispiele möglicher Alignment der Sequenzen `CACT` und `ACGCTT` sind

```
C  −  A  −  C  T
A  C  G  C  T  T
```

und

```
−  C  A  −  C  T
A  C  G  C  T  T
```

sowie

```
−  C  A  C  T  −
A  C  G  C  T  T
```

□

In obigem Beispiel erscheint das erste Alignment schlechter als das zweite, und das wiederum schlechter als das dritte. Das liegt daran, dass im ersten ein, im zweiten zwei, und im dritten drei Nukleotide in beiden Sequenzen übereinstimmen. Wir definieren daher eine Distanzfunktion auf Sequenzpaaren, um so bessere von schlechteren Alignments unterscheiden zu können.

Definition 7.8 (Distanz von Sequenzen)
Seinen x' und y' ein Alignment der Länge k von zweier Sequenzen x und y. Wir definieren die *Distanz* von x' und y' als

$$d(x, y) = \sum_{j=1}^{k} d(x'_j, y'_j),$$

wobei die Distanzfunktion auf Symbolen aus $\Sigma \cup \{-\}$ als

$$d(x'_j, y'_j) = \begin{cases} 1 & \text{wenn } x'_j \neq y'_j \\ 0 & \text{sonst.} \end{cases}$$

gegeben ist.

Obige Definition ist die einfachst mögliche; im Fall $d(x'_j, y'_j) = 1$ sind eigentlich drei Fälle zusammengefasst, die das Auseinanderdriften zweier Genomsequenzen beschreiben können:

Löschen $d(x'_j, -)$ ist der Abstand, der sich durch das Löschen von y'_j ergibt.

Einfügen $d(-, y'_j)$ ist der Abstand, der sich durch das Einfügen von y'_j ergibt.

Ersetzen $d(x'_j, y'_j)$ ist für $x'_j \neq y'_j$ der Abstand, der sich durch Ersetzen von x'_j durch y'_j ergibt.

Da das Alignmentproblem symmetrisch ist, entspricht das Einfügen in eine Sequenz dem Löschen in der anderen und umgekehrt.

In realistischeren Modellen des Sequenzvergleichs (speziell bei Proteinvergleichen) werden diese drei Operationen unterschiedlich gewichtet (auch für unterschiedliche Symbole!). So ist etwa das Ersetzen einer Aminosäure durch eine funktional ähnliche geringer gewichtet als das Löschen oder Ersetzen durch eine funktional unterschiedliche.

Mit Definition 7.8 können wir nun festlegen, welche Alignments optimal sind.

Definition 7.9 (Optimale Alignments)
Ein Aligment x^* und y^* zweier Sequenzen x und y ist *optimal*, wenn gilt

$$d(x^*, y^*) \leq d(x', y')$$

für alle Alignments x' und y' von x und y.

Mit dynamischer Optimierung können optimale Alignments (die meist nicht eindeutig sind) berechnet werden. Dabei wird wie in den letzten Beispielen von kürzestmöglichen Teillösungen ausgegangen, die zu einer vollständigen Lösung erweitert werden.

Wir definieren dazu folgende vereinfachende Schreibweise für Anfangsteile einer Sequenz $x = x_1 \dots x_n$:

$$ {}_0x_j = x_1 \dots x_j. $$

Damit können wir rekursiv Distanzen definieren. Wir beginnen bei leeren Sequenzen, die wir als ${}_0x_0$ schreiben:

$$\begin{aligned} d({}_0x_0,\, {}_0y_0) &= 0 \\ d({}_0x_0,\, {}_0y_i) &= d({}_0x_0,\, {}_0y_{i-1}) + d(-, y_i) \\ d({}_0x_j,\, {}_0y_0) &= d({}_0x_{j-1},\, {}_0y_0) + d(x_j, -). \end{aligned}$$

Mit unseren vereinfachten Distanzen aus Definition 7.8 ist unmittelbar einsichtig, dass $d({}_0x_0,\, {}_0y_i) = i$ und $d({}_0x_j,\, {}_0y_0) = j$ ist.

Der Vergleich beliebiger Sequenzen beruht dann auf dem Optimalitätsprinzip. Um zu sehen, dass das Optimalitätsprinzip gilt, betrachten wir das Ende von Alignments. Diese Enden fallen in eine der drei Kategorien

$$\begin{array}{cc} \dots & x_n \\ \dots & - \end{array}, \qquad \begin{array}{cc} \dots & - \\ \dots & y_m \end{array}, \qquad \begin{array}{cc} \dots & x_n \\ \dots & y_m \end{array}.$$

In jedem dieser Fälle gilt: Wenn das Alignment optimal ist, dann muss auch das Alignment ohne diese letzten Symbole optimal sein, da sonst ein besseres Alignment auf dem ersten Teil durch Anfügen des letzten Symbols besser als das optimale Alignment wäre.

Durch Anwendung des Optimalitätsprinzips kann die Rekursion zur Bestimmung allgemeiner Alignments einfach formuliert werden:

$$\begin{aligned} d({}_0x_i,\, {}_0y_j) = \min\big\{ & d({}_0x_{i-1},\, {}_0y_j) + d(x_i, -) \\ & d({}_0x_i,\, {}_0y_{j-1}) + d(-, y_j) \\ & d({}_0x_{i-1},\, {}_0y_{j-1}) + d(x_i, y_j) \big\} \end{aligned}$$

Dabei entspricht die erste Zeile dem Einfügen des Symbols x_i in die Sequenz x, die zweite Zeile dem Einfügen von y_j in die Sequenz y, und die dritte Zeile dem gleichzeitigen Einfügen von x_i in x und y_j in y. Das optimale Alignment ist dann durch folgenden Satz gegeben.

Satz 7.7 Seien $x = x_1 \dots x_n$ und $y = y_1 \dots y_m$ zwei Sequenzen. Dann ist $d({}_0x_n,\, {}_0y_m)$ die Distanz des optimalen Alignments zwischen x und y.

Um die Minimaldistanz zwischen zwei Sequenzen und damit auch das optimale Alignment auszurechnen konstruiert man eine $(n+1) \times (m+1)$ Matrix, die man von links oben nach rechts unten füllt; im Feld $(n+1, m+1)$ ist dann der minimale Abstand zwischen den Sequenzen abzulesen. Zur Berechnung der Matrixeinträge ist folgende Visualisierung hilfreich:

$$\begin{array}{ccc} d({}_0x_{i-1},\, {}_0y_{j-1}) & & d({}_0x_{i-1},\, {}_0y_j) \\ & \searrow & \downarrow \\ d({}_0x_i,\, {}_0y_{j-1}) & \rightarrow & d({}_0x_i,\, {}_0y_j) \end{array}$$

Der Wert $d({}_0x_i, {}_0y_j)$ in Position $(i+1, j+1)$ der Matrix hängt laut Definition von den Werten der drei Matrixeinträge links, links oberhalb und oberhalb dieser Position ab. Wenn die Sequenzen x und y links bzw. oberhalb der Matrix angeschrieben werden, so entspricht ein Schritt von links nach rechts dem Einfügen eines Leerzeichens in x, ein Schritt von oben nach unten dem Einfügen eines Leerzeichens in y, und ein Schritt von links oben nach rechts unten dem Übereinstimmen bzw. Ersetzen eines Symbols. Wir betrachten dazu ein Beispiel.

Beispiel 7.19 Zu bestimmen sei das optimale Alignment der beiden Sequenzen $x =$ `ACCGTAC` und $y =$ `GCCTAA`. Wir konstruieren dafür eine Matrix mit 8 Zeilen und 7 Spalten, deren erste Zeile und Spalte aus $0, 1, \ldots, 6$ bzw. $0, 1, \ldots, 7$ besteht. Die restlichen Einträge kann man durch sukzessives Anwenden der rekursiven Definition von $d({}_0x_i, {}_0y_j)$ einfüllen. Wir erhalten dann folgende Matrix:

		G	C	C	T	A	A
	0	1	2	3	4	5	6
A	1	**1**	2	3	4	4	5
C	2	2	**1**	2	3	4	5
C	3	3	2	**1**	2	3	4
G	4	3	3	**2**	2	3	4
T	5	4	4	3	**2**	3	4
A	6	5	5	4	3	**2**	3
C	7	6	5	5	4	3	**3**

Damit ist der minimale Abstand zwischen dieser beiden Sequenzen 3. Wenn wir nun von rechts unten nach links oben diejenigen Entscheidungen zurückverfolgen, die in jedem Schritt entlang des Alignments minimalen Abstands gemacht wurden, so erhalten wir die Folge von Matrixeinträgen, die oben durch Quadrate markiert sind. Da die Minimumbestimmung in der der Definition von $d({}_0x_i, {}_0y_j)$ nicht eindeutig sein muss, kann es auch mehrere optimale Alignments geben. Das in obiger Matrix hervorgehobene Alignment ist

```
A C C G T A C
G C C - T A A
```

□

Beispiel 7.20 Dieses Beispiel zeigt, dass optimale Alignments nicht eindeutig sein müssen. Gegeben seinen dazu die beiden Sequenzen `ACAA` und `AGCA` mit untenstehender Alignmentmatrix:

		A	G	C	A
	0	1	2	3	4
A	1	0	1	2	3
C	2	1	1	1	2
A	3	2	2	2	1
A	4	3	3	2	2

Die minimale Distanz ist somit 2; durch verschiedene Pfade von rechts unten nach links oben ergeben sich diese drei optimalen Alignments:

```
A G C A      A G C A -      A G C - A
A C A A ,    A - C A A ,    A - C A A .
```

□

Kapitel 8

Parameterbestimmung in stochastischen Modellen

8.1 Grundbegriffe der Wahrscheinlichkeitsrechnung

Viele Modelle in den Naturwissenschaften und der Medizin beruhen auf Daten, die nicht deterministisch, sondern stochastischer Natur sind. Da die Daten damit eine gewisse Unschärfe haben, variieren auch die auf den Daten basierenden Modelle. Ein Hauptziel der Wahrscheinlichkeitsrechnung ist es, diese Unschärfen zu beschreiben, und damit Aussagen über die von den Daten abgeleiteten Größen zu treffen.

In dieser kurzen Zusammenfassung werden wir uns primär mit denjenigen Aspekten der Wahrscheinlichkeitsrechnung beschäftigen, die für die Bestimmung von Parametern in Modellen wichtig sind. Dafür benötigen wir einige elementare Konzepte. Unter *Zufallsexperimenten* versteht man Versuche, die sowohl *wiederholbar* als auch *zufällig* sind; die *Ausgänge* der Experimente sind also nicht vorhersehbar. Wir verwenden *Zufallsvariable*, um die Ausgänge dieser Experimente zu beschreiben. Zufallsvariable nehmen mit bestimmten Wahrscheinlichkeiten als Werte die möglichen Ausgänge eines Zufallsexperiments an. Wenn die möglichen Ergebnisse eines Zufallsexperiments diskrete Werte sind (nicht notwendigerweise endlich viele), kann das gesamte Zufallsexperiment durch Auflisten der Zufallsvariable und der zugehörigen Werte beschrieben werden.

Beispiel 8.1 Das Zufallsexperiment "Werfen eines Würfels" wird durch folgende Tabelle beschrieben:

k	1	2	3	4	5	6
$P(Z=k)$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

Dabei bezeichnet Z die Zufallsvariable mit Wertebereich $\{1,\ldots,6\}$ und den Wahrscheinlichkeiten $P(Z=k)$, die in obiger Tabelle angegeben sind. □

Mit Hilfe des kartesischen Produkts von Mengen

$$A \times B = \big\{(x,y)\,|\,x \in A \wedge y \in B\big\}$$

lassen sich auch wiederholte Experimente bzw. Experimente mit mehr als einem interessanten Ausgang beschreiben. Wenn die Teilexperimente unabhängig voneinander durchgeführt werden (dies wird weiter unten noch genauer definiert), dann kann man die Wahrscheinlichkeiten der Teilexperimente multiplizieren.

Beispiel 8.2 Das Experiment des *einmaliges* Werfens *zweier* Würfel besteht aus den zwei Teilexperimenten, die in Beispiel 8.1 beschrieben sind. Der Wertebereich der Zufallsvariablen, die die Ausgänge dieses Experiments beschreibt, besteht somit aus zwei Komponenten: der Augenzahl des ersten und der des zweiten Würfels. Man erhält

(i,j)	(1,1)	(1,2)	(1,3)	...	(6,5)	(6,6)
$P(Z=(i,j))$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	...	$\frac{1}{36}$	$\frac{1}{36}$.

□

In Beispielen 8.1 und 8.2 waren alle Ereignisse gleich wahrscheinlich; dies muss aber nicht immer der Fall sein. Sehr wohl aber müssen alle Wahrscheinichkeiten einer Zufallsvariable zu eins aufsummieren. Die Zuteilung der Wahrscheinlichkeitswerte auf die möglichen Ausgänge eines Zufallsexperiments wird *Verteilung* genannt. Wir werden uns im Folgenden zunächst auf diskrete Zufallsvariable beschränken und stetige Zufallsvariable erst in Abschnitt 8.2 behandelt. Weiters verwenden wir die Bezeichnungen W_X für den Wertebereiche der Zufallsvariablen X. Für eine Teilmenge $A \subseteq W_X$ sei

$$P(A) = \sum_{x \in A} P(X = x)$$

die Wahrscheinlichkeit einer Teilmenge von W_X.

Definition 8.1 (Wahrscheinlichkeit, Verteilung)
Sei W_X der Wertebereich der Zufallsvariable X. Dann nennt man eine Funktion

$$P : W_X \to [0,1]$$

eine *Wahrscheinlichkeit* (oder *Verteilung*) von X, wenn folgende zwei Bedingungen gelten:

$$P(W_X) = 1$$
$$P(A_1 \cup A_2 \cup A_3 \cup \dots) = P(A_1) + P(A_2) + P(A_3) + \dots$$

für paarweise disjunkte Teilmengen $A_i \subseteq W_X$.

Wir werden im Folgenden die Schreibweise $P(X)$ für die Verteilung von X verwenden, und austauschbar $P(X = x)$ oder nur $P(x)$ für die Wahrscheinlichkeit eines Einzelereignisses.

Beispiel 8.3 Sei $W_X = \{x_1, \dots, x_7\}$ der Ereignisraum einer Zufallsvariable X. Dann ist durch die Tabelle

W_X	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$P(x_i)$	0.17	0.11	0.01	0	0.09	0.5	0.12

eine Verteilung von X gegeben. □

Ein Bespiel einer Funktion, die Definition 8.1 erfüllt, ist die *relative Häufigkeit*. Diese ist wie folgt definiert.

Definition 8.2 (Relative Häufigkeit)
Ein Zufallsexperiment mit Ereignisraum W_X werde m-mal unabhängig wiederholt; die Zufallsvariable X_i gebe das Ergebnis des i-Versuchs an. Dann bezeichnet man

$$m_i = \left| \{j \mid X_j = x_i \text{ für } j = 1, \ldots, m\} \right|$$

als die absolute Häufigkeit des Auftretens von x_i in der Versuchsreihe; und den Anteil

$$h(x_i) = \frac{m_i}{m},$$

als *relative Häufigkeit* von x_i.

Beispiel 8.4 Eine Münze wird 100 mal geworfen. Das Ereignis KOPF erscheint dabei 46 mal. Damit ist $h(\text{KOPF}) = 0.46$. Wenn die Versuchsreihe auf 1000-maliges Werfen erweitert wird und KOPF 487 mal erscheint, ergibt sich eine relative Häufigkeit von $h(\text{KOPF}) = 0.487$. □

Sei W_X der Wertebereich einer Zufallsvariablen X, und $A, B \subseteq W_X$ zwei Teilmengen. Einige elementare Fakten der Wahrscheinlichkeitsrechnung, wie etwa

$$\begin{aligned} P(\emptyset) &= 0 \\ P(W_X \setminus A) &= 1 - P(A) \\ P(A \cup B) &= P(A) + P(B) - P(A \cap B) \end{aligned}$$

folgen mehr oder weniger direkt aus den beiden Bedingungen in Definition 8.1.

Bei allen Beispielen, die wir bis jetzt betrachtet haben, war die Wahrscheinlichkeit jedes Elementarereignisses gleich groß. Verteilungen dieser Art haben einen speziellen Namen.

Definition 8.3 (Gleichverteilung)
Sei W_X der Ereignisraum einer diskreten Zufallsvariable. Dann nennt man die Verteilung mit

$$P(x) = \frac{1}{|W_X|}$$

für alle $x \in W_X$ die *Gleichverteilung* auf W_X.

Nicht jede Verteilung ist die Gleichverteilung, wie wir an folgendem Beispiel sehen können.

Beispiel 8.5 Beim zweimaligen Werfen eines Würfels gebe die Zufallsvariable X die Summe der beiden Würfel an; der Wertebereich W_X ist somit $\{2, \ldots, 12\}$. Die Verteilung von X ist aber *nicht* die Gleichverteilung: So ist das Auftreten der Augensumme 2 weniger wahrscheinlich als das der Augensumme 3: der erste Fall wird nur durch die Würfelkombination $(1, 1)$ erreicht, der zweite durch $(1, 2)$ und $(2, 1)$. Man kann (etwa durch Abzählen) nachrechnen, dass man folgende Verteilung von X erhält:

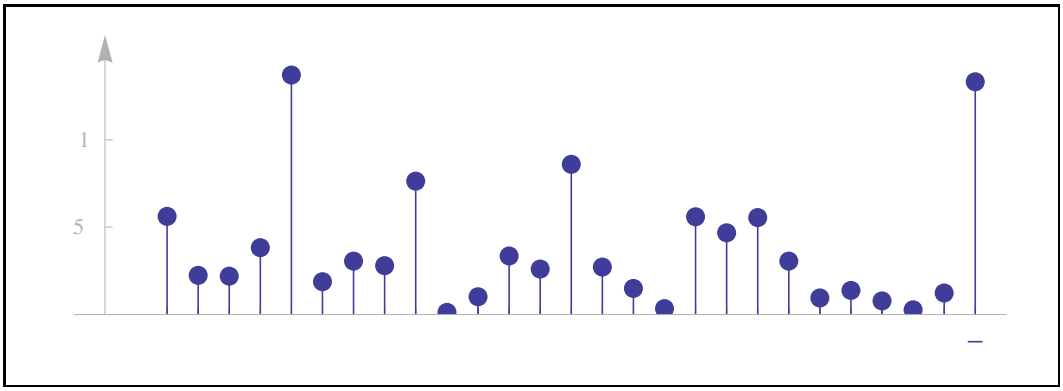

Abbildung 8.1: Graphische Repräsentation der relativen Häufigkeit der Buchstaben `a` bis `z` und des Leerzeichens − in diesem Skriptum.

W_X	2	3	4	5	6	7	8	9	10	11	12
$P(x_i)$	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$.

Für alle anderen Werte x ist $P(x) = 0$. □

Beispiel 8.6 [1] In einem (vereinfachten) Text habe das Experiment "Lesen eines Buchstabens" 27 verschiedene Ausgänge, nämlich die Buchstaben `a` bis `z` und das Leerzeichen, das hier mit − bezeichnet wird. Eine Zufallsvariable X gebe an, welcher dieser Buchstaben gelesen wurde. Die Verteilung dieser Zufallsvariable ist graphisch in Abbildung 8.1 zu sehen. Wenn im Text nur diese Zeichen vorhanden sind, müssen die relativen Häufigkeiten zu 1 aufsummieren. □

Mehrdimensionale Zufallsvariable messen mehr als nur eine Kenngröße eines Zufallsexperiments. So besteht etwa die Zufallsvariable Z in Beispiel 8.2 (zweimaliges Würfels) aus zwei Komponenten (Z_1, Z_2), wobei Z_1 und Z_2 das Ergebnis des ersten bzw. zweiten Würfels angeben.

Definition 8.4 (Zweidimensionale Zufallsvariable)
Seien W_X und W_Y die Ereignisräume zweier Zufallsvariablen X und Y. Dann ist Z eine zweidimensionale Zufallsvariable mit dem Ereignisraum $W_X \times W_Y$.

Verteilungen mehrdimensionaler Zufallsvariable werden als *gemeinsame Verteilung* dieser Zufallsvariable bezeichnet.

Definition 8.5 (Gemeinsame Verteilung zweier Zufallsvariable)
Gegeben sei eine zweidimensionale Zufallsvariable $Z = (Z_1, Z_2)$ mit Wertebereich $W_Z = W_{Z_1} \times W_{Z_2}$. Dann nennt man $P(Z_1, Z_2)$, die Verteilung von Z auf W_Z, die *gemeinsame Verteilung* von Z_1 und Z_2.

[1] Dieses und davon abgeleitete Beispiele sind aus [MacKay, 2003] entnommen.

Beispiel 8.7 Von Bakterienkulturen werden gleichzeitig zwei Kennzahlen Z_1 und Z_2 ermittelt. Die folgende Tabelle gibt an, wie oft unter 10000 Bakterienkulturen bestimmte Kombinationen der beiden Kennzahlen Z_1 und Z_2 aufgetreten sind:

		Z_1				
		0	1	2	3	4
	0	3817	1611	375	127	65
	1	1657	939	199	96	8
Z_2	2	435	228	134	3	2
	3	115	89	7	4	2
	4	78	7	1	0	1

Wenn jede der Kombinationen von Beobachtungen (z_1, z_2) gleich wahrscheinlich ist, kann man aus dieser Tabelle direkt etwa $P(Z_1 = 0, Z_2 = 3) = 0.0115$ ablesen. Durch Aufsummieren von Zeilen bzw. Spalten kann man beispielsweise folgende Schlüsse ziehen: $P(Z_1 = 0) = 0.6102, P(Z_2 = 3) = 0.0217$. Dieses Aufsummieren wird im Folgenden noch speziell betrachtet werden. □

Beispiel 8.8 (Fortsetzung von Beispiel 8.6) Wir betrachten einen Text, den wir in überlappende Segmente der Länge 2 zerlegen. Von jedem Paar von Buchstaben gebe die Zufallsvariable X den ersten, und die Zufallsvariable Y den zweiten Buchstaben an; es gilt somit

$$W_X = W_Y = \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}, -\}.$$

Die gemeinsame Verteilung von X und Y ist dann die relative Häufigkeit von Buchstabenkombinationen. Für den Text dieses Skriptums ist das Ergebnis einer solchen Analyse graphisch in Abbildung 8.2 zu sehen; dabei bilden die ersten Buchstaben die Zeilen, und die zweiten Buchstaben die Spalten der Abbildung. Man kann erkennen, dass über alle Buchstabenkombinationen der Wert von $P(X = \mathtt{e}, Y = \mathtt{n})$ am größten ist. Weitere häufige Kombinationen sind etwa `er` oder der Buchstabe `n` am Wortende. □

Aus der gemeinsamen Verteilung zweier Zufallsvariable können alle Informationen über die einzelnen Zufallsvariablen gewonnen werden. Es ist zu beachten, dass dies umgekehrt nicht der Fall ist: Wenn man also die Verteilungen zweier Zufallsvariable gegeben hat, kann man nicht immer deren gemeinsame Verteilung berechnen.

Das Berechnen von Verteilungen aus gemeinsamen Verteilungen nennt man *Marginalisieren*. Man definiert für diskrete Zufallsvariable den Begriff der *Randverteilung* folgendermaßen.

Definition 8.6 (Randverteilung)
Sei $P(X, Y)$ die gemeinsame Verteilung der Zufallsvariablen X und Y mit Wertebereichen W_X bzw. W_Y. Dann definiert man über

$$P(X = x) = \sum_{y \in W_Y} P(X = x, Y = y)$$

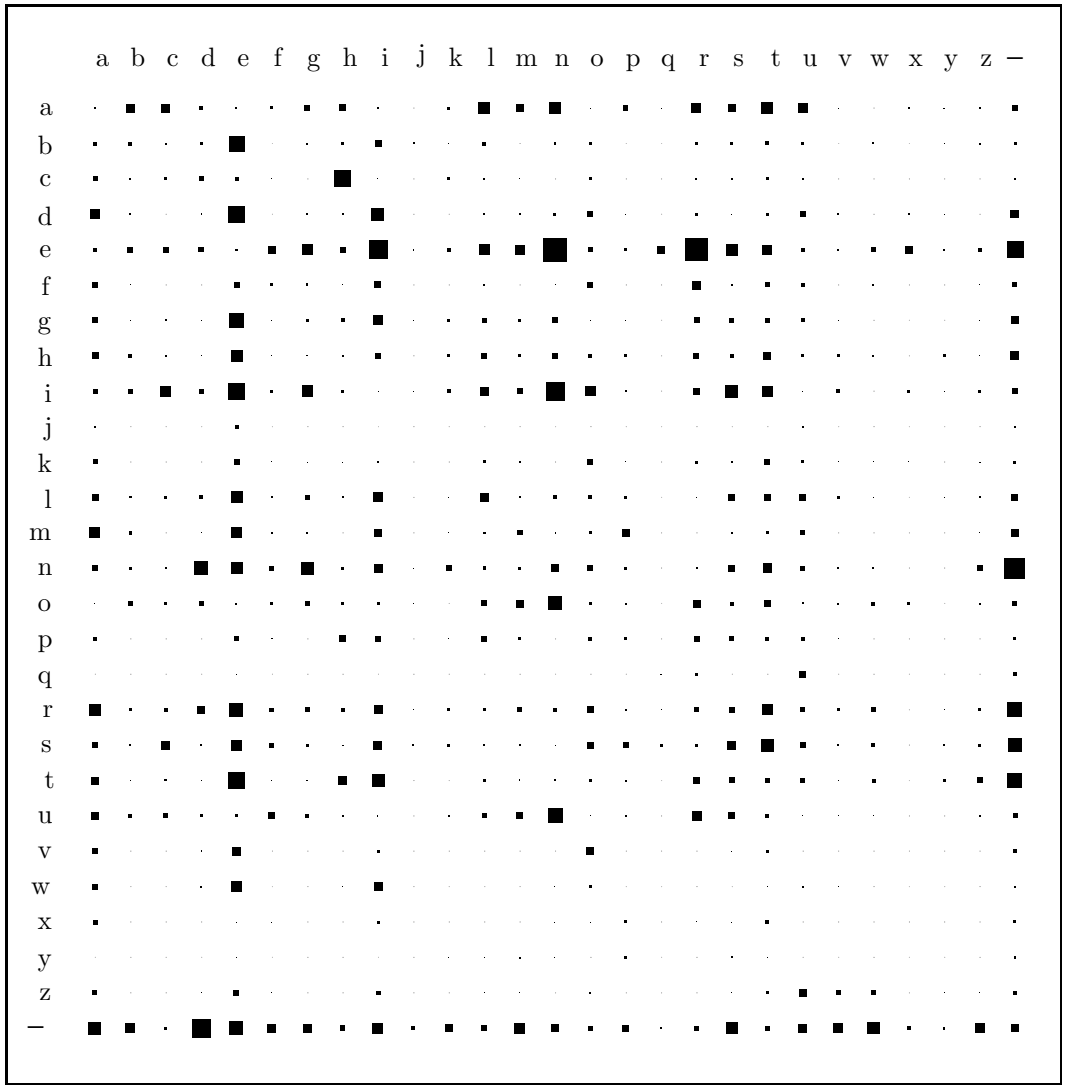

Abbildung 8.2: Gemeinsame Verteilung der beiden Zufallsvariablen, die den ersten bzw. zweiten Buchstaben einer zweielementigen Buchstabenkombination angeben.

und

$$P(Y = y) = \sum_{x \in W_X} P(X = x, Y = y)$$

die *Randverteilungen* $P(X)$ und $P(Y)$ von X bzw. Y.

Diese Verteilungen werden meist kürzer als $P(X) = \sum_{y \in W_Y} P(X, Y = y)$ und $P(Y) = \sum_{x \in W_X} P(X = x, Y)$ geschrieben.

Beispiel 8.9 (Fortsetzung von Beispiel 8.8) In Abbildung 8.2 sind die relativen Häufigkeiten von 162 428 Buchstabenkombinationen zu sehen. Die relativen Häufigkeiten der einzelnen Buchstaben ergeben sich daraus durch Marginalisieren:

Es ist etwa

$$P(X = \texttt{e}) = \sum_{y \in W_Y} P(X = \texttt{e}, Y = y) = 0.144$$

für das Alphabet $W_Y = \{\texttt{a}, \ldots, \texttt{z}, -\}$. Der gleiche Wert würde sich ergeben, wenn man entlang der Spalten (also über alle möglichen Werte von X) aufsummiert. Die gemeinsame Verteilung der Buchstabenkombinationen ist in dem Sinn außergewöhnlich, dass sowohl $P(X)$ als auch $P(Y)$ identisch sind (und mit der Verteilung aus Beispiel 8.8 übereinstimmen). □

Der für die weiteren Ausführungen fundamental wichtigste Begriff ist der der *bedingten Wahrscheinlichkeit*. Dabei wird einer gemeinsamen Verteilung einer der beiden Werte fixiert, wodurch für den anderen wiederum eine Verteilung gegeben ist. Formal definiert man diesen Begriff folgendermaßen.

Definition 8.7 (Bedingte Wahrscheinlichkeit)
Seien X und Y zwei Zufallsvariable mit gemeinsamer Verteilung $P(X, Y)$. Dann nennt man für $P(Y = y) \neq 0$ den Ausdruck

$$P(X = x \,|\, Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)}$$

die *bedingte Wahrscheinlichkeit* von $X = x$ gegeben $Y = y$.

Oft schreibt man für die Gleichung in obiger Definition allgemeiner und kürzer

$$P(X \,|\, Y) = \frac{P(X, Y)}{P(Y)}.$$

Zu beachten ist, dass die bedingte Verteilung wiederum eine Verteilung ist, und damit Definition 8.1 genügt. Meist wird der Nenner $P(Y = y)$ in obiger Definition mit Definition 8.6 als

$$P(Y = y) = \sum_{x \in W_X} P(X = x, Y = y)$$

geschrieben. Durch diesen *Normalisierungsfaktor* wird erreicht, dass bedingte Verteilungen zu 1 aufsummieren.

Beispiel 8.10 (Fortsetzung von Beispiel 8.9) Für die gemeinsame Verteilung von Buchstabenkombinationen, die in Abbildung 8.2 zu sehen ist, können auch bedingte Wahrscheinlichkeiten berechnet werden. So ist etwa

$$P(Y = \texttt{h} \,|\, X = \texttt{c}) = 0.664$$

die Wahrscheinlichkeit, dass der Buchstabe `h` nach dem Buchstaben `c` erscheint. Graphisch sind diese bedingten Wahrscheinlichkeiten in Abbildung 8.3 links zu sehen. In dieser Abbildung werden Verteilungen bedingt auf X-Werte dargestellt; daher repräsentiert jede *Zeile* dieser Graphik eine Wahrscheinlichkeitsverteilung.

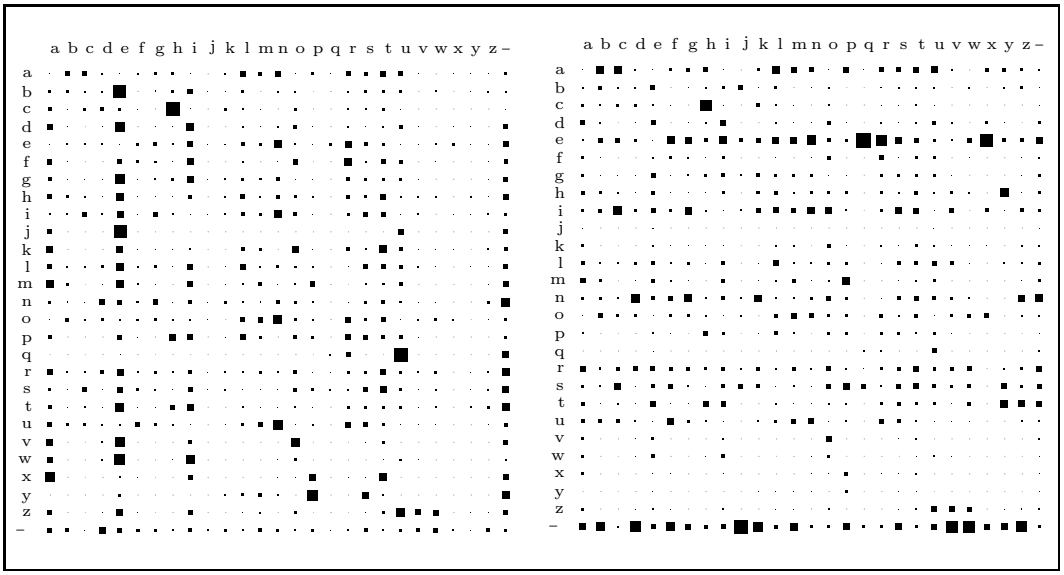

Abbildung 8.3: Die bedingten Wahrscheinlichkeiten $P(Y \mid X)$ (links) und $P(X \mid Y)$ (rechts) für die gemeinsame Verteilung aus Abbildung 8.2. Die Werte von X bilden die Zeilen, die von Y die Spalten der Matrizen. Die Einträge jeder Zeile der linken und jeder Spalte der rechten Abbildung summieren zu 1.

In Abbildung 8.3 rechts sind die bedingten Wahrscheinlichkeiten $P(X \mid Y)$ graphisch aufgelistet. Es ist etwa

$$P(X = \texttt{c} \mid Y = \texttt{h}) = 0.545$$

die Wahrscheinlichkeit, dass ein `c` vor einem `h` auftritt. Wie man sieht, ist dies *nicht* derselbe Wert wie $P(Y = \texttt{h} \mid X = \texttt{c})$, da nach `c` sehr oft `h` vorkommt, aber vor `h` auch öfter andere Buchstaben auftreten. In Abbildung 8.3 rechts sind auf Y bedingte Wahrscheinlichkeiten dargestellt; somit ist jede *Spalte* dieser Abbildung eine Wahrscheinlichkeitsverteilung. □

Über die bedingten Wahrscheinlichkeiten können wichtige Konzepte und Rechenregeln der Wahrscheinlichkeitsrechnung eingeführt werden. Die erste ergibt sich direkt aus einer Umformulierung von Definition 8.7.

Satz 8.1 (Multiplikationsregel) Für zwei Zufallsvariable X und Y mit gemeinsamer Verteilung $P(X, Y)$ gilt

$$P(X, Y) = P(X \mid Y)P(Y) = P(Y \mid X)P(X).$$

Da es in manchen Anwendungsgebieten leichter ist, bedingte Wahrscheinlichkeiten zu bestimmen, ist es über obigen Satz möglich, aus den bedingten Wahrscheinlichkeiten die gemeinsame Verteilung zweier Zufallsvariable zu bestimmen.

Zusammen mit Definition 8.6 (Randverteilung) folgt aus obigem Satz

$$P(X) = \sum_{y \in W_Y} P(X|Y = y)P(Y = y);$$

diese Gleichung werden wir später noch öfters benötigen.

Weiters kann man definieren, dass zwei Zufallsvariable *unabhängig* sind, wenn das Wissen um den Ausgang der einen das Wissen um die andere nicht beeinflusst, wenn also $P(X \mid Y) = P(X)$ bzw. $P(Y \mid X) = P(Y)$ ist. Mit der Multiplikationsregel lässt sich dies noch anders ausdrücken.

Definition 8.8 (Unabhängigkeit von Zufallsvariablen)
Seien X und Y zwei Zufallsvariable mit gemeinsamer Verteilung $P(X, Y)$. Dann nennt man X und Y *unabhängig*, wenn gilt:

$$P(X, Y) = P(X)P(Y).$$

Wir haben das Resultat dieses Satzes bereits eingesetzt, ohne speziell darauf hinzuweisen: Wenn nämlich auf Grund von praktischen Überlegungen die Unabhängigkeit zweier Zufallsvariable gegeben ist, kann man mit Definition 8.8 die Wahrscheinlichkeit des gemeinsamen Auftretens als Produkt berechnen.

Beispiel 8.11 Zwei Würfel werden je einmal geworfen; X gebe die Augenzahl des ersten, und Y die Augenzahl des zweiten Würfels an. Es ist $P(X = k) = \frac{1}{6}$ und $P(Y = j) = \frac{1}{6}$ für $1 \leq k, j \leq 6$. Da die beiden Zufallsvariablen unabhängig sind, gilt $P(X = k, Y = j) = \frac{1}{36}$. □

Der wichtigste Satz im Kontext der bedingten Wahrscheinlichkeiten ist der folgende, der im weiteren Verlauf noch öfter verwendet werden wird.

Satz 8.2 (Satz von Bayes) Seien X und Y zwei Zufallsvariable mit gemeinsamer Verteilung $P(X, Y)$. Dann ist

$$\begin{aligned} P(X \mid Y) &= \frac{P(Y \mid X)P(X)}{P(Y)} \\ &= \frac{P(Y \mid X)P(X)}{\sum_{x \in W_X} P(Y|X = x)P(X = x)}. \end{aligned}$$

Mit dem Satz von Bayes lässt sich also $P(X \mid Y)$ in Bezug zu $P(Y \mid X)$ setzen. Dies ist vor allem im biomedizinischen Bereich oft von Vorteil.

Beispiel 8.12 Ein Patient unterzieht sich einem Screening-Test zur Diagnose einer Krankheit. Der Test habe eine Sensitivität von 98% und eine Spezifizität von 99%. Die Prävelenz der Krankheit in der Bevölkerung sei 0.1%. Der Test liefere für diesen Patienten ein positives Ergebnis. Was ist die Wahrscheinlichkeit, dass der Patient tatsächlich krank ist?

Zur Lösung dieses Problems verwenden wir die Terminologie der Wahrscheinlichkeitsrechnung. Die Zufallsvariable D (disease) habe die zwei möglichen Werte d^+ und d^- (erkrankt bzw. gesund); die Zufallsvariable T (Test) habe die zwei Ausgänge t^+ und t^- (Test positiv bzw. negativ). Gesucht ist die Wahrscheinlichkeit einer Erkrankung, gegeben ein positives Testresultat, also $P(d^+ \mid t^+)$. Die zur Verfügung stehenden Informationen sind

$$P(t^+ \mid d^+) = 0.98 \text{ (Sensitivität)}, \quad P(t^- \mid d^-) = 0.99 \text{ (Spezifizität)}, \text{ und}$$
$$P(d^+) = 0.01 \text{ (Prävalenz)}.$$

Mit dem Satz von Bayes ergibt sich für die gesuchte Größe

$$P(d^+ \mid t^+) = \frac{P(t^+ \mid d^+)P(d^+)}{P(t^+)}.$$

Der Nenner $P(t^+)$ dieses Bruchs ist zwar nicht direkt gegeben, kann aber wie in Satz 8.2 angegeben geschrieben werden als

$$\begin{aligned} P(t^+) &= P(t^+|d^+)P(d^+) + P(t^+|d^-)P(d^-) \\ &= 0.98 \cdot 0.01 + (1 - 0.99) \cdot (1 - 0.01) \\ &= 0.0197. \end{aligned}$$

Man erhält

$$\begin{aligned} P(d^+ \mid t^+) &= \frac{P(t^+ \mid d^+)P(d^+)}{P(t^+)} \\ &= \frac{0.98 \cdot 0.01}{0.0197} \\ &= 0.497. \end{aligned}$$

Damit ist die Wahrscheinlichkeit, dass der Patient erkrankt ist, gerade einmal knapp 50%. Dieser niedrige Wert ist durch die niedrige Prävalenz der Krankheit in der Bevölkerung zu erklären. □

8.2 Verteilungen von Zufallsvariablen

In diesem Abschnitt werden wir einige bekannte Verteilungen von Zufallsvariablen betrachten; diese Ausführungen sollen zur Verwendung dieser Verteilungen in den Abschnitten 8.3 und 8.5 hinarbeiten.

Beispiel 8.13 Eine unfaire Münze mit $P(\text{Kopf}) = \frac{1}{3}$ wird zehnmal geworfen. Die Zufallsvariable Z gebe an, wie oft unter diesen zehn Versuchen das Ereignis Kopf eintritt. Man bestimme die Verteilung von Z.

Wir betrachten zuerst einige "außergewöhnliche" Ereignisse. Die Wahrscheinlichkeit, dass *nie* Kopf geworfen wird, ist $\left(\frac{2}{3}\right)^{10}$ (da jeder Münzwurf unabhängig von den anderen ist); ebenso ist die Wahrscheinlichkeit *immer* Kopf zu werfen $\left(\frac{1}{3}\right)^{10}$. In Verallgemeinerung dieser Situation ist die Wahrscheinlichkeit einer Sequenz

Kopf Kopf Zahl Kopf Kopf Kopf Kopf Kopf Kopf Kopf,

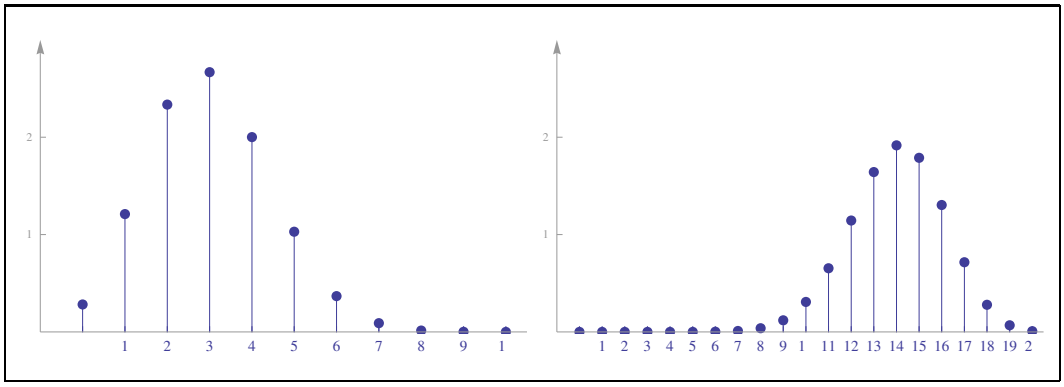

Abbildung 8.4: Dichtefunktionen zweier binomialverteilter Zufallsvariablen, links mit Parametern $n = 10$ und $p = \frac{1}{3}$, rechts mit $n = 20$ und $p = \frac{7}{10}$.

in der neunmal KOPF und einmal ZAHL geworfen wird $\left(\frac{1}{3}\right)^9 \cdot \frac{2}{3}$. Es gibt aber *zehn verschiedene* Möglichkeiten, neunmal KOPF und einmal ZAHL zu werfen (die zehn Positionen, an denen ZAHL in der Liste auftreten kann). Somit ist $P(Z = 9) = 10 \cdot \left(\frac{1}{3}\right)^9 \cdot \frac{2}{3}$.

Man kann sich überlegen, dass es 45 verschiedene Möglichkeiten gibt, zwei ZAHL und acht KOPF in einer Reihe anzuordnen. Im allgemeinen gilt, dass es $\binom{n}{k}$ verschiedene Möglichkeiten gibt, k gleiche Objekte auf n Plätze aufzuteilen. Damit ist eine Formel für die Verteilung von Z in dieser Aufgabe gegeben durch

$$P(Z = k) = \begin{cases} \binom{10}{k} \left(\frac{1}{3}\right)^k \left(\frac{2}{3}\right)^{n-k} & \text{für } 0 \leq k \leq 10 \\ 0 & \text{sonst.} \end{cases}$$

Dabei bezeichnet

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

den *Binomialkoeffizienten* "n über k". □

Im Folgenden werden wir die Notation der bedingten Wahrscheinlichkeit verwenden, um die Abhängigkeit einer Verteilung von Parameterwerten herauszustreichen. Dieser Ansatz (nämlich Parameter wie Zufallsvariable zu verwenden) ist nicht unumstritten. Wir wählen hier den pragmatischen Weg und verwenden ihn trotzdem, da er es erlaubt, auf elegante Weise Aussagen über die Abhängigkeit von Daten von Parametern (und umgekehrt!) zu treffen.

Definition 8.9 (Binomialverteilung)
Eine Zufallsvariable Z heißt *binomialverteilt*, wenn die Verteilung die Form

$$P(X = k \,|\, p, n) = \begin{cases} \binom{n}{k} p^k (1-p)^{n-k} & \text{für } 0 \leq k \leq n \\ 0 & \text{sonst} \end{cases}$$

hat.

Graphische Repräsentationen von Binomialverteilungen mit unterschiedlichen Werten von n und p sind in Abbildung 8.4 zu sehen.

Die wichtigste Verteilung in den Naturwissenschaften und der Biomedizin ist die *Normalverteilung*, auch *Gaußverteilung* genannt. Diese Verteilung unterscheidet sich von den bisher behandelten Verteilungen dadurch, dass sie eine *stetige Verteilung* ist. Von einer stetigen Verteilung spricht man dann, wenn der Wertebereich einer Zufallsvariable die reellen Zahlen sind (und nicht eine diskrete Menge). Dazu benötigt man den Begriff der *Dichtefunktion*.

Definition 8.10 (Dichte)
Eine Funktion $f : \mathbb{R} \to \mathbb{R}^+$ heißt *Dichte*, wenn

$$\int_{\mathbb{R}} f(x)\,dx = 1$$

gilt.

Ebenso wie *jede beliebige* Aufteilung von nichtnegativen Zahlen, die zu eins summieren, eine Wahrscheinlichkeitsverteilung ist, ist jede Funktion mit der Eigenschaft aus obiger Definition eine Dichtefunktion.

Definition 8.11 (Verteilung einer stetigen Zufallsvariable)
Sei X eine stetige Zufallsvariable mit Dichte f. Dann ist für ein Intervall $[a, b]$ die Verteilung von X gegeben durch

$$P(X \in [a, b]) = \int_a^b f(x)dx.$$

Aus dieser Definition kann man den Hauptunterschied zwischen diskreten und stetigen Zufallsvariable ablesen: dass für diskrete Zufallsvariable $P(X = k)$ für zumindest manche Werte k eine positive Zahl ist. Für eine stetige Zufallsvariable X ist wegen

$$P(X = a) = P(X \in [a, a]) = \int_a^a f(x)dx = 0$$

die Wahrscheinlichkeit eines Einzelereignisses a immer null. Wahrscheinlichkeiten von stetigen Zufallsvariablen sind also nicht auf *Punkten*, sondern auf *Intervallen* definiert. Wir werden im Folgenden zur Vereinheitlichung der Notation, und um nicht zwischen diskreten und stetigen Verteilungen unterscheiden zu müssen, auch Dichten mit P bezeichnen. Dann bezeichnen $P(X \in [a, b])$ und $P([a, b])$ die Wahrscheinlichkeit des Intervalls $[a, b]$, $P(x)$ aber die Dichte einer Zufallsvariable.

Beispiel 8.14 Auf einer Kreisscheibe wird vom Mittelpunkt zu einem beliebigen Punkt am Rand eine Linie gezogen. Anschließend wird die Kreisscheibe an die Wand gehängt und wie ein Glücksrad gedreht. Die Zufallsvariable Z bezeichne dabei den Winkel φ (in Radiant), den die aufgezeichnete Linie nach Stillstand mit der Horizontalen einnimmt.

Für jedes Intervall ist die Wahrscheinlichkeit, dass φ in diesem Intervall liegt, proportional zur Größe des Intervalls (bis zu einem größtmöglichen Wert von 2π, welches dem gesamten Wertebereich von Z entspricht).

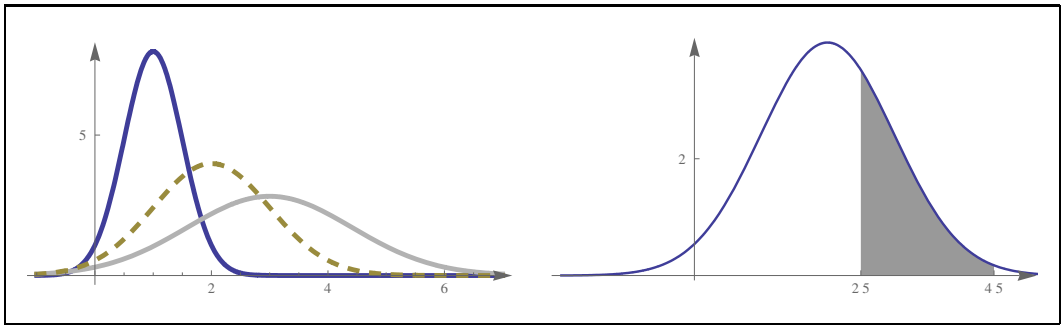

Abbildung 8.5: Links drei Dichten einer Normalverteilung mit Parametern $\mu = 1, \sigma^2 = \frac{1}{4}$ (normal), $\mu = 2, \sigma^2 = 1$ (gestrichelt) und $\mu = 3, \sigma^2 = 2$ (grau). Rechts ist grau der Wert von $P([2.5, 4.5] \,|\, \mu = 2, \sigma^2 = 1) = 0.302$ für eine normalverteilte Zufallsvariable zu sehen.

Über die Bedingung, dass $P(Z \in [0, 2\pi]) = 1$ sein muss, ergibt sich für die Dichte von Z die Funktion

$$P(x) = \begin{cases} \frac{1}{2\pi} & \text{für } x \in [0, 2\pi] \\ 0 & \text{sonst.} \end{cases}$$

Die Wahrscheinlichkeit, dass φ im Intervall $[a, b]$ liegt, ist somit

$$P([a, b]) = \int_a^b P(x)dx = \frac{b-a}{2\pi}.$$

Damit ist etwa die Wahrscheinlichkeit eines Viertelkreises $P([\pi, \frac{3}{2}\pi]) = \frac{1}{4}$, wie zu erwarten war. □

Wir sind nun in der Lage, die Normalverteilung zu definieren. Diese Verteilung spielt deswegen eine so große Rolle, weil sie die Verteilung einer Summe unabhängiger, identisch verteilter Zufallsvariablen ist (wir werden darauf aber nicht weiter eingehen).

Definition 8.12 (Normalverteilung)
Eine stetige Verteilung mit Dichte

$$P(x \,|\, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

heißt *Normalverteilung* (oder *Gaußverteilung*) mit Parametern μ und σ^2. Abkürzend schreiben wir für diese Verteilung meist $\mathrm{N}(\mu, \sigma^2)$.

Die Verteilung einer $N(\mu, \sigma^2)$-verteilten Zufallsvariable kann nur numerisch über

$$P([a, b] \,|\, \mu, \sigma^2) = \int_a^b \frac{1}{\sqrt{2\pi\sigma^2}} \, e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx$$

berechnet werden, da es zu diesem Integral keine Stammfunktion gibt.

Einige Formen der Dichtefunktion einer normalverteilten Zufallsvariable sind in Abbildung 8.5 links zu sehen. Rechts ist die Wahrscheinlichkeit eines Intervalls für eine normalverteilte Zufallsvariable dargestellt.

Die Parameter μ und σ^2 einer Normalverteilung können folgendermaßen interpretiert werden:

- Der Parameter μ gibt das Maximum der Dichtefunktion an; somit sind Werte um μ am wahrscheinlichsten. Der Durchschnitt einer Reihe von $\mathrm{N}(\mu, \sigma^2)$-verteilten Zufallsvariablen nähert sich diesem Wert an.
- Der Parameter σ^2 ist ein Maß für die Streuung der Verteilung. Für kleines σ^2 sind die Realisierungen von $\mathrm{N}(\mu, \sigma^2)$-verteilten Zufallsvariablen eher in der Nähe von μ; für größere Werte von σ^2 werden größere Abstände von μ immer wahrscheinlicher.

8.3 Maximum Likelihood Schätzungen

Wir werden uns in diesem Abschnitt mit der Problemstellung beschäftigen, aus einer Menge von Daten Rückschlüsse auf die Verteilung zu ziehen, die diese Daten generiert hat.

Wir werden eine Folge $x_1, \ldots, x_n$ betrachten, die als Realisierungen einer Folge $X_1, \ldots, X_n$ von unabhängigen, identisch verteilten Zufallsvariablen aufgefasst werden. Das nächste Beispiel soll den Unterschied zwischen einer *konkreten Stichprobe* $x_1, \ldots, x_n$ und einer *mathematischen Stichprobe* $X_1, \ldots, X_n$ klarmachen.

Beispiel 8.15 Ein Würfel werde zehnmal geworfen. Dann beschreibt die Zufallsvariable X_i den Ausgang des i-ten Würfelns; alle X_i sind unabhängig und identisch verteilt, nämlich gleichverteilt auf der Menge $\{1, \ldots, 6\}$. Eine mögliche konkrete Ausprägung von $X_1, \ldots, X_{10}$ wäre

$$(x_1, \ldots, x_{10}) = (1, 4, 2, 2, 5, 6, 4, 3, 6, 2),$$

eine andere

$$(x_1, \ldots, x_{10}) = (3, 6, 4, 1, 3, 5, 2, 4, 6, 1). \qquad \square$$

Das Ziel von Parameterschätzmethoden ist es nun, auf Basis einer konkreten Stichprobe den (bestmöglichen) Wert eines oder mehrerer Parameter derjenigen Verteilung anzugeben, die die konkrete Stichprobe erzeugt hat. Dazu benötigen wir noch etwas Terminologie.

Definition 8.13 (Likelihood-Funktion)
Seien X eine Zufallsvariable mit Dichte $P(x \mid \theta)$ und $x_1, \ldots, x_n$ eine konkrete Stichprobe von n unabhängigen, wie X verteilten Zufallsvariablen. Dann nennt man

$$L(\theta) := P(\{x_1, \ldots, x_n\} \mid \theta) = \prod_{k=1}^{n} P(x_k \mid \theta)$$

die *Likelihood-Funktion* von θ.

Es erscheint nun sinnvoll, als bestmöglichen Wert des Parameters θ denjenigen zu wählen, der die Daten maximal wahrscheinlich macht (bzw. im Fall stetiger Verteilungen die Dichten maximiert).

Definition 8.14 (Maximum-likelihood-Schätzer)
Seien $P(x \mid \theta)$ und $x_1, \ldots, x_n$ wie in Definition 8.13. Dann nennt man

$$\theta_{\mathrm{ML}} := \arg\max_{\theta} L(\theta)$$

den *Maximum-Likelihood-Schätzer* von θ.

Je nach Verteilung kann die Bestimmung von θ_{ML} mathematisch kompliziert sein. Ein einfaches Beispiel soll helfen, diese Begriffe zu erläutern. Wir werden schnell sehen, dass selbst einfache Aufgabenstellungen einiges an technischem Aufwand benötigen.

Beispiel 8.16 Eine Münze werde 1000-mal geworfen. Bei jedem einzelnen Wurf sei θ die Wahrscheinlichkeit des Ereignisses Kopf. Auf Basis der Versuchsreihe $x_1, \ldots, x_{1000}$ soll der beste Wert von θ bestimmt werden.

Zur Lösung dieser Aufgabestellung benötigen wir die Verteilung von X_i, die beim i-ten Wurf das Auftreten von Kopf oder Zahl angibt; es muss $P(X_i = \text{Kopf}) = \theta$ und $P(X_i = \text{Zahl}) = 1 - \theta$ gelten. Wenn wir Kopf durch 1 und Zahl durch 0 repräsentieren, kann man die Verteilung von X_i schreiben als

$$P(X_i = x \mid \theta) = \theta^x \, (1-\theta)^{(1-x)}.$$

Mit dieser Definition erhält man wie gewünscht

$$P(\text{Kopf}) = P(X_i = 1 \mid \theta) = \theta^1 \, (1-\theta)^{(1-1)} = \theta$$

und

$$P(\text{Zahl}) = P(X_i = 0 \mid \theta) = \theta^0 \, (1-\theta)^{(1-0)} = 1 - \theta.$$

Die Likelihood-Funktion von θ ist für observierte Daten $x_1, \ldots, x_n$ (die alle entweder 0 oder 1 sind) somit

$$\begin{aligned} L(\theta) &= P(\{x_1, \ldots, x_n\} \mid \theta) \\ &= \prod_{i=1}^{n} \theta^{x_i} (1-\theta)^{(1-x_i)} = \theta^{x_1 + \cdots + x_n} (1-\theta)^{n - x_1 - \cdots - x_n}. \end{aligned}$$

Dieser Ausdruck kann noch vereinfacht werden: So gibt $x_1 + \cdots + x_n$ an, wie oft in der Versuchsreihe Kopf aufgetreten ist. Mit $K = x_1 + \cdots + x_n$ ergibt sich für die Likelihood-Funktion von θ

$$L(\theta) = \theta^K (1-\theta)^{n-K}.$$

Diese Funktion ist für eine konkrete Versuchsreihe, in der 512-mal Kopf und 488-mal Zahl aufgetreten ist, in Abbildung 8.6 links zu sehen. Der senkrechte Strich bei 0.512 zeigt die Position des Maximums dieser Funktion.

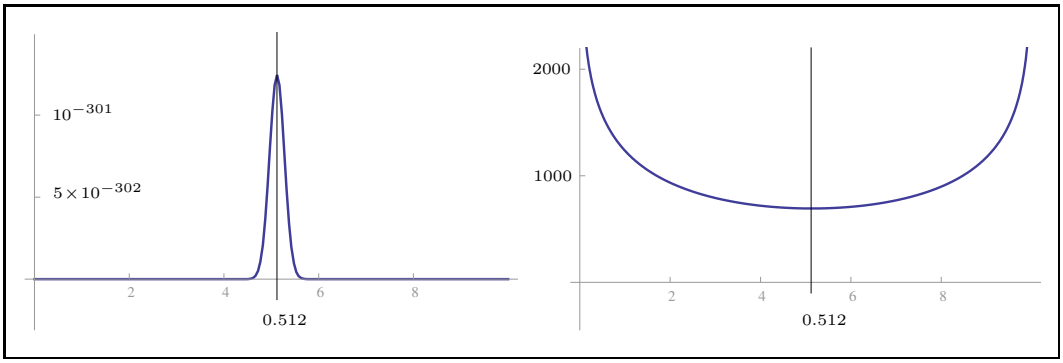

Abbildung 8.6: Links die Likelihood-Funktion $L(\theta)$ aus Beispiel 8.16, rechts $-\log L(\theta)$. Man kann erkennen, dass sich die Position des Optimums durch Logarithmieren nicht ändert.

Uns bleibt somit nur mehr, diesen plausiblen Wert auch numerisch zu bestimmen. Wir suchen den Wert θ, der den Ausdruck $\theta^K(1-\theta)^{n-K}$ maximiert. Rein rechnerisch ist es bei dieser Aufgabenstellung (wie auch bei vielen anderen Optimierungsproblemen der Statistik) leichter, das Minimum von $-\log\bigl(\theta^K(1-\theta)^{n-K}\bigr)$ zu bestimmen: Durch das Logarithmieren wird nämlich das Produkt aus Definition 8.13 in eine Summe umgewandelt, und damit leichter zu behandeln. Die wichtige Erkenntnis dabei ist, dass sich die Position des Optimums auf der x-Achse durch das Logarithmieren nicht ändert. Dies ist graphisch in Abbildung 8.6 rechts zu sehen: Das Minimum von $-\log f(x)$ ist immer an der selben Stelle wie das Maximum von $f(x)$. Die Formulierung als Minimierungsproblem (statt des gegebenen Maximierungsproblems) hat hauptsächlich historische Gründe, da früher Optimierungssoftware oftmals nur Minimierungsalgorithmen implementiert hat.

Für $L(\theta) = \theta^K(1-\theta)^{n-K}$ erhalten wir mit der Rechenregel $\log(ab) = \log(a) + \log(b)$ die Vereinfachung

$$-\log L(\theta) = -K\log\theta - (n-K)\log(1-\theta).$$

Das Minimum dieses Ausdrucks befindet sich an der Stelle, an der die Ableitung null wird. Es gilt wegen $\log'(x) = 1/x$

$$-\frac{d\,\log L(\theta)}{d\,\theta} = -\frac{K}{\theta} + \frac{n-K}{1-\theta}.$$

Nullsetzen dieses Ausdrucks liefert

$$-K(1-\theta) + \theta(n-K) = 0$$

und mit kurzem Umformen schließlich das erwartete

$$\theta = \frac{K}{n}.$$

Wir erhalten somit für unser Zahlenmaterial den Maximum-Likelihood-Schätzwert $\theta_{\mathrm{ML}} = 0.512$. □

Das letzte Beispiel zeigt gleichzeitig die Mächtigkeit und den gravierenden Nachteil des Maximum-Likelihood-Ansatzes auf: Es ist rein aus den Daten möglich, einen optimalen (und noch dazu plausiblen) Wert des unbekannten Parameters einer Verteilung zu bestimmen. Da dieser Wert aber *nur* von den Daten abhängt, unterliegt er auch der Schwankungsbreite der Daten: So kann bei der nächsten Versuchsreihe etwa $\theta_{\mathrm{ML}} = 0.496$ und bei der nächsten $\theta_{\mathrm{ML}} = 0.523$ bestimmt werden. Wir werden in Abschnitt 8.5 eine Ansatz kennenlernen, mit dem diese Unscärfen vermieden werden.

In den nächsten Beispielen werden wir uns mit Maximum-Likelihood-Schätzern für die Parameter der Normalverteilung beschäftigen.

Beispiel 8.17 Von einer konkreten Stichprobe $x_1, \ldots, x_n$ sei bekannt, dass sie von einer Normalverteilung generiert wurde. Von dieser Normalverteilung sei weiters σ^2 bekannt, nicht aber μ (es ist in vielen technischen Bereichen tatsächlich möglich, nur einen von mehreren Parametern einer Verteilung zu kennen). Man bestimme aus $x_1, \ldots, x_n$ den Maximum-Likelihood-Schätzer μ_{ML} für μ.

In diesem Beispiel ist die Likelihood-Funktion

$$L(\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x_i-\mu)^2}{2\sigma^2}},$$

und für $-\log L(\mu)$ gilt

$$\begin{aligned}
-\log L(\mu) &= -\log\left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right) \\
&= -\sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right) \\
&= -\sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \sum_{i=1}^{n} \log\left(e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right) \\
&= n\log(\sqrt{2\pi\sigma^2}) + \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2} \\
&= n\log(\sqrt{2\pi\sigma^2}) + \frac{1}{2\sigma^2}\sum_{i=1}^{n} (x_i-\mu)^2.
\end{aligned}$$

Wir sind am Minimum dieses Ausdrucks interessiert; da der erste Teil nicht von den Daten $x_1, \ldots, x_n$ abhängt kann er weggelassen werden. Nullsetzen der Ableitung des verbleibenden Terms liefert wegen

$$-\frac{d\,\log L(\mu)}{d\,\mu} = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\big(-2(x_i-\mu)\big)$$

die Gleichung

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu) = 0,$$

die umgeformt

$$\sum_{i=1}^{n} x_i = n\mu$$

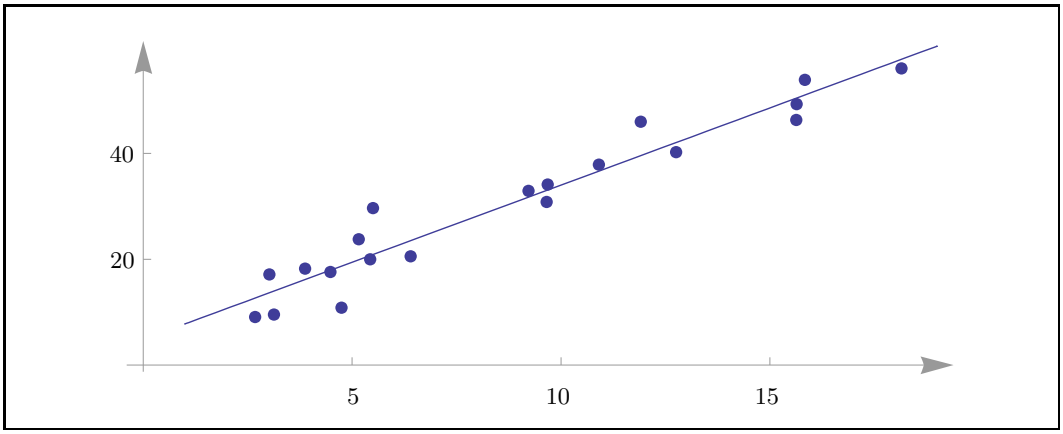

Abbildung 8.7: Illustration zur linearen Regression: Da die Werte y_i fehlerbehaftet sind, liegen die Datenpunkte (x_i, y_i) nicht auf der bestapproximierenden Gerade.

und damit

$$\mu_{\mathrm{ML}} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

ergibt. Der im Maximum-Likelihood-Sinn beste Schätzwert des Parameters μ einer Normalverteilung ist damit der Mittelwert der Stichprobe $x_1, \ldots, x_n$. Mit diesem Schätzwert kann durch eine ähnliche Rechnung der Maximum-Likelihood-Schätzwert

$$\sigma^2_{\mathrm{ML}} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{\mathrm{ML}})^2$$

hergeleitet werden. □

8.4 Lineare und logistische Regression

Als Anwendung der Maximum-Likelihood-Methode, die wir in Abschnitt 8.3 kennengelernt haben, betrachten wir nun zwei *Regressionsprobleme*. Von Regressionsproblemen spricht man normalerweise, wenn man auf Basis von Daten ein Modell zur Vorhersage eines reellzahligen Wertes konstruiert (im Gegensatz zu *Klassifikationsproblemen*, bei denen das Modell die Klassenzugehörigkeit eines Datenpunktes entscheidet). Wir werden auch erkennen, dass die logistische Regression trotz ihres Namens eigentlich ein Klassifikationsproblem löst.

Diese Unterscheidung wird klarer, wenn wir die Aufgabenstellungen genauer formalisieren. In beiden Fällen sind Datenpunkte $x_1, \ldots, x_n$ gegeben. Bei der *linearen* Regression sind zusätzlich noch reelle Werte $y_1, \ldots, y_n$ gegeben, die eine Funktion auf den Werten $x_1, \ldots, x_n$ annimmt; bei der *logistischen* Regression sind dies binäre Werte $t_1, \ldots, t_n$, die die Zugehörigkeit der Datenpunkte zu einer von zwei Klassen anzeigen.

Wir werden zuerst den Fall der linearen Regression genauer betrachten. Ein Beispiel eines linearen Regressionsproblems (und seiner Lösung) ist in Abbildung 8.7 zu sehen. Es soll auf Basis der Daten $x_1, \ldots, x_n$ und $y_1, \ldots, y_n$ ein Modell der

Abhängigkeit der y-Werte von den x-Werten gefunden werden, um für einen neuen neuen Wert x_k einen "möglichst guten" Wert y_k vorhersagen zu können. Dabei stehen die Werte y_i mit den Werten x_i durch

$$y_i = f(x_i) + E_i$$

in Zusammenhang. Dabei ist f das Modell, das den Zusammenhang von x_i und y_i ausdrückt, und die E_i die ZV, die erklären sollen, warum kein direkter (deterministischer) Zusammenhang zwischen x_i und y_i besteht. In Abbildung 8.7 ist f eine lineare Funktion $y = kx + d$; aus den konkreten Daten $x_1, \ldots, x_n$ und $y_1, \ldots, y_n$ sollen die bestmöglichen Werte dieser beiden Parameter bestimmt werden. Ohne Fehler (wenn der Zusammenhang also nur $y_i = f(x_i)$ wäre) würden alle Datenpunkte in Abbildung 8.7 auf der Gerade liegen. Wir nehmen nun an, dass die E_i normalverteilt sind mit konstanten Parametern $\mu = 0$ und σ^2. Das bedeutet, dass kein systematischer Fehler gemacht wird, und dass die Streubreite des Fehlers bei allen Datenpunkten gleich groß ist. Bei anderen Verteilungsannahmen ist die folgende Herleitung entsprechend zu modifizieren.

Wir betrachten die konkrete Stichprobe $y_1, \ldots, y_n$ als Realisierungen von ZV $Y_1, \ldots, Y_n$. Da die Fehler E_i $\mathrm{N}(0, \sigma^2)$-verteilt sind, folgt aus $y_i = f(x_i) + E_i$, dass die Y_i ebenfalls normalverteilt sind, und zwar mit Parametern $\mu_i = f(x_i)$ und konstantem σ^2.

Wir drücken im Folgenden die Abhängigkeit des Modells $f(x)$ von den Parametern explizit aus, um den Maximum-Likelihood-Ansatz anwenden zu können. Für den einfachsten Fall wählen wir, wie schon in Abbildung 8.7 das Modell

$$f(x;\, \theta) = \theta_1 x + \theta_2.$$

Andere (komplizierte) Modell sind mit diesem Ansatz ebenso leicht behandelbar, solange sie *linear in den Parametern* sind. Das bedeutet, dass f (als Funktion von θ gesehen) linear ist.

Unter Verwendung des Modells $f(x;\, \theta)$ und unter der Annahme, dass die gemessenen y-Werte nur durch $\mathrm{N}(0, \sigma^2)$-verteilte Fehler vom vorhergesagten Wert $f(x;\, \theta)$ abweichen, ergibt sich als Dichtefunktion für die Verteilung der ZV Y_i in Abhängigkeit von der ZV X_i der Ausdruck

$$P(y \,|\, x;\, \theta, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(y - f(x;\,\theta))^2}{2\sigma^2}},$$

woraus man die Likelihood-Funktion

$$\begin{aligned} L(\theta) &= P(\{y_1, \ldots, y_n\} \,|\, \{x_1, \ldots, x_n\};\, \theta, \sigma^2) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(y_i - f(x_i;\,\theta))^2}{2\sigma^2}} \end{aligned}$$

bestimmen kann. Maximieren von $\log L(\theta)$ verläuft analog zur Herleitung des Resultats in Beispiel 8.17 und liefert

$$\log L(\theta) = \log\left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(y_i - f(x_i;\,\theta))^2}{2\sigma^2}} \right)$$

$$
\begin{aligned}
&= [\dots] \\
&= -n \log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - f(x_i;\, \theta))^2.
\end{aligned}
$$

Zur Bestimmung des Minimums muss dieser Ausdruck nun nach θ abgeleitet und nullgesetzt werden. Obige Notation eignet sich nicht gut für weitere Berechnungen; wir wechseln nun zu Matrixnotation. Mit $\theta = (\theta_1, \theta_2)$ kann man das angenommene lineare Modell schreiben als

$$
f(x;\, \theta) = \theta_1 x + \theta_2 = (\theta_1, \theta_2) \cdot \begin{pmatrix} x \\ 1 \end{pmatrix}.
$$

Die gesamte Summe $\sum_{i=1}^{n} (y_i - f(x_i;\, \theta))^2$ kann man mit den Notationen

$$
Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \qquad \text{und} \quad X = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_n & 1 \end{pmatrix}
$$

in Matrizenschreibweise angeben als

$$
\sum_{i=1}^{n} (y_i - f(x_i;\, \theta))^2 = (Y - X \cdot \theta)^T \cdot (Y - X \cdot \theta).
$$

Der Vorteil der Matrizenschreibweise liegt darin, dass man nach Vektoren wie nach "normalen" Variablen ableiten kann. Man erhält

$$
\begin{aligned}
\operatorname{grad} \log L(\theta) &= \operatorname{grad}_\theta \Big(- n \log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2} (Y - X \cdot \theta)^T \cdot (Y - X \cdot \theta) \Big) \\
&= 0 - \frac{1}{2\sigma^2} \big(-2 X^T \cdot (Y - X \cdot \theta) \big).
\end{aligned}
$$

Nullsetzen dieses Ausdrucks liefert

$$
X^T \cdot (Y - X \cdot \theta) = 0,
$$

welcher durch Ausmultiplizieren auf die Form

$$
X^T \cdot Y = X^T \cdot X \cdot \theta
$$

gebracht werden kann, woraus schließlich

$$
\theta_{\mathrm{ML}} = \big(X^T \cdot X \big)^{-1} \cdot X^T \cdot Y
$$

folgt.

Beispiel 8.18 Durch die Datenpunkte

x_i	5	6	7	10	12	15	18	20
y_i	7.4	9.3	10.6	15.4	18.1	22.2	24.1	22.8

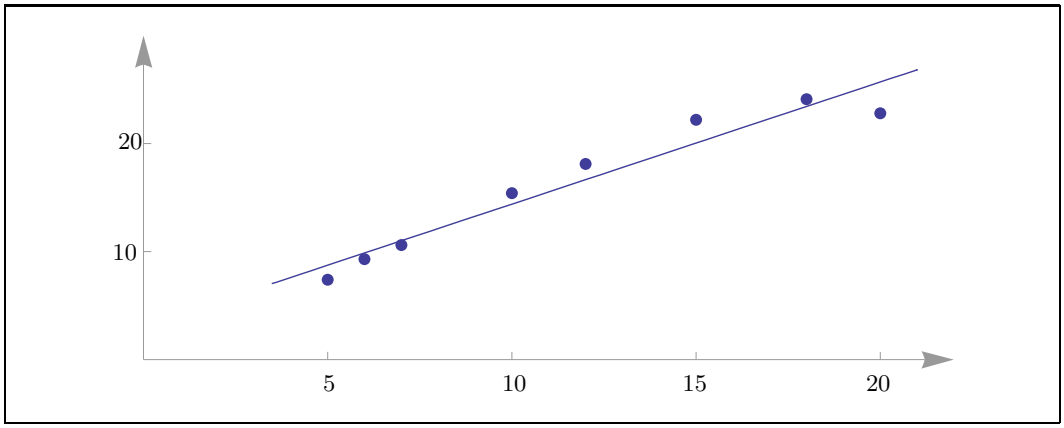

Abbildung 8.8: Die acht Datenpunkte aus Beispiel 8.18, zusammen mit dem durch den Maximum-Likelihood-Schätzwert θ_{ML} definierten linearen Modell.

soll ein Modell der Form $f(x;\,\theta) = \theta_1 x + \theta_2$ gelegt werden. Die Maximum-Likelihood-Schätzwerte für den Parametervektor θ erhält man wie in obiger Herleitung. Für das Zahlenmaterial dieses Beispiels ist

$$X = \begin{pmatrix} 5 & 6 & 7 & 10 & 12 & 15 & 18 & 20 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}^T$$

und

$$Y = (7.4, 9.3, 10.6, 15.4, 18.1, 22.2, 24.1, 22.8)^T .$$

Damit ist

$$\left(X^T \cdot X\right)^{-1} = \begin{pmatrix} 0.0045 & -0.0524 \\ -0.0524 & 0.734 \end{pmatrix},$$

und insgesamt

$$\theta_{\mathrm{ML}} = \left(X^T \cdot X\right)^{-1} \cdot X^T \cdot Y = \begin{pmatrix} 1.13 \\ 3.09 \end{pmatrix}.$$

Die Datenpunkte und die Gerade $y = 1.13x + 3.09$ sind graphisch in Abbildung 8.8 zu sehen; der empirische Korrelationskoeffizient $r_{x,y} = 0.968$ weist auch darauf hin, dass die die Datenpunkte durch die lineare Funktion gut approximiert werden. □

An dieser Stelle soll noch Folgendes vermerkt sein: Ein hoher Wert von $r_{x,y}$ bedeutet nicht notwendigerweise, dass sich der Zusammenhang zwischen zwei ZV X und Y optimal durch eine Gerade $Y = aX + b$ beschreiben lässt. Der Korrelationskoeffizient $r_{x,y}$ gibt vielmehr an, um wieviel besser sich der Zusammenhang zwischen X und Y durch $Y = aX + b$ als nur durch eine Konstante $Y = b$ ausdrücken lässt. Dadurch ist allerdings noch nicht ausgesagt, dass nicht etwa eine höhergradige Abhängigkeit besteht. Dies wird im nächsten Beispiel illustriert.

Beispiel 8.19 (Fortsetzung von Beispiel 8.18) Wir untersuchen nochmals die selben Daten wie im letzten Beispiel. Durch graphische Betrachtung der Datenpunkte kann man sich überzeugen, dass sich der Zusammenhang zwischen X und Y besser durch

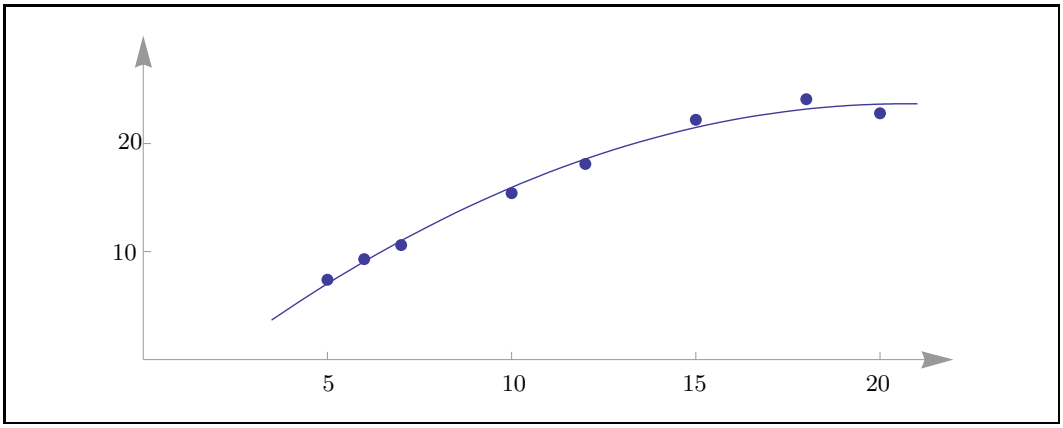

Abbildung 8.9: Quadratische Approximation der Datenpunkte aus Beispiel 8.18, ebenfalls durch den Maximum-Likelihood Ansatz bestimmt.

eine quadratische Funktion wiedergeben lässt. Dies entspricht dem quadratischen Modell

$$f(x;\,\theta) = \theta_2 x^2 + \theta_1 x + \theta_0 = (\theta_2, \theta_1, \theta_0) \cdot \begin{pmatrix} x^2 \\ x \\ 1 \end{pmatrix}$$

bei der Herleitung der linearen Regression; es muss nur die Matrix X entsprechend abgeändert werden.

Obwohl der Zusammenhang zwischen X und Y nichtlinear ist, so ist die Funktion $f(x;\,\theta)$ doch *linear in den Parametern*, sodass lineare Regression auch hier den optimalen Parametervektor θ bestimmen kann.

Die quadratische Approximation $y = -0.07x^2 + 2.80x - 5.27$ liefert eine bessere Annäherung an die Datenpunkte, wie graphisch in Abbildung 8.9 zu sehen ist. Dies lässt sich auch dadurch nachrechnen, dass man sowohl für das lineare als auch für das quadratische Modell die Summe der Fehlerquadrate berechnet (die ja durch die Modelle minimiert werden). Dies ergibt beim linearen Modell einen Wert von 18.86, beim quadratischen Modell aber nur 2.92. □

Bei der *logistischen Regression* sind die vorherzusagenden Werte y_i nicht mehr reelle Zahlen, sondern binäre Klassenzugehörigkeitsindikatoren. Wir wollen nun auf Basis von $x_1, \ldots, x_n$ und $y_1, \ldots, y_n$ ein Modell finden, das für einen neuen Wert x_k die Klassenzugehörigkeit y_k bestimmt; dabei seien die beiden Klassen als 0 bzw. 1 kodiert. Mit dem hier präsentierten Ansatz ist es sogar möglich, eine *Wahrscheinlichkeit* der Klassenzugehörigkeit anzugeben; dies ist in den meisten Fällen einer rein dichotomen Entscheidung vorzuziehen. Das gesuchte Modell soll für gegebenen x-Wert

$$P(y = 1 \,|\, x)$$

repräsentieren; wegen $P(y = 0 \,|\, x) = 1 - P(y = 1 \,|\, x)$ können damit bei einem dichotomen Problem die Wahrscheinlichkeiten der Zugehörigkeit zu beiden Klassen bestimmt werden. Die Problemstellung der logistischen Regression ist in Abbildung 8.10 links dargestellt: Zwei Punktmengen sollen durch eine gerade Linie

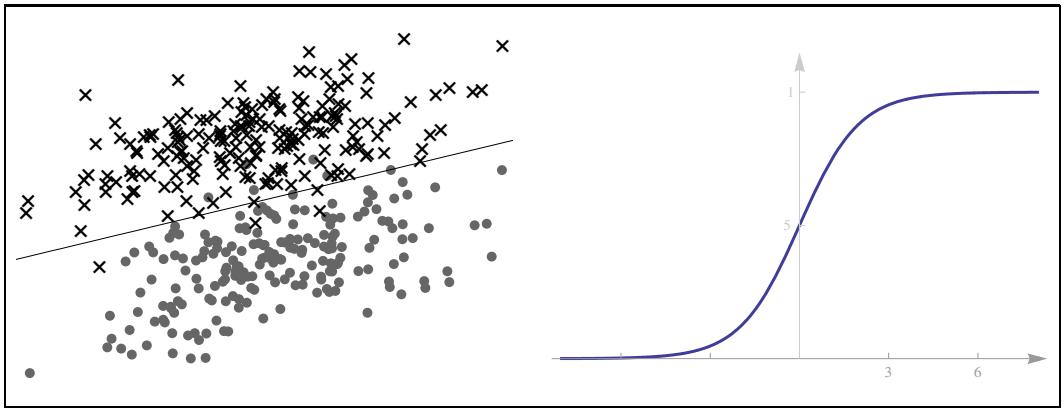

Abbildung 8.10: Illustration zur logistischen Regression: Links eine Menge von Datenpunkten, die zwei Klassen angehören, und die mittels logistischer Regression bestimmte optimale Trennlinie. Rechts ist der Graph der logistischen Funktion $f(x) = 1/(1 + e^{-x})$ zu sehen.

möglichst optimal getrennt werden; das Optimalitätskriterium wird später noch genauer angegeben. Die optimale Trennlinie repräsentiert die Menge aller Punkte x mit

$$P(y = 1 \,|\, x) = P(y = 0 \,|\, x) = 0.5,$$

also die Menge aller Punkte, bei denen die Klassenzugehörigkeit nicht eindeutig festgestellt werden kann. Diese Linie trennt die Menge der Datenpunkte in zwei Teilmengen. Wir benötigen nun noch eine Funktion, um den beiden Regionen von Punkten mit

$$P(y = 1 \,|\, x) > 0.5 \qquad \text{bzw.} \quad P(y = 1 \,|\, x) < 0.5$$

Wahrscheinlichkeiten zuordnen zu können. Dazu verwendet man die *logistische Funktion*

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

deren Wertebereich (wie für eine Wahrscheinlichkeit gefordert) zwischen 0 und 1 liegt. Der Graph dieser Funktion ist in Abbildung 8.10 rechts zu sehen. Dabei wird im Modell x durch den Abstand von der Trennlinie ersetzt, sodass Punkte mit sehr großem Abstand Wahrscheinlichkeiten nahe bei 0 bzw. 1 haben, und Punkte nahe der Trennlinie Wahrscheinlichkeiten um die 0.5 ergeben. Da der Abstand eines zweidimensionalen Punktes $p = (p_1, p_2)$ von einer Trennlinie $\theta_1 x_1 + \theta_2 x_2 + \theta_0 = 0$ proportional zu $\theta_1 p_1 + \theta_2 p_2 + \theta_0$ ist, kann das logistische Modell mit den Abkürzungen $\bar{x} = (1, x_1, x_2)^T$ und $\theta = (\theta_0, \theta_1, \theta_2)^T$ geschrieben werden als

$$P(y = 1 \,|\, x;\, \theta) = \frac{1}{1 + e^{-\theta^T \cdot \bar{x}}}.$$

Man kann nachrechnen, dass unter bestimmten Verteilungsannahmen dieses Modell korrekt ist; wenn diese Annahmen nicht erfüllt sind, liefert das Modell immer noch eine gute Approximation.

Um im Folgenden die Notation zu vereinfachen, schreiben wir für die Modellausgabe kürzer

$$p(x;\,\theta) = P(y = 1 \,|\, x;\,\theta)$$

und setzen auch $x := \bar{x}$, unterscheiden also nicht mehr zwischen x und seiner um die Konstante 1 erweiterten Form.

Für den Maximum-Likelihood-Ansatz benötigen wir einen Ausdruck für die $P(y \,|\, x;\,\theta)$, und nicht nur $P(y = 1 \,|\, x;\,\theta)$ (vergleiche dazu auch das Münzwerfen in Beispiel 8.16). Dies ist etwa durch

$$P(y \,|\, x;\,\theta) = p(x;\,\theta)^y \cdot \big(1 - p(x;\,\theta)\big)^{1-y}$$

möglich, da daraus wie gewünscht

$$P(1 \,|\, x;\,\theta) = \frac{1}{1 + e^{-\theta^T \cdot \bar{x}}} \qquad \text{und} \qquad P(0 \,|\, x;\,\theta) = 1 - P(1 \,|\, x;\,\theta)$$

folgen.

Mit diesem Ausdruck erhält man die Likelihood-Funktion

$$\begin{aligned} L(\theta) &= P\big(\{y_1, \ldots, y_n\} \,|\, \{x_1, \ldots, x_n\};\,\theta\big) \\ &= \prod_{i=1}^{n} \Big(p(x_i;\,\theta)^{y_i} \cdot \big(1 - p(x_i;\,\theta)\big)^{1-y_i} \Big) \,. \end{aligned}$$

An dieser Stelle benötigen wir zwei unterschiedliche Schreibweisen für die Größen, die in diesem Ausdruck vorkommen. Es gelten

$$p(x;\,\theta) = \frac{1}{1 + e^{-\theta^T \cdot x}} = \frac{e^{\,\theta^T \cdot x}}{1 + e^{\,\theta^T \cdot x}}$$

und

$$1 - p(x;\,\theta) = \frac{1}{1 + e^{\,\theta^T \cdot x}} \,.$$

Damit ist

$$\begin{aligned} \log L(\theta) &= \sum_{i=1}^{n} \Big(y_i \log p(x_i;\,\theta) + (1 - y_i) \log \big(1 - p(x_i;\,\theta)\big) \Big) \\ &= \sum_{i=1}^{n} \Big(y_i \log \big(e^{\,\theta^T \cdot x_i}\big) - y_i \log(1 + e^{\,\theta^T \cdot x_i}) - (1 - y_i) \log(1 + e^{\,\theta^T \cdot x_i}) \Big) \\ &= \sum_{i=1}^{n} \Big(y_i\, \theta^T \cdot x_i - \log \big(1 + e^{\,\theta^T \cdot x_i}\big) \Big) \,. \end{aligned}$$

Wie bei linearer Regression berechnen wir hier die Ableitung von $\log L(\theta)$, um das Maximum dieser Funktion zu bestimmen. Dies liefert

$$\begin{aligned} \operatorname{grad} \log L(\theta) &= \sum_{i=1}^{n} \Big(y_i x_i - \frac{1}{1 + e^{\,\theta^T \cdot x_i}}\, e^{\,\theta^T \cdot x_i}\, x_i \Big) \\ &= \sum_{i=1}^{n} x_i \big(y_i - p(x_i;\,\theta)\big) \,. \end{aligned}$$

Das Problem an dieser Stelle ist, dass nach Nullsetzen dieses Ausdrucks θ nicht freigestellt werden kann, es also keine direkte (exakte) Lösung der Gleichung $\operatorname{grad} \log L(\theta) = 0$ gibt. Deshalb verwendet man zur Optimierung der Likelihood-Funktion mehrdimensionale Optimierungsverfahren. Für die Parameterbestimmung bei logistischer Regression eignet sich besonders gut eine Variante des Quasi-Newton-Verfahrens aus Abschnitt 7.4, bei der die Hesse-Matrix direkt invertiert (und die Inverse nicht nur approximiert) wird.

Nochmaliges Ableiten liefert

$$\begin{aligned}
\operatorname{Hess} \log L(\theta) &= \operatorname{grad} \sum_{i=1}^{n} x_i \big(y_i - p(x_i;\, \theta)\big) = -\sum_{i=1}^{n} x_i \cdot \operatorname{grad} p(x_i;\, \theta) \\
&= -\sum_{i=1}^{n} x_i \cdot \operatorname{grad} \frac{1}{1 + e^{-\theta^T \cdot x_i}} = -\sum_{i=1}^{n} x_i \cdot \frac{x_i^T\, e^{-\theta^T \cdot x_i}}{\big(1 + e^{-\theta^T \cdot x_i}\big)^2} \\
&= -\sum_{i=1}^{n} x_i \cdot x_i^T \frac{e^{-\theta^T \cdot x_i}}{\big(1 + e^{-\theta^T \cdot x_i}\big)^2} = -\sum_{i=1}^{n} x_i \cdot x_i^T \frac{1}{1 + e^{-\theta^T \cdot x_i}} \frac{e^{-\theta^T \cdot x_i}}{1 + e^{-\theta^T \cdot x_i}} \\
&= -\sum_{i=1}^{n} x_i \cdot x_i^T\, p(x_i;\, \theta)\big(1 - p(x_i;\, \theta)\big)\,.
\end{aligned}$$

Das iterative Lösen von $\operatorname{grad} \log L(\theta) = 0$ über das Newtonverfahren lässt sich (wie bei der linearen Regression) am einfachsten in Matrizenschreibweise formulieren. Seien dazu Y der Spaltenvektor der y_i, X die Datenmatrix, die zeilenweise aus den Vektoren x_i besteht (mit einer letzten Spalte von Einsern), P der Spaltenvektor von $p(x_i;\, \theta_k)$, und W die Diagonalmatrix mit Einträgen $p(x_i;\, \theta_k)\big(1 - p(x_i;\, \theta_k)\big)$. Dann ist

$$\operatorname{grad} \log L(\theta) = X^T(Y - P) \qquad \text{und} \qquad \operatorname{Hess} \log L(\theta) = -X^T W X,$$

und der Newton-Schritt $x_{k+1} = x_k - f'(x_k)/f''(x_k)$ zur Bestimmung der Nullstelle von $f'(x)$ ist für das logistische Regressionsmodell

$$\theta_{k+1} = \theta_k + (X^T W X)^{-1} X^T (Y - P)\,.$$

Zu beachten ist, dass P und W von θ_k abhängen, und sich in jedem Iterationsschritt ändern. Als Startwert setzt man meist $\theta_0 = 0$.

Beispiel 8.20 Gegeben seien 400 Datenpunkte in zwei Klassen, für die ein logistisches Regressionsmodell erstellt werden soll. Nach numerischer Optimierung des Logarithmus der Likelihood-Funktion $L(\theta)$ erhält man den Parametervektor $\theta = (2.36, 3.05, -7.66)$. Damit ist die Menge aller Punkte

$$\big\{(x_1, x_2) \,|\, 2.36x_1 + 3.05x_2 - 7.66 = 0\big\}$$

die optimale Trennlinie zwischen den beiden Klassen, da an diesen Punkten x für die Klassenzugehörigkeit

$$P(y = 1 \,|\, x;\, \theta) = \frac{1}{1 + e^{-\theta^T \cdot \bar{x}}} = \frac{1}{1 + e^0} = \frac{1}{2}$$

gilt. In Abbildung 8.11 sind die beiden Punktmengen und diese Trennlinie dargestellt. Ebenfalls zu sehen sind die beiden Geraden, für die $P(y = 1 \,|\, x;\, \theta) = 0.75$ bzw. $P(y = 1 \,|\, x;\, \theta) = 0.25$ ist. □

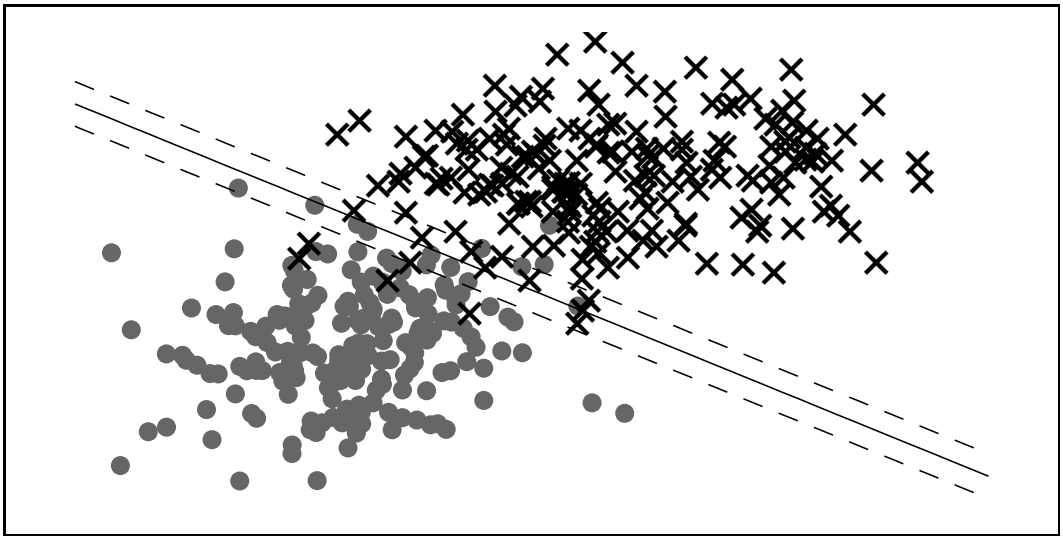

Abbildung 8.11: Die zwei Punktmengen aus Beispiel 8.20. Zu sehen sind außerdem die drei Linien mit $P(y = 1 \,|\, x;\, \theta) = 0.5$ (durchgezogen) und $P(y = 1 \,|\, x;\, \theta) = 0.25$ bzw. $P(y = 1 \,|\, \theta, x) = 0.75$ (gestrichelt).

8.5 Maximum A-Posteriori Schätzungen

Der im letzten Abschnitt behandelte Ansatz, die optimalen Parameterwerte eines Modells durch Maximierung der Likelihood-Funktion

$$L(\theta) = P(\{x_1, \ldots, x_n\} \,|\, \theta)$$

mit Daten $x_1, \ldots, x_n$ zu bestimmen, erscheint zumindest plausibel. Mit diesem Ansatz wird derjenige Parameterwert θ_{ML} gefunden, mit dem die Daten maximal wahrscheinlich werden.

Eine kurze Überlegung zeigt, dass man das Problem der Parameterschätzung auch von einer anderen Warte aus betrachten könnte: Man könnte ebenso gut den "wahrscheinlichsten" Parameterwert bestimmen, also denjenigen, für den die bedingte Wahrscheinlichkeit $P(\theta \,|\, \{x_1, \ldots, x_n\})$ maximal wird. Wenn wir diesen Ansatz wählen, bekommen wir nicht nur *einen* besten Wert, sondern eine ganze *Verteilung*: damit lässt sich (über die Streuung der Verteilung) auch die Unschärfe des optimalen Parameterwerts quantifizieren.

Zur Bestimmung der Verteilung $P(\theta \,|\, \{x_1, \ldots, x_n\})$ kann man den Satz von Bayes verwenden (Satz 8.2): Es gilt

$$P(\theta \,|\, \{x_1, \ldots, x_n\}) = \frac{P(\{x_1, \ldots, x_n\} \,|\, \theta)\, P(\theta)}{P(\{x_1, \ldots, x_n\})}$$

Man bezeichnet $P(\theta)$ als die *A-priori-Verteilung* von θ, und $P(\theta \,|\, \{x_1, \ldots, x_n\})$ als *A-posteriori-Verteilung* von θ. Der Nenner $P(\{x_1, \ldots, x_n\})$ ist ein *Normalisierungsfaktor*: Da für die A-Posteriori-Verteilung (wie für jede Verteilung bzw. Dichte)

$$\int_0^1 P(\theta \,|\, \{x_1, \ldots, x_n\})\, d\theta = 1$$

gelten muss, folgt aus

$$\int_0^1 P(\theta \,|\, \{x_1, \ldots, x_n\})\, d\theta = \frac{1}{P(\{x_1, \ldots, x_n\})} \int_0^1 P(\{x_1, \ldots, x_n\} \,|\, \theta)\, P(\theta) = 1$$

die Bedingung

$$P(\{x_1, \ldots, x_n\}) = \int_0^1 P(\{x_1, \ldots, x_n\} \,|\, \theta)\, P(\theta).$$

Da der Normalisierungsfaktor nicht von θ abhängt, muss er nicht in allen Berechnungen mitgeführt werden, sondern kann erst am Ende bestimmt werden. Mit den oben eingeführten Bezeichnungen kann die Bestimmung der A-posteriori-Verteilung auch als

$$\text{A-posteriori} = \frac{\text{Likelihood} \times \text{A-priori}}{\text{Normalisierungsfaktor}}$$

geschrieben werden. Wegen der Verwendung des Satzes von Bayes wird diese Methode zur Parameterschätzung auch *Bayesianischer Ansatz* genannt.

Um die A-posteriori-Verteilung rechnerisch bestimmen zu können, benötigen wir somit neben dem Normalisierungsfaktor noch die Likelihood-Funktion und eine A-priori-Verteilung von θ. Die Likelihood-Funktion wurde bereits in Abschnitt 8.3 behandelt; eine A-priori-Verteilung von θ gibt das Wissen um θ *vor* Eintreffen der Daten wieder. Die Notwendigkeit einer A-priori-Verteilung von θ ist zugleich der größte Nachteil und der größte Vorteil des Bayesianischen Ansatzes: der größte Nachteil, da (laut Kritikern) dadurch subjektives Empfinden in den Analyseprozess eingehen kann; der größte Vorteil, da (laut Befürwortern) alle Annahmen und deren Unsicherheiten genau quantifiziert werden müssen.

Wir werden auf diese Problematik im Folgenden nicht näher eingehen sondern anhand von Beispielen zeigen, welchen Einfluss die A-priori-Verteilung und die Daten auf die A-posteriori-Verteilung haben. Wenn wir aus der Bayesianischen Analyse nur *einen besten* Parameterwert ermitteln sollen, so ist dieser das Maximum der A-posteriori-Verteilung, also

$$\theta_{\text{MAP}} := \arg\max_{\theta} P(\theta \,|\, \{x_1, \ldots, x_n\}).$$

Dieser Wert wird *Maximum-A-posteriori-Schätzer* von θ genannt.

Beispiel 8.21 Für eine unfaire Münze sei $P(\text{Kopf}) = 0.6$. Wenn wir mit dieser Münze Datenmaterial erhalten, bei dem die relative Häufigkeit von Kopf 0.6 ist, dann wissen wir aus Beispiel 8.16, dass der beste Schätzwert θ_{ML} (im Maximum-Likelihood-Sinn) für die unbekannte Wahrscheinlichkeit $P(\text{Kopf})$ gleich 0.6 ist. Wir werden nun ein ähnliches Problem mit dem Ansatz von Bayes behandeln, um die Unterschiede und Ähnlichkeiten dieser Betrachtungsweise herauszuarbeiten.

Wenn wir über die unbekannte Münze keine Informationen haben, so können wir unser Unwissen durch eine A-priori-Verteilung

$$P(\theta) = \begin{cases} 1 & \text{für } 0 \le \theta \le 1 \\ 0 & \text{sonst} \end{cases}$$

ausdrücken, bei der jeder Parameterwert zwischen 0 und 1 gleich wahrscheinlich ist. Durch Eintreffen der Daten wird diese Verteilung modifiziert, und man erhält

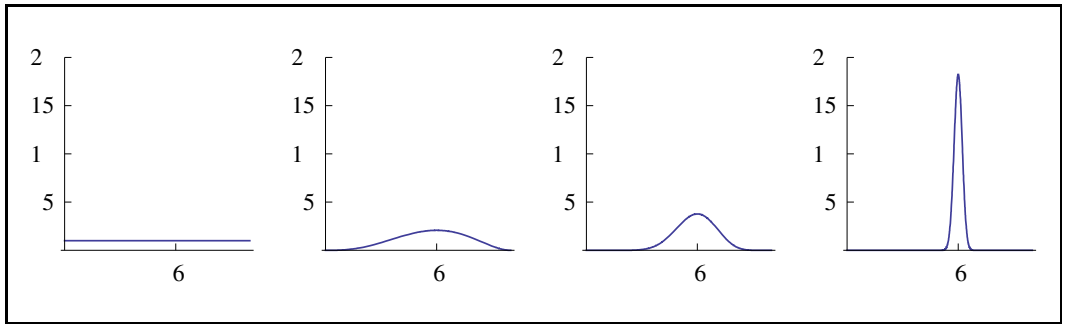

Abbildung 8.12: Die A-posteriori-Verteilungen für $\theta = P(\text{KOPF})$ aus Beispiel 8.21. Links die A-priori-Verteilung $P(\theta) = 1$, dann die A-posteriori-Verteilungen für $n = 5, K = 3$, für $n = 20, K = 12$, und schließlich für $n = 500, K = 300$.

daraus die A-posteriori-Verteilung von θ. Mit K-maligem KOPF unter n Würfen ist die Likelihood-Funktion

$$L(\theta) = \theta^K (1-\theta)^{n-K},$$

wie wir in Beispiel 8.16 ausgearbeitet haben. Für den Normalisierungsfaktor gilt

$$\begin{aligned} P(\{x_1, \ldots, x_n\}) &= \int_0^1 P(\{x_1, \ldots, x_n\} \,|\, \theta) P(\theta)\, d\theta \\ &= \int_0^1 \theta^K (1-\theta)^{n-K} \cdot 1 \, d\theta \\ &= \frac{K!(n-K)!}{(K + (n-K) + 1)!} \end{aligned}$$

Die Umformung in der letzten Zeile werden wir nicht herleiten, sondern als gegeben annehmen.

Mit diesen Herleitungen können wir die A-posteriori-Verteilung angeben als

$$P(\theta \,|\, \{x_1, \ldots, x_n\}) = \frac{(n+1)!}{K!(n-K)!}\, \theta^K (1-\theta)^{n-K} \cdot 1.$$

Diese Verteilung ist für verschiedene Werte von n und K (für die immer $K/n = 0.6$ gilt) in Abbildung 8.12 zu sehen. Man kann erkennen, dass die Verteilung umso stärker um 0.6 konzentriert ist, je mehr Münzwürfe durchgeführt werden. In allen drei dargestellten Fällen ist $\theta_{\text{MAP}} = 0.6$. Mit unserer Wahl der A-priori-Verteilung bestimmen somit allein die Daten den Wert von θ_{MAP}.

Wir können die A-priori-Verteilung aber auch dazu verwenden, um bestimmte Überzeugungen oder Vorwissen auszudrücken. Wenn wir die Berechnung der A-posteriori-Verteilung algebraisch durchführen wollen, muss das Produkt der A-priori-Verteilung und der Likelihood-Funktion wieder als Verteilung darstellbar sein; andernfalls kann man numerische Methoden zur Evaluierung der A-posteriori-Verteilung verwenden. Für die Likelihood-Funktion $L(\theta) = \theta^K (1-\theta)^{n-K}$ bietet sich die *Beta-Verteilung* als A-priori-Verteilung $P(\theta)$ an, die für ganzzahlige Parameter α und β die Dichte

$$P(\theta \,|\, \alpha, \beta) = \frac{(\alpha+\beta-1)!}{(\alpha-1)!(\beta-1)!}\, \theta^{\alpha-1} (1-\theta)^{\beta-1}$$

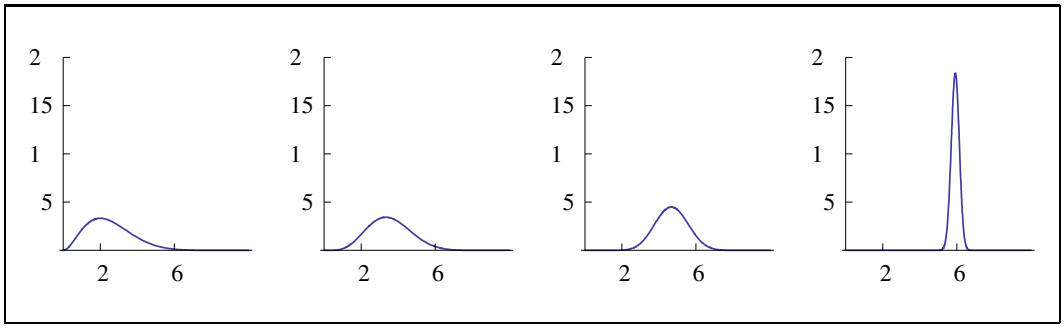

Abbildung 8.13: Die A-posteriori-Verteilungen für $\theta = P(\text{KOPF})$ in Beispiel 8.21. Links die A-priori-Verteilung $P(\theta) = \text{Beta}(3, 9)$, dann die A-posteriori-Verteilungen für $n = 5, K = 3$, für $n = 20, K = 12$, und schließlich für $n = 500, K = 300$.

besitzt. Parameter wie α und β, die die Verteilung von anderen Parametern angeben, werden *Hyperparameter* genannt. Die a-priori-Meinung, θ wäre ungefähr 0.2, kann etwa durch eine Beta-Verteilung mit Parametern $\alpha = 3$ und $\beta = 9$ ausgedrückt werden; diese Verteilung ist in Abbildung 8.13 links zu sehen.

Wir werden für dieses Beispiel den Normalisierungsfaktor erst nach Bestimmen der A-posteriori-Verteilung normalisieren. Dann ist

$$\begin{aligned} P(\theta \,|\, \{x_1, \ldots, x_n\}) &\propto P(\{x_1, \ldots, x_n\} \,|\, \theta)\, P(\theta \,|\, \alpha, \beta) \\ &= \theta^K (1-\theta)^{n-K} \frac{(\alpha+\beta-1)!}{(\alpha-1)!(\beta-1)!}\, \theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &\propto \theta^K (1-\theta)^{n-K}\, \theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &= \theta^{K+\alpha-1}(1-\theta)^{n-K+\beta-1} \end{aligned}$$

In dieser Herleitung haben wir auf alle multiplikativen Faktoren verzichtet, da diese einfacher in einem abschließenden Normalisierungsschritt berechnet werden können. Wenn wir den letzten Term mit der Dichte der Beta-Verteilung vergleichen, so erkennen wir, dass der Term einer nicht-normalisierten Beta-Verteilung mit Parametern $\alpha + K$ und $\beta + n - K$ entspricht Für diese Verteilung ist der Normalisierungsfaktor aus der Dichte ablesbar, weswegen er nicht berechnet werden muss. Damit erhalten wir als normalisierte A-posteriori-Verteilung

$$P(\theta \,|\, \{x_1, \ldots, x_n\}) = \frac{1}{Z}\, \theta^{\alpha+K-1}(1-\theta)^{\beta+n-K-1}.$$

mit dem Normalisierungsfaktor

$$\frac{1}{Z} = \frac{1}{\int_0^1 \theta^{\alpha+K-1}(1-\theta)^{\beta+n-K-1}\, d\theta} = \frac{(\alpha+\beta+n-1)!}{(\alpha+K-1)!(\beta+n-K-1)!}.$$

Die A-posteriori-Verteilungen für dieselben Kombinationen von n und K wie in Abbildung 8.12 sind in Abbildung 8.13 zu sehen. Man kann erkennen, dass unsere A-priori-Meinung, der Parameter θ wäre ungefähr 0.2, durch eine steigende Anzahl von Versuchen mit $K/n = 0.6$ widerlegt wird. Für $n = 500$ in Abbildung 8.13 rechts ist $\theta_{\text{MAP}} \approx 0.6$. Dies bedeutet, dass bei zunehmender Anzahl von Daten die Bedeutung der A-priori-Verteilung abnimmt. Man kann diese Tatsache auch daran

erkennen, dass sich die rechtesten Graphiken in den Abbildungen 8.12 und 8.13 nicht unterscheiden, obwohl sie auf unterschiedlichen A-priori-Verteilungen beruhen. □

Zusammenfassend kann man den Bayesianischen Ansatz zur Parameterschätzung folgendermaßen charakterisieren:

- Eine A-posteriori-Verteilung eines Parameters wird durch den Einfluss von Daten auf die A-priori-Verteilung des Parameters bestimmt.
- Das Maximum der A-priori-Verteilung gibt eine Vermutung über den Parameterwert an; die Streuung dieser Verteilung gibt an, wie sicher man sich dieser Vermutung ist.
- Je mehr Daten zur Verfügung stehen, desto geringer ist der Einfluss der A-priori-Verteilung auf die A-posteriori-Verteilung.
- Für $n \to \infty$ konvergiert θ_{MAP} gegen θ_{ML}.
- Für $n \to \infty$ geht die Streuung der A-posteriori-Verteilung, und damit die Unsicherheit im Wert von θ_{MAP}, gegen null.

Kapitel 9

Erzeugung von Zufallszahlen

In vielen Anwendungen, besonders in der Simulation realer Ereignisse, benötigt man Zufallszahlen. Es liegt auf der Hand, dass man mit einer streng deterministischen Maschine, wie es der Computer ist, keine echt zufällige Zahlenreihe erzeugen kann. Trotzdem werden in vielen Anwendungsbereichen (wie etwa Spielen) vom Computer generierte Zahlen verwendet, die sich praktisch nicht von echten Zufallszahlen unterscheiden. Man bezeichnet solche vom Computer generierte Zahlen als *Pseudozufallszahlen*. Am häufigsten werden dabei Zahlen benötigt, die auf einer endlichen Menge (oder auf einem Intervall) gleichverteilt sind. Da vom Computer nur endlich viele Zahlen dargestellt werden können, ist die Verteilung dieser Zahlen immer eine diskrete Verteilung. Für die meisten Anwendungen kann man aber aufgrund der großen Menge von Maschinenzahlen die generierten Pseudozufallszahlen auch als stetig verteilt betrachten. Nachdem nun auf den Unterschied zwischen "echten" Zufallszahlen und Pseudozufallszahlen hingewiesen wurde, werden wir im Folgenden nur mehr von Zufallszahlen sprechen.

Die Generierung von Zufallszahlen ist in praktisch allen höheren Programmiersprachen über Systembefehle leicht möglich. Dies ist aber nur dann empfehlenswert, wenn man keine allzuhohen Ansprüche an die Qualität dieser Zahlen stellt, da die Implementierung etwa in C und C++ nicht standardisiert ist und auf veralteten Algorithmen basiert. Der theoretische Hintergrund der einfachsten Methode zur Generierung von gleichverteilten Zufallszahlen wird in Abschnitt 9.1 erläutert. Die Aufgabenstellung wird interessanter, wenn Zufallszahlen anderer Verteilungen generiert werden sollen. Dies wird in den Abschnitten 9.3 und 9.4 besprochen. Nach einem Einschub über mehrdimensionale Integrale in Abschnitt 9.5 behandeln wir in Abschnitt 9.6 und Abschnitt 9.7 die Generierung von normalverteilten Zufallszahlen.

9.1 Gleichverteilte Zufallszahlen

Zur Generierung gleichverteilter Zufallszahlen hat man bis vor Kurzem meist *lineare Kongruenzverfahren* verwendet, die aber im Vergleich zu neueren (besseren) Methoden deutliche qualitative Nachteile aufweisen. Bei allen (auch den besseren) Verfahren wird stets die nächste Zufallszahl als Funktion der letzten Zufallszahl berechnet. Um gleiche Zahlenfolgen zu generieren, kann der Startwert der Folge (der *seed value*) über eigene Befehle gesetzt werden.

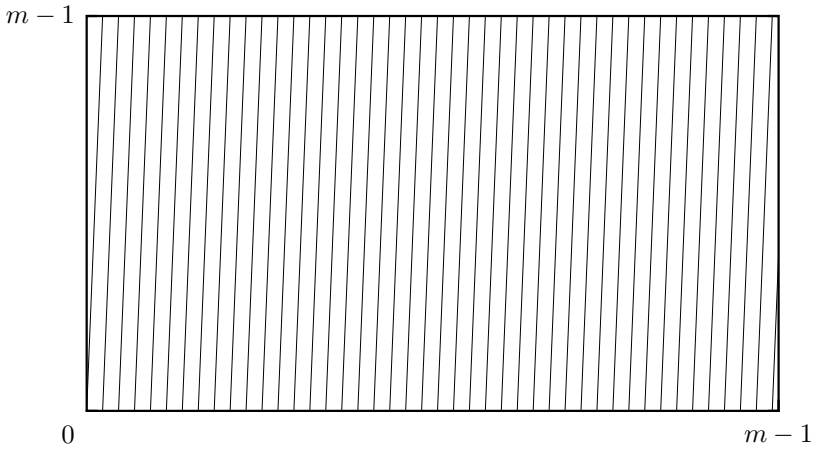

Abbildung 9.1: Beispiel einer linearen Kongruenzfunktion; der Graph ist der Übersicht wegen wie für eine stetige Funktion dargestellt.

Die jeweils nächste Zufallszahl x_{k+1} errechnet man bei linearen Kongruenzverfahren aus der letzten Zufallszahl x_k über die Formel

$$x_{k+1} = a\,x_k + b \pmod{m}.$$

Der Ausdruck (mod m) gibt an, dass vom vorangegangenen Term der Rest bei Division durch m zu nehmen ist. Der Wert x_{k+1} ergibt sich also als lineare Funktion von x_k, die auf den Bereich $\{0, \ldots, m-1\}$ beschränkt ist.

Graphisch lässt sich diese Verfahren leicht wie in Abbildung 9.1 veranschaulichen. Durch geeignete Wahl von a, b und m ergibt sich eine stückweise lineare Funktion, die sehr stark unstetig und deswegen genügend "chaotisch" ist, um wirklich zufällige Zahlen zu simulieren. Es ist aber unmittelbar klar, dass mit diesem Verfahren maximal m verschiedene Zufallszahlen erzeugt werden können. So ist etwa in ANSI C nur vorgeschrieben, dass $m \geq 2^{15} = 32768$ sein muss.

Eine versteckte Schwierigkeit bei der Verwendung linearer Kongruenzmethoden liegt in der richtigen Wahl von a und b, die die *Periode* des Zufallszahlengenerators bestimmen – also wieviele unterschiedliche Zufallszahlen generiert werden, bevor eine bereits generierte Zahl zum zweiten Mal erzeugt wird und sich somit die gesamte Sequenz wiederholt. Bei jeder Wahl von m kann es durch ungeeignete Werte von a und b vorkommen, dass die Periode der erzeugten Zahlenfolge viel kleiner als m ist. Bei millionenfachem Aufruf des Zufallszahlengenerators werden dann nicht Millionen unterschiedlicher Zufallszahlen erzeugt, sondern sehr wenige Zahlen sehr oft.

9.2 Kombinationsmethoden

Am Stand der Technik wird heutzutage empfohlen, die Ausgaben von zwei oder mehreren guten Verfahren zu kombinieren, um so noch bessere Ergebnisse zu erzielen. Als Einzelkomponenten bieten sich die unten präsentierten Verfahren an. Die Qualität der Ergebnisse kann durch eine Reihe von anspruchsvollen statistischen

Methoden überprüft werden, die die Gleichverteilung und Unabhängigkeit der erzeugten Zufallszahlen testen. Bei geeigneter Kombination der einzelnen Verfahren erfüllen nicht nur die Zufallszahlen, sondern auch einzelne bits aus diesen Zahlen die statistischen Tests.

Alle Details in diesem Abschnitt sind dem Standardwerk [Press *et al.*, 2007] entnommen.

64 bit xor Shift Bei diesem Verfahren wird eine Maschinenzahl durch eine Folge von drei Shift und xor Operationen verändert. Die genaue Vorschrift, die über die Größe der Shift-Operationen parametrisiert ist, lässt sich folgendermaßen angeben, wobei x_n die letzte generierte Zufallszahl ist:

$$\begin{aligned} x_n &= x_n \; xor \; (x_n >> \; a) \\ x_n &= x_n \; xor \; (x_n << \; b) \\ x_{n+1} &= x_n \; xor \; (x_n >> \; c). \end{aligned}$$

Dabei bezeichne *xor* die bitweise xor Funktion, und $<<$ und $>>$ die Shift-Operationen nach links bzw. rechts. Im Folgenden wird diese Methode als SR (für Shift Right) bezeichnet. Analog dazu kann man auch mit einem Shift nach links beginnen:

$$\begin{aligned} x_n &= x_n \; xor \; (x_n << \; a) \\ x_n &= x_n \; xor \; (x_n >> \; b) \\ x_{n+1} &= x_n \; xor \; (x_n << \; c). \end{aligned}$$

Wir bezeichnen diese Methode mit SL. Geeignete Kombinationen von a, b und c sind (unter anderem) 17, 31 und 8 bzw. 21, 35 und 4.

Multiplikation mit Übertrag Bei dieser Methode werden die unteren 32 bit einer Zahl mit einer Konstanten multipliziert. Die jeweils nächste Zahl ergibt sich als Produkt

$$x_{n+1} = a\,(x_n \;\&\; 2^{32-1}) + (x_n >> \; 32).$$

Geeignete Konstante a sind etwa 4294957665 oder 4294963023. Dabei sei $\&$ die bitweise logische and-Operation. Wir bezeichnen diese Methode im Folgenden mit M.

Linear Kongruenzmethode Obwohl diese Methode für sich selbst genommen nicht empfehlenswert ist, so kann sie doch in Kombination mit anderen Verfahren sinnvoll eingesetzt werden. Hier wird die Folge von Zufallszahlen in 64 bit erzeugt durch

$$x_{n+1} = a\,x_n + b.$$

Diese Methode wird hier mit L bezeichnet. Geeignete Werte für a und b sind etwa 2862933555777941757 und 7046029254386353087.

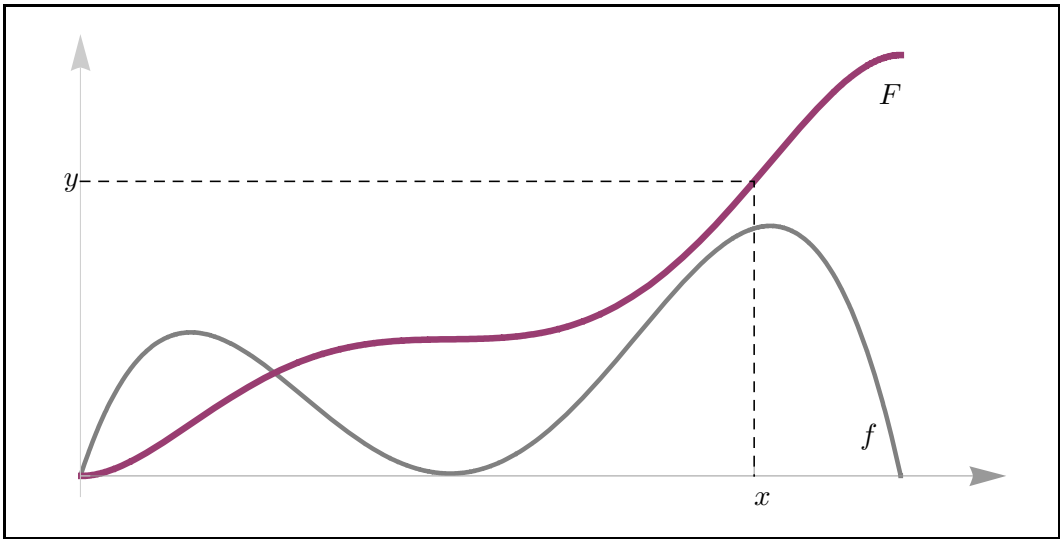

Abbildung 9.2: Das Urbild $x = F^{-1}(y)$ von $y \sim \mathrm{U}([0,1])$ unter der Verteilungsfunktion F ist mit Dichte f verteilt.

Kombination dieser Verfahren Obwohl diese Verfahren für sich selbst genommen schon als (mehr oder weniger gute) Methoden zur Erzeugung von Zufallszahlen verwendet werden können, wird die Qualität der von ihnen erzeugten Zahlenfolge durch geeignete Kombination noch weiter erhöht. Dabei werden Generatoren auch ineinander geschachtelt, die Ausgabe eines Generators dient also als Eingabe (seed value) eines anderen. Man beachte, dass in diesem Fall der "äußere" Generator natürlich keinen Zustand zu haben braucht, da dieser ja durch den Aufruf des "inneren" Generators geliefert wird.

Eine empfohlene Kombination der obigen drei Verfahren ist

$$(\mathrm{SL}(L) + \mathrm{SR})\ \mathit{xor}\ M.$$

Auch weniger komplexe Kombinationen genügen bereits hohen Qualitätsansprüchen und sind dann vorzuziehen, wenn man sehr viele Zufallszahlen sehr schnell und effizient generieren muss.

9.3 Inversionsmethode

Die einfachste Möglichkeit, Zufallszahlen anderer Verteilungen aus gleichverteilten Zufallszahlen zu generieren ist die *Inversionsmethode* (manchmal auch *Transformationsmethode* genannt). Dabei wird allerdings vorausgesetzt, dass die Verteilungsfunktion F analytisch berechenbar und invertierbar ist – dies schließt die Inversionsmethode für einige Verteilungen von vornherein aus.

Bei der Inversionsmethode wird entlang der y-Achse eine auf $[0,1]$ gleichverteilte Zufallszahl y generiert. Im Folgenden werden wir die Notation $\mathrm{U}([a,b])$ für die Gleichverteilung auf dem Intervall $[a,b]$ verwenden. Es sei f die Dichte und $F(x)$ die Verteilungsfunktion der Verteilung, aus der Zufallszahlen erzeugt werden sollen. Zur Erinnerung: Die Verteilungsfunktion einer Verteilung mit Dichte $f(x)$ ist definiert

als

$$F(x) = \int_{-\infty}^{x} f(t)\, dt.$$

Dann hat $x = F^{-1}(y)$, also der Punkt, an dem $\int_{-\infty}^{x} f(t)\, dt = y$ ist, die gewünschte Verteilung. Diese Methode ist graphisch in Abbildung 9.2 illustriert.

Der folgende Satz besagt, dass x wirklich die gewünschte Verteilung aufweist.

Satz 9.1 Die mit der Inversionsmethode erzeugte Zufallszahl x ist mit Dichte f verteilt, d.h., für beliebiges Intervall $[a, b]$ ist

$$P(\{x \in [a, b]\}) = \int_{a}^{b} f(t)\, dt.$$

Diesen Satz kann man durch Verwenden der Definition von Verteilungsfunktionen bzw. der Gleichverteilung von y nachrechnen. Es gilt

$$\begin{aligned} P(\{x \in [a, b]\}) &= P(\{y \in [F(a), F(b)]\}) \\ &= F(b) - F(a) = \int_{-\infty}^{b} f(t)\, dt - \int_{-\infty}^{a} f(t)\, dt \\ &= \int_{a}^{b} f(t)\, dt. \end{aligned}$$

Beispiel 9.1 Die *Exponentialverteilung* ist durch die Dichte

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{für } 0 \leq x < \infty \\ 0 & \text{sonst} \end{cases}$$

gegeben. Für $\lambda = 1$ lassen sich Zufallszahlen dieser Verteilung besonders leicht generieren. Man erhält $F(x) = 1 - e^{-x}$ und $F^{-1}(y) = -\log(1 - y)$. Somit ist für $y \sim \mathrm{U}([0, 1])$ die Zahl $-\log(1 - y)$ und damit auch $-\log(y)$ exponentialverteilt mit Parameter $\lambda = 1$. □

9.4 Verwerfungsmethode

Die in diesem Abschnitt vorgestellte Methode, mit der Zufallszahlen beliebiger Verteilung aus anderen Zufallszahlen generiert werden können, basiert auf einem einfachen geometrischen Argument.

Nehmen wir an, wir wollen eine Zufallszahl erzeugen, deren Verteilung die Dichte f besitzt. Wenn wir dann eine zweidimensionale Zufallszahl (x, y) generieren können, die in der Fläche unter dem Graphen von f gleichverteilt ist, dann besitzt x die gewünschte Verteilung. Damit dieses Argument klar wird, betrachten wir zuerst einen Spezialfall.

Die Verteilung der zu generierenden Zufallszahlen besitze eine Dichte f, die auf das Intervall $[a, b]$ beschränkt ist; weiters sei $m = \max_{x \in [a,b]} f(x)$ der größte Wert,

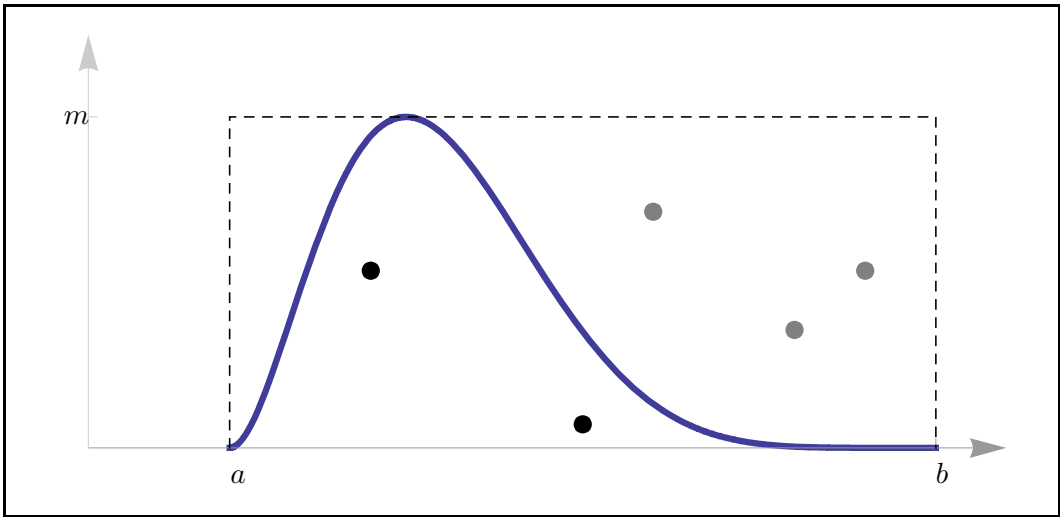

Abbildung 9.3: Bei der Verwerfungsmethode sind die Punkte • unter dem Graphen von f gleichverteilt; ihre x-Koordinaten sind mit Dichte f verteilt.

den f in diesem Intervall annimmt. In diesem Fall ist es nicht schwierig eine zweidimensionale Zufallszahl zu erzeugen, die unter dem Graphen von f gleichverteilt ist: Man wählt eine auf dem Rechteck $[a, b] \times [0, m]$ gleichverteilte Zufallszahl (x, y) und verwirft sie, wenn sie nicht unter dem Graphen liegt (also $y > f(x)$ gilt). Die verbleibenden Punkte sind unter dem Graphen gleichverteilt; ihre x-Koordinaten haben die gewünschte Verteilung. Diese Vorgangsweise ist in Abbildung 9.3 dargestellt.

Man kann nachrechnen, dass die x-Koordinaten der akzeptierten Punkte tatsächlich mit Dichte f verteilt sind.

Satz 9.2 Gegeben sei eine Dichtefunktion f, die auf dem endlichen Intervall $[a, b]$ konzentriert ist und deren Funktionswerte zwischen 0 und m liegen. Die x-Koordinaten der Punkte, die durch die Verwerfungsmethode unter dem Graphen der Funktion f bestimmt wurden, sind mit Dichte f verteilt, d.h., es gilt für ein beliebiges Intervall $[c, d] \subseteq [a, b]$

$$P(\{x \in [c, d]\} \mid \text{“Punkt } (x, y) \text{ akzeptiert”}) = \int_c^d f(t)\, dt.$$

Die Gültigkeit dieses Satzes kann folgendermaßen überprüft werden: Mit der Definition der bedingten Wahrscheinlichkeit ist

$$P(\{x \in [c, d]\} \mid \text{“Punkt } (x, y) \text{ akzeptiert”}) = \frac{P(\{x \in [c, d]\} \cap \text{“Punkt } (x, y) \text{ akzeptiert”})}{P(\text{“Punkt } (x, y) \text{ akzeptiert”})}.$$

Ein zufällig im Rechteck $[a, b] \times [0, m]$ ausgewählter Punkt (x, y) hat gemeinsame

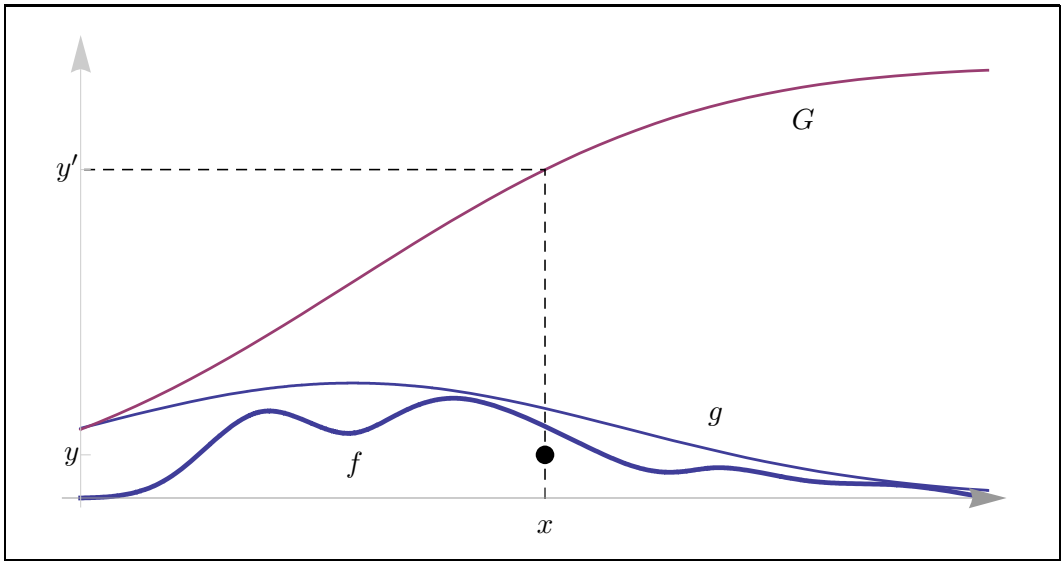

Abbildung 9.4: Illustration zur Verwerfungsmethode für Dichten f, die nicht auf einem endlichen Intervall definiert sind. Die Zufallszahlen, die unter g, aber nicht unter f liegen, werden verworfen. Die x-Koordinaten der akzeptierten Punkte sind mit Dichte f verteilt.

Dichte $f_{X,Y} = \frac{1}{m(b-a)}$. Der Punkt wird akzeptiert, wenn $y \leq f(x)$ ist. Damit gilt

$$\begin{aligned} P(\{x \in [c,d]\} \cap \text{“Punkt } (x,y) \text{ akzeptiert”}) &= \int_c^d \int_0^{f(t)} \frac{1}{m(b-a)}\, ds\, dt \\ &= \int_c^d \frac{f(t)}{m(b-a)}\, dt \end{aligned}$$

und, da $\int_a^b f(t)dt = 1$ ist, auch

$$\begin{aligned} P(\text{“Punkt } (x,y) \text{ akzeptiert”}) &= \int_a^b \int_0^{f(t)} \frac{1}{m(b-a)}\, ds\, dt \\ &= \int_a^b \frac{f(t)}{m(b-a)}\, dt = \frac{1}{m(b-a)}. \end{aligned}$$

Dies kann man sich auch anschaulich graphisch überlegen: Das Verhältnis der akzeptierten Punkten zu allen generierten Punkten ist das Verhältnis der Fläche unter der Kurve zum Rechteck mit Fläche $m(b-a)$; die Fläche unter der Kurve ist per Definition 1.

Aus obigen drei Überlegungen ergibt sich wie gewünscht

$$P(\{x \in [c,d]\} \mid \text{“Punkt } (x,y) \text{ akzeptiert”}) = \int_c^d f(t)dt.$$

Satz 9.2 kann nur in den Spezialfällen angewandt werden, bei denen die Dichtefunktion nur auf einem endlichen Bereich definiert ist. Bei vielen stetigen Dichten (wie etwa der Normalverteilung) ist dies nicht der Fall. Die oben beschreibene Vorgehensweise kann hierfür folgendermaßen verallgemeinert werden.

Wenn der Definitionsbereich der Dichte f ganz $\mathbb{R}$ ist, kann man entlang der x-Achse keine gleichverteilte Zufallszahl wählen. Statt dessen verwendet man eine Funktion g, die überall größer als f ist, deren Integral über $\mathbb{R}$ aber auch endlich ist. Dies ist wegen $\int_{\mathbb{R}} f(t)dt = 1$ immer möglich. Jetzt wendet man die Verwerfungsmethode auf zweidimensionale Zufallszahlen an, die unter g gleichverteilt sind. Dann akzeptiert man den Punkt (x, y), wenn er unter dem Graphen von f liegt und verwirft ihn sonst.

Das Problem reduziert sich also darauf, einen Punkt gleichverteilt unter der Kurve von g zu bestimmen. Dazu verwendet man die im letzten Abschnitt besprochene Inversionsmethode. Um diese anwenden zu können, muss man g so wählen, dass die Verteilungsfunktion $G(x) = \int_{-\infty}^{x} g(t)dt$ von g invertierbar ist. Da die Fläche unter g laut Voraussetzung endlich ist, ist G beschränkt – sei etwa $\lim_{x\to\infty} G(x) = A$. Wir wählen also $y' \sim \mathrm{U}([0, A])$ und bestimmen $x = G^{-1}(y')$. Zu dieser x-Koordinate benötigen wir noch einen y-Wert. Dazu generieren wir eine weitere gleichverteilte Zufallszahl y, diesmal auf dem Intervall $[0, g(x)]$. Wenn dann noch $y \leq f(x)$ gilt, y also unter der ursprünglichen Dichtefunktion liegt, dann ist der Punkt (x, y) gleichverteilt unter der Kurve von f und x weist die gewünschte Verteilung auf. Eine graphische Veranschaulichung dieser Methode ist in Abbildung 9.4 gegeben.

9.5 Einschub: Mehrdimensionale Integrale

Für die effiziente Erzeugung normalverteilter Zufallszahlen benötigt man einen Trick, der auf mehrdimensionaler Integration beruht. In diesem Abschnitt werden die theoretischen Grundlagen dieser Methode erarbeitet. Der Fundamentalsatz der Analysis (Satz 4.1) wird als bekannt vorausgesetzt; zur Erinnerung ein kurzes Beispiel.

Beispiel 9.2 Da $-\cos'(x) = \sin(x)$ ist, gilt

$$\int_0^{\pi} \sin(x)\, dx = -\cos(\pi) - (-\cos(0)) = -(-1) - (-1) = 2. \qquad \square$$

Ebenso bekannt sein sollte die Substitutionsmethode zum Evaluieren von Integralen, die die Umkehrung der Kettenregel der Differentialrechnung darstellt. Die Anwendung dieser Methode wird anhand des folgenden Beispiels demonstriert.

Beispiel 9.3 Zu bestimmen ist das Integral $\int_0^{2\pi} x\cos(x^2)dx$. Mit der Substitution $y = x^2$ erhält man $dy/dx = 2x$, für die Integrationsgrenzen $a = 0^2 = 0$ und $b = (2\pi)^2 = 4\pi^2$, und somit

$$\int_0^{2\pi} x\cos(x^2)\, dx = \int_0^{4\pi^2} x\cos(y)\, dy/2x = \frac{1}{2}\int_0^{4\pi^2} \cos(y)\, dy$$

$$= \frac{1}{2}\sin(y)\Big|_0^{4\pi^2} = \frac{1}{2}\sin(4\pi^2). \qquad \square$$

Die in obigem Beispiel gezeigte Substitution funktioniert auch, wenn man eine "einfache" Variable durch einen komplizierteren Ausdruck ersetzt. Dies wird im nächsten Beispiel illustriert.

Beispiel 9.4 Zu berechnen sei $\int_1^2 \log(x)\,dx$. Mit der Substitution $x = \exp(y)$ erhält man $dx/dy = \exp(y)$, für die Integrationsgrenzen aus den Bedingungen $1 = \exp(a)$ bzw. $2 = \exp(b)$ die Werte $a = \log(1) = 0$ bzw. $b = \log(2)$, sowie insgesamt

$$\int_1^2 \log(x)\;dx = \int_0^{\log(2)} \log(\exp(y))\exp(y)\;dy = \int_0^{\log(2)} y\exp(y)\;dy.$$

Dieses Integral ist schwieriger als das ursprüngliche, lässt sich aber durch partielle Integration mit $u = y$ und $v' = \exp(y)$ über die Regel

$$\int u\,v' = u\,v - \int u'\,v$$

lösen als

$$\int_0^{\log(2)} y\exp(y)\;dy = y\exp(y)\Big|_0^{\log(2)} - \int_0^{\log(2)} 1\exp(y)\;dy$$

$$= y\exp(y)\Big|_0^{\log(2)} - \exp(y)\Big|_0^{\log(2)} = 2\log(2) - 1. \qquad \square$$

Wir werden im Folgenden noch sehen, dass das Ersetzen von einfachen Ausdrücken durch kompliziertere bei mehrdimensionalen Integralen sehr zielführend sein kann.

Nach diesen Ausführungen über Integrationsregeln im eindimensionalen Fall betrachten wir nun die in diesem Kontext wichtige Erweiterung auf den mehrdimensionalen Fall. Der folgende Satz besagt, dass die Integrationsreihenfolge keine Rolle spielt.

Satz 9.3 (Satz von Fubini) Sei $f : [a,b] \times [c,d] \to \mathbb{R}$ eine stetige Funktion. Dann gilt

$$\int_a^b \int_c^d f(x_1,x_2)\;dx_2\,dx_1 = \int_c^d \int_a^b f(x_1,x_2)\;dx_1\,dx_2.$$

Wenn die Integrationsregion nicht rechteckig ist, sondern durch die Bedingungen $a \le x_1 \le b$ und $g_1(x_1) \le x_2 \le g_2(x_2)$ bzw. $h_1(x_2) \le x_1 \le h_2(x_2)$ und $c \le x_2 \le d$ definiert ist, so gilt

$$\int_a^b \int_{g_1(x_1)}^{g_2(x_1)} f(x_1,x_2)\;dx_2\,dx_1 = \int_c^d \int_{h_1(x_2)}^{h_2(x_2)} f(x_1,x_2)\;dx_1\,dx_2.$$

Beispiel 9.5 Es ist

$$\int_{-1}^1 \int_0^2 (1 - 6x^2y)\;dx\,dy = \int_{-1}^1 (x - 2x^3y)\Big|_0^2\;dy = \int_{-1}^1 (2 - 16y)\;dy$$

$$= (2y - 8y^2)\Big|_{-1}^1 = 4.$$

In der anderen Reihenfolge (zuerst nach y, dann nach x integriert) ergibt sich ebenfalls

$$\int_0^2 \int_{-1}^1 (1 - 6x^2 y)\, dy\, dx = \int_0^2 (y - 3x^2 y^2)\Big|_{-1}^1 dx = \int_0^2 2\, dx$$

$$= 2x\Big|_0^2 = 4. \qquad \square$$

Folgendes Beispiel illustriert die Anwendung des zweiten Teils vom Satz von Fubini.

Beispiel 9.6 Es ist

$$\int_0^1 \int_0^x (3 - x - y)\, dy\, dx = \int_0^1 \left(3y - xy - \frac{y^2}{2}\right)\Big|_0^x dx$$

$$= \int_0^1 \left(3x - x^2 - \frac{x^2}{2}\right) dx$$

$$= \left(\frac{3}{2}x^2 - \frac{1}{2}x^3\right)\Big|_0^1 = 1.$$

Dieselbe Region kann durch $0 \le y \le 1$ und $y \le x \le 1$ begrenzt werden. Folglich kann man obiges Integral auch evaluieren als

$$\int_0^1 \int_y^1 (3 - x - y)\, dx\, dy = \int_0^1 \left(3x - \frac{x^2}{2} - xy\right)\Big|_y^1 dy$$

$$= \int_0^1 \left(\frac{3}{2}y^2 - 4y + \frac{5}{2}\right) dy$$

$$= \left(\frac{1}{2}y^3 - 2y^2 + \frac{5}{2}y\right)\Big|_0^1 = 1. \qquad \square$$

Der Satz von Fubini lässt sich auf mehr als zwei Dimensionen verallgemeinern, wie wir in den folgenden Beispielen sehen werden. Zuvor benötigen wir noch folgendes Resultat über die Substitutionsregel in höheren Dimensionen, das sich ebenfalls auf höhere Dimensionen verallgemeinern lässt.

Satz 9.4 Sei $f : \mathbb{R}^2 \to \mathbb{R}$ eine reellwertige Funktion, $x_1 = g(u, v)$ und $x_2 = h(u, v)$ eine Variablensubstitution, $B \subseteq \mathbb{R}^2$ ein Integrationsbereich in den Variablen (u, v), sowie A der entsprechende Bereich in den Variablen (x_1, x_2). Dann gilt

$$\iint_A f(x_1, x_2)\, dx_1\, dx_2 = \iint_B f(g(u, v), h(u, v))\, |\det(J(g, h))|\, du\, dv.$$

Dabei bezeichnet $J(g, h)$ die *Jacobi-Matrix* der Variablentransformation (siehe Definition 6.2).

Wir werden diesen Satz nicht beweisen. Als Plausibilitätargument sei aber erwähnt, dass durch die Determinante der Jacobi-Matrix die Veränderung berücksichtigt wird, die sich für das Flächenelement $dx_1 dx_2$ durch die Koordinatentransformation ergibt.

Beispiel 9.7 Sei $f(x, y) = x + y$, und der Integrationsbereich gegeben durch die Bedingungen $-1 \leq x \leq 1$ und $0 \leq y \leq 1 - x^2$ (der obere Halbkreis). Dann gilt

$$\begin{aligned}\int_{-1}^{1}\int_{0}^{\sqrt{1-x^2}} (x+y)\, dy\, dx &= \int_{-1}^{1} \left(xy + \frac{y^2}{2}\right)\Big|_0^{\sqrt{1-x^2}}\, dx \\ &= \int_{-1}^{1} \left(x\sqrt{1-x^2} + \frac{1-x^2}{2}\right) dx \\ &= -\frac{1}{3}(1-x^2)^{\frac{3}{2}} + \frac{1}{2}x - \frac{x^3}{6}\Big|_{-1}^{1} = \frac{2}{3}.\end{aligned}$$

Mit Polarkoordinaten, also über die Transformation

$$x = g(r, \varphi) = r\cos\varphi \qquad \text{und} \qquad y = h(r, \varphi) = r\sin\varphi,$$

ist die Jacobi-Matrix

$$J(g, h) = \begin{pmatrix} \cos\varphi & \sin\varphi \\ -r\sin\varphi & r\cos\varphi \end{pmatrix}$$

mit Determinante $\det(J(g, h)) = r\cos^2\varphi + r\sin^2\varphi = r$.

Der Integrationsbereich des oberen Halbkreises ist in Polarkoordinaten über die Bedingungen $0 \leq r \leq 1$ und $0 \leq \varphi \leq \pi$ parametrisiert. Somit ergibt sich für das Integral der Wert

$$\begin{aligned}\int_0^1 \int_0^\pi (r\cos\varphi + r\sin\varphi)\; r\; d\phi\, dr &= \int_0^1 r^2(\sin\varphi - \cos\varphi)\Big|_0^\pi\, dr \\ &= 2\int_0^1 r^2\, dr = \frac{2}{3}r^3\Big|_0^1 = \frac{2}{3}.\end{aligned}$$

□

Beispiel 9.8 Als weiteres Beispiel sei die Volumnsberechnung des Torus angeführt. Es wird dabei keine "echte" Funktion integriert, sondern nur die Größe des Integrationsbereichs bestimmt. Dies kann man erreichen, indem man in diesem Bereich die konstante 1-Funktion integriert. Einfache Beispiele in ein bzw. zwei Dimensionen machen die Gültigkeit dieses Ansatzes schnell klar, werden hier aber weggelassen.

Zu bestimmen ist das das Integral

$$\iiint_T 1\; dx\, dy\, dz,$$

wobei T eine Parametrisierung des Torus mit Parametern wie in Abbildung 9.5 im Standard- (Euklid'schen) Koordinatensystem sei.

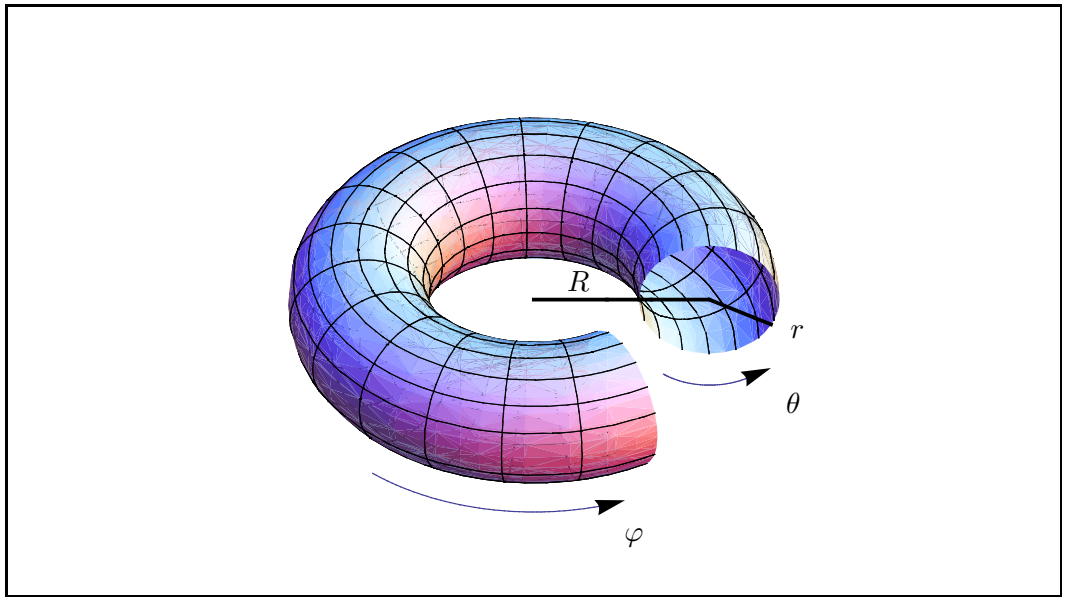

Abbildung 9.5: Parametrisierung des Torus aus Beispiel 9.8.

Man erhält eine einfachere Parametrisierung des Torus, wenn man einen Kreis in der x-z Ebene um die z-Achse rotiert. Es ergibt sich die Koordinatentransformation

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f(s,\theta,\varphi) \\ g(s,\theta,\varphi) \\ h(s,\theta,\varphi) \end{pmatrix} = \begin{pmatrix} (R + rs\cos\theta)\cos\varphi \\ (R + rs\cos\theta)\sin\varphi \\ rs\sin\theta \end{pmatrix}.$$

Der zusätzliche Parameter $s \in [0,1]$ wird benötigt, da wir ja das *Volumen* des Torus, und nicht nur die Oberfäche berechnen wollen.

Die Jacobi-Matrix dieser Koordinatentransformation ist

$$J(f,g,h) = \begin{pmatrix} r\cos\theta\cos\varphi & -rs\sin\theta\cos\varphi & -(R + rs\cos\theta)\sin\varphi \\ r\cos\theta\sin\varphi & -rs\sin\theta\sin\varphi & (R + rs\cos\theta)\cos\varphi \\ r\sin\theta & rs\cos\theta & 0 \end{pmatrix}$$

mit der Determinante

$$\det(J(f,g,h)) = -r^2 s(R + rs\cos\theta).$$

Mit Satz 9.4 ist das Volumen nun folgendermaßen zu berechnen:

$$\begin{aligned} &\int_0^1 \int_0^{2\pi} \int_0^{2\pi} r^2 s(R + rs\cos\theta)\, d\varphi\, d\theta\, ds \\ &\qquad = 2\pi r^2 \int_0^1 \int_0^{2\pi} s(R + rs\cos\theta)\, d\theta\, ds \\ &\qquad = 2\pi r^2 \int_0^1 (sR\theta + rs^2\sin\theta)\Big|_0^{2\pi}\, ds \\ &\qquad = 4\pi^2 r^2 R \int_0^1 s\, ds = 2\pi^2 r^2 R. \end{aligned}$$

□

9.6 Standard-normalverteilte Zufallszahlen

Es gibt mehrere Methoden, um eindimensionale normalverteilte aus gleichverteilten Zufallszahlen zu generieren. Eine besonders einfache (aber nicht sehr effiziente) Methode basiert auf dem Zentralen Grenzverteilungssatz, laut dem die Summe unabhängiger identisch verteilter Zufallsvariable normalverteilt ist. Wir benötigen dazu folgende als bekannt vorausgesetzte Fakten.

Satz 9.5 Sei $X \sim \mathrm{U}[a,b]$. Dann gilt

$$E(X) = \frac{a+b}{2} \qquad \text{und} \qquad \mathrm{Var}(X) = \frac{(b-a)^2}{12}.$$

Seien nun $X_1, \ldots, X_{12}$ zwölf Zufallsvariable, die unabhängig U([0,1])-verteilt sind. Dann ist $Z = \sum_{i=1}^{12} X_i - 6$ standard-normalverteilt, da

$$E(Z) = E\left(\sum_{i=1}^{12} X_i\right) - 6 = 6 - 6 = 0$$

und

$$\mathrm{Var}(Z) = \mathrm{Var}\left(\sum_{i=1}^{12} X_i\right) = 12\,\frac{1^2}{12} = 1.$$

Die Inversionsmethode aus Abschnitt 9.3 kann für die Erzeugung normalverteilter Zufallszahlen nicht verwendet werden, da die Verteilungsfunktion nicht explizit angegeben werden kann. Dennoch kann man anhand eines Tricks sehr einfach N(0,1)-verteilte Zufallszahlen erzeugen. Die notwendigen Grundlagen dafür wurden in Abschnitt 9.5 erarbeitet. Wir benötigen außerdem noch eine weitere spezielle Verteilung, die wie folgt definiert ist.

Definition 9.1 (Rayleigh-Verteilung)
Die Dichte der Rayleigh-Verteilung mit Parameter σ ist definiert als

$$f(x) = \begin{cases} \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)} & \text{für } x \geq 0 \\ 0 & \text{sonst.} \end{cases}$$

Man kann nachrechnen, dass es sich bei obiger Definition tatsächlich um eine Dichtefunktion handelt. Wir benötigen im Folgenden nur den Speziallfall $\sigma = 1$. Für diesen Fall ist die Verteilungfunktion gegeben durch

$$F(x) = \int_0^x t\, e^{-t^2/2}\, dt = -e^{-t^2/2}\Big|_0^x = -e^{-x^2/2} + 1.$$

Wie benötigt folgt daraus sofort $\lim_{x\to\infty} F(x) = 1$.

Nach all diesen Vorbereitungen können wir nun die sogenannte *Box-Muller Methode* behandeln, mit der man leicht normalverteilte Zufallszahlen aus gleichverteil-

ten generieren kann. Obwohl die Dichtefunktion

$$f(x) = \frac{1}{\sqrt{2\pi}}\, e^{-x^2/2}$$

einer N(0,1)-Verteilung nicht symbolisch integriert werden kann, ist dies für die gemeinsame Dichte *zweier* N(0,1)-Verteilungen möglich. Die gemeinsame Dichte

$$f(x,y) = \frac{1}{\sqrt{2\pi}}\, e^{-x^2/2} \cdot \frac{1}{\sqrt{2\pi}}\, e^{-y^2/2} = \frac{1}{2\pi}\, e^{-(x^2+y^2)/2}$$

ist sehr wohl symbolisch integrierbar: Mit den Polarkoordinaten $x = r\cos\varphi$ und $y = r\sin\varphi$ erhält man die Jacobi-Determinante r (siehe Beispiel 9.7) und für die Stammfunktion

$$\int\!\!\int \frac{1}{2\pi}\, e^{-(x^2+y^2)/2}\, dx\, dy = \int\!\!\int \frac{1}{2\pi} r e^{-r^2/2}\, dr\, d\varphi.$$

Dieser Ausdruck ist mit unseren Vorarbeiten als Stammfunktion der gemeinsamen Verteilung einer Gleichverteilung auf $[0, 2\pi]$ (Dichte $f(\varphi) = \frac{1}{2\pi}$) und einer Rayleigh-Verteilung mit $\sigma = 1$ zu identifizieren.

Dies bedeutet: Wenn wir zwei unabhängige Zufallszahlen generieren können, nämlich

- eine U($[0, 2\pi]$)-verteilte für φ, und
- eine Rayleigh-verteilte mit $\sigma = 1$ für r,

dann sind $x = r\cos\varphi$ und $y = r\sin\varphi$ beide standard-normalverteilt.

Beide oben angeführten Fälle sind einfach zu handhaben: Für gegebene Zufallszahl $u_1 \sim \mathrm{U}([0,1])$ ist

$$\varphi = 2\pi u_1$$

auf $[0, 2\pi]$ gleichverteilt. Die Verteilungsfunktion der Rayleigh-Verteilung ist analytisch invertierbar. Man erhält aus $F(r) = 1 - e^{-r^2/2}$ und der Überlegung, dass für $u_2 \sim \mathrm{U}([0,1])$ auch $1 - u_2 \sim \mathrm{U}([0,1])$ gilt, über die Inversionsmethode

$$r = F^{-1}(u_2) = \sqrt{-2\ln(u_2)}.$$

Damit ergeben sich aus zwei U([0,1])-verteilten Zufallszahlen u_1 und u_2 zwei N(0,1)-verteilte Zufallszahlen x_1 und x_2 als

$$x_1 = \sqrt{-2\ln(u_2)}\cos(2\pi u_1) \qquad \text{und} \qquad x_2 = \sqrt{-2\ln(u_2)}\sin(2\pi u_1).$$

9.7 Normalverteilte Zufallszahlen

Aus einer N(0,1)-verteilten Zufallszahlen x kann man leicht über die Formel

$$y = \sigma\, x + \mu$$

eine $\mathrm{N}(\mu, \sigma^2)$-verteilte Zufallszahl y generieren. Der mehrdimensionale Fall ist interessanter. Die Dichte einer n-dimensionalen Normalverteilung ist für $x \in \mathbb{R}^n$ gegeben durch

$$f(x) = \frac{1}{(2\pi)^{n/2}\sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right),$$

wobei $\mu \in \mathbb{R}^n$ der Erwartungswert, und Σ die symmetrisch, positiv definite $n \times n$ Kovarianzmatrix der Verteilung ist. Eine symmetrische Matrix ist genau dann positiv definit, wenn alle ihre Eigenwerte positiv sind.

In Analogie zum eindimensionalen Fall kann man erwarten, dass für $x \sim \mathrm{N}(0, I_n)$ durch $y = \sqrt{\Sigma}\, x + \mu$ eine $\mathrm{N}(\mu, \Sigma)$-verteilter Zufallsvektor erzeugt werden kann. Das Problem dabei liegt selbstverständlich darin, dass die Operation "$\sqrt{\Sigma}$" erst zu definieren ist. Eine Möglichkeit dazu ist die *Cholesky-Zerlegung*. Diese liefert eine Matrix L in unterer Dreiecksform mit der Eigenschaft

$$\Sigma = L \cdot L^T.$$

Die Cholesky-Zerlegung kann iterativ durch ein Verfahren berechnet werden, das dem Gauß'schen Eliminationsverfahren ähnlich ist. Die Berechnungsvorschrift lässt sich herleiten aus der Bedingung

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix} \cdot \begin{pmatrix} l_{11} & l_{21} & \dots & l_{n1} \\ 0 & l_{22} & \dots & l_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & l_{nn} \end{pmatrix}.$$

Man erhält

$$\begin{aligned}
a_{11} &= l_{11}^2 && \Rightarrow && l_{11} = \sqrt{a_{11}} \\
a_{21} &= l_{21} l_{11} && \Rightarrow && l_{21} = a_{21}/l_{11} \\
a_{31} &= l_{31} l_{11} && \Rightarrow && l_{31} = a_{31}/l_{11} \\
&\vdots \\
a_{n1} &= l_{n1} l_{11} && \Rightarrow && l_{n1} = a_{n1}/l_{11} \\
a_{22} &= l_{21}^2 + l_{22}^2 && \Rightarrow && l_{22} = \sqrt{a_{22} - l_{21}^2} \\
a_{32} &= l_{31} l_{21} + l_{32} l_{22} && \Rightarrow && l_{32} = (a_{32} - l_{31} l_{21})/l_{22} \\
&\vdots \\
a_{n2} &= l_{n1} l_{21} + l_{n2} l_{22} && \Rightarrow && l_{n2} = (a_{n2} - l_{n1} l_{21})/l_{22}
\end{aligned}$$

und daraus die allgemeinen Berechnungsvorschriften

$$\begin{aligned}
l_{ii} &= \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2} && \text{für } i = 1, \dots, n \\
l_{ji} &= \frac{1}{l_{ii}} \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik} \right) && \text{für } j = i+1, \dots, n.
\end{aligned}$$

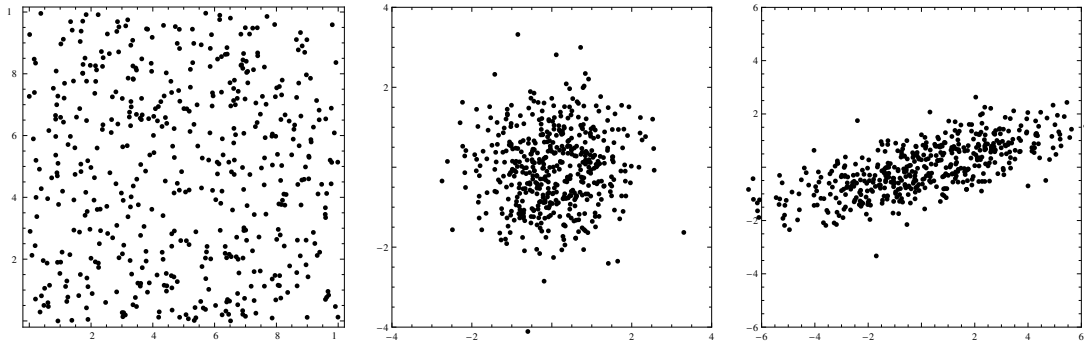

Abbildung 9.6: Illustration zur Generierung normalverteilter Zufallszahlen. Links eine zweidimensionale Gleichverteilung, in der Mitte und rechts die daraus resultierenden Normalverteilungen.

Beispiel 9.9 Für die Cholesky-Zerlegung der Matrix

$$A = \begin{pmatrix} 4 & -2 & 4 & -2 & -4 \\ -2 & 5 & -8 & -3 & 0 \\ 4 & -8 & 14 & 7 & 0 \\ -2 & -3 & 7 & 18 & 11 \\ -4 & 0 & 0 & 11 & 14 \end{pmatrix}$$

erhält man spaltenweise folgende Matrizen (dabei bezeichne $\cdot$ ein noch leeres Feld):

$$\xrightarrow[\substack{\text{andere Elemente durch} \\ 2 \text{ dividieren}}]{\text{Diagonalelement } \sqrt{4}} \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ -1 & \cdot & 0 & 0 & 0 \\ 2 & \cdot & \cdot & 0 & 0 \\ -1 & \cdot & \cdot & \cdot & 0 \\ -2 & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

$$\xrightarrow[\substack{-3=\frac{1}{2}(-8-(-2)) \\ -2=\frac{1}{2}(-3-1) \\ -1=\frac{1}{2}(0-2)}]{\text{Diagonalelement } \sqrt{5-(-1)^2}} \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 2 & -3 & \cdot & 0 & 0 \\ -1 & -2 & \cdot & \cdot & 0 \\ -2 & -1 & \cdot & \cdot & \cdot \end{pmatrix}$$

$$\xrightarrow[\substack{3=\frac{1}{1}(7-(-2)-6) \\ 1=\frac{1}{1}(0-(-4)-3)}]{\text{Diagonalelement } \sqrt{14-2^2-(-3)^2}} \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 2 & -3 & 1 & 0 & 0 \\ -1 & -2 & 3 & \cdot & 0 \\ -2 & -1 & 1 & \cdot & \cdot \end{pmatrix}$$

$$\xrightarrow[2=\frac{1}{2}(11-2-2-3)]{\text{Diagonalelement } \sqrt{18-(-1)^2-(-2)^2-3^2}} \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 2 & -3 & 1 & 0 & 0 \\ -1 & -2 & 3 & 2 & 0 \\ -2 & -1 & 1 & 2 & \cdot \end{pmatrix}$$

$$\xrightarrow{\text{Diagonalelement } \sqrt{14-(-2)^2-(-1)^2-1^2-2^2}} \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 2 & -3 & 1 & 0 & 0 \\ -1 & -2 & 3 & 2 & 0 \\ -2 & -1 & 1 & 2 & 2 \end{pmatrix}.$$

Man kann leicht überprüfen, dass für diese Matrix L die Bedingung $L \cdot L^T = A$ erfüllt ist. □

Zum Abschluss dieser Überlegungen ist in Abbildung 9.6 graphisch dargestellt, wie aus einer Menge von zweidimensionalen Gleichverteilungen zuerst eine Standard-Normalverteilung erzeugt werden kann, und dann daraus eine Verteilung, die eine gewünschte Streuung aufweist. Zur Erzeugung der Standard-Normalverteilung wird dabei die Box-Muller Methode verwendet. Die Kovarianzmatrix der schiefen Verteilung (rechts) ist $\Sigma = \begin{pmatrix} 8 & 2 \\ 2 & 1 \end{pmatrix}$.

Weiterführende Literatur

[Burden and Faires, 2004] R.L. Burden and J.D. Faires. *Numerical Analysis*. Brooks Cole, 8th edition, 2004.

[Kreyszig, 2011] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, 10th edition, 2011.

[MacKay, 2003] D.J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[Press *et al.*, 2007] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3rd edition, 2007.