Höhere Mathematik für Ingenieure IV

Volker John

Sommersemester 2007

Inhaltsverzeichnis

Teil I

Weiterführende Integralrechung

Kapitel 1

Kurvenintegrale

1.1 Kurven

Sei $I = [a, b] \subset \mathbb{R}$ ein Intervall. Eine Weg κ ist eine Abbildung dieses Intervalls in den $\mathbb{R}^d, d \geq 1$,

$$\kappa \ : \ I \to \mathbb{R}^d.$$

Dabei nennt man $\kappa(a)$ den Anfangspunkt, $\kappa(b)$ den Endpunkt und das Bild $\kappa([a, b])$ die Spur des Weges. Der Weg wird von $\kappa(a)$ nach $\kappa(b)$ durchlaufen. Die obige Abbildungsvorschrift nennt man Parametrisierung des Weges.

Beispiel 1.1 Zwei Wege können die gleiche Spur besitzen:

$$\begin{aligned} \kappa_1 \quad &: \quad I = [-1, 1], \quad \kappa_1(t) = \begin{pmatrix} t \\ \sqrt{1 - t^2} \end{pmatrix}, \quad t \in I, \\ \kappa_2 \quad &: \quad I = [0, \pi], \quad \kappa_2(t) = \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix}, \quad t \in I, \end{aligned}$$

haben als Spur jeweils den Einheitshalbkreis in der oberen Halbebene. □

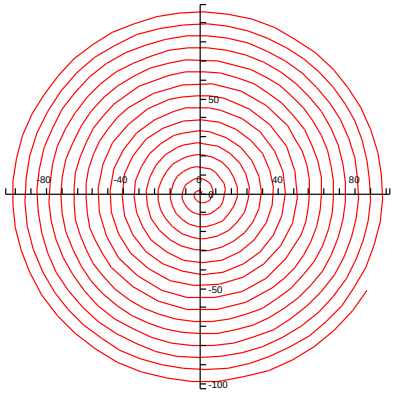
Zwei Wege mit der gleichen Spur sollen als die gleiche Kurve bezeichnet werden, falls der Druchlaufsinn der Spuren sowie die Anzahl der Durchläufe gleich sind. Im Beispiel 1.1 ist der Durchlaufsinn der Spuren unterschiedlich, also handelt es sich nicht um die gleichen Kurven.

Wir werden hier nur Kurven betrachten, bei denen der Tangentenvektor $d\kappa/dt =: \dot{\kappa}(t)$ an keiner Stelle $t \in I$ verschwindet. Man spricht dann von einer regulären Kurve.

Beispiel 1.2 Die Archimedische Spirale ist für $t \geq 0$ durch

$$t \mapsto \kappa(t) = \begin{pmatrix} t \cos t \\ t \sin t \end{pmatrix}$$

gegeben.



Es gilt

$$\dot{\kappa}(t) = \begin{pmatrix} \cos t - t \sin t \\ \sin t + t \cos t \end{pmatrix}.$$

Daraus folgt

$$\|\dot{\kappa}(t)\|_2 = \sqrt{(\cos t - t \sin t)^2 + (\sin t + t \cos t)2} = \sqrt{1 + t^2} \neq 0.$$

Damit ist die Archimedische Spirale auf $[0, \infty)$ regulär. *Weitere Beispiel Übungsaufgaben* □

Wird die betrachtete Kurve durch eine Funktion $f : [a, b] \to \mathbb{R}$ definiert, so kann man als Parametrisierung

$$t \mapsto \kappa(t) = \begin{pmatrix} t \\ f(t) \end{pmatrix}, \quad t \in [a, b] \tag{1.1}$$

verwenden, siehe κ_1 in Beispiel 1.1.

1.2 Skalares Kurvenintegral

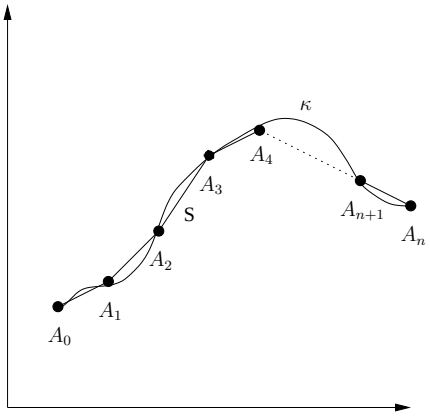
Das skalare Kurvenintegral wurde früher auch Kurvenintegral 1. Art genannt. Es ist eine Verallgemeinerung des Integrals definiert auf einem Intervall der x–Achse auf ein Integral welches auf einer Kurve definiert ist. Es dient vor allem der Berechnung der Länge von Kurven.

Wir betrachten eine Kurve κ im $\mathbb{R}^d$ und endlich viele Teilpunkte

$$\kappa(a) = A_0, A_1, \ldots, A_{n-1}, A_n = \kappa(b)$$

auf dieser Kurve, die vom Anfangs– zum Endpunkt numeriert sind. Zu den Teilpunkten $A_0, A_1, \ldots, A_n \in \kappa$ gehört das Sehnenpolygon S mit der Länge

$$l(S) = \sum_{i=0}^{n-1} \|A_{i+1} - A_i\|_2 = \sum_{i=0}^{n-1} \sigma_i.$$



Um eine Berechnungsvorschrift für die Länge von κ zu erhalten, gehen wir analog wie bei der Herleitung des (Riemann)–Integrals über Intervallen vor. Wir betrachten eine Zerlegung der Kurve durch immer mehr Punkte A_i und zwar so, dass $\max_i \|A_{i+1} - A_i\|_2 \to 0$. Strebt die Länge der Sehnenpolygone einem endlichen Grenzwert zu, der nicht von der Wahl der Punkte A_i abhängig ist, so ist dieser Grenzwert gleich der Länge der Kurve und man schreibt

$$l(\kappa) = \int_\kappa ds,$$

wobei ds als skalares Bogenelement bezeichnet wird. Kurven mit endlicher Länge $l(\kappa) < \infty$ werden rektifizierbar genannt.

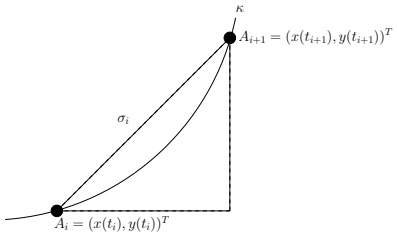


Abbildung 1.1: Sehne σ_i.

Betrachten wir den zweidimensionalen Fall, $\kappa(t) = (x(t), y(t))^T$, dann lässt sich σ_i wie in Abb. 1.1 veranschaulichen. Man hat (Pythagoras)

$$\begin{aligned} \sigma_i &= \sqrt{(x(t_{i+1}) - x(t_i))^2 + (y(t_{i+1}) - y(t_i))^2} \\ &= \sqrt{\left(\frac{x(t_{i+1}) - x(t_i)}{t_{i+1} - t_i}\right)^2 + \left(\frac{y(t_{i+1}) - y(t_i)}{t_{i+1} - t_i}\right)^2} (t_{i+1} - t_i), \end{aligned}$$

woraus man im Grenzprozess

$$ds = \sqrt{(\dot{x}(t))^2 + (\dot{y}(t))^2} dt = \|\dot{\kappa}(t)\|_2 \, dt$$

erhält. Diese Beziehung gilt sinngemäß auch in höheren Dimensionen. Man erhält also

$$l(\kappa) = \int_a^b \|\dot{\kappa}(t)\|_2 \; dt. \tag{1.2}$$

Beispiel 1.3 Ein Kreis K mit Radius r und Mittelpunkt (x_0, y_0) ist durch

$$t \mapsto \kappa(t) = \begin{pmatrix} r\cos t + x_0 \\ r\sin t + y_0 \end{pmatrix}, \quad t \in [0, 2\pi],$$

gegeben. Der Umfang des Kreises kann einfach mit (1.2) berechnet werden

$$l(K) = \int_0^{2\pi} \sqrt{(-r\sin(t))^2 + (r\cos(t))^2}\ dt = r\int_0^{2\pi} \sqrt{\sin^2(t) + \cos^2(t)}\ dt = 2\pi r.$$

Weitere Beispiele Übungsaufgaben □

Im Falle einer skalaren Funktion einer skalaren Veränderlichen, in welchem man die Parametrisierung (1.1) verwenden kann, wird aus der Formel (1.2) für die Kurvenlänge

$$l(f) = \int_a^b \sqrt{1 + (f'(t))^2}\ dt.$$

Die Berechnung der Kurvenlänge ist ein Spezialfall des skalaren Kurvenintegrals. Im allgemeinen skalaren Kurvenintegral ist eine Funktion auf der Kurve gegeben $f : \kappa([a,b]) \to \mathbb{R}$ und man sucht den Inhalt der Fläche zwischen f und der Kurve. Mit der obigen Vorgehensweise erhält man die Berechnungsvorschrift

$$\int_\kappa f(s)\ ds = \int_a^b f(\kappa(t))\,\|\dot\kappa(t)\|_2\ dt.$$

Die Länge von κ erhält man für $f \equiv 1$.

Einfache Eigenschaften des skalaren Kurvenintegrals sind:

- Linearität

$$\int_\kappa (\lambda f(s) + \mu g(s))\ ds = \lambda \int_\kappa f(s)\ ds + \mu \int_\kappa g(s)\ ds, \quad \lambda, \mu \in \mathbb{R},$$

- Integralabschätzung

$$\int_\kappa f(s)\ ds \le l(\kappa) \sup_{\mathbf{x}\in\kappa} |f(\mathbf{x})|.$$

- Das skalare Kurvenintegral ist unabhängig vom Durchlaufsinn der Kurve. Bei der Kurvenlänge ist das unmittelbar aus (1.2) einzusehen. Da $a < b$ für jede Parametrisierung ist und der Integrand positiv ist, so ist der Integralwert auch positiv. *(Übungsaufgabe)*

1.3 Vektorielles Kurvenintegral

Eine große Bedeutung in der Praxis besitzt das vektorielle Kurvenintegral. Dieses wurde früher auch Kurvenintegral 2. Art genannt.

Das vektorielle Kurvenintegral ist für Vektorfelder definiert. Unter einem Vektorfeld verstehen wir eine Abbildung $\mathbf{v} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^d$. Anschaulich denkt man sich den Vektor $\mathbf{v}(\mathbf{x})$ im Punkt $\mathbf{x}$ angeheftet. Beispiele von Vektorfeldern sind Geschwindigkeitsfelder, Gravitationsfelder, elektrische oder magnetische Felder.

Sei $\kappa : [a,b] \to \mathbb{R}^d$ eine reguläre Kurve. Dann ist das vektorielle Kurvenintegral definiert durch

$$\int_\kappa \mathbf{v}\cdot d\mathbf{x} := \int_\kappa v_1\ dx_1 + \ldots + v_d\ dx_d = \int_a^b \mathbf{v}(\kappa(t))\cdot\dot\kappa(t)\ dt.$$

Das Symbol $d\mathbf{x} = \dot\kappa(t)\ dt$ steht dabei für das vektorielle Bogenelement.

Das vektorielle Kurvenintegral kann folgende Bedeutungen in der Physik besitzen:

Vektorfeld	Kurvenintegral
Kraftfeld	Arbeit
Geschwindigkeitsfeld	Zirkulation
elektrische Feldstärke	elektrische Spannung
infinitesimale Wärmeänderung	Wärmemenge

Beispiel 1.4 Ein Massenpunkt im Koordinatenursprung $\mathbf{0}$ erzeugt ein Gravitationsfeld, das bis auf einen konstanten Faktor gegeben ist durch

$$\mathbf{G}(\mathbf{x}) = -\frac{\mathbf{x}}{\|\mathbf{x}\|_2^3} = -\frac{1}{(x_1^2+x_2^2+x_3^2)^{3/2}} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

Wird ein zweiter Massenpunkt der Masse 1 längs der Kurve $\kappa \subset \mathbb{R}^3 \setminus \mathbf{0}$, $\kappa : [a,b] \to \mathbb{R}^3$ bewegt, so ist die an ihm geleistete Arbeit das Wegintegral

$$\int_\kappa \mathbf{G} \cdot d\mathbf{x} = -\int_a^b \frac{\kappa(t)}{\|\kappa(t)\|_2^3} \cdot \dot{\kappa}(t)\ dt.$$

Man erhält mit der Kettenregel

$$\frac{d}{dt}\frac{1}{\|\kappa(t)\|_2} = \frac{d}{dt}\frac{1}{\left(\kappa_1^2(t)+\kappa_2^2(t)+\kappa_3^2(t)\right)^{1/2}} = -\frac{\kappa(t)\cdot\dot{\kappa}(t)}{\|\kappa(t)\|_2^3}.$$

Somit ergibt sich für die geleistete Arbeit

$$\int_\kappa \mathbf{G} \cdot d\mathbf{x} = \int_a^b \frac{d}{dt}\frac{1}{\|\kappa(t)\|_2}\ dt = \frac{1}{\|\kappa(b)\|_2} - \frac{1}{\|\kappa(a)\|_2}.$$

Man sieht insbesondere, dass in diesem Kraftfeld die Arbeit nicht von der konkreten Kurve abhängt, sondern nur von deren Anfangs– und Endpunkt. Ist die Kurve geschlossen, stimmen also Anfangs– und Endpunkt überein, so wird keine Arbeit geleistet. □

Beispiel 1.5 Fließt durch einen Draht, der in der x_3–Achse liegt, ein konstanter Strom, so erzeugt dieser nach dem Biot-Savartschen Gesetz ein Magnetfeld außerhalb des Drahts, das bis auf einen konstanten Faktor durch

$$\mathbf{H}(\mathbf{x}) = \frac{1}{x_1^2+x_2^2} \begin{pmatrix} -x_2 \\ x_1 \\ 0 \end{pmatrix}$$

gegeben ist. Die Integration über eine geschlossene Kurve, etwa einen Kreis in der x_1-x_2–Ebene $\kappa(t) = (r\cos(t), r\sin(t), 0)$, $t \in [0, 2\pi]$, ergibt

$$\int_\kappa \mathbf{H} \cdot d\mathbf{x} = \int_0^{2\pi} \frac{1}{r^2} \begin{pmatrix} -r\sin(t) \\ r\cos(t) \\ 0 \end{pmatrix} \begin{pmatrix} -r\sin(t) \\ r\cos(t) \\ 0 \end{pmatrix} dt = 2\pi.$$

In diesem Beispiel verschwindet also das vektorielle Kurvenintegral über die geschlossene Kurve nicht. □

Eigenschaften des vektoriellen Kurvenintegrals sind:

- Additivität. Ist $\kappa = \kappa_1 + \kappa_2$ mit κ_1, κ_2 regulär, so gilt

$$\int_\kappa \mathbf{v} \cdot d\mathbf{x} = \int_{\kappa_1} \mathbf{v} \cdot d\mathbf{x} + \int_{\kappa_2} \mathbf{v} \cdot d\mathbf{x},$$

- Linearität
$$\int_\kappa (\lambda \mathbf{v} + \mu \mathbf{w}) \cdot d\mathbf{x} = \lambda \int_\kappa \mathbf{v} \cdot d\mathbf{x} + \mu \int_\kappa \mathbf{w} \cdot d\mathbf{x},$$
- Integralabschätzung
$$\left| \int_\kappa \mathbf{v} \cdot d\mathbf{x} \right| \leq l(\kappa) \sup_{\mathbf{x} \in \kappa} \|\mathbf{v}(\mathbf{x})\|_2 .$$
- Das Vorzeichen des vektoriellen Kurvenintegrals hängt vom Durchlaufsinn der Kurve ab.

Wichtige Felder in der Physik sind diejenigen, für welche der Wert des vektoriellen Kurvenintegrals nur vom Anfangs– und Endpunkt des Weges, aber nicht vom Weg selber abhängt. Vektorfelder $\mathbf{v} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^d$, die diese Eigenschaft besitzen, werden konservativ genannt. Eine einfache Schlussfolgerung besteht darin, dass ein Vektorfeld genau dann konservativ ist, wenn das Kurvenintegral über alle geschlossenen, stückweise regulären Kurven in Ω verschwindet.

Satz 1.6 *Ein Vektorfeld $\mathbf{v} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^d$ ist genau dann konservativ, wenn es eine stetig differenzierbare Abbildung $U : \Omega \to \mathbb{R}$ gibt, so dass*

$$\mathbf{v} = \nabla U = grad\ U,$$

d.h. $\mathbf{v}$ ist ein Gradientenfeld. In diesem Fall ist das Kurvenintegral über eine Kurve κ vom Punkt P_1 nach P_2 gegeben durch

$$\int_\kappa \mathbf{v} \cdot d\mathbf{x} = U(P_2) - U(P_1).$$

U heißt Potential von $\mathbf{v}$.

Beweis: Jedes Gradientenfeld ist konservativ. Mit Kettenregel gilt

$$\frac{d}{dt} U(\kappa(t)) = \nabla U(\kappa(t)) \cdot \dot{\kappa}(t) = \mathbf{v}(\kappa(t)) \cdot \dot{\kappa}(t).$$

Das Kurvenintegral über κ ist dann

$$\begin{aligned} \int_\kappa \mathbf{v} \cdot d\mathbf{x} &= \int_a^b \mathbf{v}(\kappa(t)) \cdot \dot{\kappa}(t)\ dt = \int_a^b \frac{d}{dt} U(\kappa(t))\ dt = U(\kappa(b)) - U(\kappa(a)) \\ &= U(P_2) - U(P_1). \end{aligned}$$

Für den Beweis der Umkehrung, dass jedes konservative Vektorfeld ein Gradientenfeld ist, sei auf die Literatur verwiesen. ∎

Ein einfaches notwendiges Kriterium um zu entscheiden, ob ein Vektorfeld ein Gradientenfeld ist, basiert auf dem Satz von Schwarz. Dieser besagt, dass unter gewissen Bedingungen, für die gemischten Ableitungen einer Funktion $U : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ gilt

$$\frac{\partial^2 U}{\partial x_i \partial x_j}(\mathbf{x}) = \frac{\partial^2 U}{\partial x_j \partial x_i}(\mathbf{x}), \quad i, j = 1, \ldots, d, \quad \mathbf{x} \in \Omega,$$

das heißt, die Reihenfolge der Differentiation ist egal. Ist $\mathbf{v} = (v_1, \ldots, v_d)^T = \nabla U$, dann folgt

$$\frac{\partial v_i}{\partial x_j} = \frac{\partial v_j}{\partial x_i}, \quad i, j = 1, \ldots, d. \tag{1.3}$$

Ein Vektorfeld, welches Bedingung (1.3) erfüllt, wird rotationsfrei genannt. Im $\mathbb{R}^3$ gibt es den sogenannten Rotationsoperator

$$\text{rot } \mathbf{v} := \nabla \times \mathbf{v} = \begin{vmatrix} \mathbf{e}_{x_1} & \mathbf{e}_{x_2} & \mathbf{e}_{x_3} \\ \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_3} \\ v_1 & v_2 & v_3 \end{vmatrix} = \begin{pmatrix} \frac{\partial v_3}{\partial x_2} - \frac{\partial v_2}{\partial x_3} \\ \frac{\partial v_1}{\partial x_3} - \frac{\partial v_3}{\partial x_1} \\ \frac{\partial v_2}{\partial x_1} - \frac{\partial v_1}{\partial x_2} \end{pmatrix}, \tag{1.4}$$

mit welchem ein rotationsfreies Vektorfeld kurz mit $\nabla \times \mathbf{v} = \mathbf{0}$ charakterisiert werden kann. Das Symbol $|\cdot|$ ist hier die Determinante.

Die Schwierigkeit bei dieser Charakterisierung steckt im Detail. Der Schluss von Rotationsfreiheit eines Vektorfelds auf ein Gradientenfeld hängt von der Beschaffenheit des Gebiets Ω ab. Ein Gebiet ist eine offene und zusammenhängende Menge. Dieses wird sternförmig genannt, wenn es einen Punkt $\mathbf{x}_0 \in \Omega$ gibt, so dass mit jedem Punkt $\mathbf{x} \in \Omega$ auch die Verbindungsstrecke zwischen $\mathbf{x}_0$ und $\mathbf{x}$ in Ω liegt. Anschaulich bedeutet dies, dass man vom Zentrum $\mathbf{x}_0$ aus jeden Punkt $\mathbf{x} \in \Omega$,,sehen" kann. Ein Gebiet Ω heißt einfach, wenn es eine eine eineindeutige Abbildung von Ω auf ein sternförmiges Gebiet gibt, die hinreichend oft differenzierbar ist. Anschaulich entsteht ein einfaches Gebiet durch ,,Verbiegen" eines sternförmigen Gebiets.

Beispiel 1.7 Ein einfaches Gebiet muss nicht sternförmig sein. So kann man beispielsweise das sternförmige Rechteck $(a, b) \times (c, d)$ eineindeutig auf einen Kreisring mit Schlitz abbilden, siehe Abbildung 1.2. □

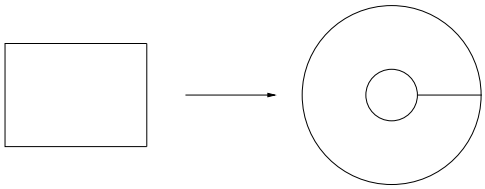


Abbildung 1.2: Eineindeutige Abbildung eines Rechtecks auf einen Kreisring mit einem Schlitz.

Für zweidimensionale Gebiete hat man eine andere, einfache Charakterisierung eines einfachen Gebiets: Für jede in Ω liegende einfach geschlossene Kurve muss der von dieser Kurve berandete endliche Bereich ganz zu Ω gehören. Man nennt diese Gebiete auch einfach zusammenhängend. Eine ähnliche Charakterisierung gibt es in drei Dimensionen: In jede einfache geschlossene stückweise reguläre Kurve κ in Ω kann eine stückweise glatte, sich nicht durchdringende Fläche eingespannt werden, die κ als Rand besitzt und ganz in Ω enthalten ist.

Beispiel 1.8 Für das Vektorfeld $\mathbf{H}(\mathbf{x})$ aus Beispiel 1.5 gilt

$$\nabla \times \mathbf{H}(\mathbf{x}) = \left(0, 0, \frac{x_1^2 + x_2^2 - 2x_1^2}{(x_1^2 + x_2^2)^2} - \frac{-(x_1^2 + x_2^2) + 2x_2^2}{(x_1^2 + x_2^2)^2}\right)^T = \mathbf{0}.$$

Das Vektorfeld ist also rotationsfrei. Es ist jedoch nicht auf der x_3–Achse $(0, 0, x_3)$ definiert. Man kann zeigen, dass das Definitionsgebiet nicht einfach ist. Die anschauliche Beschreibung mit der eingespannten Fläche kann man nicht erfüllen. Diese kann man aber in Beispiel 1.4 erfüllen, bei welchem die Funktion auf $\mathbb{R}^3 \setminus \{\mathbf{0}\}$ definiert ist. Nimmt man da bespielsweise einen Kreis in der x_1–x_2–Ebene als geschlossen Kurve, so kann man da eine Halbkugel einspannen und das obige anschauliche Kriterium ist erfüllt.

Dieses Beispiel zeigt, dass die Rotationsfreiheit nur eine notwendige, aber keine hinreichende Bedingung ist. □

Es gilt:

Satz 1.9 Hinreichende Bedingung für die Existenz eines Potentials. *Ein stetig differenzierbares Vektorfeld* $\mathbf{v}$ *sei auf dem einfachen Gebiet* Ω *rotationsfrei. Dann gibt es auf* Ω *ein Potential* U *mit* $\mathbf{v} = \nabla U$.

Für den Beweis dieses Satzes sei auf die Literatur verwiesen.
Die Berechnung des Potentials wird an einem Beispiel veranschaulicht.

Beispiel 1.10 Gegeben sei $\mathbf{v} : \mathbb{R}^3 \to \mathbb{R}^3$ mit

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} x_1 + x_3 \\ -x_2 - x_3 \\ x_1 - x_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}.$$

Der $\mathbb{R}^3$ ist ein sternförmiges Gebiet. Es reicht, die Rotationsfreiheit von $\mathbf{v}$ zu überprüfen. Die Bedingungen dafür sind nach (1.4)

$$\frac{\partial v_1}{\partial x_2} = \frac{\partial v_2}{\partial x_1} (= 0), \quad \frac{\partial v_1}{\partial x_3} = \frac{\partial v_3}{\partial x_1} (= 1), \quad \frac{\partial v_2}{\partial x_3} = \frac{\partial v_3}{\partial x_2} (= -1).$$

Damit existiert ein Potential U. Dieses wird mittels Integration berechnet. Aus der Eigenschaft des Gradientenfeldes folgt

$$U(\mathbf{x}) = \int v_1(\mathbf{x}) \; dx_1 = \frac{x_1^2}{2} + x_1 x_3 + C_1(x_2, x_3).$$

Differenziert man dies nach x_2 und vergleicht mit v_2, so erhält man

$$-x_2 - x_3 = v_2 = \frac{\partial U}{\partial x_2} = \frac{\partial C_1(x_2, x_3)}{\partial x_2}.$$

Inegration ergibt

$$C_1(x_2, x_3) = \int -x_2 - x_3 \; dx_2 = -\frac{x_2^2}{2} - x_2 x_3 + C_2(x_3),$$

also

$$U(\mathbf{x}) = \frac{x_1^2}{2} + x_1 x_3 - \frac{x_2^2}{2} - x_2 x_3 + C_2(x_3).$$

Nun differenziert man U nach x_3 und vergleicht mit v_3:

$$x_1 - x_2 = v_3 = \frac{\partial U}{\partial x_3} = x_1 - x_2 + \frac{\partial C_2(x_3)}{\partial x_3}.$$

Daraus folgt, dass $C_2(x_3)$ eine Konstante C ist und man erhält

$$U(\mathbf{x}) = \frac{x_1^2}{2} - \frac{x_2^2}{2} + x_1 x_3 - x_2 x_3 + C.$$

Durch Differentiation rechnet man schnell nach, dass U das gesuchte Potential ist.
□

Kapitel 2

Das Flächenintegral

2.1 Motivation, Zurückführung auf ein Doppelintegral

Wir betrachten einen zylindrischen Körper K, der von der Fläche

$$z = f(x, y),$$

seitlich von einer Zylinderfläche mit Erzeugenden parallel zur z–Achse und schließlich von einem ebenen Gebiet Ω in der x–y–Ebene begrenzt wird, siehe Abbildung 2.1. Es soll das Volumen V von K bestimmt werden.

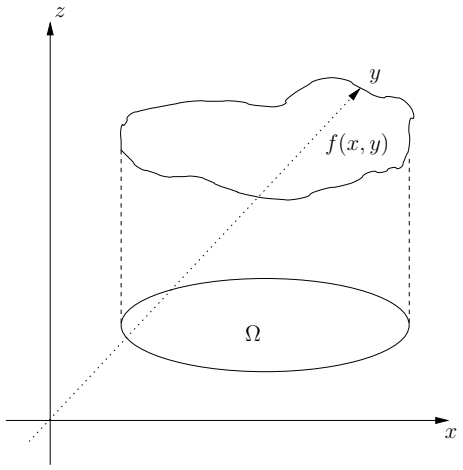


Abbildung 2.1: Volumen unter einer Fläche.

Zur Lösung wird so verfahren, wie wir es vom Riemann–Integral kennen. Wir zerlegen das Grundgebiet Ω in Elemente, berechnen das Volumen über jedem Element näherungsweise (zum Beispiel durch Volumina von Quadern), summieren und führen schließlich einen Grenzübergang (Elemente immer kleiner) durch. Das soll hier nicht im Detail erläutert werden. Falls der Grenzwert für alle zulässigen Zerlegungen existiert und gleich ist, erhält man das Volumen als das Flächenintegral von f über Ω

$$V = \int_\Omega f(x, y)\ d\Omega = \iint_\Omega f(x, y)\ d\Omega.$$

Zur Berechnung des Flächenintegrals versucht man, dieses auf ein Doppelintegral von eindimensionalen Integralen zurückzuführen. Sei $\Omega = (a, b) \times (c, d)$ ein Rechteck.

Dann gilt

$$\int_\Omega f(x,y)\ d\Omega = \int_a^b \left(\int_c^d f(x,y)\ dy \right) dx = \int_c^d \left(\int_a^b f(x,y)\ dx \right) dy, \qquad (2.1)$$

sofern alle Integrale existieren. Das heißt, die Berechnung des Flächenintegrals über ein Rechteck kann man sich wie folgt vorstellen (mittlere Formel). Für festes $x_0 \in (a,b)$ berechnet man zuerst das Integral bezüglich y in (c,d). Danach „summiert" man für alle $x_0 \in (a,b)$ auf, das heißt, man integriert die Ergebnisse aus dem ersten Schritt in (a,b).

Beispiel 2.1 Man berechne für $\Omega = (2,5) \times (1,3)$

$$\int_\Omega (5x^2y - 2y^3)\ d\Omega.$$

Nach (2.1) gilt

$$\int_\Omega (5x^2y - 2y^3)\ d\Omega = \int_1^3 \int_2^5 \left(5x^2y - 2y^3\right)\ dx\ dy.$$

Man berechnet zunächst das innere Integral

$$\int_2^5 \left(5x^2y - 2y^3\right)\ dx = \frac{5}{3}x^3y - 2xy^3 \Big|_2^5 = 195y - 6y^3.$$

Das setzt man zur Berechnung des äußeren Integrals ein

$$\int_1^3 \left(195y - 6y^3\right)\ dy = \frac{195}{2}y^2 - \frac{6}{4}y^4 \Big|_1^3 = 660.$$

□

Formel (2.1) lässt sich auf den Fall verallgemeinern, dass Ω oben und unten von zwei stetigen Funktionen $\varphi(x) \le \psi(x)$, $x \in [a,b]$, und seitlich von den beiden Ordinaten $x = a$ und $x = b$ begrenzt wird, siehe Abbildung 2.2. Die Längen der seitlichen Grenzen kann Null sein. Dann gilt

$$\int_\Omega f(x,y)\ d\Omega = \int_a^b \left(\int_{\varphi(x)}^{\psi(x)} f(x,y)\ dy \right) dx, \qquad (2.2)$$

sofern alle Integrale existieren.

Beispiel 2.2 Sei Ω der Kreis um den Koordinatenursprung mit Radius r. Man berechne

$$\int_\Omega y^2\sqrt{r^2 - x^2}\ d\Omega.$$

Für festes x mit $|x| \le r$ durchläuft y die Werte $-\sqrt{r^2-x^2}$ bis $\sqrt{r^2-x^2}$. Deshalb folgt mit (2.2)

$$\begin{aligned} \int_\Omega y^2\sqrt{r^2 - x^2}\ d\Omega &= \int_{-r}^{r} \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} y^2\sqrt{r^2 - x^2}\ dy\ dx \\ &= \int_{-r}^{r} \sqrt{r^2 - x^2} \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} y^2\ dy\ dx. \end{aligned}$$

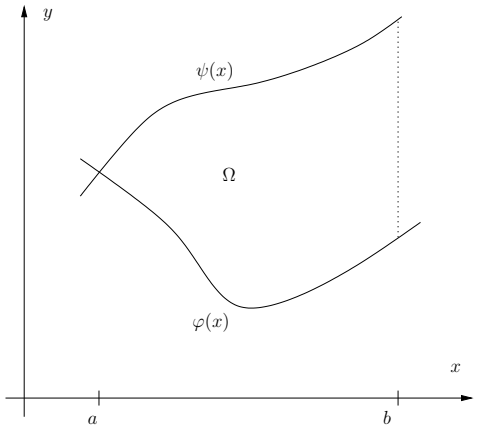


Abbildung 2.2: Durch Kurven begrenzte Fläche.

Für das innere Integral erhält man

$$\int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} y^2 \, dy = \frac{2}{3}(r^2 - x^2)^{3/2}.$$

Einsetzen ergibt

$$\int_\Omega y^2 \sqrt{r^2 - x^2} \, d\Omega = \frac{2}{3} \int_{-r}^{r} (r^2 - x^2)^2 \, dx = \frac{32}{45} r^5.$$

□

2.2 Der Gaußsche Integralsatz

Der Gaußsche Integralsatz stellt eine Beziehung zwischen einem Flächenintegral und einem Kurvenintegral her. Aus der eindimensionalen Integralrechnung ist bereits die Formel der partiellen Integration bekannt

$$\int_a^b u'v \, dx = u(b)v(b) - u(a)v(a) - \int_a^b uv' \, dx.$$

Im Spezialfall $v \equiv 1$ erhält man den Hauptsatz der Differential– und Integralrechnung

$$\int_a^b u' \, dx = u(b) - u(a).$$

Die rechte Seite kann man als „Integrale" über die nulldimensionalen Gebiete (Punkte) a und b auffassen. Die Dimension des Integrationsgebiets der ersten beiden Terme auf der rechten Seite ist also um Eins niedriger als auf der linken Seite. Solch eine Beziehung soll jetzt in zwei Dimensionen hergeleitet werden.

Zunächst wird der Divergenz–Operator definiert. Sei $\mathbf{v}$ ein auf einem Gebiet $\Omega \subset \mathbb{R}^d$ definiertes Vektorfeld, dann ist die Divergenz von $\mathbf{v}$ die Summe der ersten partiellen Ableitungen der i–ten Komponente von $\mathbf{v}$ nach der i–ten Koordinate

$$\text{div } \mathbf{v}(\mathbf{x}) = \nabla \cdot \mathbf{v}(\mathbf{x}) = \frac{\partial \mathbf{v}_1}{\partial x_1}(\mathbf{x}) + \ldots + \frac{\partial \mathbf{v}_d}{\partial x_d}(\mathbf{x}) = \sum_{i=1}^{d} \frac{\partial \mathbf{v}_i}{\partial x_i}(\mathbf{x}).$$

Wie auch in den anderen Abschnitten, wird der Gaußsche Integralsatz für einen Spezialfall hergeleitet und dann für den allgemeinen Fall nur angegeben. Als Spezialfall betrachten wir, dass $\Omega = (a, b) \times (c, d)$ ein Rechteck ist. Wir wollen

$$\int_\Omega \nabla \cdot \mathbf{v}(x, y)\ d\Omega = \int_a^b \int_c^d \nabla \cdot \mathbf{v}(x, y)\ dydx$$

berechnen. Man erhält, mit der eindimensionalen Formel der partiellen Integration,

$$\begin{aligned}
&\int_a^b \int_c^d \nabla \cdot \mathbf{v}(x, y)\ dydx \\
&= \int_c^d \left(\int_a^b \frac{\partial \mathbf{v}_1}{\partial x}(x, y)\ dx \right) dy + \int_a^b \left(\int_c^d \frac{\partial \mathbf{v}_2}{\partial y}(x, y)\ dy \right) dx \\
&= \int_c^d (\mathbf{v}_1(b, y) - \mathbf{v}_1(a, y))\ dy + \int_a^b (\mathbf{v}_2(x, d) - \mathbf{v}_2(x, c))\ dx. \qquad (2.3)
\end{aligned}$$

Wir bezeichnen den Rand von Ω mit $\partial\Omega$ und desweiteren sei $\mathbf{n}$ die äußere Einheitsnormale an $\partial\Omega$. Für den konkreten Fall des Rechtecks ist $\mathbf{n} = (0, -1)^T$ für die untere Seite $y = c$, $\mathbf{n} = (1, 0)^T$ für die rechte Seite $x = b$, $\mathbf{n} = (0, 1)^T$ für die obere Seite $y = d$ und $\mathbf{n} = (-1, 0)^T$ für die linke Seite $x = a$. Mit diesen Bezeichnungen kann man die Gleichung (2.3) kurz schreiben

$$\int_\Omega \nabla \cdot \mathbf{v}(x, y)\ d\Omega = \int_{\partial\Omega} \mathbf{v} \cdot \mathbf{n}\ ds. \qquad (2.4)$$

Die Gleichung (2.4) wird Gaußsche Integralsatz genannt. Er gilt für viel allgemeinere Gebiete als Rechtecke. Wichtig ist, dass das Gebiet in einem gewissen Sinne einen vernünftigen Rand besitzt.

Seien $u : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}^d$, eine skalare Funktion und $\mathbf{v} : \Omega \to \mathbb{R}^d$ ein Vektorfeld. Dann folgt mit Produktregel

$$\nabla \cdot (u\mathbf{v}) = \sum_{i=1}^d \frac{\partial (u\mathbf{v}_i)}{\partial x_i} = \sum_{i=1}^d \frac{\partial u}{\partial x_i}\mathbf{v}_i + u\frac{\partial \mathbf{v}_i}{\partial x_i} = \nabla u \cdot \mathbf{v} + u\nabla \cdot \mathbf{v}.$$

Mit dieser Beziehung folgt

$$\int_\Omega \nabla \cdot (u\mathbf{v})(x, y)\ d\Omega = \int_\Omega \nabla u \cdot \mathbf{v}\ d\Omega + \int_\Omega u\nabla \cdot \mathbf{v}\ d\Omega$$

und mit dem Gaußschen Satz folgt

$$\int_\Omega \nabla \cdot (u\mathbf{v})(x, y)\ d\Omega = \int_{\partial\Omega} u\mathbf{v} \cdot \mathbf{n}\ ds.$$

Durch Einsetzen und Umstellen erhält man

$$\int_\Omega \nabla u \cdot \mathbf{v}\ d\Omega = \int_{\partial\Omega} u\mathbf{v} \cdot \mathbf{n}\ ds - \int_\Omega u\nabla \cdot \mathbf{v}\ d\Omega \qquad (2.5)$$

womit die Ähnlichkeit zur Formel der partiellen Integration in einer Dimension offensichtlich ist. Die Beziehung (2.5) wird auch Gaußscher Integralsatz oder erste Greensche Formel genannt.

Der Laplace–Operator einer skalaren Funktion $u : \Omega \to \mathbb{R}$ ist die Summe aller nicht gemischten zweiten Ableitungen

$$\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \ldots + \frac{\partial^2 u}{\partial x_d^2} = \sum_{i=1}^d \frac{\partial^2 u}{\partial x_i^2} = \nabla \cdot \nabla u.$$

Die letzte Beziehung rechnet man leicht nach. Nach der ersten Greenschen Formel (2.5) gilt

$$\begin{aligned}\int_\Omega (\Delta u)v \; d\Omega &= \int_\Omega (\nabla \cdot \nabla u)v \; d\Omega = \int_{\partial\Omega} \nabla u \cdot \mathbf{n} v ds - \int_\Omega \nabla u \cdot \nabla v \; d\Omega \\ &= \int_{\partial\Omega} \frac{\partial u}{\partial \mathbf{n}} v ds - \int_\Omega \nabla u \cdot \nabla v \; d\Omega.\end{aligned}$$

Auch diese Beziehung wird oft erste Greensche Formel genannt. Der Ausdruck $\nabla u \cdot \mathbf{n} = \frac{\partial u}{\partial \mathbf{n}}$ ist die Richtungsableitung in Normalenrichtung, oder kurz die Normalenableitung auf dem Rand. Auf das Flächenintegral der rechten Seite kann man noch einmal die erste Greensche Formel (2.5) anwenden. Man erhält

$$\begin{aligned}\int_\Omega (\Delta u)v \; d\Omega &= \int_{\partial\Omega} \nabla u \cdot \mathbf{n} v ds - \int_\Omega \nabla u \cdot \nabla v \; d\Omega \\ &= \int_{\partial\Omega} \nabla u \cdot \mathbf{n} v ds - \int_{\partial\Omega} u \nabla v \cdot \mathbf{n} ds + \int_\Omega u(\Delta v) \; d\Omega. \qquad (2.6)\end{aligned}$$

Diese Formel wird zweite Greensche Formel genannt.

2.3 Variablensubstitution in Flächenintegralen

Ein wichtiges Hilfsmittel zur Berechnung eindimensionaler bestimmter Integrale ist die Variablensubstitution

$$\int_a^b f(g(x))g'(x) \; dx = \int_{g(a)}^{g(b)} f(u) \; du,$$

wobei $u = g(x)$ ist. Man kann auch in Flächenintegralen die Variablen substituieren. Damit kann man die Integration beispielsweise auf ein einfacheres Gebiet überführen, oder die natürlichen Koordinaten eines Problems (z.B. Polarkoordinaten) nutzen.

Die Herleitung der Formel für die Variablensubstitution ist recht langwierig und muss aus Zeitgründen entfallen. Seien Ω_1 und Ω_2 zwei Gebiete, die gewisse Eigenschaften bezüglich ihrer Ränder erfüllen und zwischen denen es eine eineindeutige Abbildung gibt

$$D \; : \; \Omega_2 \to \Omega_1, \; \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \varphi(\xi,\eta) \\ \psi(\xi,\eta) \end{pmatrix}.$$

Insbesondere dürfen die Gebiete gleich sein. Die Funktionaldeterminante der Abbildung ist

$$J(\xi,\eta) = \begin{vmatrix} \frac{\partial\varphi}{\partial\xi} & \frac{\partial\psi}{\partial\xi} \\ \frac{\partial\varphi}{\partial\eta} & \frac{\partial\psi}{\partial\eta} \end{vmatrix}.$$

Dann lautet die Formel für die Variablensubstitution

$$\int_{\Omega_1} f(x,y) \; dxdy = \int_{\Omega_2} f(\varphi(\xi,\eta),\psi(\xi,\eta)) \, |J(\xi,\eta)| \; d\xi d\eta, \qquad (2.7)$$

wobei man beachten muss, dass im Integral der Betrag der Funktionaldeterminante steht.

Beispiel 2.3 Seien $f(x,y) = xy$ und Ω_1 sei der Viertelkreis $x^2+y^2 \le R^2, x \ge 0, y \ge 0$. Bei der Integration über Gebiete, die etwas mit Kreisen zu tun haben, bieten sich

Polarkoordinaten an: $x = r\cos(\phi), y = r\sin(\phi)$. Für die Funktionaldeterminante erhält man

$$J(\xi,\eta) = \begin{vmatrix} \cos(\phi) & \sin(\phi) \\ -r\sin(\phi) & r\cos(\phi) \end{vmatrix} = r\left(\cos^2(\phi) + \sin^2(\phi)\right)\big) = r.$$

Der Viertelkreis ist gegeben durch $0 \le r < R$, $0 \le \phi \le \pi/2$. Man erhält also

$$\begin{aligned} \int_{\Omega_1} f(x,y)\ d\Omega &= \int_0^R \int_0^{\pi/2} r^2\cos(\phi)\sin(\phi) r\ d\phi dr \\ &= \int_0^R r^3 \frac{-\cos(2\phi)}{4}\Big|_0^{\pi/2}\ dr = \frac{1}{2}\int_0^R r^3\ dr = \frac{R^4}{8}. \end{aligned}$$

□

2.4 Erweiterungen der Integrale auf höhere Dimensionen

Eine natürliche Erweiterung des Flächenintegrals auf ein d–dimensionales Gebiet, $d \ge 3$, ist das Volumenintegral. Dieses braucht man zum Beispiel, um die Volumina von d–dimensionalen Körpern zu berechnen. Man versucht, das d–dimensionale Volumenintegral auf ein d–fach Integral zurückzuführen. Hat man zum Beispiel über einen Quader $\Omega = \times_{i=1}^d (a_i, b_i)$ zu integrieren, so erhält man in Analogie zu (2.1)

$$\int_\Omega f(x_1,\ldots,x_d)\ d\Omega = \int_{a_1}^{b_1} \left(\cdots \left(\int_{a_d}^{b_d} f(x_1,\ldots,x_d)\ dx_d \right) \cdots \right)\ dx_1.$$

Unter entsprechenden Voraussetzungen an die zu integrierende Funktion, kann man die Integrationsreihenfolge beliebig vertauschen (Satz von Fubini). **check**

Eine zweite Erweiterung des Flächenintegralbegriffs besteht in der Integration auf $d-1$–dimensionalen Hyperflächen. Für $d = 3$ spricht man vom Oberflächenintegral und man hat beispielsweise über den Rand eines Quaders oder die Oberfläche einer Kugel zu integrieren. Die allgemeine Behandlung des Oberflächenintegrals ist recht kompliziert und kann aus Zeitgründen nicht geschehen. Die formale Schreibweise ist wie in zwei Dimensionen

$$\int_{\partial\Omega} f\ d\mathbf{s}.$$

Die in zwei Dimensionen behandelten Beziehungen gelten sinngemäß auch in höheren Dimensionen:

- eine mehrfache Integralformel wie (2.2),
- der Gaußsche Integralsatz (2.4),
- die erste Greensche Formel (2.5),
- die zweite Greensche Formel (2.6),
- die Formel der Variablensubstitution (2.7).

Dazu gibt es Übungsaufgaben.

Teil II

Einführung in die Numerische Mathematik

Kapitel 1

Einführung

Die Numerische Mathematik ist eine etwas andere Art von Mathematik, als sie bisher in der Vorlesung geboten wurde. Sie ist vor allem für die Lösung von Problemen in den Anwendungen von entscheidender Bedeutung.

1.1 Aufgabenstellungen und Ziele der Numerischen Mathematik

Viele Probleme, die in der Mathematik auftreten, kann man mit ,,rein mathematischen Methoden" nur sehr schwer oder gar nicht lösen. Beispiele sind:

- die Berechnung vieler bestimmter Integrale, da man die Stammfunktion nicht findet,
- die Berechnung von Nullstellen von Funktionen, da man nicht nach der unbekannten Variablen auflösen kann,
- die Lösung großer linearer Gleichungssystem per Hand ist sehr zeitaufwendig,
- die Lösung vieler Differentialgleichung geht nicht analytisch.

Innerhalb der Numerischen Mathematik werden Verfahren (Algorithmen) zur approximativen Lösung dieser Probleme entwickelt, welche mit Hilfe von Computern abgearbeitet werden können. Wichtige Fragestellungen, die im Rahmen der Numerischen Mathematik untersucht werden, sind:

- Wie teuer sind diese Verfahren? Wie lange dauert ihre Abarbeitung?
- Wie genau ist das berechnete Ergebnis?
- Unter welchen Bedingungen konvergieren Interationsverfahren? Wie schnell ist die Konvergenz?
- Wie robust sind die Verfahren gegenüber Datenfehlern?

In dieser Vorlesung kann aus Zeitgründen auf Details zur Beantwortung dieser Fragen nicht eingegangen werden. Es werden lediglich wichtige Antworten präsentiert.

1.2 Computerzahlen und numerische Verfahren

Auf einem Computer kann nur eine endliche Menge von reellen Zahlen dargestellt werden. Zahlen, die man nicht darstellen kann, müssen gerundet werden. Deswegen ist jede Rechenoperation mit Computerzahlen potentiell mit Rundungsfehlern behaftet.

Alle modernen Computer nutzen zur Darstellung von Computerzahlen die Ba-

sis 2, das heißt das Binärsystem. In dieser Basis wird eine Zahl $x \in \mathbb{R}$ durch

$$x = \sum_{k=-\infty}^{\infty} x_k 2^k \tag{1.1}$$

dargestellt. Die Koeffizienten x_k werden Ziffern genannt. Im Binärsystem hat man nur die Ziffern $x_k \in \{0, 1\}$. Zum Vergleich, im gewöhnlichen Dezimalsystem hat man

$$x = \sum_{k=-\infty}^{\infty} y_k 10^k$$

mit den Ziffern $y_k \in \{0, 1, \ldots, 9\}$. Auf einem Computer kann man nur Zahlen mit endlich vielen Ziffern darstellen, das bedeutet, anstelle von (1.1) hat man

$$x = \sum_{k=-m}^{M} x_k 2^k, \quad 0 < m, M < \infty.$$

Beispiel 1.1 *Binärdarstellungen einiger Zahlen.*

$$\begin{aligned} 5 &= 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 101, \\ 0.5 &= 1 \cdot 2^{-1} = 0.1, \\ 123.75 &= 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} \\ &= 1111011.11. \end{aligned}$$

□

Moderne Programmiersprachen (C, C++, FORTRAN90) unterscheiden zwischen ganzen Zahlen (integer) und Fließkommazahlen (double). Zur Speicherung von Integerzahlen werden im allgemeinen 4 Byte = 32 Bit genutzt. Ein Bit braucht man für das Vorzeichen, so dass man noch 31 Bits für Ziffern hat. Für double–Zahlen werden im allgemeinen 8 Byte = 64 Bit genutzt. Dabei verwendet man die Darstellung

$$x = \pm m 2^e,$$

wobei man ein Bit für das Vorzeichen braucht, im allgemeinen 52 Bit für die Ziffern m (Mantisse) und 11 Bit für den Exponenten e verwendet.

Infolge der endlich vielen Computerzahlen gelten nicht mehr alle bekannten Rechengesetze. Das trifft zum Beispiel auf das Assoziativitätsgesetz der Addition zu. Man erhält auf dem Computer

$$1 + 10^{20} - 10^{20} = 0, \quad 1 + \left(10^{20} - 10^{20}\right) = 1.$$

(*mit MATLAB demonstrieren.*)

Falls man numerische Verfahren im Detail untersuchen will, muss man wissen, wie sich die Rundungsfehler auf das Ergebnis auswirken. Solche Untersuchungen können hier aus Zeitgründen nicht präsentiert werden. Es stellt sich heraus, dass die meisten Operationen gutartig sind, das heißt, kleine Rundungsfehler in den Eingangsdaten führen nur zu kleinen Fehlern in den berechneten Ergebnissen. Es gibt jedoch eine wichtige Ausnahme: Die Subtraktion zweier fast gleich großer Zahlen kann zu einem großen relativen Fehler im Ergebnis führen, siehe Übungsaufgabe. Dieses Phänomen nennt man Auslöschung.

Die Kosten der Verfahren werden oft in der Anzahl der für die Berechnung benötigten Operationen (Addition, Subtraktion, Multiplikation, Division) gemessen. Diese Operationen nennt man Flops (floating point operations).

1.3 Klassifizierung von Problemen

Wir werden in dieser Vorlesung nur korrekt gestellte Problem betrachten. Ein Problem heißt korrekt gestellt, falls

- es eine eindeutige Lösung besitzt,
- die Lösung stetig von den Daten abhängt.

Anderenfalls heißt das Problem schlecht gestellt. Schlecht gestellte Probleme, zum Beispiel sogenannte inverse Probleme, spielen in der Praxis eine große Rolle. Ihre numerische Behandlung ist jedoch ziemlich kompliziert und jenseits dem Anliegen dieser Vorlesung.

Stetige Abhängigkeit von den Daten bedeutet, dass kleine Störungen in den Daten nur kleine Änderungen in der Lösung bewirken. Diese Eigenschaft ist sehr wichtig für die numerische Lösung von Problemen, da man durch das Runden eigentlich immer kleine Störungen der Daten hat. Man kann ein abstraktes Maß definieren, welches die Störungen der Lösung in Abhängigkeit von den Datenstörungen misst – die sogenannte Konditionszahl. Wir werden dieses Maß im Spezialfall bei der Lösung linearer Gleichungssysteme kennenlernen. Grob gesprochen ist die Konditionszahl der Quotient des maximalen Lösungsfehlers und eines vorgegebenen Datenfehlers. Das heißt, eine kleine Konditionszahl ist günstig. Korrekt gestellte Probleme mit großer Konditionszahl werden schlecht konditioniert genannt.

Kapitel 2

Numerische Lösung linearer Gleichungssysteme

Dieses Kapitel behandelt numerische Verfahren zur Lösung linearer Gleichungssysteme der Gestalt

$$A\mathbf{x} = \mathbf{b}, \quad A \in \mathbb{R}^{n\times n},\ \mathbf{x}, \mathbf{b} \in \mathbb{R}^n \tag{2.1}$$

mit

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}.$$

Viele Methoden zur Lösung von Problemen erfordern im Kern die Lösung linearer Systeme der Form (2.1). Daher sind effiziente und zuverlässige Verfahren zur numerischen Lösung von (2.1) von großer Wichtigkeit.

In der Praxis unterscheidet man folgende Typen linearen Systemen:

1.) die Dimension n ist klein, $n \lesssim 10000$,
2.) die Dimension n ist groß, $10000 \lesssim n \lesssim 10^8$, und die Elemente von A sind überwiegend Null (A ist schwach besetzt),
3.) die Dimension von A ist groß und A ist nicht schwach besetzt.

In der Vorlesung werden vorwiegend Verfahren für Matrizen vom Typ 1 behandelt. Im letzten Abschnitt werden auch einige Verfahren kurz vorgestellt, die man prinzipiell für alle Typen verwenden kann.

2.1 Theorie

In diesem Abschnitt werden wichtige Eigenschaften von Matrizen und Vektoren kurz zusammengefasst und einige Begriffe eingeführt.

Aus der linearen Algebra ist folgender Sachverhalt bekannt.

Satz 2.1 *Sei $A \in \mathbb{R}^{n\times n}$. Die folgenden Aussagen sind äquivalent:*

- *A ist eine reguläre Matrix, das heißt der Rang von A ist $rg(A) = n$,*
- *alle Eigenwerte von A sind ungleich Null,*
- *das System (2.1) besitzt genau eine Lösung,*
- *das System (2.1) mit der rechten Seite $\mathbf{b} = \mathbf{0}$ besitzt nur die triviale Lösung $\mathbf{x} = \mathbf{0}$,*
- *die Determinante von A ist ungleich Null,* $\det(A) \neq 0$,
- *die Matrix A ist invertierbar.*

Damit das Problem (2.1) korrekt gestellt ist, muss die Lösung eindeutig sein. Also muss A regulär sein. Außerdem gilt für eine reguläre Matrix A und eine Störung $\delta\mathbf{b}$

$$A\mathbf{x} = \mathbf{b} + \delta\mathbf{b} \quad \Longleftrightarrow \quad \mathbf{x} = A^{-1}\mathbf{b} + A^{-1}(\delta\mathbf{b}).$$

Da die Matrizenmultiplikation stetig ist, folgt

$$\lim_{\delta\mathbf{b}\to\mathbf{0}} A^{-1}(\delta\mathbf{b}) = A^{-1}\mathbf{0} = \mathbf{0}.$$

Damit hängt die Lösung stetig von den Störungen der rechten Seite ab. Außerdem kann man zeigen, dass hinreichend kleine Störungen der Matrixkoeffizienten verändern nicht die Regularität der Matrix. Das wird am Ende dieses Abschnitts noch genauer angegeben. Es gilt, dass die Lösung auch stetig von Störungen der Matrixkoeffizienten abhängt. Damit ist gezeigt, dass Problem (2.1) genau dann korrekt gestellt ist, wenn A regulär ist.

Für lineare Gleichungssysteme (2.1) mit regulärer Matrix A soll auch der Begriff der Kondition des Problems genauer besprochen werden. Dazu benötigt man Vektor– und Matrixnormen.

Vektornormen. $\mathbf{x} \in \mathbb{R}^n$, l^p–Norm, $p \in [1, \infty)$:

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

$p = 1$ – Summennorm, $p = 2$ – Euklidische Norm; für $p = \infty$ hat man die Maximumnorm

$$\|\mathbf{x}\|_\infty := \max_{i=1,\ldots,n} |x_i|$$

Matrixnormen. $A \in \mathbb{R}^{m\times n}$. Die induzierte Matrixnorm ist gegeben durch

$$\|A\|_p := \max_{\mathbf{x}\in\mathbb{R}^n, \mathbf{x}\neq\mathbf{0}} \frac{\|A\mathbf{x}\|_p}{\|\mathbf{x}\|_p} = \max_{\|\mathbf{x}\|_p=1} \|A\mathbf{x}\|_p .$$

Man findet

$$\begin{aligned}
\|A\|_1 &= \max_{j=1,\ldots,n} \sum_{i=1}^{m} |a_{ij}| \quad \text{Spaltensummennorm,} \\
\|A\|_2 &= \sqrt{\lambda_{\max}\left(A^T A\right)} \quad \text{Spektralnorm,} \\
\|A\|_\infty &= \max_{i=1,\ldots,m} \sum_{j=1}^{n} |a_{ij}| \quad \text{Zeilensummennorm.}
\end{aligned}$$

λ – Eigenwert. Außerdem

$$\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 \right)^{1/2} \quad \text{Frobenius–Norm.}$$

Eine Vektornorm und eine Matrixnorm nennt man verträglich, falls für alle Matrizen und Vektoren gilt

$$\|A\mathbf{x}\|_{\text{vektor}} \leq \|A\|_{\text{matrix}} \|\mathbf{x}\|_{\text{vektor}} .$$

Man kann zeigen, dass eine l^p–Vektornorm und ihre induzierte Matrixnorm verträglich sind. *Übungsaufgabe*

Definition 2.2 Die Konditionszahl einer regulären Matrix $A \in \mathbb{R}^{n\times n}$ ist definiert als

$$\kappa(A) = \|A\| \left\|A^{-1}\right\|,$$

wobei $\|\cdot\|$ eine der oben angegebenen Matrixnormen ist. □

Unterschiedliche Normen ergeben unterschiedliche Konditionszahlen. Diese werden durch einen Index unterschieden.

Eigenschaften der Konditionszahl.

- $\kappa(A) \geq 1$, da (I – Einheitsmatrix)

$$1 = \|I\| = \left\|AA^{-1}\right\| \leq \|A\| \left\|A^{-1}\right\| = \kappa(A),$$

- $\kappa(A^{-1}) = \kappa(A)$,
- sei $\alpha \in \mathbb{R} \setminus \{0\}$

$$\kappa(\alpha A) = \|\alpha A\| \left\|(\alpha A)^{-1}\right\| = |\alpha| \left|\alpha^{-1}\right| \|A\| \left\|A^{-1}\right\| = \kappa(A),$$

- ist A eine orthogonale Matrix, dass heißt $A^T = A^{-1}$, dann ist $\kappa_2(A) = 1$, da

$$\|A\|_2 = \left\|A^T\right\|_2 = \left\|A^{-1}\right\|_2$$

und

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)} = \sqrt{\lambda_{\max}(I)} = 1,$$

$\kappa_2(\cdot)$ – Spektralkonditionszahl.

- ist A symmetrisch und positiv definit, das heißt alle Eigenwerte von A sind positiv, dann gilt *Übungsaufgabe*

$$\kappa_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)},$$

Definiert man den relativen Abstand einer regulären Matrix A zur Menge der singulären Matrizen wie folgt

$$\operatorname{dist}_p(A) := \min\left\{ \frac{\|\delta A\|_p}{\|A\|_p} \ : \ A + \delta A \text{ ist singulär} \right\},$$

so kann man zeigen, dass

$$\operatorname{dist}_p(A) = \frac{1}{\kappa_p(A)}.$$

Dass bedeutet, dass bei Matrizen mit einer großen Konditionszahl schon kleine Störungen dazu führen können, dass die gestörte Matrix singular ist. Des weiteren ist die Matrix $A + \delta A$ auf jeden Fall regulär, falls

$$\frac{\|\delta A\|_p}{\|A\|_p} < \frac{1}{\kappa_p(A)} \quad \Longleftrightarrow \quad \|\delta A\|_p \left\|A^{-1}\right\|_p < 1.$$

Auf Grund der allgegenwärtigen Rundungsfehler wird ein numerisches Verfahren zur Lösung von (2.1) nur eine Näherungslösung berechen, die jedoch die Lösung eines gestörten linearen Systems ist

$$(A + \delta A)(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}.$$

In der Literutur findet man Abschätzungen für den relativen Fehler $\|\delta\mathbf{x}\| \,/\, \|\mathbf{x}\|$ mit Hilfe der Konditionszahl der Matrix A, siehe [QSS04].

2.2 Numerische Lösung linearer Systeme mit Dreiecksmatrix

Die Matrix $A \in \mathbb{R}^{n\times n}$ habe eine der beiden Formen

$$L = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & 0 & \\ \dots & \dots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix}, \quad U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & 0 & \ddots & \vdots \\ & & & u_{nn} \end{pmatrix},$$

L – untere Dreiecksmatrix (lower), U – obere Dreiecksmatrix (upper).

Da A regulär sein soll, folgt $l_{ii} \neq 0$ beziehungsweise $u_{ii} \neq 0$ für $i = 1, \dots, n$. Die Komponenten des Lösungsvektors können in diesen Fällen nacheinander berechnet werden, zum Beispiel für $L\mathbf{x} = \mathbf{b}$:

$$\begin{aligned} x_1 &= b_1/l_{11}, \\ x_2 &= (b_2 - l_{21}x_1)/l_{22}, \\ &\vdots \\ x_i &= \frac{1}{l_{ii}}\left(b_i - \sum_{j=1}^{i-1} l_{ij}x_j\right), \quad i = 1, \dots, n. \end{aligned}$$

Diese Vorgehensweise wird auch in dieser Form implementiert. *Übungsaufgabe* Die Durchführung dieses Verfahrens erfordert $n(n+1)/2$ Multiplikationen/Divisionen und $n(n-1)/2$ Additionen/Subtraktionen. *Übungsaufgabe* Die Gesamtzahl der benötigten Flops ist somit n^2.

Man beachte, dass man nach der Berechnung von x_i den Wert b_i der rechten Seite nicht mehr benötigt. Deshalb kann man die Lösungskomponente x_i sofort auf dem Speicherplatz von b_i speichern. Man benötigt also keinen zusätzlichen Speicherplatz für die Lösung.

Für $L\mathbf{x} = \mathbf{b}$ ist die Reihenfolge der berechneten Komponenten $x_1 \to x_2 \to \cdots \to x_n$ und man spricht von Vorwärtssubstitution. Löst man $U\mathbf{x} = \mathbf{b}$, so erhält man $x_n \to x_{n-1} \to \cdots \to x_1$ und dies wird Rückwärtssubstitution genannt.

Benötigt man den Vektor $\mathbf{b}$ nicht mehr, so kann man im Verfahren die berechneten Komponenten x_i auf dem Platz von b_i speichern.

Führt man eine Rundungsfehleranalyse für die Vorwärts–(Rückwärts–)substitution durch, erhält man folgendes Ergebnis, [KS88, S. 146f.]: Die mit der Vorwärtssubstitution berechnete Lösung $\mathbf{x}$ genügt der Gleichung

$$(L + \delta L)\mathbf{x} = \mathbf{b}$$

mit einer unteren Dreiecksmatrix δL, die komponentenweise klein ist im Sinne von

$$|(\delta L)_{ij}| \leq \varepsilon_M j\, |l_{ij}|, \quad i, j = 1, \dots, n,\ i \geq j,$$

wobei ε_M das Maschinenepsilon (siehe Übungen) ist. Man beachte, dass nur die Matrix L mit durch Rundungsfehler hervorgerufene Störungen behaftet ist. Die Aussage bedeutet, dass die Vorwärts–(Rückwärts–)substitution numerisch gutartige (gut konditionierte) Verfahren sind. Diese sollen im folgenden für die Lösung eines allgemeinen linearen Gleichungssystems der Form (2.1) genutzt werden.

2.3 Das Gauß–Verfahren und die LU–Zerlegung

Das bekannte Gauß–Verfahren ist bei richtiger Anwendung ein stabiles und effizientes Verfahren zur Lösung kleiner linearer Gleichungssysteme.

Vorgehen. Man formt das System $A\mathbf{x} = \mathbf{b}$ in ein äquivalentes System $U\mathbf{x} = \tilde{\mathbf{b}}$ um, wobei U eine obere Dreiecksmatrix ist, welches sich durch Rückwärtssubstitution leicht lösen lässt. Bei den Umformungen nutzt man aus, dass sich die Lösung des Systems $A\mathbf{x} = \mathbf{b}$ nicht ändert, wenn

- ein Vielfaches einer Gleichung zu einer anderen Gleichung addiert wird,
- zwei Gleichungen vertauscht werden,
- eine Gleichung mit einer rellen Zahl $\neq 0$ multipliziert wird.

Man hat also zwei Phasen beim Lösen von $A\mathbf{x} = \mathbf{b}$, zuerst die Erzeugung des Dreieckssystems (Vorwärtselimination) und dann die Rückwärtssubstitution.

1. Vorwärtselimination. (Erzeugung von $U\mathbf{x} = \tilde{\mathbf{b}}$)

- Falls $a_{11} \neq 0$, dann eliminiert man x_1 aus den letzten $(n-1)$ Gleichungen, indem man von der i–ten Gleichung das

$$m_{i1} = \frac{a_{i1}}{a_{11}} \; - \; \text{fache}$$

der 1. Gleichung subtrahiert, $i = 2, \ldots, n$, subtrahiert. Dann erhält man das modifizierte System

$$\left(\begin{array}{c|ccc} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ \hline 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{array}\right) \left(\begin{array}{c} x_1 \\ \hline x_2 \\ \vdots \\ x_n \end{array}\right) = \left(\begin{array}{c} b_1^{(1)} \\ \hline b_2^{(2)} \\ \vdots \\ b_n^{(2)} \end{array}\right)$$

mit

$$\begin{aligned} a_{ij}^{(1)} &= a_{ij}, \quad b_i^{(1)} = b_i, \\ a_{ij}^{(2)} &= a_{ij}^{(1)} - m_{i1} a_{1j}^{(1)}, \quad b_i^{(2)} = b_i^{(1)} - m_{i1} b_1^{(1)}. \end{aligned}$$

Das System ohne erste Zeile und Spalte wird mit $A^{(2)}\mathbf{x}^{(2)} = \mathbf{b}^{(2)}$ bezeichnet.

- Falls $a_{22}^{(2)} \neq 0$, so wendet man das gleiche Vorgehen auf $A^{(2)}\mathbf{x}^{(2)} = \mathbf{b}^{(2)}$ an und eliminiert x_2. Dieses Vorgehen wird fortgesetzt, falls $a_{ii}^{(i)} \neq 0$, $i = 3, \ldots n-1$. Als Ergebnis erhält man ein oberes Dreieckssystem

$$\left(\begin{array}{ccccc} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1,n-1}^{(1)} & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2,n-1}^{(2)} & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{n-1,n-1}^{(n-1)} & a_{n-1,n}^{(n-1)} \\ 0 & 0 & \cdots & 0 & a_{nn}^{(n)} \end{array}\right) \left(\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{array}\right) = \left(\begin{array}{c} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_{n-1}^{(n-1)} \\ b_n^{(n)} \end{array}\right). \quad (2.2)$$

2. Rückwärtssubstitution. Die Lösung $\mathbf{x}$ wird nun durch Rückwärtssubstitution aus (2.2) berechnet.

Dieses Vorgehen funktioniert, solange die Elemente $a_{ii}^{(i)} \neq 0$, $i = 1, \ldots, n$, sind. Das ist aber für reguläre Matrizen nicht selbstverständlich, zum Beispiel bei

$$A = \left(\begin{array}{cc} 0 & 1 \\ 1 & 1 \end{array}\right)$$

ist $a_{11}^{(1)} = 0$.

Sei beim k–ten Schritt das Element $a_{kk}^{(k)} = 0$. Eines der Elemente der restlichen Spalte $a_{ik}^{(k)}$, $i = k+1, \ldots, n$, muss dann ungleich Null sein, etwa $a_{k'k}^{(k)}$. Ansonsten

wäre die k–te Spalte von den vorhergehenden Spalten linear abhängig und A wäre damit singulär. Man vertausche die Zeilen k und k', auch in der rechten Seite, und setzte die Vorwärtselimination fort. Mit dieser Vertauschung vertauscht man zwei Gleichungen des linearen Systems.

Das Diagonalenelement $a_{kk}^{(k)}$, welches man zur Definition der Multiplikatoren

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \tag{2.3}$$

verwendet, und welches man eventuell erst durch eine Zeilenvertauschung erhält, nennt man Pivotelement. Zur Fehlerdämpfung stellt es sich als günstig heraus, wenn die Multiplikatoren (2.3) vom Betrage her möglichst klein sind. Das bedeutet, $\left|a_{kk}^{(k)}\right|$ sollte möglichst groß sein. Die gebräuchlichste Strategie um betragsmäßig kleine Multiplikatoren zu bekommen ist die sogenannte Spaltenpivotsuche: Suche das betragsmäßig größte Element in der restlichen k–ten Spalte

$$\left|a_{k'k}^{(k)}\right| \geq \left|a_{ik}^{(k)}\right|, \quad i \geq k.$$

Dann tauscht man die Zeilen k und k', auch in $\mathbf{b}$. Es gibt auch Pivotstrategien, bei denen man Spalten vertauscht.

Bei dieser Pivotstrategie gilt $|m_{ik}| \leq 1$. Erhält man zum Schluss der Vorwärtselimination auch mit Pivotsuche kleine Pivotelemente, so ist das ein Hinweis darauf, dass die Matrix A schlecht konditioniert ist.

Das Gaußsche Verfahren soll nun so formuliert werden, dass man die Matrix A überspeichern kann, ohne sie damit zu verlieren. Man wird sie indirekt behalten. Sei $n = 2$. Wir definieren mit $m_{21} = a_{21}/a_{11}$

$$L = \begin{pmatrix} 1 & 0 \\ m_{21} & 1 \end{pmatrix}, \quad U = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} \\ 0 & a_{22}^{(2)} \end{pmatrix},$$

mit $a_{22}^{(2)} = a_{22}^{(1)} - m_{21}a_{12}^{(1)}$. Lässt man der Einfachheit halber die oberen Indizes weg sofern sie (1) sind, so erhält man

$$LU = \begin{pmatrix} a_{11} & a_{12} \\ m_{21}a_{11} & m_{21}a_{12} + a_{22}^{(2)} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & m_{21}/a_{12} + a_{22} - m_{21}/a_{12} \end{pmatrix} = A.$$

Diese Beobachtung lässt sich für $A \in \mathbb{R}^{n\times n}$ verallgemeinern. Seien

$$L = \begin{pmatrix} 1 & & & \\ m_{21} & 1 & 0 & \\ \vdots & & \ddots & \\ m_{n1} & \cdots & m_{n,n-1} & 1 \end{pmatrix}, \quad U = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn}^{(n)} \end{pmatrix},$$

dann gilt

$$PA = LU, \tag{2.4}$$

wobei P eine Permutationsmatrix ist, die die Zeilen– (und eventuellen Spalten–) vertauschungen beschreibt, welche bei der Vorwärtselimination erfolgen. Man braucht diese Matrix nicht zu speichern, falls alle erforderlichen Vertauschungen in der rechten Seite gleich ausgeführt werden $\mathbf{b} \to P\mathbf{b}$. Beschränkt man sich auf Zeilenvertauschungen, so braucht man für die Beschreibung der Vertauschungen nur einen Vektor zu speichern. Hat man keine Vertauschungen, so gilt $A = LU$. Die Zerlegung (2.4) nennt man LU–Zerlegung oder LR–Zerlegung von A.

Bei der allgemeinen numerischen Herangehensweise zur Lösung von (2.1) entkoppelt man die Vorwärtselimination der Matrix A und die Anwendung dieser Elimination auf die rechte Seite $\mathbf{b}$. Somit hat man drei Schritte:

1. berechne die LU–Zerlegung von A: $PA = LU$,
2. berechne $\mathbf{y}$ aus
$$PA\mathbf{x} = L(U\mathbf{x}) = L\mathbf{y} = P\mathbf{b}$$
durch Vorwärtssubstitution,
3. berechne $\mathbf{x}$ aus
$$U\mathbf{x} = \mathbf{y}$$
durch Rückwärtssubstitution.

Wenn die Matrix A nicht mehr explizit benötigt wird, können L und U sofort auf dem Speicherplatz von A gespeichert werden. Man beachte, dass man die Diagonale von L nicht zu speichern braucht, da die Einträge $(= 1)$ sowieso bekannt sind. Mit Hilfe von L, U und der Permutationsmatrix P kann man jedoch gegebenenfalls A wieder rekonstruieren. Braucht man die rechte Seite $\mathbf{b}$ nicht mehr, so kann dieser Speicherplatz sofort für $\mathbf{y}$ und dann für die Lösung $\mathbf{x}$ verwendet werden. Man kann das Gauß–Verfahren also so implementieren, dass man keinen zusätzlichen Speicher benötigt für die LU–Zerlegung von A und für die Lösung benötigt.

Die Kosten zur Lösung von $A\mathbf{x} = \mathbf{b}$ sind wie folgt

1. $\frac{2}{3}n^3 - \frac{1}{2}n^2 - \frac{7}{6}n$ Flops, *Übungsaufgabe*
2. n^2 Flops,
3. n^2 Flops,

woraus Gesamtkosten von
$$\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{7}{6}n \text{ Flops} \approx \frac{2}{3}n^3 \text{ Flops}$$
resultieren.

Man kann nicht mathematisch beweisen, dass die LU–Zerlegung mit Spaltenpivotsuche stabil ist. Im Gegenteil, man kann (pathologische) Beispiele konstruieren, bei denen eine Fehlerschranke, die man beweisen kann, mit wachsender Dimension n des Gleichungssystems exponentiell anwächst. In den allermeisten praktischen Fällen ist dies jedoch nicht der Fall. Auf Grund dieser Erfahrungen wird die Gauß–Elimination mit Spaltenpivotsuche als praktisch stabil angesehen und im allgemeinen verwendet.

Die LU–Zerlegung ohne Pivotsuche ist jedoch im allgemeinen instabil. Es gibt allerdings Klassen von Matrizen, für die man zeigen kann, dass die LU–Zerlegung auch ohne Pivotsuche stabil ist, beispielsweise symmetrische und positiv definite Matrizen. Dann kann man sich die Pivotsuche sparen und spart damit Rechenzeit.

Analysiert man das Gesamtverfahren 1)–3) zur Lösung des linearen Gleichungssystems (2.1), so erhält man folgende Aussagen:

Satz 2.3 *Löst man das lineare Gleichungssystem $A\mathbf{x} = \mathbf{b}$ mittels Gauß–Elimination mit Spaltenpivotsuche, so ist die berechnete Lösung $\mathbf{x}$ die Lösung des gestörten Systems*
$$(A + \delta A)\mathbf{x} = \mathbf{b}$$
und für die Fehlermatrix δA gilt die Abschätzung
$$\|\delta A\|_\infty \leq (\varepsilon_M n^3 + 3n^2)\rho\varepsilon_M \|A\|_\infty ,$$
wobei ε_M das Maschinenepsilon und
$$\rho = \max_{i,j,k} \left(\frac{\left|a_{ij}^{(k)}\right|}{\|A\|_\infty} \right)$$
sind.

Sei $\mathbf{x}^*$ *die Lösung von* $A\mathbf{x} = \mathbf{b}$*, dann gilt für den Fehler*

$$\|\mathbf{x}^* - \mathbf{x}\|_\infty \leq \kappa_\infty(A)(\varepsilon_M n^3 + 3n^2)\rho\varepsilon_M \|\mathbf{x}\|_\infty .$$

Die Abschätzung von $\|\delta A\|_\infty$ im Satz stellt für große n im allgemeinen eine starke Überschätzung dar. In der Praxis ist $\|\delta A\|_\infty$ kaum größer als $\varepsilon_M n \|A\|_\infty$. Ebenso ist die Abschätzung für den Fehler im allgemeinen zu pessimistisch. Man erkennt jedoch den Einfluss der Konditionszahl von A.

Bemerkung 2.4

1. Hat man mehrere Systeme mit der Matrix A und unterschiedlichen rechten Seiten $\mathbf{b}_1, \ldots, \mathbf{b}_l$ zu lösen, so führt man zuerst die LU–Zerlegung wie oben beschrieben durch ($\approx 2n^3/3$ Flops). Man speichert L, U und P. Danach wendet man Vorwärts– und Rückwärtssubstitution für alle rechten Seiten an (je $2n^2$ Flops). Sind alle rechten Seiten direkt nach der LU–Zerlegung bekannt, kann man die Substitutionen für diese rechten Seiten sogar gleichzeitig durchführen. Ist dies nicht der Fall so braucht man ab der zweiten rechten Seite die teure LU–Zerlegung nicht mehr zu berechnen.
2. Die inverse Matrix erhält man, falls man die in Bemerkung 2.4.1 beschriebene Vorgehensweise mit $\mathbf{b}_j = \mathbf{e}_j$ (j–ter Einheitsvektor), $j = 1, \ldots, n$, durchführt. Die Berechnung der inversen Matrix ist jedoch praktisch fast nie von Interesse. Hat man in einer Aufgabenstellung „$A^{-1}\mathbf{b}$" zu berechnen, so heißt das „löse das System $A\mathbf{x} = \mathbf{b}$". Die Matrix A^{-1} wird hierbei nicht gebraucht.
3. Sind die Matrizen L und U auf dem Speicherplatz von A gespeichert und braucht man den Wert des Produkts $\mathbf{x} = A\mathbf{y}$, so erhält man diesen durch

   $$\mathbf{z} = U\mathbf{y}, \quad \mathbf{w} = L\mathbf{z}, \quad \mathbf{x} = P^{-1}\mathbf{w}.$$

   Bei Spaltenpivotsuche ist P^{-1} nur eine Vertauschung der Komponenten. Die Berechnung des Produkts auf diesem Weg benötigt die gleiche Anzahl an Flops wie die Berechnung von $A\mathbf{y}$ bei abgespeicherter Matrix A.
4. Die Determinante der Matrix A erhält man direkt aus ihrer LU–Zerlegung. Für Permutationsmatrizen gilt $\det(P) = 1$. Damit hat man

   $$\det(A) = \det(PA) = \det(LU) = \det(L)\det(U) = \det(U) = \prod_{i=1}^{n} a_{ii}^{(i)}.$$

   Die Determinanten der Dreiecksmatrizen L und U ist jeweils das Produkt der Hauptdiagonalelemente.
5. Ist A eine symmetrische und (positiv) definite Matrix, so verwendet man eine Variante des Gauß–Verfahrens, bei welcher man eine Zerlegung der Gestalt

   $$A = C^T C, \quad C - \text{ obere Dreiecksmatrix},$$

   erhält, das sogenannte Cholesky–Verfahren. Wenn A symmetrisch ist, reicht es, das obere Dreieck von A zu speichern. Bei der normalen LU–Zerlegung müsste man jedoch sowohl L als auch U speichern, also eine volle quadratische Matrix. Beim Cholesky–Verfahren ist dies nicht nötig und man kann C auf dem Speicherplatz von A speichern.
6. In MATLAB steckt das Gaußsche Verfahren mit Spaltenpivotsuche hinter dem Befehl `x = A\b`. (*Demo)*

□

2.4 Klassische Iterationsverfahren zur Lösung linearer Gleichungssysteme

Bei der Diskretisierung partieller Differentialgleichungen, die zum Beispiel physikalische Prozesse beschreiben, muss man letztlich oft große lineare Systeme mit schwach besetzten Matrizen lösen. Dazu sind direkte Verfahren, wie das Gauß–Verfahren, im allgemeinen ungeeignet. Zum einen ist die Rechenzeit zu lang. Ist beispielsweise $n = 10^6$ und hat man einen Computer mit einem Gigaflop (10^9 Flops/s), dann braucht man mit dem Gauß–Verfahren zur Lösung des linearen Systems ungefähr 21.1 Jahre. Wendet man das Gauß–Verfahren auf eine schwach besetzte Matrix an, dann muss man im allgemeinen damit rechnen, dass die Faktoren L und U nicht mehr schwach besetzt sind. Deshalb ist der Speicherbedarf im allgemeinen n^2. Für $n = 10^6$ wären das etwa 7450 Gigabyte (mit 8 Byte double Zahlen).

Für große Systeme nutzt man iterative Verfahren. Dabei kann die bekannte Idee der Fixpunktiteration verwendet werden. Dazu wird die Matrix $A \in \mathbb{R}^{n\times n}$ in die Form

$$A = M - N, \quad M, N \in \mathbb{R}^{n\times n}$$

zerlegt, wobei M invertierbar sein soll. Aus (2.1) erhält man dann die äquivalente Fixpunktgleichung

$$M\mathbf{x} = \mathbf{b} + N\mathbf{x} \quad \Longleftrightarrow \quad \mathbf{x} = M^{-1}\left(\mathbf{b} + N\mathbf{x}\right).$$

Ist eine Startnäherung $\mathbf{x}^{(0)} \in \mathbb{R}^n$ gegeben, so lautet die Fixpunktiteration zur Lösung von $A\mathbf{x} = \mathbf{b}$:

$$\mathbf{x}^{(k+1)} = M^{-1}\left(\mathbf{b} + N\mathbf{x}^{(k)}\right), \quad k = 0, 1, 2, \ldots \tag{2.5}$$

Über die Konvergenz dieser Iteration gibt der Banachsche Fixpunktsatz, siehe später Satz 3.4, Auskunft. Für den Spezialfall (2.5) erhält man:

Satz 2.5 *Das iterative Verfahren (2.5) konvergiert genau dann für alle $\mathbf{x}^{(0)} \in \mathbb{R}^n$ gegen die Lösung von $A\mathbf{x} = \mathbf{b}$, wenn für die zugehörige Iterationsmatrix $M^{-1}N$ gilt*

$$\max\left\{|\lambda| \ : \ \lambda \text{ ist Eigenwert von } M^{-1}N\right\} < 1.$$

In jedem Iterationsschritt von (2.5) muss man ein lineares Gleichungssystem der Gestalt

$$M\mathbf{y} = \mathbf{b} + N\mathbf{x}^{(k)}$$

lösen. Das soll natürlich einfach gehen. Im einfachsten Fall wählt man

$$M = \omega^{-1}D = \omega^{-1}\mathrm{diag}(A),$$

wobei $\mathrm{diag}(A)$ die Diagonalmatrix mit den Diagonalelementen von A ist und $\omega > 0$ ein Parameter (Dämpfungsfaktor). Man muss bei diesem Fall natürlich voraussetzen, dass alle Diagonalenelemente von A ungleich Null sind. Es folgt

$$N = M - A = \omega^{-1}D - A$$

und man erhält die Iteration

$$\begin{aligned}\mathbf{x}^{(k+1)} &= M^{-1}\left(\mathbf{b} + N\mathbf{x}^{(k)}\right) = \omega D^{-1}\left(\mathbf{b} + \omega^{-1}D\mathbf{x}^{(k)} - A\mathbf{x}^{(k)}\right)\\ &= \mathbf{x}^{(k)} + \omega D^{-1}\left(\mathbf{b} - A\mathbf{x}^{(k)}\right).\end{aligned}$$

Dieses Verfahren wird für $\omega = 1$ Jacobi–Verfahren und für $\omega \in (0, 1)$ gedämpftes Jacobi–Verfahren genannt.

Eine andere Möglichkeit besteht darin, M als Dreiecksmatrix zu wählen. Zerlege

$$A = L + D + U,$$

wobei D die Diagonalmatrix mit den Diagonalelementen von A ist, L das strikte untere Dreieck von A und U das strikte obere Dreieck. Man muss wieder voraussetzen, dass alle Diagonalelemente von A ungleich Null sind. Wählt man

$$M = L + \omega^{-1} D,$$

so ergibt sich folgendes Iterationsverfahren

$$\begin{aligned}
\left(L + \omega^{-1} D\right) \mathbf{x}^{(k+1)} &= \mathbf{b} + \left(L + \omega^{-1} D\right) \mathbf{x}^{(k)} - (L + D + U)\, \mathbf{x}^{(k)} \quad \Longleftrightarrow \\
\omega^{-1} D \mathbf{x}^{(k+1)} &= \mathbf{b} + \omega^{-1} D \mathbf{x}^{(k)} - (D + U)\, \mathbf{x}^{(k)} - L \mathbf{x}^{(k+1)} \quad \Longleftrightarrow \\
\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \omega D^{-1} \left(\mathbf{b} - L \mathbf{x}^{(k+1)} - (D + U)\, \mathbf{x}^{(k)} \right). \qquad (2.6)
\end{aligned}$$

Schreibt man dieses Verfahren in Komponentenschreibweise, so sieht man, dass man die rechte Seite berechnen kann, obwohl sich darin die neue Iterierte $\mathbf{x}^{(k+1)}$ befindet:

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij} x_j^{(k)} \right), \quad i = 1, \ldots, n.$$

Man nutzt die neu berechneten Komponenten von $\mathbf{x}^{(k+1)}$ sofort zur Berechnung weiterer Komponenten. Dieses Verfahren heißt für $\omega = 1$ Gauß–Seidel–Verfahren und für $\omega > 0, \omega \neq 1$, spricht man vom SOR–Verfahren (successive overrelaxation, aufeinander folgende Entspannung). (*MATLAB–Demo*)

Es gibt viele Untersuchungen zur Konvergenz dieser Iterationsverfahren. Ein bemerkenswertes Ergebnis ist das folgende:

Satz 2.6 *Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Dann konvergiert das SOR–Verfahren für alle $\mathbf{x}^{(0)} \in \mathbb{R}^n$ genau dann, wenn $\omega \in (0, 2)$.*

Bemerkung 2.7

1. Der Satz gibt keine Aussage über die Konvergenzgeschwindigkeit und über die Existenz oder Wahl eines optimalen Parameters ω. Es gibt Fälle, in denen man zeigen kann, dass ein optimaler Parameter ω existiert. Die Berechnung dieses Parameters erfordert jedoch im allgemeinen Informationen über die Matrix A, deren Beschaffung mit großem Aufwand verbunden ist.
2. Es zeigt sich, dass die Lösung großer Gleichungssysteme mit den hier vorgestellten Iterationsverfahren zwar im Prinzip geht, jedoch im allgemeinen sehr ineffizient ist (sehr viele Iterationen, sehr lange Rechenzeiten). Man hat inzwischen wesentlich bessere Iterationsverfahren zur Lösung linearer Systeme konstruiert. Die hier vorgestellten einfachen Iterationsverfahren können jedoch eine wichtige Teilkomponente komplizierterer Iterationsverfahren sein.

□

Kapitel 3

Nullstellenberechnung von nichtlinearen Funktionen

In dieser Vorlesung wird nur die Nullstellenberechnung reeller Funktionen einer reellen Variablen $f : \mathbb{R} \to \mathbb{R}$ betrachtet. Man nennt die Nullstellen von nichtlinearen Funktionen auch Wurzeln.

In der Praxis sind Nullstellenberechnungen mehrdimensionaler Funktionen von großer Bedeutung. Die Behandlung von Verfahren für solche Funktionen ist jedoch aus Zeitgründen nicht möglich.

3.1 Theorie

Man unterscheidet zwei Normalformen von nichtlinearen Gleichungen, die Nullstellengleichung

$$f_1(x) = 0 \tag{3.1}$$

und die Fixpunktgleichung

$$f_2(x) = x. \tag{3.2}$$

Eine Zahl $\alpha \in \mathbb{R}$ wird Fixpunkt von f_2 genannt, falls $f_2(\alpha) = \alpha$.

Jede Nullstellengleichung lässt äquivalent sich in eine Fixpunktgleichung überführen. Sei $g(x)$ eine stetige Funktionen mit $g(x) \neq 0$, dann wird (3.1) mit $g(x)$ multipliziert und auf beiden Seiten wird x addiert:

$$F_1(x) := g(x) f_1(x) + x = x.$$

Eine Nullstelle von f_1 ist ein Fixpunkt von F_1 und umgekehrt. Subtrahiert man in (3.2) x, so erhält man die Nullstellengleichung

$$F_2(x) := f_2(x) - x = 0.$$

Das heißt, ein Fixpunkt von f_2 ist eine Nullstelle von F_2 und umgekehrt.

Bemerkung 3.1 Kondition des Wurzelfindens. Sei $f_1 \in C^1(a, b)$, betrachte

$$f_1(x) - d = 0, \quad d \in \mathbb{R}.$$

Man kann zeigen, dass das Problem des Findens der Wurzel nur dann wohldefiniert ist, wenn f_1 in einer Umgebung von d invertierbar ist. Sei $\alpha = f_1^{-1}(d)$, f_1^{-1} – inverse Funktion (Umkehrfunktion). Mit der Differentiationsregel für die inverse Funktion erhält man

$$\left(f_1^{-1}\right)'(d) = \frac{1}{f_1'(\alpha)}.$$

Dann ist das Wurzelfinden einer nichtlinearen Gleichung gut konditioniert, falls die Wurzel α einfach ist, das heißt $f_1'(\alpha) \neq 0$, und $f_1'(\alpha)$ hinreichend weit von Null entfernt ist. Ansonsten ist das Problem schlecht konditioniert. □

Numerische Verfahren zur Lösung nichtlinearer Gleichungen sind im allgemeinen iterativ. Beginnend mit einem Startwert $x^{(0)}$ wird vom Verfahren eine Folge $\{x^{(k)}\}_{k\geq 0}$ generiert, mit dem Ziel, dass

$$\lim_{k\to\infty} x^{(k)} = \alpha.$$

Man möchte nun ein Maß für die Geschwindigkeit der Konvergenz haben, womit man dann die Effizienz von Verfahren beschreiben kann.

Definition 3.2 Eine Folge $\{x^{(k)}\}$ konvergiert gegen α mit Ordnung $p \geq 1$, falls eine Konstante $C > 0$ existiert, so dass gilt

$$\frac{\left|x^{(k+1)} - \alpha\right|}{\left|x^{(k)} - \alpha\right|^p} \leq C \quad \forall\ k \geq k_0,$$

wobei $k_0 \in \mathbb{N}$ ein geeigneter Index ist. Man sagt dann, dass die Methode von p–ter Ordnung ist.

Ist $p = 1$, so muss notwendigerweise $C < 1$ sein, damit $\{x^{(k)}\}$ gegen α konvergiert. In diesem Fall wird C Konvergenzfaktor der Methode genannt. □

Ob ein Iterationsverfahren überhaupt konvergiert, kann auch vom Startwert $x^{(0)}$ abhängen.

Definition 3.3 Sei I die Menge aller zulässigen Startwerte. Gibt es eine Umgebung $U(\alpha)$ der Wurzel α (offene Menge, die α enthält), so dass ein Iterationsverfahren für alle $x^{(0)} \in U(\alpha)$ konvergiert, so nennt man das Verfahren lokal konvergent. Konvergiert das Verfahren für alle $x^{(0)} \in I$, dann wird es global konvergent genannt. □

Zur Lösung der Fixpunktgleichung (3.2) kann man das Iterationsverfahren

$$x^{(k+1)} = f_2\left(x^{(k)}\right), \quad k = 0, 1, \ldots \tag{3.3}$$

verwenden. *MATLAB-Demo*

Zur Existenz eines Fixpunktes und zur Konvergenz der Iteration (3.3) gibt es folgenden berühmten Satz.

Satz 3.4 Fixpunktsatz von Banach (1922). *Die Funktion f_2 sei im abgeschlossenen Intervall $I = [a, b] \subset \mathbb{R}$ kontraktiv, dass heißt:*

- *I wird durch f_2 in sich abgebildet, $f_2(I) \subseteq I$,*
- *f_2 ist Lipschitz–stetig mit einer Lipschitz–Konstanten $L \in [0, 1)$:*

$$|f_2(x) - f_2(y)| \leq L\,|x - y| \quad \forall\ x, y \in I.$$

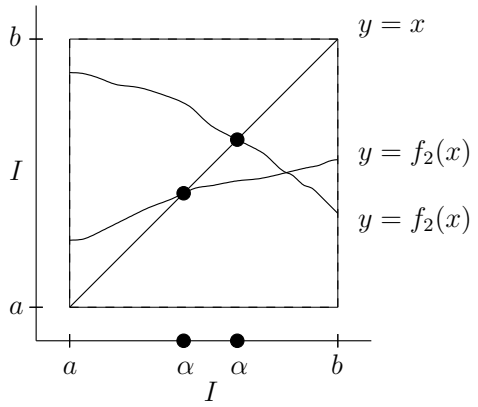
Dann besitzt f_2 in I genau einen Fixpunkt α und die nach der Iterationsvorschrift (3.3) berechnete Folge konvergiert gegen α für jeden Startwert $x^{(0)} \in I$. Es gelten die Fehlerabschätzungen

$$\begin{aligned}
\left|\alpha - x^{(k)}\right| &\leq \frac{L^k}{1-L}\left|x^{(1)} - x^{(0)}\right|, \\
\left|\alpha - x^{(k)}\right| &\leq \frac{L}{1-L}\left|x^{(k)} - x^{(k-1)}\right|.
\end{aligned}$$

Beweis: HMI I, Literatur. ■

Bemerkung 3.5

1. Die Aussagen des Banachschen Fixpunktsatzes gelten allgemein in Banach–Räumen, also insbesondere auch für Funktionen $f : \mathbb{R}^n \to \mathbb{R}^n$, $n \in \mathbb{N}$.
2. Ist f_2 in I stetig differenzierbar bis zum Rand, $f_2 \in C^1([a,b])$, so ist f_2 Lipschitz–stetig und es gilt
$$L = \max_{x \in [a,b]} |f_2'(x)|.$$
3. Geometrische Deutung des Banachschen Fixpunktsatzes.



Sei f_2 differenzierbar in $[a,b]$ und sei $L < 1$. Der Betrag des Anstiegs von f_2 ist kleiner als 1, siehe Bemerkung 2. Falls f_2 die Gerade $y = x$ einmal geschnitten hat, kann es keinen zweiten Schnittpunkt geben, weil f_2 dann schneller ansteigen müsste als $y = x$. Das geht aber nicht, da $y' = (x)' = 1$. Dass überhaupt ein Schnittpunkt von f_2 und $y = x$ existiert, folgt aus
   - f_2 bildet I in I ab,
   - f_2 ist stetig,
   - I ist abgeschlossen.

   Man kann leicht Beispiele dafür konstruieren, dass der Banachsche Fixpunktsatz nicht mehr gilt, wenn auch nur eine dieser drei Voraussetzungen nicht erfüllt ist.
4. Sei $L > 0$. Aus (3.3) und der Taylor–Entwicklung von f_2 in α folgt
$$\begin{aligned} x^{(k+1)} - \alpha &= f_2\left(x^{(k)}\right) - f_2(\alpha) = f_2(\alpha) + f_2'(\xi)\left(x^{(k)} - \alpha\right) - f_2(\alpha) \\ &= f_2'(\xi)\left(x^{(k)} - \alpha\right) \end{aligned}$$
   mit $\xi \in [a,b]$. Mit Bemerkung 2 hat man
$$\frac{\left|x^{(k+1)} - \alpha\right|}{\left|x^{(k)} - \alpha\right|} = |f_2'(\xi)| \leq L.$$
   Die Konvergenzordnung ist also linear und der Konvergenzfaktor ist L. Die Konvergenzgeschwindigkeit ist desto besser, je kleiner L ist.

□

Numerische Verfahren, die auf dem Prinzip der Fixpunktiteration beruhen, werden in den Abschnitten 3.3 und 3.4 behandelt.

3.2 Das Bisektions–Verfahren

Bei diesem Verfahren handelt es sich um das einfachste und sicherste Verfahren zur Berechnung einer Nullstelle einer stetigen Funktion im Intervall $I = [a, b]$. Dieses Verfahren lässt sich jedoch nicht auf höhere Dimensionen erweitern. Es beruht auf dem Zwischenwertsatz: Jede stetige Funktion $f : [a, b] \to \mathbb{R}$ nimmt jeden Funktionswert zwischen $f(a)$ und $f(b)$ an (falls $f(a) \leq f(b)$, ansonsten umgekehrt). Damit folgt aus $f(a) f(b) < 0$, dass f mindestens eine Nullstelle in $[a, b]$ besitzt.

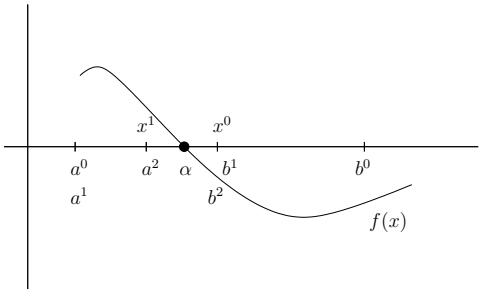
Beim Bisektions–Verfahren versucht man, ein genügend kleines Intervall zu finden, in dem eine Nullstelle von f liegt. Man muss zunächst Werte a und b finden, so dass $f(a)$ und $f(b)$ unterschiedliches Vorzeichen besitzen. Setze dann $I_0 := [a, b]$. Dann werden Teilintervalle $I_k := [a^{(k)}, b^{(k)}]$, $k \geq 0$, $I_k \subset I_{k-1}$, $k \geq 1$, gesucht, so dass $f\left(a^{(k)}\right) f\left(b^{(k)}\right) < 0$. Das geschieht mit dem folgenden Algorithmus:

Algorithm 3.6 Bisektions–Verfahren.

- Setze $a^{(0)} := a$, $b^{(0)} := b$, $x^{(0)} := \dfrac{a^{(0)} + b^{(0)}}{2}$.
- Iteration: falls $k \geq 0$:
  - falls $f\left(x^{(k)}\right) f\left(a^{(k)}\right) < 0$, setze $a^{(k+1)} := a^{(k)}, b^{(k+1)} := x^{(k)}$,
  - falls $f\left(x^{(k)}\right) f\left(b^{(k)}\right) < 0$, setze $a^{(k+1)} := x^{(k)}, b^{(k+1)} := b^{(k)}$,

  anschließend setze $x^{(k+1)} := \dfrac{a^{(k+1)} + b^{(k+1)}}{2}$.
- Abbruch: Man bricht das Bisektions–Verfahren ab, falls
  - $f\left(x^{(k+1)}\right) = 0$, das heißt, man hat eine Nullstelle gefunden,
  - oder $\left|b^{(k+1)} - a^{(k+1)}\right| = |I_{k+1}| < \varepsilon$, wobei $\varepsilon > 0$ eine vorgegebene Toleranz ist, das heißt das Restintervall ist hinreichend klein. Nehme als Approximation an die Nullstelle den Mittelpunkt $x^{(k+1)}$ von I_{k+1}.

MATLAB–Demo □



Nun wird die Konvergenzgeschwindigkeit des Bisektions–Verfahrens betrachtet. Es sind

$$|I_0| = b - a, \quad |I_k| = \frac{|I_0|}{2^k} = \frac{b-a}{2^k}, \ k \geq 0.$$

Sowohl die k–te Iterierte $x^{(k)}$ als auch die Lösung α liegen im Intervall I_k. Da $x^{(k)}$ der Mittelpunkt von I_k ist, gilt für den absoluten Fehler

$$\left|e^{(k)}\right| = \left|x^{(k)} - \alpha\right| \leq \frac{|I_k|}{2} = \frac{b-a}{2^{k+1}} \quad \Longrightarrow \quad \lim_{k\to\infty} \left|e^{(k)}\right| = 0.$$

Das gilt für beliebige Startwerte $a^{(0)}$, $b^{(0)}$ mit $f\left(a^{(0)}\right) f\left(b^{(0)}\right) < 0$. Das Bisektions–Verfahren ist also global konvergent.

Sei $\varepsilon > 0$ gegeben. Gesucht ist die Anzahl von Bisektions–Schritten, damit $\left|x^{(m)} - \alpha\right| < \varepsilon$. a $x^{(k)}$ der Mittelpunkt von I_k ist, ist das sicher erfüllt falls

$$|I_m| < 2\varepsilon \quad \Longleftrightarrow \quad \frac{b-a}{2^{m+1}} < \varepsilon \quad \Longleftrightarrow \quad m > \frac{\log\left((b-a)/\varepsilon\right)}{\log(2)} - 1.$$

Man kann die Formel so interpretieren, als dass jede Bisektion die Genauigkeit um eine Dualstelle verbessert. Daraus folgt, dass man zur Verbesserung um eine Dezimalstelle $\log_2(10) \approx 3.32$ Bisektionen benötigt. Insgesamt ist die Konvergenz der Bisektion zwar sicher, aber langsam.

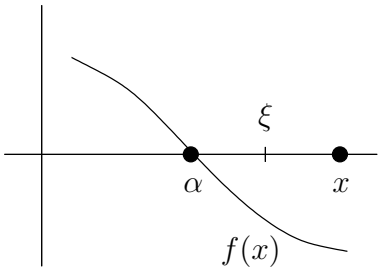
Die Konvergenz braucht nicht monoton zu sein, siehe Beispiele in der Literatur. Aus diesem Grunde passt das Bisektions–Verfahren nicht in den Rahmen der Definition 3.2 über die Konvergenzordnung. Man kann also nicht von einer Methode erster Ordnung im Sinne dieser Definition sprechen.

Die Bisektion eignet sich vor allem als Annäherungsverfahren an die Nullstelle, um einen Startwert für schnellere Iterationsverfahren zu finden, die nur lokal konvergent sind.

3.3 Das Sekanten–Verfahren und Varianten

Sei $\alpha \in I = [a,b]$ eine Nullstelle von $f \in C^1(I)$. Sei x nahe genug an α, o.B.d.A. mit $x > \alpha$. Dann erhält man mit Taylor–Entwicklung

$$f(x) = f(\alpha) + (x-\alpha) f'(\xi), \quad \xi \in (\alpha, x).$$



Diese Beziehung kann man in die Fixpunktgleichung

$$\alpha = x - \frac{f(x) - f(\alpha)}{f'(\xi)} = x - \frac{f(x)}{f'(\xi)} \tag{3.4}$$

umformen, falls $f'(\xi) \neq 0$. Diese Eigenschaft ist zum Beispiel in einer Umgebung von α gesichert, falls α eine einfache Nullstelle von f ist. Die Verfahren, die in diesem Abschnitt vorgestellt werden, approximieren $f'(\xi)$ in (3.4) unter Benutzung von Werten von f.

Die Grundidee der Approximation einer Ableitung besteht in der Verwendung einer finiten Differenz:

$$f'(x) \approx \frac{f(x_1) - f(x_2)}{x_1 - x_2},$$

wobei x_1, x_2 sich in einer Umgebung von x befinden. Sind x_1 oder x_2 relativ weit voneinanger oder von x entfernt, wird die Approximation relativ ungenau sein. Liegen andererseits x_1, x_2 sehr dicht beieinander, kann es sowohl im Zähler als auch im Nenner der finiten Differenz Auslöschung geben, was wiederum zu ungenauen Ergebnissen führen kann.

Beim Sekanten–Verfahren verwendet man

$$f'(\xi) \approx \frac{f\left(x^{(k)}\right) - f\left(x^{(k-1)}\right)}{x^{(k)} - x^{(k-1)}}, \quad k \geq 0,$$

wobei $x^{(k-1)}, x^{(k)}$ die beiden letzen Iterierten sind, siehe Abbildung 3.1.

Man benötigt also zunächst zwei Startwerte $x^{(-1)}, x^{(0)}$. Sind diese gegeben, so lautet das Sekanten–Verfahren

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f\left(x^{(k)}\right) - f\left(x^{(k-1)}\right)} f\left(x^{(k)}\right), \quad k \geq 0. \tag{3.5}$$

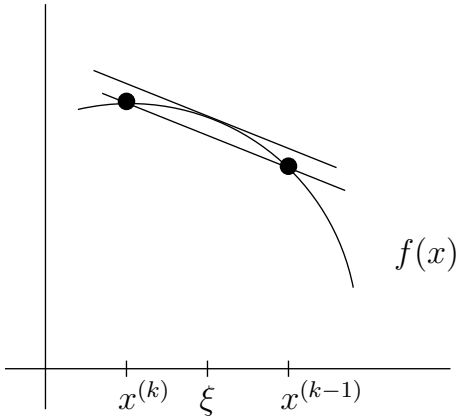


Abbildung 3.1: Approximation einer Tangente durch eine Sekante.

MATLAB–Demo

Zur die Konvergenz dieses Verfahrens gibt es folgende Aussage:

Satz 3.7 *Seien $U(\alpha)$ eine geeignete Umgebung von α, $f \in C^2(U(\alpha))$, $f'(\alpha) \neq 0$ und $f''(\alpha) \neq 0$. Falls die Startwerte $x^{(-1)}, x^{(0)} \in U(\alpha)$ hinreichend nahe an α gewählt werden, konvergiert die mit (3.5) berechnete Folge $\{x^{(k)}\}$ gegen α und die Konvergenzordnung ist $p = \left(1+\sqrt{5}\right)/2 \approx 1.62$.*

Beweis: Literatur, zum Beispiel [SK06]. ■

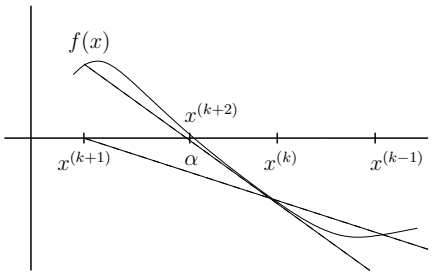


Abbildung 3.2: Graphische Illustration zum Sekantenverfahren.

Bemerkung 3.8

1. In der Praxis bricht man das Verfahren im allgemeinen ab, falls $\left|f\left(x^{(m)}\right)\right| < \varepsilon$ oder $\left|x^{(m)} - x^{(m-1)}\right| < \varepsilon$ für eine vorgegebene Toleranz $\varepsilon > 0$.
2. Das Sekanten–Verfahren ist lokal konvergent, aber im allgemeinen nicht global.
3. Man benötigt neben der Berechnung von $f\left(x^{(k)}\right)$ im k–ten Iterationsschritt keine zusätzliche Funktionswertberechnung, da man $f\left(x^{(k-1)}\right)$ noch aus dem vorangegangenen Iterationsschritt kennt.
4. $x^{(k+1)}$ liegt nicht unbedingt im Intervall $[x^{(k)}, x^{(k-1)}]$, siehe Abbildung 3.2. Das geschieht, wenn $f\left(x^{(k)}\right)$ und $f\left(x^{(k-1)}\right)$ gleiches Vorzeichen besitzen.
5. Verallgemeinerungen auf komplexe Nullstellen und Funktionen in Banach–Räumen sind möglich.
6. Eine weitere Verallgemeinerung besteht darin, statt einer Sekante eine Parabel durch drei Punkte zu nehmen – Verfahren von Muller, Konvergenzordnung 1.84.

□

Falls $f\left(x^{(k)}\right)$ und $f\left(x^{(k-1)}\right)$ gleiches Vorzeichen besitzen und fast gleichgroß sind, kann $x^{(k+1)}$ sehr weit entfernt von α liegen. Es kann sein, dass f für $x^{(k+1)}$ gar nicht definiert ist oder sehr große Werte annimmt, die zu einem Überlauf führen.

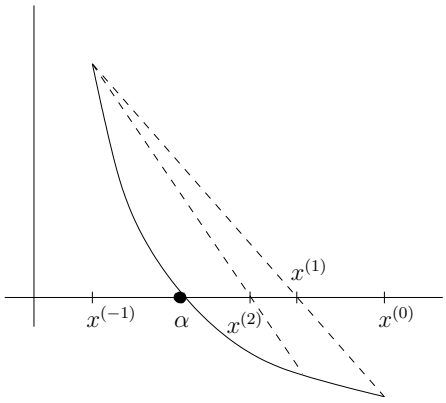
Ein Verfahren, bei denen alle Iterierten $x^{(k)}$ im Intervall $[x^{(-1)}, x^{(0)}]$ liegen, und welches ebenfalls Sekanten benutzt, ist die Regula falsi (Regel des falschen Ansatzes). Anstatt die Sekante immer durch die letzten beiden Iterierten zu nehmen, verwendet man die Sekante durch $\left(x^{(k)}, f\left(x^{(k)}\right)\right)$ und $\left(x^{(k')}, f\left(x^{(k')}\right)\right)$, wobei k' der größte Index kleiner als k ist, für welchen $f\left(x^{(k)}\right) f\left(x^{(k')}\right) < 0$ gilt. Für die Regula falsi benötigt man zwei Startwerte $x^{(-1)}, x^{(0)}$ mit $f\left(x^{(0)}\right) f\left(x^{(-1)}\right) < 0$. Die Iterationsvorschrift lautet

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k')}}{f\left(x^{(k)}\right) - f\left(x^{(k')}\right)} f\left(x^{(k)}\right), \quad k \geq 0. \tag{3.6}$$

MATLAB–Demo

Satz 3.9 *Sei $f \in C\left([x^{(-1)}, x^{(0)}]\right)$. Dann gilt für die mit (3.6) erzeugte Folge $\{x^{(k)}\}$, dass $x^{(k)} \in [x^{(-1)}, x^{(0)}]$, $k \geq 1$ und $\lim_{k\to\infty} x^{(k)} = \alpha$. Sind $f \in C^2\left([x^{(-1)}, x^{(0)}]\right)$, α eine einfache Nullstelle ($f'(\alpha) \neq 0$) und $f''(\alpha) \neq 0$, dann ist die Konvergenzordnung der Regula falsi gleich Eins.*

Beweis: Literatur, z.B. [SK06]. ■



In diesem Fall ist $k' = 1$ für alle k. Das ist der schlechteste Fall, der dafür sorgt, dass im allgemeinen die Konvergenzordnung nicht größer als Eins ist.

Bemerkung 3.10

1. Die Regula falsi ist global konvergent. Da sie im allgemeinen jedoch langsam ist, eignet sie sich for allem zur Beschaffung von Startwerten für lokal konvergente, schnellere Verfahren.
2. Gemessen an Funktionswertberechnungen, ist der Aufwand von Regula falsi und Sekanten–Verfahren gleich.
3. Es gibt Varianten der Regula falsi, die eine bessere Konvergenzordnung besitzen, indem sie vermeiden, dass k' fixiert wird: Illinois–Algorithmus $p = \sqrt[3]{3} \approx 1.442$; Pegasus–Verfahren, $p \approx 1.642$.
4. Des weiteren gibt es Verfahren, die Bisektion (Sicherheit) und Sekanten–Verfahren (Schnelligkeit) kombinieren, zum Beispiel das Verfahren von Dekker und Brent (1973), siehe [QSS04].

□

3.4 Das Newton–Verfahren

- Newton 1669,
- veröffentlicht von Wallis 1685,
- in der heutigen Form von Raphson 1697, deshalb auch manchmal Newton–Raphson–Verfahren.

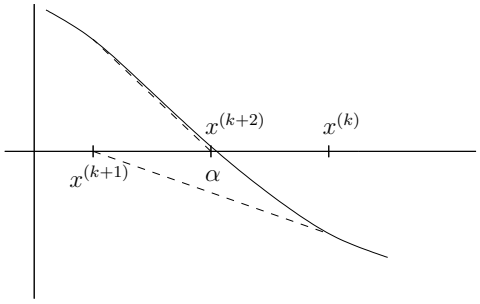
Sei $f : [a,b] \to \mathbb{R}$ stetig differenzierbar. Das Newton–Verfahren verwendet zur Approximation von $f'(\xi)$ in (3.4) den Wert der Ableitung von f an der Stelle $x^{(k)}$. Sei also $x^{(0)}$ gegeben, dann lautet das Newton–Verfahren

$$x^{(k+1)} = x^{(k)} - \frac{f\left(x^{(k)}\right)}{f'\left(x^{(k)}\right)}, \quad k \geq 0. \tag{3.7}$$

MATLAB–Demo

Satz 3.11 *Seien $f \in C^3([a,b])$, $\alpha \in [a,b]$ und $f'(\alpha) \neq 0$ (einfache Nullstelle). Dann existiert eine Umgebung $U(\alpha)$, so dass für jeden Startwert $x^{(0)} \in U(\alpha)$ die mit (3.7) berechnete Folge $\{x^{(k)}\}$ gegen α konvergiert. Die Konvergenzordnung ist mindestens 2.*

Beweis: Literatur, zum Beispiel [SK06]. ■



In $\left(x^{(k)}, f\left(x^{(k)}\right)\right)$ wird die Tangente angelegt. Der Schnittpunkt der Tangente mit der x–Achse ist $x^{(k+1)}$.

Bemerkung 3.12

1. Man bricht die Iteration ab, falls $\left|x^{(m)} - x^{(m-1)}\right| < \varepsilon$ oder $\left|f\left(x^{(m)}\right)\right| < \varepsilon$ für eine vorgegebene Toleranz $\varepsilon > 0$.
2. Das Newton–Verfahren ist lokal konvergent, im allgemeinen jedoch nicht global.
3. Bei einer l–fachen Nullstelle, $l \geq 2$, konvergiert das Newton–Verfahren (3.7) linear mit dem Konvergenzfaktor $C = (l-1)/l$. Ist die Vielfachheit der Nullstelle bekannt, so konvergiert das modifizierte Newton–Verfahren

$$x^{(k+1)} = x^{(k)} - l\frac{f\left(x^{(k)}\right)}{f'\left(x^{(k)}\right)}, \quad k \geq 0,$$

   quadratisch.
4. Das Newton–Verfahren benötigt zwei Funktionsaufrufe pro Iteration, $f\left(x^{(k)}\right)$ und $f'\left(x^{(k)}\right)$. Diesbezüglich ist es doppelt so teuer wie das Sekanten–Verfahren. Zwei Schritte des Sekanten–Verfahrens haben die Konvergenzordnung $\left(1+\sqrt{5}\right)^2/4 \approx 2.62$. Damit konvergiert, gemessen am Aufwand, das Sekanten–Verfahren schneller.

5. *Optimale Fixpunktiteration.* Die Nullstellengleichung $f(x) = 0$ lässt sich äquivalent in die Fixpunktgleichung

$$x = x + g(x) f(x), \quad g(x) \neq 0,$$

umformen. Der Banachsche Fixpunktsatz 3.4 besagt, dass die Fixpunktiteration

$$x^{(k+1)} = x^{(k)} + g\left(x^{(k)}\right) f\left(x^{(k)}\right)$$

desto schneller konvergiert, je kleiner die Lipschitz–Konstante der rechten Seite der Fixpunktgleichung ist. Seien $f(x), g(x)$ in $U(\alpha)$ stetig differenzierbar, dann folgt

$$L = \max_{x \in U(\alpha)} \left|(x + g(x) f(x))'\right| = \max_{x \in U(\alpha)} \left|1 + g'(x) f(x) + g(x) f'(x)\right|.$$

Man kann hoffen, dass L in $\overline{U(\alpha)}$ klein ist, wenn für $x = \alpha$ der Ausdruck im Betrag verschwindet. Da $f(\alpha) = 0$, folgt

$$1 + g(\alpha) f'(\alpha) = 0 \quad \Longrightarrow \quad g(\alpha) = -\frac{1}{f'(\alpha)}.$$

Eine Funktion, die diese Bedingung erfüllt, ist $g(x) = -1/f'(x)$, womit man das Newton–Verfahren erhält.
Man erhält also das Newton–Verfahren, indem man die Funktion $g(x)$ in der Fixpunktgleichung so wählt, dass man auf eine möglichst kleine Lipschitz–Konstante in $\overline{U(\alpha)}$ hoffen kann.

6. Ist $f \in C^2\left(\overline{U(\alpha)}\right)$, dann erhält man aus dem Banachschen Fixpunktsatz folgende hinreichende Bedingung für die Konvergenz des Newton–Verfahrens:

$$\begin{aligned} 1 \quad &> \quad L = \max_{x \in \overline{U(\alpha)}} \left|\left(x - \frac{f(x)}{f'(x)}\right)'\right| = \max_{x \in \overline{U(\alpha)}} \left|1 - \frac{(f'(x))^2 - f(x) f''(x)}{(f'(x))^2}\right| \\ &= \quad \max_{x \in \overline{U(\alpha)}} \left|\frac{f(x) f''(x)}{(f'(x))^2}\right|. \end{aligned}$$

7. Die Übertragung des Newton–Verfahrens auf komplexe Nullstellen und auf Banach–Räume ist möglich.

□

3.5 Nullstellen von Polynomen

Dieser Abschnitt stellt ein Verfahren vor, mit dem man alle Nullstellen eines Polynoms

$$p_n(x) = \sum_{i=0}^{n} a_i x^i = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0, \tag{3.8}$$

$a_i \in \mathbb{R}$, $i = 0, \ldots, n$, berechnen kann.

Der Fundamentalsatz der Algebra besagt, dass $p_n(x)$ genau n Nullstellen in $\mathbb{C}$ besitzt, wobei mehrfache Nullstellen ihrer Vielfachheit gemäß gezählt werden. Des weiteren ist bekannt:

- $n \in \{1, 2\}$ – es gibt einfache Berechnungsvorschriften für die Nullstellen,
- $n \in \{3, 4\}$ – es gibt relativ komplizierte Berechnungsvorschriften für die Nullstellen,

- $n \geq 5$ – es gibt im allgemeinen keine expliziten Berechnungsvorschriften für die Nullstellen.

Das heißt, für Polynome mit $n \geq 3$ ist die Nutzung numerischer Verfahren zu empfehlen, für $n \geq 5$ ist sie im allgemeinen notwendig.

Bevor wir zur Nullstellenberechnung kommen, wollen wir erst einmal ein Schema kennenlernen, mit dessen Hilfe man Funktionswerte eines Polynoms schnell berechnen kann. Nutzt man den naivsten Weg, jeden Summanden von (3.8) einzeln zu berechnen, so hat man für den i–ten Summanden i Multiplikationen und zum Schluss n Additionen. Damit ist die Anzahl der Flops

$$\sum_{i=0}^{n} i + n = \frac{n}{2}(n+1) + n = \frac{n^2}{2} + \frac{3}{2}n.$$

Günstiger ist bereits, wenn man die Potenzen von x zwischendurch speichert:

$p := a_0 + a_1 x$
$t := x$
`for` $i = 2 : n$
  $t := tx$
  $p := p + a_i t$
`end`

Dieser Weg benötigt n Additionen und $(2n-1)$ Multiplikationen, also $3n-1$ Flops.

Die Darstellung (3.8) ist nicht die einzig mögliche für ein Polynom. Äquivalent dazu ist

$$p_n(x) = a_0 + x(a_1 + x(a_2 + x(\ldots + x(a_{n-1} + a_n x)\ldots))). \tag{3.9}$$

Nutzt man (3.9), so kann man $p_n(x)$ mit n Additionen und n Multiplikationen, also $2n$ Flops berechnen. Man nennt (3.9) auch eingebettete Multiplikation und (3.9) ist die Basis des sogenannten Horner–Schemas (synthetische Division) zur Berechnung von $p_n(z)$:

$b_n := a_n$
`for` $i = n-1 : -1 : 0$
  $b_i := a_i + b_{i+1} z$
`end`

Beispiel 3.13 $n = 3$:

$$p_3(z) = a_0 + z\underbrace{\left(\underbrace{a_1 + z\underbrace{\left(a_2 + \underbrace{a_3}_{b_3} z\right)}_{b_2}}_{b_1}\right)}_{b_0}.$$

□

Dieses Verfahren, veröffentlich von Horner (1818), findet man schon bei Newton über 100 Jahre früher. Damals hat man das Verfahren handschriftlich durchgeführt und dabei folgendes Schema entwickelt:

p_n	a_n	a_{n-1}	a_{n-2}	$\ldots$	a_1	a_0
z	–	$b_n z$	$b_{n-1} z$	$\ldots$	$b_2 z$	$b_1 z$
	b_n	b_{n-1}	b_{n-2}	$\ldots$	b_1	b_0

Das ist das sogenannte Horner–Schema.

Beispiel 3.14 Berechne $p_3(z) = -2z^3 + 20z^2 - 2z - 13$ an der Stelle $z = 3$:

p_3	-2	20	-2	-13
3		-6	42	120
	-2	14	40	107

Es ist $p_3(3) = 107$. □

Man kann das Polynom $p_n(x)$ eindeutig zerlegen in

$$p_n(x) = p_0 + (x - z)\, p_{n-1}(x),$$

wobei p_0 eine Konstante ist und $p_{n-1}(x)$ ein Polynom vom Grad $n-1$ (Polynomdivision). Setzt man $x = z$, so folgt $p_n(z) = b_0 = p_0$, also folgt

$$p_{n-1}(x) = \frac{p_n(x) - b_0}{x - z}.$$

Man findet durch Nachrechnen *Übungsaufgabe*

$$p_{n-1}(x) = b_1 + b_2 x + \ldots + b_n x^{n-1} = \sum_{i=1}^{n} b_i x^{i-1} =: q_{n-1}(x).$$

Einsetzen führt zu der Darstellung

$$p_n(x) = b_0 + (x - z)\, q_{n-1}(x). \tag{3.10}$$

Ist z eine Nullstelle von $p_n(x)$, dann ist $b_0 = 0$ und man hat

$$p_n(x) = (x - z)\, q_{n-1}(x).$$

Die restlichen $(n-1)$ Nullstellen von $p_n(x)$ sind nun gerade die Nullstellen von $q_{n-1}(x)$. Man hat die Nullstelle z abgespalten. Diesen Vorgang nennt man Deflation. Auf dieser Basis kann man folgende allgemeine Strategie zur Berechnung aller Nullstellen von $p_n(x)$ angeben:

1. finde eine Nullstelle z von $p_n(x)$ mit einem geeigneten Verfahren,
2. berechne die Koeffizienten von $q_{n-1}(x)$ mit dem Horner–Schema,
3. setze $p_{n-1}(x) := q_{n-1}(x)$, $n := n - 1$, gehe zu 1.

Beim ersten Punkt kann man das Newton–Verfahren verwenden. Dazu benötigt man $p_n(z)$ und $p_n'(z)$ für die gegenwärtige Iterierte z. Aus (3.10) folgt

$$p_n'(x) = q_{n-1}(x) + x q_{n-1}'(x) - z q_{n-1}'(x),$$

also

$$p_n'(z) = q_{n-1}(z).$$

Diesen Wert kann man gemeinsam mit $p_n(z)$ mit dem erweiterten Horner–Schema berechnen:

p_n	a_n	a_{n-1}	a_{n-2}	$\ldots$	a_1	a_0
z	–	$b_n z$	$b_{n-1} z$	$\ldots$	$b_2 z$	$b_1 z$
p_{n-1}	b_n	b_{n-1}	b_{n-2}	$\ldots$	b_1	$b_0 = p_n(z)$
z	–	$c_n z$	$c_{n-1} z$	$\ldots$	$c_2 z$	
	c_n	c_{n-1}	c_{n-2}	$\ldots$	$c_1 = p_n'(z)$	

Beispiel 3.15 Berechne $p_4(z)$, $p_4'(z)$ für

$$p_4(x) = 3x^4 - 5x^2 + 26x - 17, \quad z = 2.$$

p_4	3	0	-5	26	-17
2	–	6	12	14	80
p_3	3	6	7	40	63
2	–	6	24	62	
	3	12	31	102	

Damit sind $63 = p_4(2)$ und $p_4'(2) = 102$. □

Dieses Vorgehen kann man erweitern, um alle Ableitungen von $p_n(x)$ an der Stelle z zu berechnen, vollständiges Horner–Schema, siehe Literatur.

Bemerkung 3.16

1. Die Verbindung von Newton–Verfahren und Horner–Schema zur Berechnung einer Nullstelle von $p_n(x)$ nennt man Newton–Horner–Verfahren. Ein Iterationsschritt benötigt $4n$ Flops ($p_n(z) : 2n, q_{n-1}(z) : 2(n-1)$, eine Division, eine Subtraktion).
2. Beginnt man mit einer reellen Startnäherung, so erhält man beim Newton–Horner–Verfahren nur reelle Iterierte. Komplexe Nullstellen kann man so nicht finden. Sowohl das Newton–Verfahren als auch das Horner–Schema besitzen für komplexe Zahlen die gleiche Gestalt wie für reelle Zahlen. Verwendet man eine komplexe Startnäherung, so erhält man im allgemeinen komplexe Iterierte. Dazu benötigt man Computerarithmetik für komplexe Zahlen. Diese wird in MATLAB automatisch erledigt, falls komplexe Eingangsdaten vorliegen. *MATLAB–Demo von Newton–Verfahren im Komplexen*
3. Eine fortgesetzte Deflation erhöht den Einfluss von Rundungsfehlern. Um diesen gering zu halten, sollte man zwei Dinge beachten:
   - Berechne zuerst die betragsmäßig kleinen Nullstellen, weil diese am anfälligsten gegen Rundungsfehler sind.
   - Hat man eine Approximation für eine Nullstelle mit dem Newton–Horner–Verfahren gefunden, so nehme man diese Approximation als Startnäherung für das Newton–Verfahren angewandt auf das Originalpolynom $p_n(x)$: Newton–Horner–Verfahren mit Verfeinerung.
4. In MATLAB erhält man alle Nullstellen eines Polynoms mit dem Befehl `roots`.

□

3.6 Abbruchkriterien für iterative Verfahren

Ein wesentliches Problem in der Praxis sind Kriterien, mit denen man die Iteration abbricht. Eine erste Möglichkeit ist die Kontrolle des Residuums:

Sei $\varepsilon > 0$ vorgegeben, dann breche die Iteration ab, falls $\left|f\left(x^{(k)}\right)\right| < \varepsilon$.

Es gibt jedoch Situationen, in denen dieses Kriterium entweder zu stark oder zu optimistisch ist, siehe Abbildung 3.3

Aus diesem Grunde ist es oft besser, die Änderung der Korrektur zu betrachten:

Sei $\varepsilon > 0$ vorgegeben, dann breche die Iteration ab, falls $\left|x^{(k+1)} - x^{(k)}\right| < \varepsilon$.

Darüber hinaus sollte immer eine maximale Anzahl von Iterationen als Abbruchkriterium vorgegeben sein.

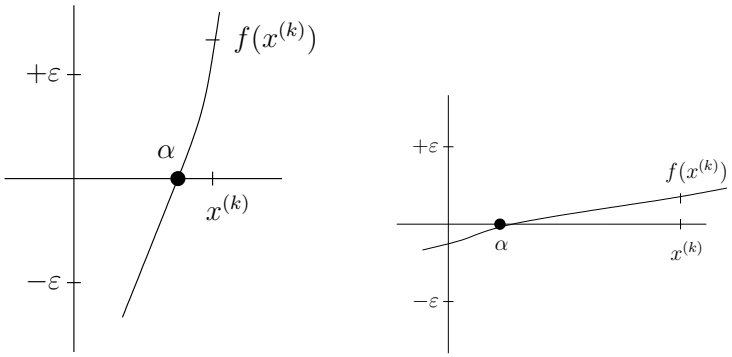


Abbildung 3.3: Links: Residuumkriterium zu stark, da $f'(\alpha)$ sehr groß ist; rechts: Residuumkriterium zu optimistisch, da $f'(\alpha) \approx 0$.

Kapitel 4

Interpolation

4.1 Aufgabenstellung

Gegeben seien $(n+1)$ Paare (x_i, y_i), $i = 0, \ldots n$, wobei die x_i paarweise verschieden sind. Die Interpolationsaufgabe besteht darin, eine (einfache) Funktion Φ zu finden, so dass $\Phi(x_i) = y_i$, $i = 0, \ldots n$, ist.

Man bezeichnet $\{x_i\}$ als die Menge der Stützstellen und die Menge $\{y_i\}$ als die Menge der Stützwerte. Man sagt, dass die Funktion $\Phi(x)$ die Stützwerte $\{y_i\}$ an den Stützstellen $\{x_i\}$ interpoliert.

Natürlich gibt es unendlich viele Funktionen, die die Interpolationsaufgabe erfüllen. Deshalb muss man die Klasse der Funktionen festlegen, in der man die Interpolierende $\Phi(x)$ sucht. Man unterscheidet zum Beispiel:

- $\Phi(x)$ ist ein Polynom – Polynominterpolation,
- $\Phi(x) = a_0 + a_1 e^{ix} + \ldots + a_n e^{inx}$ – trigonometrische Interpolation ($i = \sqrt{-1}$),
- $\Phi(x)$ ist stückweise ein Polynom – Spline–Interpolation,
- $\Phi(x)$ ist eine rationale Funktion – rationale Interpolation.

4.2 Polynominterpolation

Wir bezeichnen mit

$$P_n := \left\{p(x) \ : \ p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0, \ a_i \in \mathbb{R}, i = 0, \ldots, n\right\}$$

den Raum aller Polynome vom Höchstgrad n. Es werden $(n+1)$ Paare (x_i, y_i), $i = 0, \ldots, n$, $x_i \neq x_j$ für $i \neq j$, betrachtet. Die Aufgabe der Polynominterpolation besteht nun darin, ein $p \in P_n$ zu finden, so dass gilt

$$p(x_i) = y_i, \quad i = 0, \ldots, n.$$

Als erstes stellt sich die Frage nach der Existenz und Eindeutigkeit eines solchen Polynoms. Diese wird mit dem folgenden Satz beantwortet.

Satz 4.1 *Zu gegebenen paarweise verschiedenen Stützstellen $x_0, \ldots, x_n \in \mathbb{R}$ und zugehörigen Werten $y_0, \ldots, y_n \in \mathbb{R}$ existiert genau ein Polynom $p \in P_n$ mit $p(x_i) = y_i$, $i = 0, \ldots, n$.*

Beweis: Literatur, zum Beispiel [QSS04]. ∎

Man kann zeigen, dass dieses Polynom sich wie folgt darstellen lässt:

$$p_n(x) = \sum_{k=0}^{n} \frac{\omega_{n+1}(x)}{(x - x_k)\omega'_{n+1}(x_k)} y_k, \tag{4.1}$$

wobei $\omega_{n+1}(x)$ das sogenannte Knotenpolynom vom Grad $n+1$ ist

$$\omega_{n+1}(x) = \prod_{j=0}^{n} (x - x_j)\,. \tag{4.2}$$

Mit der Produktregel erhält man *Übungsaufgabe*

$$\frac{\omega_{n+1}(x)}{(x - x_k)\omega'_{n+1}(x_k)} = \prod_{j=0, j\neq k}^{n} \frac{x - x_j}{x_k - x_j}. \tag{4.3}$$

Damit erhält man die zu (4.1) äquivalente Darstellung

$$p_n(x) = \sum_{k=0}^{n} \left(\prod_{j=0, j\neq k}^{n} \frac{x - x_j}{x_k - x_j} \right) y_k. \tag{4.4}$$

Mit dieser Darstellung erkennt man auch leicht, dass $p_n(x_i) = y_i$ für alle Stützstellen. Setzt man $x = x_i$ in (4.4), dann hat man im Zähler des Produkts für $k \neq i$ den Faktor $x_i - x_i = 0$. Es bleibt somit nur der Summand mit dem Summationsindex $k = i$ übrig

$$p_n(x_i) = \left(\prod_{j=0, j\neq i}^{n} \frac{x_i - x_j}{x_i - x_j} \right) y_i = y_i.$$

Die Formeln (4.1) oder (4.4) bezeichnet man als Langrange–Form des Interpolationspolynoms.

Zur Untersuchung der Genauigkeit der Polynominterpolation geht man wie folgt vor. Man nimmt sich eine Funktion $f(x)$, gibt sich $(n+1)$ Paare von Stützstellen und Stützwerten $(x_i, f(x_i))$ vor, berechnet das Interpolationspolynom $p_n f(x)$ durch diese Punkte und vergleicht dieses mit der vorgegebenen Funktion. *Bild* Man kann die folgende Abschätzung für den Interpolationsfehler beweisen.

Satz 4.2 *Seien $x_0, \ldots, x_1$ paarweise verschiedene Stützstellen und sei z Element das Definitionsbereichs von $f(x)$. Sei weiter $f \in C^{n+1}(I_x)$, wobei I_x das kleinste Intervall ist, mit $x_i \in I_x$, $i = 0, \ldots, n$. Dann ist der Interpolationsfehler im Punkt z durch*

$$E_n(z) := f(z) - p_n f(z) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(z)$$

gegeben, wobei $\xi \in I_x$ ist und $\omega_{n+1}(z)$ in (4.2) definiert ist.

Beweis: Literatur, [QSS04]. ∎

Die Aussage impliziert nicht, dass für $n \to \infty$ gilt, dass $p_n f(x) \to f(x)$ für alle $x \in I_x$. Man kann Funktionen und zugehörige Mengen von Stützstellen so finden (zum Beispiel mit gleichem Abstand), dass es Teilintervalle von I_x gibt, in denen $p_n f(x) \not\to f(x)$ für alle Argumente x aus diesen Teilintervallen.

Zur Untersuchung der Stabilität der Polynominterpolation betrachtet man neben den Paaren $(x_i, f(x_i))$ Paare mit gestörten Werten $(x_i, \tilde{f}(x_i))$. Dann gilt

$$\begin{aligned} \left\| p_n f - p_n \tilde{f} \right\|_{C(I_x)} &:= \max_{x \in I_x} \left| \sum_{k=0}^{n} \left(\prod_{j=0, j\neq k}^{n} \frac{x - x_j}{x_k - x_j} \right) \left(f(x_k) - \tilde{f}(x_k) \right) \right| \\ &\leq \max_{k=1,\ldots,n} \left| f(x_k) - \tilde{f}(x_k) \right| \underbrace{\left\| \sum_{k=0}^{n} \left(\prod_{j=0, j\neq k}^{n} \frac{x - x_j}{x_k - x_j} \right) \right\|_{C(I_x)}}_{\Lambda_n(x)}. \end{aligned}$$

Man nennt $\Lambda_n(x)$ Lebesgue–Konstante und diese Konstante spielt die Rolle einer Konditionszahl für die Polynominterpolation. Das bedeutet, dass kleine Änderungen in den Daten nur dann zu kleinen Änderungen im Ergebnis führen, falls $\Lambda_n(x)$ für alle $x \in I_x$ klein ist. Man kann jedoch zeigen, dass $\Lambda_n \to \infty$ für $n \to \infty$. Für äquidistante Stützstellen hat man beispielsweise

$$\Lambda_n \approx \frac{2^{n+1}}{e\ n \log(n)}, \quad e \ - \text{ Eulersche Zahl.}$$

Daraus folgt, dass die Polynominterpolation für große n instabil werden kann. *MATLAB–Demo*

Im Prinzip ist mit den Darstellungen (4.1) oder (4.4) die Aufgabe der Polynominterpolation gelöst. Diese Darstellungen sind aber aus numerischer Sicht unvorteilhaft, weil:

- unnötig viele Flops zur Auswertung von $p_n f(x)$ an einer bestimmten Stelle x benötigt werden,
- es in diesen Darstellungen unmöglich ist, auf einfache Art und Weise zusätzliche Stützstellen zu integrieren.

4.3 Die Newton–Interpolation

In diesem Abschnitt wird ein Interpolation eingeführt, die die beiden Nachteile der Standard–Lagrange–Interpolation nicht besitzt, die sogenannte Newton–Interpolation.

Seien $(n+1)$ Paare von Stützstellen und Stützwerten $(x_i, f(x_i))$, $i = 0, \ldots, n$, gegeben und $p_n f(x)$ sei das zugehörige Interpolationspolynom. Nun soll $p_n f(x)$ als eine Summe des Interpolationspolynoms $p_{n-1} f(x)$ zu den Stützstellen $(x_i, f(x_i))$, $i = 0, \ldots, n-1$, und eines Polynom n–ten Grades $q_n(x)$ dargestellt werden

$$p_n f(x) = p_{n-1} f(x) + q_n(x). \tag{4.5}$$

Das Polynom $q_n(x)$ soll nur von den Stützstellen x_i abhängen und nur einen unbekannten Koeffizienten besitzen. Falls es eine solche Darstellung (4.5) gibt, ist die Hinzunahme zusätzlicher Stützstellen kein Problem.

Aus (4.5) folgt

$$q_n(x_i) = p_n f(x_i) - p_{n-1} f(x_i) = 0, \quad i = 0, \ldots, n-1.$$

Damit sind n Nullstellen von $q_n(x)$ bekannt. Mehr kann ein Polynom n–ten Grades nicht besitzen und somit hat $q_n(x)$ die Darstellung

$$q_n(x) = a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}) = a_n \omega_n(x).$$

Damit hat $q_n(x)$ den einzigen unbekannten Koeffizienten a_n. Zur Bestimmung dieses Koeffizienten verwendet man die Stützstelle $(x_n, f(x_n))$ und erhält

$$q_n(x_n) = a_n \omega_n(x_n) = p_n f(x_n) - p_{n-1} f(x_n) = f(x_n) - p_{n-1} f(x_n).$$

Daraus folgt

$$a_n = \frac{f(x_n) - p_{n-1} f(x_n)}{\omega_n(x_n)}. \tag{4.6}$$

Definition 4.3 Seien $(x_i, f(x_i)) \in \mathbb{R} \times \mathbb{R}$, $i = 0, \ldots, n$, paarweise verschiedene Stützstellen. Die k–te dividierte Differenz $f[x_i, x_{i+1}, \ldots, x_{i+k}]$ wird rekursiv definiert durch

$$\begin{aligned} f[x_i] &= f(x_i) \quad \text{für } i = 0, \ldots, n, \\ f[x_i, x_{i+1}, \ldots, x_{i+k}] &= \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}. \end{aligned}$$

□

Um den Zusammenhang zwischen den dividierten Differenzen und der Interpolationsaufgabe herzustellen, betrachten wir zunächst den Fall $n = 1$. Dann sind $p_0 f(x) = f(x_0)$ für alle x und

$$a_1 = \frac{f(x_1) - p_0 f(x_1)}{\omega_1(x_1)} = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = f[x_0, x_1].$$

Damit ergibt sich mit (4.5)

$$p_1 f(x) = f[x_0] + f[x_0, x_1](x - x_0).$$

Für $n = 2$ erhält man

$$\begin{aligned} a_2 &= \frac{f(x_2) - p_1 f(x_2)}{\omega_2(x_2)} = \frac{f[x_2] - f[x_0] - f[x_0, x_1](x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \\ &= \frac{1}{x_2 - x_0} \left(\frac{(f(x_2) - f(x_1)) + (f(x_1) - f(x_0)) - f[x_0, x_1](x_2 - x_0)}{x_2 - x_1} \right) \\ &= \frac{1}{x_2 - x_0} \left(\frac{f[x_1, x_2](x_2 - x_1) + f[x_0, x_1](x_1 - x_0) - f[x_0, x_1](x_2 - x_0)}{x_2 - x_1} \right) \\ &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = f[x_0, x_1, x_2], \end{aligned}$$

also

$$p_2 f(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1).$$

Durch Induktion erhält man für a_n in (4.6)

$$a_n = f[x_0, x_1, \ldots, x_n]$$

und für das Interpolationspolynom

$$p_n f(x) = \sum_{k=0}^{n} \omega_k(x) f[x_0, x_1, \ldots, x_k]. \tag{4.7}$$

Diese Darstellung wird Newtonsche Interpolationsformel genannt. Die Eindeutigkeit der Polynominterpolation sichert, dass man das gleiche Polynom wie bei der Lagrange–Interpolation erhält.

Zur Berechnung der dividierten Differenzen nutzt man ein Dreiecksschema:

	$k=0$		$k=1$		$k=2$		$k=n$
x_0	$\underline{f[x_0]} = f(x_0)$	$\searrow$					
x_1	$f[x_1] = f(x_1)$	$\rightarrow$	$\underline{f[x_0, x_1]}$				
		$\searrow$		$\searrow$			
x_2	$f[x_2] = f(x_2)$	$\rightarrow$	$f[x_1, x_2]$	$\rightarrow$	$\underline{f[x_0, x_1, x_2]}$		
$\vdots$							
		$\searrow$		$\searrow$			
x_n	$f[x_n] = f(x_n)$	$\rightarrow$	$f[x_{n-1}, x_n]$	$\rightarrow$	$f[x_{n-2}, x_{n-1}, x_n]$	$\ldots$	$\underline{f[x_0, \ldots, x_n]}$

Beispiel 4.4 Man berechne das Newton–Interpolationspolynom durch die in der Tabelle angegebenen Paare von Stützstellen und Stützwerten:

x_i	$f(x_i) = f[x_i]$			
-1	$\underline{2}$			
0	4	$\frac{4-2}{0-(-1)} = \underline{2}$		
2	6	$\frac{6-4}{2-0} = 1$	$\frac{1-2}{2-(-1)} = \underline{-\frac{1}{3}}$	
3	12	$\frac{12-6}{3-2} = 6$	$\frac{6-1}{3-0} = \frac{5}{3}$	$\frac{5/3-(-1/3)}{3-(-1)} = \underline{\frac{1}{2}}$

Es folgt

$$p_3 f(x) = 2 + 2(x+1) - \frac{1}{3}(x+1)x + \frac{1}{2}(x+1)x(x-2).$$

□

Zur Berechnung der Koeffizienten im Newton–Interpolationspolynom (4.7) benötigt man nur die unterstrichenen Werte. Deswegen braucht man zu ihrer Berechnung nur einen Vektor der Länge $n+1$. Man geht spaltenweise vor. Zunächst belegt man den Vektor mit $f[x_0], \ldots, f[x_n]$. Im zweiten Schritt überschreibt man $f[x_1], \ldots, f[x_n]$ mit $f[x_0, x_1], \ldots, f[x_{n-1}, x_n]$ und so weiter. Wenn man allerdings später noch Stützstellen hinzufügen will, braucht man das gesamte Dreieck.

Der Rechenaufwand pro dividierter Differenz ist 3 Flops. Zur Berechnung aller Koeffizienten des Newton–Interpolationspolynoms hat man

$$n + (n-1) + (n-2) + \ldots + 1 = \frac{n}{2}(n+1)$$

dividierte Differenzen zu berechnen, so dass der Gesamtaufwand

$$\frac{3}{2}n^2 + \frac{3}{2}n \text{ Flops}$$

beträgt.

Bezüglich der Approximationsgüte des Newton–Interpolationspolynoms gilt die Aussage von Satz 4.1.

4.4 Spline–Interpolation

spline (engl.) – längliches, dünnes Stück Holz oder Metall

Die Polynominterpolation besitzt zwei sehr negative Eigenschaften:

- für große n, das heißt viele Stützstellen kann sie instabil werden, insbesondere bei äquidistanten Stützstellen,
- für nichtglatte Funktionen kann der Interpolationsfehler zwischen den Stützstellen beliebig groß werden.

Matlab–Demo Diese beiden Situationen (viele äquidistante Stützstellen oder nichtglatte Funktionen) treten in den Anwendung jedoch häufig auf. In diesen Fällen verwendet man oft die Spline–Interpolation.

Definition 4.5 Seien $x_0, \ldots, x_n \in [a,b]$ paarweise verschiedene Punkte mit $a = x_0 < x_1 < \ldots < x_n = b$. Die Funktion $s_k(x) : [a,b] \to \mathbb{R}$ wird Spline vom Grad k bezüglich der Stützstellen $\{x_j\}$ genannt, falls

$$s_k(x)|_{[x_j, x_{j+1}]} \in P_k, \quad j = 0, \ldots, n-1, \tag{4.8}$$

$$s_k(x) \in C^{k-1}([a,b]). \tag{4.9}$$

□

Das bedeutet ein Spline vom Grad k ist

- eine stückweise polynomiale Funktion, in jedem Teilintervall ein Polynom vom Grad k,
- im Gesamtintervall noch $(k-1)$–mal differenzierbar.

In jedem Teilintervall kann man den Spline als

$$s_{k,j}(x) = \sum_{i=0}^{k} s_{ij}(x - x_j)^i, \quad x \in [x_j, x_{j+1}],\ j = 0, \ldots, n-1,$$

darstellen. Man hat also $n(k+1)$ unbekannte Koeffizienten s_{ij}. Aus (4.9) folgt

$$s_{k,j-1}^{(m)}(x_j) = s_{k,j}^{(m)}(x_j), \quad j = 1, \ldots, n-1;\ m = 0, \ldots, k-1.$$

Das sind $k(n-1)$ Bedingungen an die Koeffizienten. Im allgemeinen soll der Spline eine Funktion f approximieren, deren Werte in den Stützstellen berechnet werden können. Damit hat man weitere $n+1$ Bedingungen: $s_k(x_j) = f(x_j)$, $j = 0, \ldots, n$. Es fehlen noch $k-1$ Bedingungen.

In der Praxis nutzt man oft kubische Splines, das heißt $k = 3$. Ein kubischer Spline ist zweimal stetig bis zum Rand differenzierbar, insbesondere ist seine Krümmung wohldefiniert. Für die zwei fehlenden Bedingungen setzt man zum Beispiel die Randwerte

$$s_3''(a) = s_3''(b) = 0. \tag{4.10}$$

Betrachten wir nun eine Funktion f, die durch einen kubischen Spline interpoliert werden soll. Wir verwenden die Bezeichnungen

$$f_i = s_3(x_i), \quad m_i = s_3'(x_i), \quad M_i = s_3''(x_i), \quad i = 0, \ldots, n.$$

Die zweite Ableitung $s_3''(x)$ ist eine stetige, stückweise lineare Funktion (Polygonzug). Mit den obigen Bezeichnungen gilt

$$s_3''(x) = M_{i-1}\frac{x_i - x}{x_i - x_{i-1}} + M_i\frac{x - x_{i-1}}{x_i - x_{i-1}}, \quad x \in [x_{i-1}, x_i].$$

Die Stetigkeit von $s_3''(x)$ überprüft man durch Einsetzen der Intervallgrenzen. Zweimaliges integrieren liefert eine Darstellung des Splines in $[x_{i-1}, x_i]$

$$s_3(x) = \frac{M_{i-1}}{6}\frac{(x_i - x)^3}{x_i - x_{i-1}} + \frac{M_i}{6}\frac{(x - x_{i-1})^3}{x_i - x_{i-1}} + c_{i-1}(x - x_{i-1}) + d_{i-1}. \tag{4.11}$$

Einsetzen der Endpunkte des Teilintervalls ergibt die Konstanten:

$$\begin{aligned} f_{i-1} &= s_3(x_{i-1}) = \frac{M_{i-1}}{6}(x_i - x_{i-1})^2 + d_{i-1}, \quad \Longrightarrow \\ d_{i-1} &= f_{i-1} - \frac{M_{i-1}}{6}(x_i - x_{i-1})^2, \end{aligned} \tag{4.12}$$

und

$$\begin{aligned} f_i &= s_3(x_i) = \frac{M_i}{6}(x_i - x_{i-1})^2 + c_{i-1}(x_i - x_{i-1}) + d_{i-1} \Longrightarrow \\ c_{i-1} &= \frac{f_i - f_{i-1}}{x_i - x_{i-1}} - \frac{x_i - x_{i-1}}{6}(M_i - M_{i-1}). \end{aligned} \tag{4.13}$$

Man rechnet schnell nach, dass mit diesen Konstanten die Stetigkeit von $s_3(x)$ in den Stützstellen gewährleistet ist.

Aus der Stetigkeitsbedingung für die erste Ableitung erhält man $(n-1)$ Gleichungen für $M_0, \ldots, M_n$. Es gilt für $x \in [x_{i-1}, x_i]$

$$s_3'(x) = -\frac{M_{i-1}}{2}\frac{(x_i - x)^2}{x_i - x_{i-1}} + \frac{M_i}{2}\frac{(x - x_{i-1})^2}{x_i - x_{i-1}} + c_{i-1}.$$

Für den rechten Randpunkt erhält man

$$\begin{aligned} s_3'(x_{i-0}) &= \frac{M_i}{2}(x_i - x_{i-1}) + \frac{f_i - f_{i-1}}{x_i - x_{i-1}} - \frac{x_i - x_{i-1}}{6}(M_i - M_{i-1}) \\ &= \left(\frac{M_{i-1}}{6} + \frac{M_i}{3}\right)(x_i - x_{i-1}) + \frac{f_i - f_{i-1}}{x_i - x_{i-1}}. \end{aligned}$$

Betrachtet man nun den linken Endpunkt des Intervalls $[x_i, x_{i+1}]$, so erhält man auf die gleiche Weise

$$\begin{aligned} s_3'(x_{i+0}) &= -\frac{M_i}{2}(x_{i+1} - x_i) + \frac{f_{i+1} - f_i}{x_{i+1} - x_i} - \frac{x_{i+1} - x_i}{6}(M_{i+1} - M_i) \\ &= \left(-\frac{M_i}{3} - \frac{M_{i+1}}{6}\right)(x_{i+1} - x_i) + \frac{f_{i+1} - f_i}{x_{i+1} - x_i}. \end{aligned}$$

Setzt man diese Bedingungen gleich, so erhält man $(n-1)$ Gleichungen für $M_0, \ldots, M_n$. Für die zwei fehlenden Gleichungen setzt man

$$2M_0 + \lambda_0 M_1 = g_0, \quad \mu_n M_{n-1} + 2M_n = g_n$$

mit vorgegebenen Werten $\lambda_0, \mu_n \in [0, 1]$ und g_0, g_n. Zur Erfüllung von (4.10) setzt man alle diese Werte gleich Null.

Zur Berechnung der Koeffizienten $M_0, \ldots, M_n$ hat man damit folgendes lineares Gleichungssystem zu lösen

$$\begin{pmatrix} 2 & \lambda_0 & & & & & & \\ \mu_1 & 2 & \lambda_1 & & & & & \\ & \mu_2 & 2 & \lambda_2 & & & & \\ & & & \ddots & & & & \\ & & & & \mu_{n-1} & 2 & \lambda_{n-1} & \\ & & & & & \mu_n & 2 & \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{n-1} \\ g_n \end{pmatrix}.$$

Die Koeffizienten sind

$$\begin{aligned} \mu_i &= \frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}}, \\ \lambda_i &= \frac{x_{i+1} - x_i}{x_{i+1} - x_{i-1}}, \\ g_i &= \frac{6}{x_{i+1} - x_{i-1}}\left(\frac{f_{i+1} - f_i}{x_{i+1} - x_i} - \frac{f_i - f_{i-1}}{x_i - x_{i-1}}\right), \quad i = 1, \ldots, n-1. \end{aligned}$$

Nach der Lösung dieses Systems erhält man mit (4.12), (4.13) durch Einsetzen in (4.11) die Darstellung des kubischen Splines. *MATLAB–Demo*

Satz 4.6 *Seien $f \in C^4([a, b])$,*

$$h_{\max} = \max_{i=0,\ldots,n-1}(x_{i+1} - x_i), \quad h_{\min} = \min_{i=0,\ldots,n-1}(x_{i+1} - x_i), \quad \beta = \frac{h_{\max}}{h_{\min}} \geq 1.$$

Dann gilt

$$\max_{x\in[a,b]} \left| f^{(r)}(x) - s_3^{(r)}(x) \right| \leq C_r h_{\max}^{4-r} \max_{x\in[a,b]} \left| f^{(4)}(x) \right|$$

mit $C_0 = 5/384$, $C_1 = 1/24$, $C_2 = 3/8$ und $C_3 = \frac{1}{2}(\beta + 1/\beta)$.

Das bedeutet, dass für $h_{\max} \to 0$ der kubische Spline samt seinen ersten beiden Ableitungen punktweise gegen f beziehungsweise gegen die entsprechenden Ableitungen von f konvergiert. Innerhalb der Intervalle konvergiert auch noch die dritte Ableitung des kubischen Splines, an den Stützstellen konvergiert der Mittelwert der beiden einseitigen dritten Ableitungen.

Kapitel 5

Numerische Integration

Sei $f : [a,b] \to \mathbb{R}$ eine integrierbare Funktion. Die Berechnung von

$$I(f) := \int_a^b f(x)\ dx$$

kann schwierig oder sogar analytisch nicht durchführbar sein. Jede explizite Formel, die eine Näherung für $I(f)$ darstellt, wird Quadraturformel genannt.

5.1 Interpolatorische Quadraturformeln

Eine Näherung $I_n(f)$ an $I(f)$ erhält man, indem man die Funktion $f(x)$ durch eine Approximation $f_n(x)$ ersetzt, die man einfacher integrieren kann

$$I_n(f) := \int_a^b f_n(x)\ dx.$$

Polynome sind Funktionen, die sich einfach integrieren lassen. Wählt man $(n+1)$ paarweise verschiedene Stützstellen aus $[a,b]$, so hat das Lagrange–Interpolationspolynom von f die Gestalt, siehe (4.4),

$$p_n f(x) = \sum_{i=0}^{n} \underbrace{\left(\prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \right)}_{=:l_i(x)} f(x_i).$$

Man erhält

$$I_n(f) = \sum_{i=0}^{n} f(x_i) \int_a^b l_i(x)\ dx.$$

Das ist eine spezielle Form der Quadraturformel

$$I_n(f) = \sum_{i=0}^{n} \alpha_i f(x_i). \tag{5.1}$$

Die Punkte x_i in (5.1) werden Knoten genannt und die Werte α_i Gewichte. Falls $f(x)$ eine konstante Funktion ist, $f(x) = c$, so verlangt man, dass die Quadraturformel exakt sein soll, also $I_n(f) = I(f)$. Aus dieser Bedingung folgt

$$I(f) = c\,(b-a) = I_n(f) = c \left(\sum_{i=0}^{n} \alpha_i \right),$$

also

$$\left(\sum_{i=0}^{n} \alpha_i\right) = b - a.$$

Als Ordung oder Genauigkeit einer Quadraturformel definiert man diejenige natürliche Zahl $r \geq 0$, für die gilt $I_n(f) = I(f)$ für alle $f \in P_r$. Das bedeutet, eine Quadraturformel r–ter Ordnung integriert alle Polynome vom Grad r exakt.

Bei der praktischen Berechnung von $I_n(f)$ wird man im allgemeinen nicht $f(x)$ durch $p_n f(x)$ im gesamten Intervall $[a, b]$ ersetzen. Stattdessen wird man $[a, b]$ zuerst in Teilintervalle $[x_i, x_{i+1}]$, $i = 0, \ldots, n$, $x_0 = a$, $x_n = b$, zerlegen und auf jedem Teilintervall $f(x)$ durch $p_n f(x)$ approximieren. Das liefert die sogenannten zusammengesetzten interpolatorischen Quadraturformeln.

Mittelpunktregel

In der Mittelpunktregel wird f durch eine konstante Funktion ersetzt, deren Wert der Funktionswert in der Mitte des Intervalls ist

$$I_0(f) = \int_a^b f\left(\frac{a+b}{2}\right)\, dx = f\left(\frac{a+b}{2}\right)(b-a).$$

Sei $f \in C^2([a, b])$. Taylor–Entwicklung von $f(x)$ im Intervallmittelpunkt liefert

$$f(x) = f\left(\frac{a+b}{2}\right) + f'\left(\frac{a+b}{2}\right)\left(x - \frac{a+b}{2}\right) + \frac{f''(\xi)}{2}\left(x - \frac{a+b}{2}\right)^2$$

mit $\xi \in [a, b]$. Ersetzt man $f(x)$ durch die Taylor–Entwicklung, erhält man für den Quadraturfehler

$$\begin{aligned} & I(f) - I_0(f) \\ &= \int_a^b f(x)\, dx - f\left(\frac{a+b}{2}\right)(b-a) \\ &= f\left(\frac{a+b}{2}\right)(b-a) + f'\left(\frac{a+b}{2}\right)\cdot 0 + \frac{f''(\xi)}{3}\left(\frac{b-a}{2}\right)^3 - f\left(\frac{a+b}{2}\right)(b-a) \\ &= \frac{f''(\xi)}{3}\left(\frac{b-a}{2}\right)^3. \end{aligned}$$

Bei der Rechnung wurde ausgenutzt, dass der lineare Term der Taylor–Entwicklung bezüglich des Intervallmittelpunktes eine ungerade Funktion ist und somit das Integral verschwindet. Es folgt, dass die Mittelpunktregel exakt für konstante und lineare Polynome ist, da dort die zweite Ableitung gleich Null ist. Sie besitzt die Ordnung $r = 1$.

Teilt man $[a, b]$ in m gleichlange Teilintervalle der Länge $H = (b-a)/m$ und seien $x_k = a + (2k+1)H/2$, $k = 0, \ldots, m-1$, die Mittelpunkte der Teilintervalle. Dann hat die zusammengesetzte Mittelpunktregel die Gestalt

$$I_{0,m}(f) = H \sum_{k=0}^{m-1} f(x_k).$$

Analog zu oben erhält man für den Quadraturfehler

$$I(f) - I_{0,m}(f) = \frac{b-a}{24} H^2 f''(\xi).$$

Trapezregel

Bei dieser Quadraturformel verwendet man das Lagrange–Interpolationspolynom vom Grad 1 bezüglich a und b. Man erhält

$$\begin{aligned} I_1(f) &= f(a)\int_a^b \frac{x-b}{a-b}\,dx + f(b)\int_a^b \frac{x-a}{b-a}\,dx \\ &= -\frac{f(a)}{a-b}\frac{(a-b)^2}{2} + \frac{f(b)}{b-a}\frac{(b-a)^2}{2} \\ &= \frac{b-a}{2}\left(f(a)+f(b)\right). \end{aligned}$$

Für den Quadraturfehler kann man mit Taylor–Entwicklung zeigen, dass

$$I(f) - I_1(f) = -\frac{(b-a)^3}{12} f''(\xi)$$

mit $\xi \in [a,b]$. Damit ist die Trapezregel auch von erster Ordnung. Die zusammengesetzte Trapezregel für eine gleichmäßige Zerlegung von $[a,b]$ wie oben hat die Form

$$I_m(f) = \frac{H}{2}\sum_{k=0}^{m-1}\left(f(x_k)+f(x_{k+1})\right) = H\left(\frac{f(a)}{2} + f(x_1) + \ldots + f(x_{m-1}) + \frac{f(b)}{2}\right),$$

wobei hier $x_k = a + kH$, $k = 0,\ldots,m$, die Grenzen der Teilintervalle sind.

Simpson–Regel

Bei der Simpson–Regel wird f durch das Interpolationspolynom 2. Grades bezüglich der Knoten $x_0 = a$, $x_1 = (a+b)/2$ und $x_2 = b$ ersetzt. Man erhält die Quadraturformel

$$I_2(f) = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right).$$

Der Quadraturfehler ist

$$I(f) - I_2(f) = -\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\xi),$$

falls $f \in C^4([a,b])$, mit $\xi \in [a,b]$. Die Simpson–Regel ist von dritter Ordnung genau. Die zusammengesetzte Simpson–Regel für eine äquidistante Zerlegung von $[a,b]$ hat die Gestalt

$$I_{2,m}(f) = \frac{H}{6}\left[f(x_0) + 2\sum_{r=1}^{m-1} f(x_{2r}) + 4\sum_{s=0}^{m-1} f(x_{2s+1}) + f(x_{2m})\right],$$

wobei $x_k = a + kH/2$, $k = 0,\ldots,2m$, die Knoten sind.

5.2 Weiterführende Quadraturformeln

Die im vorherigen Abschnitt angegebenen Quadraturformeln lassen sich systematisch verallgemeinern. Man erhält die sogenannte Newton–Cotes–Formeln. In diesen Formeln besitzten die Knoten in $[a,b]$ den gleichen Abstand und $f(x)$ wird durch ein Polynom n–ten Grades approximiert. Die Ordnung der Newton–Cotes–Formeln ist n oder $n+1$. Das ist jedoch nicht die maximale Ordnung, die man mit einem Polynom n–ten Grades erreichen kann. Um eine höhere Genauigkeit zu erreichen, muss man noch die Einschränkung, dass die Knoten äquidistant sein sollen, fallenlassen. Dann kommt man zu den Gaußschen Quadraturformeln, bei denen man sogar die Ordnung $2n+1$ erhält.

Literaturverzeichnis

[KS88] A. Kiełbasiński and H. Schwetlick. *Numerische lineare Algebra*. Deutscher Verlag der Wissenschaften Berlin, 1988.

[QSS04] A. Quateroni, R. Sacco, and F. Saleri. *Numerische Mathematik 1,2*. Springer-Verlag, 2004.

[SK06] H.R. Schwarz and N. Köckler. *Numerische Mathematik*. Teubner-Verlag, Stuttgart, 6. edition, 2006.

Index