

MATTHIAS GERDTS

NUMERISCHE MATHEMATIK I

**Universität Würzburg
WiSe 2009/2010**

ADDRESSE DES AUTHORS:

Matthias Gerdts

Institut für Mathematik

Universität Würzburg

Am Hubland

97074 Würzburg

E-Mail: gerdts@mathematik.uni-wuerzburg.de

WWW: www.mathematik.uni-wuerzburg.de/~gerdts

Vorläufige Version: 9. Dezember 2014

Copyright © 2009 by Matthias Gerdts

Inhaltsverzeichnis

1	Einleitung	2
1.1	Was ist Numerische Mathematik?	2
1.2	Ein einfaches Fahrzeugmodell: Differentialgleichungen, Stabilität und Eigenwerte	4
1.3	Einige Hilfsmittel	9
1.3.1	\mathcal{O} und o Notation	12
1.4	Zahldarstellung und Gleitpunktarithmetik	12
1.5	Rundung und Auslöschung	16
1.6	Algorithmen, Kondition und Stabilität	19
1.6.1	Rückwärtsstabilität	24
2	Numerische Lösung linearer Gleichungssysteme	25
2.1	Grundlagen und Lösbarkeit	27
2.2	Gauss'sches Eliminationsverfahren und LR-Zerlegung	30
2.2.1	Einfach zu lösende Gleichungssysteme	31
2.2.2	Das Gauss'sche Eliminationsverfahren	33
2.2.3	Pivoting	37
2.2.4	Die LR-Zerlegung einer Matrix	41
2.2.5	Pivoting in Matrixnotation	46
2.3	Cholesky-Zerlegung	48
2.4	Iterative Nachbesserung	52
2.5	Fehlereinfluß und Kondition	53
2.6	Iterative Lösung	62
2.6.1	Spezielle Verfahren	69
2.7	Das konjugierte Gradientenverfahren	75
2.7.1	Bestimmung der Schrittweite	76
2.7.2	Bestimmung der Suchrichtung	76
2.7.3	Konvergenz	77
3	Lineare Ausgleichsprobleme	86
3.1	Gauss'sche Normalengleichungen	88
3.2	QR-Zerlegung einer Matrix	91
3.2.1	QR-Zerlegung zur Lösung von linearen Gleichungssystemen	92

3.2.2	Lösung des linearen Ausgleichsproblems mittels QR-Zerlegung	93
3.2.3	QR-Zerlegung durch Gram-Schmidt'sche Orthogonalisierung	94
3.2.4	QR-Zerlegung nach Householder	97
3.2.5	Rangbestimmung einer Matrix mittels QR-Zerlegung und Pivoting .	103
4	Nichtlineare Gleichungen	105
4.1	Bisektion (Intervallschachtelungsverfahren)	108
4.2	Fixpunktiteration	111
4.3	Newtonverfahren	115
4.3.1	Der eindimensionale Fall	115
4.3.2	Der n -dimensionale Fall	118
4.3.3	Varianten des Newtonverfahrens	125
4.3.4	Existenz einer Nullstelle	132
4.4	Sekantenverfahren	134
5	Interpolation	137
5.1	Polynominterpolation	139
5.1.1	Newton'sche dividierte Differenzen	141
5.1.2	Fehlerdarstellung interpolierender Polynome	145
5.2	Splineinterpolation	151
5.2.1	Fehlerdarstellung interpolierender kubischer Splines	159
6	Numerische Integration	162
6.1	Interpolatorische Quadraturformeln	165
6.1.1	Zusammengesetzte Trapezregel und Euler-MacLaurin'sche Summen- formel	170
6.2	Romberg-Verfahren und Extrapolation	175
6.3	Gauss'sche Quadraturformeln	179
6.4	Adaptive Verfahren am Beispiel der Simpson-Regel	187
6.5	Mehrdimensionale Integrale	189
7	Trigonometrische Approximation und Schnelle Fouriertransformation (FFT)	191
7.1	Trigonometrische Approximation	191
7.2	Diskrete trigonometrische Approximation und Interpolation	196
7.3	Schnelle Fourier-Transformation	200
8	Numerische Differentiation	205
8.1	Approximation der ersten Ableitung mittels Taylorentwicklung	205

8.1.1	Finite Differenzen Verfahren erster Ordnung	205
8.1.2	Finite Differenzen zweiter Ordnung	206
8.2	Approximation zweiter Ableitungen mittels Taylorentwicklung	206
8.3	Richardson Extrapolation	207
A	Software	209
	Bibliography	211

Lecture plan

Vorlesungen:

Date	Hours	Pages
20.10.2009	13:30-15:00	VL, 2–9
22.10.2009	13:30-15:00	VL, 10–16
27.10.2009	13:30-15:00	VL, 17–24
29.10.2009	13:30-15:00	VL, 25–34
03.11.2009	13:30-15:00	VL, 34–41
05.11.2009	13:30-15:00	VL, 41–46
10.11.2009	13:30-15:00	VL, 47–51
12.11.2009	13:30-15:00	VL, 52–56
17.11.2009	13:30-15:00	VL, 56–62
19.11.2009	13:30-15:00	VL 62–67
24.11.2009	13:30-15:00	VL, 68–75
26.11.2009	13:30-15:00	VL, 75–82
01.12.2009	15:15-16:00	VL 83–90
03.12.2009	13:30-15:00	VL 91–97
08.12.2009	13:30-15:00	VL, 97–102
10.12.2009	13:30-15:00	VL 102–108
15.12.2009	13:30-15:00	VL, 108–116
17.12.2009	13:30-15:00	VL, 116–122
22.12.2009	13:30-15:00	VL, 122–132
07.01.2010	13:30-15:00	VL, 137–143
12.01.2010	13:30-15:00	VL, 144–151
14.01.2010	13:30-15:00	VL, 151–159
19.01.2010	13:30-15:00	VL, 159–166
21.01.2010	13:30-15:00	VL, 167–179 ohne 172–174
26.01.2010	13:30-15:00	VL, 179–185
28.01.2010	13:30-15:00	VL, 185–187, 172–174
02.02.2010	13:30-15:00	VL, 187–189, 191–194
04.02.2010	13:30-15:00	Klausur
09.02.2010	13:30-15:00	VL,
11.02.2010	13:30-15:00	VL,

Kapitel 1

Einleitung

1.1 Was ist Numerische Mathematik?

Die Numerische Mathematik (kurz: Numerik) beschäftigt sich mit der Entwicklung, Analyse und Implementierung von Algorithmen zur Lösung von mathematischen Problemen. Die Problemstellungen sind dabei häufig durch konkrete Anwendungen motiviert.

Hierbei ist es wichtig, zwischen einem

(mathematischen) Problem

(z.B. Lösung eines linearen Gleichungssystems, Approximation einer Funktion, Berechnung von Eigenwerten) und einem

(numerischen) Algorithmus

zur Lösung des mathematischen Problems zu unterscheiden. Ein Algorithmus ist eine Verfahrensvorschrift zur konkreten Lösung eines mathematischen Problems, wobei es verschiedene Algorithmen zur Lösung desselben mathematischen Problems geben kann, etwa die Gauss'sche Eliminationsmethode und die Cramer'sche Regel zur Lösung linearer Gleichungssysteme.

Die meisten Probleme, die in der Numerik behandelt werden, können entweder nicht analytisch gelöst werden (d.h. es gibt keine geschlossene Lösungsdarstellung) und erfordern daher approximative Lösungsverfahren oder sie sind zu rechenintensiv, um Lösungen per Hand ausrechnen zu können (wer möchte schon lineare Gleichungssysteme mit tausenden von Gleichungen per Hand lösen?).

Beispiel 1.1.1

Für die folgenden Probleme kann die Lösung nicht explizit angegeben werden:

- *Die nichtlinearen Gleichungen*

$$x - \cos(x) = 0, \quad \exp(x) + x - 2 = 0, \quad x^5 - 3x^4 - \pi x^3 + x^2 - 43x + \sqrt{2} = 0$$

können nicht geschlossen gelöst werden.

- *Die Werte der Integrale*

$$\int_1^5 \frac{\exp(x)}{x} dx, \quad \int_1^2 \frac{\sin(x)}{x} dx, \quad \int_0^1 \frac{1}{\sqrt{x^3 + 1}} dx$$

können nicht analytisch berechnet werden.

- *Die nichtlineare Differentialgleichung (Pendelgleichung)*

$$x''(t) = -\frac{3g}{2\ell} \sin(x(t))$$

für x kann nicht geschlossen gelöst werden.

Es gibt sehr viele solcher Beispiele. Die meisten der in der Praxis auftretenden Beispiele sind nicht geschlossen lösbar.

Das Gebiet der Numerik bekam insbesondere durch die fortschreitende Entwicklung von leistungsfähigen Computern eine immer stärkere Bedeutung, da diese die Lösung immer komplexerer Aufgabenstellungen erlauben, z.B.

- Wettervorhersage mittels numerischer Berechnungsverfahren
- Simulation und Optimierung von technischen Systemen (Crashtests, Strömungssimulation, Baustatik, ...)
- Signalverarbeitung, Approximation von Funktionen, Bildverarbeitung (jpeg/mpeg Kompression)
- ...

Im Rahmen der Vorlesung werden verschiedene mathematische Problemstellungen und Lösungsalgorithmen behandelt. Dabei geht es nicht nur um die

Herleitung und Formulierung von Algorithmen,

sondern insbesondere auch um deren Eigenschaften. Zentrale Begriffe und Fragestellungen in der Numerik sind die folgenden:

- **Stabilität:** Wie reagiert ein Algorithmus auf Rundungsfehler, Störungen oder Parameteränderungen?
- **Konvergenz und Konvergenzgeschwindigkeit:** Konvergiert ein Algorithmus gegen die Lösung eines Problems und wenn ja, wie schnell?
- **Approximationsfehler und Approximationsordnung:** Wie gut approximiert ein Algorithmus die Lösung in Abhängigkeit von Diskretisierungsparametern?

Darüber hinaus besteht ein wesentlicher Teil der Numerischen Mathematik in der konkreten

Implementierung von Algorithmen auf dem Computer.

1.2 Ein einfaches Fahrzeugmodell: Differentialgleichungen, Stabilität und Eigenwerte

Ein stark vereinfachtes Fahrzeugmodell zur Untersuchung des Fahrkomforts eines Kraftfahrzeugs ist durch das Viertel-Fahrzeugmodell gegeben, vgl. Kortüm und Lugner [KL94] und Abbildung 1.1.

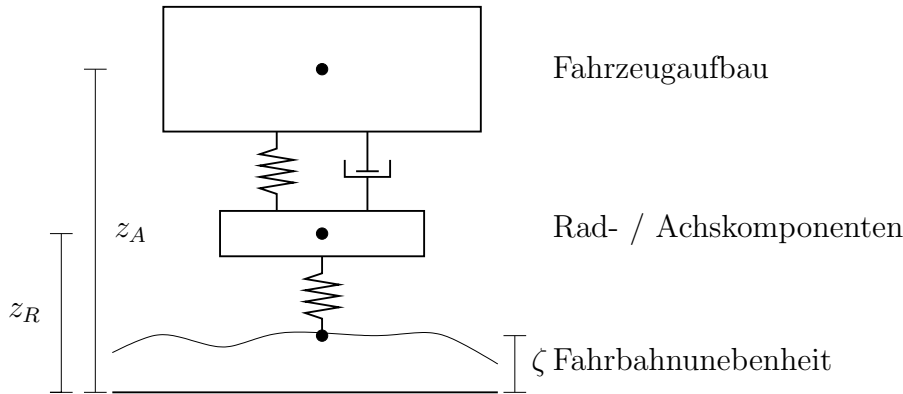


Abbildung 1.1: Viertel-Fahrzeugmodell zur Untersuchung des Fahrkomforts

Die Bewegungsgleichungen der Körperschwerpunkte z_A und z_R sind durch die Differentialgleichungen

$$\begin{aligned} m_A \ddot{z}_A &= -m_A g + F_D + F_F \\ m_R \ddot{z}_R &= -m_R g - F_D - F_F + F_S \end{aligned}$$

gegeben, wobei vereinfachend lineare Kraftgesetze für die Dämpferkraft

$$F_D = c_D(\dot{z}_R - \dot{z}_A),$$

die Federkraft

$$F_F = c_F(z_R - z_A) + F_F^0$$

und die Strassenanregung

$$F_S = c_S(\zeta - z_R) + F_S^0$$

angenommen werden.

Es bezeichnen m_A und m_R die Massen des Fahrzeugaufbaus bzw. der Rad- und Achskomponenten, g die Erdbeschleunigung, c_D und c_F die Dämpfer- bzw. Federkonstante der Radaufhängung und c_S die radiale Karkassensteifigkeit des Reifens, der in idealisierter Form als Federelement modelliert ist. Nachdem durch m_A bzw. m_R geteilt wurde, läßt

sich das Differentialgleichungssystem zweiter Ordnung dann als inhomogenes, lineares Differentialgleichungssystem erster Ordnung für den Zustand

$$x(t) = (z_A(t), z_R(t), v_A(t), v_R(t))^T \in \mathbb{R}^4$$

in Matrixnotation schreiben:

$$\dot{x}(t) = Ax(t) + h(t)$$

mit

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{c_F}{m_A} & \frac{c_F}{m_A} & -\frac{c_D}{m_A} & \frac{c_D}{m_A} \\ \frac{c_F}{m_R} & -\frac{c_F+c_S}{m_R} & \frac{c_D}{m_R} & -\frac{c_D}{m_R} \end{pmatrix},$$

$$h(t) = \begin{pmatrix} 0 \\ 0 \\ -g + \frac{F_F^0}{m_A} \\ -g - \frac{F_F^0}{m_R} + \frac{F_S^0}{m_R} + \frac{c_S}{m_R} \zeta(t) \end{pmatrix}.$$

Wir wählen konkrete Werte

$$\begin{aligned} m_A &= 300.00 \quad [kg], \\ m_R &= 60.00 \quad [kg], \\ g &= 9.81 \quad [m/s^2], \\ c_F &= 300000.00 \quad [N/m], \\ c_D &= 30000.00 \quad [Ns/m], \\ c_S &= 30000.00 \quad [N/m], \\ F_F^0 &= 64743.00 \quad [N], \\ F_S^0 &= 14211.60 \quad [N]. \end{aligned}$$

Man erhält für diesen speziellen Fall

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1000 & 1000 & -100 & 100 \\ 5000 & -5500 & 500 & -500 \end{pmatrix},$$

$$h(t) = \begin{pmatrix} 0 \\ 0 \\ 206 \\ -852 + 500 \cdot \zeta(t) \end{pmatrix}.$$

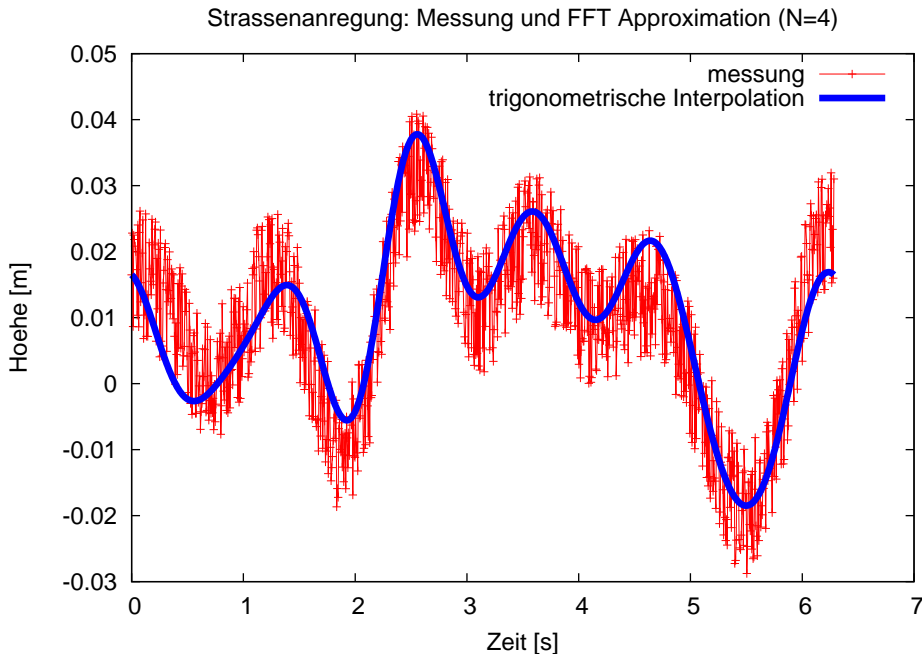
Folgende Fragestellungen können von Interesse sein:

Modellierung der Strassenanregung

Üblicherweise erhält man die Strassenanregung durch Messungen. Diese Messungen sind in der Regel bedingt durch Messungenauigkeiten verrauscht, vgl. Abbildung. Diese Messfehler möchte man in der Simulation möglichst nicht berücksichtigen und herausfiltern. Gesucht ist also ein Modell der Strassenanregung, welches die wesentlichen Schwingungen berücksichtigt. Ein gängiger Ansatz, der auch in der Datenkompression (JPEG, MP3) und der Signal- und Bildverarbeitung eine wichtige Rolle spielt, ist die **Interpolation** durch trigonometrische Polynome. Hierbei erzeugt man für ein geeignetes m und geeignete Koeffizienten $a_j, j = 0, \dots, m-1, b_j, j = 1, \dots, m-1$, eine Funktion

$$p(t) = a_0 + \sum_{j=1}^{m-1} a_j \cos(jt) + b_j \sin(jt),$$

so dass $p(t_k) = \zeta_k$ für Messungen $k = 0, \dots, m-1$ gilt. Offenbar ist p eine Überlagerung von Sinus- und Cosinusfunktionen mit verschiedenen Frequenzen jt und Amplituden a_j, b_j .



Die Abbildung zeigt die Interpolierende p für $m = 8$.

Gleichgewichtslösung

Gesucht ist eine Gleichgewichtslösung bei ebener Fahrbahn mit $\zeta(t) = 0$, d.h. es sind Werte z_A und z_R gesucht, so daß $\ddot{z}_A(t) = \ddot{z}_R(t) = \dot{z}_A(t) = \dot{z}_R(t) = 0$ bzw. $\dot{v}_A(t) = \dot{v}_R(t) = v_A(t) = v_R(t) = 0$ gilt. Diese Fragestellung führt auf das **lineare Gleichungssystem für z_A und z_R**

$$\begin{aligned} 1000z_A - 1000z_R &= 206, \\ -5000z_A + 5500z_R &= -852. \end{aligned}$$

Es hat die Lösung

$$z_A = 0.562, \quad z_R = 0.356.$$

Stabilität

Bei der Konstruktion eines Fahrzeugs ist insbesondere die Auswahl der Federn und Dämpfer (bzw. der entsprechenden Feder- und Dämpferkonstanten) derart vorzunehmen, daß das Fahrzeug ein **stabiles Fahrverhalten** aufweist. Beispielsweise soll das Fahrzeug nach dem Überfahren einer (nicht zu hohen) Bodenwelle nach einer gewissen Zeit wieder in die Gleichgewichtslage zurück gelangen und nicht in einen instabilen Fahrzustand gelangen. Die Begriffe werden genauer spezifiziert. Allgemein heißt eine Lösung $x(t)$ des allgemeinen Differentialgleichungssystems

$$x'(t) = f(t, x(t)), \quad t_0 \leq t < \infty,$$

stabil, wenn zu jedem $\varepsilon > 0$ ein $\delta > 0$ existiert, so daß alle Lösungen $y(t)$ mit

$$\|y(t_0) - x(t_0)\| < \delta$$

für alle $t \geq t_0$ existieren und

$$\|y(t) - x(t)\| < \varepsilon$$

für alle $t \geq t_0$ erfüllen. Die Lösung $x(t)$ heißt **asymptotisch stabil**, wenn sie stabil ist und wenn $\delta > 0$ existiert, so daß für alle Lösungen $y(t)$ mit $\|y(t_0) - x(t_0)\| < \delta$ gilt

$$\lim_{t \rightarrow \infty} \|y(t) - x(t)\| = 0.$$

Eine Lösung $x(t)$ heißt **instabil**, wenn sie nicht stabil ist.

Es zeigt sich, daß die Eigenwerte der Matrix A eine wesentliche Rolle bei der Beurteilung der Stabilität eines Systems spielen. Für inhomogene, lineare Differentialgleichungssysteme der Form

$$\dot{x}(t) = Ax(t) + h(t, x(t))$$

kann gezeigt werden, vgl. Demailly [Dem91], daß jede Lösung $x(t)$ stabil ist, falls

- für sämtliche Eigenwerte λ_j der Matrix A gilt, daß der Realteil $\operatorname{Re} \lambda_j < 0$ ist und
- falls die Inhomogenität h folgende Eigenschaften besitzt:
 - h ist stetig bzgl. beider Argumente,
 - es existiert eine stetige Funktion $k : [t_0, \infty) \rightarrow \mathbb{R}_+$ mit $\lim_{t \rightarrow \infty} k(t) = 0$ und

$$\|h(t, x_1) - h(t, x_2)\| \leq k(t) \|x_1 - x_2\|$$

für alle x_1, x_2 und $t_0 \leq t < \infty$.

Zur Beurteilung der Stabilität des Fahrzeugs benötigen wir also die Eigenwerte der Matrix A . In unserem Beispiel ergeben sich die vier Eigenwerte

$$\lambda_1 \approx -589.11, \lambda_2 \approx -10.58, \lambda_3 = \bar{\lambda}_4 \approx -0.15 + 8.95i.$$

Der Realteil sämtlicher Eigenwerte ist kleiner als Null, folglich hat unser Viertel-Fahrzeug ein stabiles Fahrverhalten, falls h noch obige Bedingungen erfüllt, vgl. auch Abbildung 1.2.

Simulation des Systems

Die Lösung des Differentialgleichungssystems kann nicht für beliebige Funktionen $\zeta(t)$ explizit angegeben werden. Ist man an der Lösung des Systems interessiert, benötigt man numerische Methoden, um diese zumindest näherungsweise berechnen zu können.

Wir modellieren das Überfahren einer Bodenwelle, indem das Fahrzeug zunächst zum Zeitpunkt $t = 0$ [s] aus der Gleichgewichtslage $z_A(0) = 0.562$ [m], $z_R(0) = 0.356$ [m], $v_A(0) = v_R(0) = 0$ [m/s] startend im Zeitintervall $[5, 5 + 2\pi]$ eine Bodenwelle gegeben durch

$$\zeta(t) = \begin{cases} 0.1 \cdot \sin(t - 5), & \text{falls } 5 \leq t \leq 5 + 2\pi, \\ 0, & \text{sonst} \end{cases}$$

überfährt. Abbildung 1.2 zeigt das Ergebnis der Simulation.

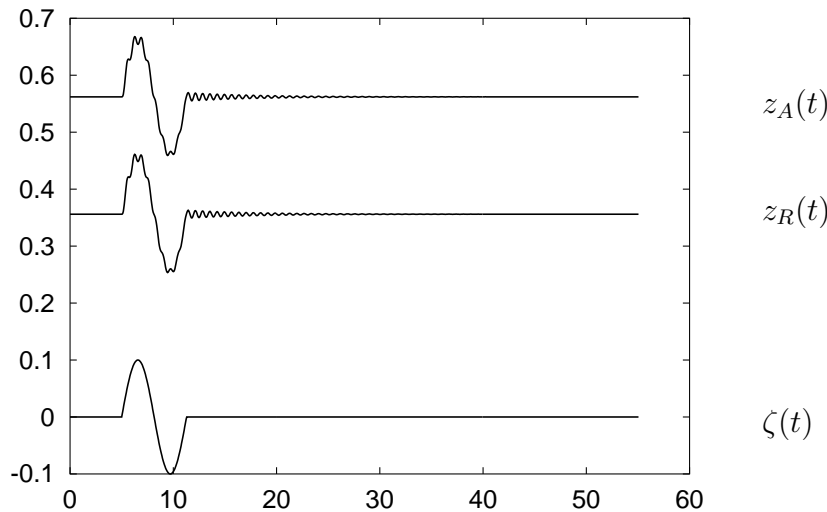


Abbildung 1.2: Simulation des Überfahrens einer Bodenwelle mit dem Viertel-Fahrzeugmodell.

Fazit

Was wurde in diesem Beispiel benötigt?

- Methoden zur Berechnung einer interpolierenden Funktion,

- Methoden zur Lösung von linearen Gleichungssystemen,
- Methoden zur Berechnung von Eigenwerten,
- Methoden zur Lösung von Differentialgleichungssystemen.

1.3 Einige Hilfsmittel

Es werden einige Hilfsmittel aus der Analysis zusammen gefaßt, die im Laufe der Vorlesung benötigt werden.

Der Zwischenwertsatz besagt, daß eine stetige Funktion jeden Wert zwischen zwei Funktionswerten $f(a)$ und $f(b)$ annimmt.

Satz 1.3.1 (Zwischenwertsatz)

Sei $f : [a, b] \rightarrow \mathbb{R}$ stetig in dem abgeschlossenen Intervall $[a, b]$ und sei y eine beliebige Zahl mit $f(a) \leq y \leq f(b)$ oder $f(b) \leq y \leq f(a)$. Dann existiert ein Punkt $c \in [a, b]$ mit $f(c) = y$.

Der Mittelwertsatz erlaubt es, $f'(x)$ in (a, b) durch

$$f'(x) \approx f'(\xi) = \frac{f(b) - f(a)}{b - a}$$

zu approximieren.

Satz 1.3.2 (Mittelwertsatz)

Sei $f : [a, b] \rightarrow \mathbb{R}$ stetig und stetig differenzierbar im offenen Intervall (a, b) . Dann gilt

$$f(b) = f(a) + (b - a)f'(\xi)$$

für ein $\xi \in (a, b)$.

Ist $f = (f_1, \dots, f_m)^\top$ vektorwertig, so kann der Mittelwertsatz zwar auf jede Komponente von f einzeln angewendet werden, jedoch mit unterschiedlichen Zwischenstellen ξ_i , $i = 1, \dots, m$. Eine alternative Darstellung für vektorwertige Funktionen f liefert der Mittelwertsatz in Integralform, welcher die Jacobimatrix von f verwendet.

Definition 1.3.3 (Jacobimatrix)

Existieren für $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ in $x \in \mathbb{R}^n$ die partiellen Ableitungen $\frac{\partial f_j}{\partial x_i}$ für $i = 1, \dots, n$, $j = 1, \dots, m$, so heißt

$$f'(x) = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(x)}{\partial x_1} & \dots & \frac{\partial f_m(x)}{\partial x_n} \end{pmatrix}$$

die Jacobimatrix von f in x .

Satz 1.3.4 (Mittelwertsatz in Integralform)

Sei $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$ stetig differenzierbar in der offenen Menge D und sei $x \in D$. Sei $h \in \mathbb{R}^n$ ein Vektor derart, daß $x + th \in D$ für alle $0 \leq t \leq 1$.

Dann gilt

$$f(x + h) - f(x) = \int_0^1 f'(x + th)h dt.$$

Neben den Mittelwertsätzen ist der Satz von Taylor in der Numerik sehr wichtig. Dieser gilt analog auch im \mathbb{R}^n .

Satz 1.3.5 (Satz von Taylor)

Sei $f : [a, b] \rightarrow \mathbb{R}$ $(n+1)$ -mal stetig differenzierbar in $[a, b]$. Dann gilt für jedes $x, x_0 \in [a, b]$

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

Es gelten folgende Restglieddarstellungen:

(i)

$$E_{n+1}(x; x_0) = \int_{x_0}^x \frac{(x-t)^n}{n!} f^{(n+1)}(t) dt.$$

(ii) Restglieddarstellung nach Lagrange:

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

wobei $\xi = \xi(x, x_0)$ zwischen x und x_0 liegt.

(iii) Restglieddarstellung nach Cauchy:

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

wobei $\xi = \xi(x, x_0)$ zwischen x und x_0 liegt.

Definition 1.3.6 (Taylorpolynom)

Das Polynom

$$T_n(x; x_0) := \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

heißt n -tes Taylorpolynom von f in x_0 .

Falls f $(n+1)$ -mal stetig differenzierbar in $[a, b]$ ist, kann f in $x \in [a, b]$ durch das n -te Taylorpolynom in $x_0 \in (a, b)$ approximiert werden. Der Approximationsfehler in x kann

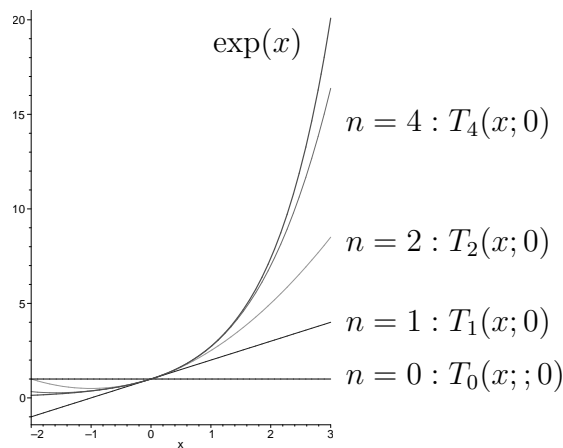
abgeschätzt werden durch

$$\begin{aligned}
 |f(x) - T_n(x; x_0)| &= |E_{n+1}(x; x_0)| \\
 &= \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)^{n+1} \right| \\
 &\leq \max_{\xi \in [a, b]} \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} \right| \cdot |x - x_0|^{n+1} \\
 &= C \cdot |x - x_0|^{n+1}.
 \end{aligned} \tag{1.1}$$

Beispiel 1.3.7

Approximation von $f(x) = \exp(x)$ in $x_0 = 0$ durch $T_n(x; 0)$:

$$\begin{aligned}
 T_0(x; 0) &= 1, \\
 T_1(x; 0) &= 1 + x, \\
 T_2(x; 0) &= 1 + x + \frac{x^2}{2!}, \\
 &\vdots \\
 T_n(x; 0) &= \sum_{i=0}^n \frac{x^i}{i!}.
 \end{aligned}$$



Bemerkung 1.3.8

Formal kann die **Taylorreihe** definiert werden als

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

falls die Ableitungen existieren. Allerdings gilt i.a. **nicht**, daß die Reihe tatsächlich existiert und gleich $f(x)$ ist! Eine hinreichende Bedingung für Gleichheit ist, daß der Fehler $E_{n+1}(x; x_0)$ gegen Null strebt für $n \rightarrow \infty$.

1.3.1 \mathcal{O} und o Notation

Häufig wird es notwendig sein, zwei Funktionen $f(x)$ und $g(x)$ miteinander zu vergleichen, wenn x gegen einen Wert x_0 strebt.

Definition 1.3.9

Seien $f, g : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ gegeben.

(a) Wir schreiben

$$f(x) = \mathcal{O}(g(x)) \quad \text{für } x \rightarrow x_0,$$

falls

$$\limsup_{x \rightarrow x_0} \frac{|f(x)|}{|g(x)|} < \infty.$$

(b) Wir schreiben

$$f(x) = o(g(x)) \quad \text{für } x \rightarrow x_0,$$

falls

$$\lim_{x \rightarrow x_0} \frac{f(x)}{g(x)} = 0.$$

Mit Hilfe der \mathcal{O} -Notation können wir (1.1) schreiben als

$$f(x) - T_n(x; x_0) = \mathcal{O}(|x - x_0|^{n+1}) \quad \text{für } x \rightarrow x_0.$$

Ist f stetig differenzierbar, so gilt

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + o(\|x - x_0\|)$$

mit $\lim_{x \rightarrow x_0} \frac{o(\|x - x_0\|)}{\|x - x_0\|} = 0$.

1.4 Zahldarstellung und Gleitpunktarithmetik

Ein Computer hat nur eine begrenzte Anzahl von Speicherplätzen zur Verfügung. Die Anzahl der auf dem Computer darstellbaren Zahlen ist damit begrenzt auf solche Zahlen, die sich mit endlich vielen Ziffern vor oder nach dem Komma beschreiben lassen. Mathematisch entspricht dies einer Teilmenge der rationalen Zahlen \mathbb{Q} . Man nennt diese Zahlen auch **Maschinenzahlen**.

Die reelle Zahl 12345.678 kann geschrieben werden als

$$12345.678 = 1 \cdot 10^4 + 2 \cdot 10^3 + 3 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0 + 6 \cdot 10^{-1} + 7 \cdot 10^{-2} + 8 \cdot 10^{-3}.$$

Allgemeiner kann eine Zahl (im Dezimalsystem) geschrieben werden als

$$\underbrace{a_n a_{n-1} \cdots a_1 a_0}_{\text{ganzzahliger Anteil}} \cdot \underbrace{b_1 b_2 b_3 \cdots}_{\text{Bruchanteil}} = \underbrace{\sum_{k=0}^n a_k \cdot 10^k}_{\text{ganzzahliger Anteil}} + \underbrace{\sum_{k=1}^{\infty} b_k \cdot 10^{-k}}_{\text{Bruchanteil}}.$$

Tatsächlich benötigen wir mitunter einen unendlichen String von Ziffern rechts des Dezimalpunkts, um eine beliebige reelle Zahl darstellen zu können, z.B.

$$\begin{aligned}\frac{1}{3} &= 0.333333\dots, \\ \pi &= 3.1415926\dots, \\ e &= 2.7182818\dots, \\ \sqrt{2} &= 1.4142135\dots\end{aligned}$$

Die obigen Zahlen sind im Dezimalsystem zur Basis $B = 10$ dargestellt. Jedes andere $B \in \mathbb{N}$, $B > 1$, ist ebenfalls zulässig und man erhält die Darstellung

$$(a_n a_{n-1} \dots a_1 a_0 . b_1 b_2 b_3 \dots)_B = \sum_{k=0}^n a_k \cdot B^k + \sum_{k=1}^{\infty} b_k \cdot B^{-k},$$

wobei $a_i, b_j \in \{0, 1, \dots, B-1\}$.

Im alltäglichen Leben sind wir es gewohnt im **Dezimalsystem** mit $B = 10$ zu rechnen. Bei der internen Darstellung im Computer wird hingegen zumeist eines der folgenden Systeme bevorzugt:

- **Binärsystem** ($B = 2$).

Die Binärzahl 10010.1001 entspricht der Dezimalzahl 18.5625, denn

$$(10010.1001)_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = 18.5625.$$

- **Oktalsystem** ($B = 8$).

- **Hexadezimalsystem** ($B = 16$).

Das Hexadezimalsystem verwendet die „Zahlen“ $a_i, b_j \in \{0, 1, \dots, 9, A, B, C, D, E, F\}$.

Es kann ebenfalls passieren, daß die Darstellung einer Zahl in einem Zahlensystem endlich ist, also nur endlich viele Ziffern benötigt, während sie unendliche viele Ziffern zur Darstellung in einem anderen Zahlensystem benötigt, z.B.

$$(0.1)_{10} = (0.00011001100110011\dots)_2.$$

Soll die reelle Zahl x im Computer realisiert werden, muß auf Grund der begrenzten Speicherkapazität die Anzahl der Ziffern zur Darstellung der Mantisse und des Exponenten auf eine gewisse Länge beschränkt werden, etwa in der Form

$$x = m \cdot B^e, \tag{1.2}$$

wobei

$$m = \pm(m_1 \cdot B^{-1} + m_2 \cdot B^{-2} + \dots + m_\ell \cdot B^{-\ell}) \in \mathbb{R} \tag{1.3}$$

die **Mantisse**, $B \in \mathbb{N}$, $B > 1$ die **Basis** und

$$e = \pm(e_1 \cdot B^{n-1} + e_2 \cdot B^{n-2} + \dots + e_{n-1} \cdot B + e_n) \in \mathbb{Z} \quad (1.4)$$

den **Exponenten** bezeichnen. Hierbei sind ℓ und n die jeweilige Länge der Mantisse bzw. des Exponenten. Für die Ziffern m_i , $i = 1, \dots, \ell$, und e_i , $i = 1, \dots, n$, gilt die Zusatzforderung

$$m_i, e_i \in \{0, 1, 2, \dots, B-1\}.$$

Damit ist schon eine gewisse Normierung vorgenommen worden. Da $0 \leq m_1 \leq B-1$ gefordert wurde, ist $|m|$ wegen

$$|m| = \left| \sum_{i=1}^{\ell} m_i \cdot B^{-i} \right| \leq \sum_{i=1}^{\ell} \frac{B-1}{B^i} = \sum_{i=1}^{\ell} \left(\frac{1}{B^{i-1}} - \frac{1}{B^i} \right) = 1 - \frac{1}{B^{\ell}} < 1$$

stets kleiner als 1.

Definition 1.4.1 (Gleitpunktdarstellung)

Die Darstellung einer Zahl x in der Form

$$x = \pm 0.m_1 m_2 \dots m_{\ell} \cdot B^{\pm e_1 e_2 \dots e_n} \quad (1.5)$$

heißt **Gleitpunktdarstellung von x** . x heißt dann auch **Gleitpunktzahl**.

Die Gleitpunktdarstellung der Zahl x ist jedoch noch nicht eindeutig, wie das folgende Beispiel zeigt:

$$\begin{aligned} 12345.678 &= 0.12345678 \cdot 10^5 \\ &= 0.012345678 \cdot 10^6 \\ &= 0.0012345678 \cdot 10^7 \\ &\vdots \end{aligned}$$

Der Dezimalpunkt verschiebt sich durch Multiplikation mit entsprechenden Zehnerpotenzen. Um Zahlen auf dem Computer effizient verwalten zu können, benötigt man eindeutige Zahldarstellungen.

Eine **normierte Gleitpunktdarstellung von x** entsteht durch die Zusatzforderung, daß für $x \neq 0$ die erste Ziffer m_1 der Mantisse m ungleich 0 sei. Die normierte Gleitpunktdarstellung von x ist eindeutig.

Definition 1.4.2 (Normierte Gleitpunktdarstellung)

Die **normierte Gleitpunktdarstellung der reellen Zahl $x \neq 0$** ist definiert als

$$x = \pm 0.m_1 m_2 m_3 \dots m_{\ell} \dots B^e = \pm m \cdot B^e$$

wobei $m_1 \neq 0$ gelte. Die Zahl $e = \pm e_1 e_2 \dots e_n \in \mathbb{Z}$ heißt **Exponent**, die Zahl m heißt **normierte Mantisse**.

Die Menge aller normierten Gleitpunktdarstellungen liefert die auf dem Computer darstellbaren Zahlen, die sogenannten Maschinenzahlen.

Definition 1.4.3 (Maschinenzahlen)

Die Menge

$$\mathcal{M}_{\ell,n}^B := \{x \in \mathbb{R} \mid x = \pm 0.m_1 m_2 \dots m_\ell \cdot B^{\pm e_1 e_2 \dots e_n}, m_1 \neq 0\} \cup \{0\},$$

wobei die Länge des Bruchanteils der Mantisse auf $\ell \in \mathbb{N}$ Ziffern und die Länge des Exponenten auf $n \in \mathbb{N}$ Ziffern beschränkt sind, heißt **Menge der Maschinenzahlen zur Basis B** (kurz: Maschinenzahlen).

Auf Grund der beschränkten Länge des Exponenten gibt es einen größtmöglichen Exponenten e_{\max} und einen kleinstmöglichen Exponenten e_{\min} und $e \in [e_{\min}, e_{\max}]$. Die Zahl $\mu = B^{e_{\min}-1}$ ist die kleinste positive Maschinenzahl und $\nu = B^{e_{\max}} \cdot (1 - B^{-\ell})$ ist die größte Maschinenzahl.

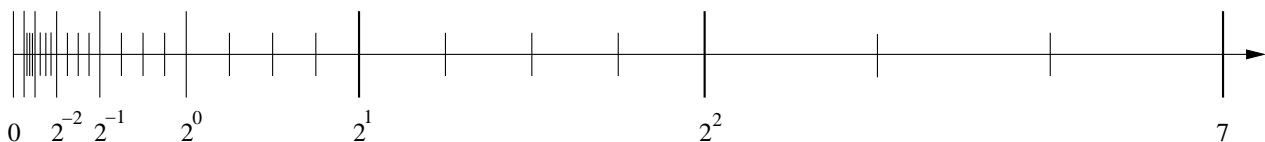
Die Maschinenzahlen sind nicht gleichmäßig verteilt, wie das folgende Beispiel zeigt.

Beispiel 1.4.4 (Verteilung der Maschinenzahlen)

Betrachte die Maschinenzahlen $\mathcal{M}_{\ell,n}^B$ mit $\ell = 3$, $n = 2$ und Basis $B = 2$. Die folgende Tabelle listet die verfügbaren Mantissen und Exponenten auf:

Mantissen	Exponenten
$(0.100)_2 = \frac{1}{2}$	$(00)_2 = 0$
$(0.101)_2 = \frac{5}{8}$	$\pm(01)_2 = \pm 1$
$(0.110)_2 = \frac{3}{4}$	$\pm(10)_2 = \pm 2$
$(0.111)_2 = \frac{7}{8}$	$\pm(11)_2 = \pm 3$

Der größtmögliche Exponent ist $e_{\max} = (11)_2 = 2^1 + 2^0 = 3$, der kleinstmögliche Exponent ist $e_{\min} = -(11)_2 = -3$. Die kleinste positive Maschinenzahl ist $\mu = 2^{-4} = \frac{1}{16}$, die größte Maschinenzahl ist $\nu = 2^3(1 - 2^{-3}) = 7$. Die Verteilung der positiven Maschinenzahlen sieht wie folgt aus:



1.5 Rundung und Auslöschung

Rundungsfehler treten zwangsläufig auf, da z.B.

$$\begin{aligned}\frac{1}{3} &= 0.3333333 \dots, \\ \pi &= 3.1415926 \dots, \\ e &= 2.7182818 \dots, \\ \sqrt{2} &= 1.4142135 \dots,\end{aligned}$$

keine Maschinenzahlen sind und somit nicht korrekt auf einem Computer dargestellt werden können.

Die Umwandlung einer gegebenen reellen Zahl x in eine Maschinenzahl wird **Rundung** genannt. Ist x bereits eine Maschinenzahl, so soll sie invariant bzgl. Rundung sein. Mathematisch ist die Rundung eine Abbildung

$$rd : \mathbb{R} \rightarrow \mathcal{M}_{\ell,n}^B, \quad x \mapsto rd(x), \quad (1.6)$$

die eine reelle Zahl

$$x = \pm 0.m_1 m_2 \dots m_k m_{k+1} \dots m_\ell m_{\ell+1} \dots \cdot B^e$$

gemäß der üblichen Vorschrift

$$rd(x) = \begin{cases} \pm 0.m_1 m_2 \dots m_\ell \cdot B^e, & \text{falls } 0 \leq m_{\ell+1} < B/2, \\ \pm 0.m_1 m_2 \dots (m_k + 1) 0 \dots 0 \cdot B^e, & \text{falls } B/2 \leq m_{\ell+1} \leq B-1, \\ & m_k \neq B-1, \\ & m_{k+1} = \dots = m_\ell = B-1, \\ \pm 0.10 \dots 0 \cdot B^{e+1}, & \text{falls } B/2 \leq m_{\ell+1} \leq B-1, \\ & m_1 = \dots = m_\ell = B-1 \end{cases}$$

auf die nächstgelegene Maschinenzahl rundet. Die bei der Rundung auftretenden Fehler können abgeschätzt werden: Es gilt

$$|rd(x) - x| \leq \frac{1}{2} \cdot B^{e-\ell}$$

für den absoluten Fehler und, falls $x \neq 0$ ist, gilt

$$\frac{|rd(x) - x|}{|x|} \leq \frac{B}{2} \cdot B^{-\ell}$$

für den relativen Fehler. Die Zahl $rd(x)$ läßt sich zudem als

$$rd(x) = x(1 + \varepsilon)$$

mit $|\varepsilon| \leq \frac{B}{2} \cdot B^{-\ell}$ darstellen. Die Schranke

$$\text{eps} = \frac{B}{2} \cdot B^{-\ell}$$

für den relativen Rundungsfehler heißt **Maschinengenauigkeit**. Sie hängt von der Computerarchitektur ab.

Es können zwei Ausnahmesituationen auftreten:

- Ein **Overflow** tritt auf, falls $rd(x) = \pm 0.m_1 \dots m_\ell \dots \cdot B^e$ mit $e > e_{max}$ dargestellt werden soll. In diesem Fall generiert der Computer üblicherweise eine Fehlermeldung und bricht das aktuelle Programm ab.
- Ein **Underflow** tritt auf, falls $rd(x) = \pm 0.m_1 \dots m_\ell \dots \cdot B^e$ mit $e < e_{min}$ dargestellt werden soll. Der Computer behandelt diesen Fall üblicherweise automatisch, indem x auf Null gesetzt wird. War $x \neq 0$, so ist der relative Fehler immer 1, während der absolute Fehler kleiner als μ und damit sehr klein ist.

Beispiel 1.5.1 (Overflow)

Wir machen einen Test mit MATLAB. Die Befehlszeile

```
X = 1000.0; for i=1:7 X=X*X, end
```

liefert die Ausgabe

```
1000000 1.0000e+12 1.0000e+24 1.0000e+48 1.0000e+96 1.0000e+192 Inf
```

Bemerkung 1.5.2 (Runden durch Abschneiden)

Ein alternativer Weg des Rundens ist das Abschneiden (chopping). Für eine Zahl $x = \pm 0.m_1 m_2 \dots m_\ell d_{\ell+1} \dots \cdot B^e$ wird $rd(x)$ definiert durch die Maschinenzahl $rd(x) = \pm 0.m_1 m_2 \dots m_\ell \cdot B^e$, d.h. die Ziffern $m_{\ell+1} \dots$ werden einfach vernachlässigt.

Absoluter Fehler:

$$|rd(x) - x| \leq B^{e-\ell}.$$

Relativer Fehler für $x \neq 0$:

$$\frac{|rd(x) - x|}{|x|} \leq B^{-\ell+1}.$$

Beachte, daß diese Fehler doppelt so groß sind wie bei der exakten Rundung.

Für Maschinenzahlen x und y werden die Grundrechenarten $+$, $-$, \cdot , $/$ durch Operationen \oplus , \ominus , \odot , \oslash gemäß

$$x \circledast y = rd(x * y), \quad * \in \{+, -, \cdot, /\}$$

realisiert. Für diese Realisierung gilt

$$x \circledast y = (x * y)(1 + \varepsilon_*), \quad |\varepsilon_*| \leq eps, \quad * \in \{+, -, \cdot, /\}.$$

Insbesondere die Subtraktion zweier Zahlen x und y kann kritisch sein, wenn $x \approx y$. Dies kann wie folgt eingesehen werden:

Der Ansatz

$$z = rd(rd(x) + rd(y)) = (x(1 + \varepsilon_1) + y(1 + \varepsilon_2))(1 + \varepsilon_3)$$

liefert in erster Näherung (d.h. durch Vernachlässigung der Fehlerterme höherer Ordnung in der Taylorentwicklung)

$$\frac{|(x + y) - z|}{|(x + y)|} \doteq \left| \varepsilon_3 + \frac{x\varepsilon_1 + y\varepsilon_2}{(x + y)} \right|.$$

Speziell für $x \approx -y$ ist der relative Fehler im schlimmsten Fall unbeschränkt. Diese Beobachtung ist der Grund für schwerwiegende numerische Probleme, die in Berechnungen auftreten können, da ein Fehler in einem Zwischenergebnis durch weitere Rechenschritte vergrößert werden kann und der Gesamtfehler sich somit aufschauelt. Dies kann zu völlig unbrauchbaren Ergebnissen führen. Dieses Phänomen heißt **Auslöschung**.

Beispiel 1.5.3

Seien $x = 0.3721448693$ und $y = 0.3720214371$ gegeben. Angenommen, unser Computer besitzt 5 Stellen Genauigkeit. Dann gelten $rd(x) = 0.37214$ und $rd(y) = 0.37202$. Subtraktion liefert $z = rd(rd(x) - rd(y)) = rd(0.37214 - 0.37202) = 0.00012$. Das korrekte Resultat ist $x - y = 0.0001234322$. Der relative Fehler beträgt somit

$$\frac{|(x - y) - z|}{|(x - y)|} = \frac{0.0000034322}{0.0001234322} \approx 3 \cdot 10^{-2}.$$

Dieser Fehler ist sehr groß im Vergleich zu den relativen Fehlern von $rd(x)$ und $rd(y)$, die jeweils durch $0.5 \cdot 10^{-4}$ beschränkt sind. Durch Subtraktion zweier in etwa gleich großer Zahlen haben wir also 3 Stellen an Genauigkeit verloren!

Gelegentlich ist es möglich, den Verlust an Genauigkeit zu vermeiden, indem alternative, analytisch äquivalente Rechenvorschriften verwendet werden.

Beispiel 1.5.4

Betrachte die Funktion $f(x) = \sqrt{x^2 + 1} - 1$. Für $x \approx 0$ entsteht bei der Subtraktion ein Verlust an Genauigkeit, da dann $\sqrt{x^2 + 1} \approx 1$ gilt. Jedoch kann f dargestellt werden als

$$f(x) = \left(\sqrt{x^2 + 1} - 1 \right) \cdot \frac{\sqrt{x^2 + 1} + 1}{\sqrt{x^2 + 1} + 1} = \frac{x^2}{\sqrt{x^2 + 1} + 1}.$$

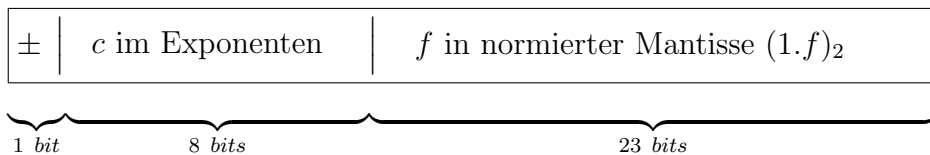
Wenn wir wiederum mit 5 Stellen Genauigkeit rechnen und $x = 10^{-3}$ wählen, dann wird $f(x)$ gemäß der ersten Formel fälschlicherweise zu 0 berechnet, während die zweite Formel

$0.5 \cdot 10^{-6}$ liefert. Der exakte Wert ist $0.4999998750000625 \cdot 10^{-6}$. Dies führt auf einen relativen Fehler von 1 für die erste Formel bzw. von ungefähr $0.25 \cdot 10^{-6}$ für die zweite Formel.

Der IEEE Standard

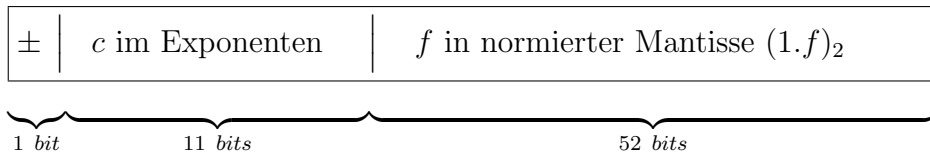
Die Zahldarstellung auf Computern orientiert sich in der Regel am IEEE Standard. Dieser Standard besitzt folgende Merkmale:

- Binärsystem, korrekte Rundung
- größte ganze Zahl: $2^{31} - 1 = 2147483647$
- Gleitpunktdarstellung in einfacher Genauigkeit (single precision) $x = \pm(1.f)_2 \cdot 2^{c-127}$



Es gilt $0 < c < (11111111)_2 = 255$ und $-127 < c - 127 < 128$. Die Werte $c = 0$ und $c = 255$ sind reserviert für die Darstellung von u.a. ± 0 und $\pm \infty$.

- Gleitpunktdarstellung in doppelter Genauigkeit (double precision) $x = \pm(1.f)_2 \cdot 2^{c-1023}$



Es gilt $0 < c < (11111111111)_2 = 2047$ und $-1023 < c - 1023 < 1024$. Die Werte $c = 0$ und $c = 2047$ sind reserviert für die Darstellung von u.a. ± 0 und $\pm \infty$.

	Single Precision	Double Precision
Länge Exponent	8 Bits	11 Bits
Länge Mantisse	23 Bits	52 Bits
Maschinengenauigkeit	$2^{-23} \approx 1.192 \cdot 10^{-7}$	$2^{-52} \approx 2.220 \cdot 10^{-16}$
kleinste positive Zahl	$2^{-126} \approx 1.175 \cdot 10^{-38}$	$2^{-1022} \approx 2.225 \cdot 10^{-308}$
größte Zahl	$(2 - 2^{-23})2^{127} \approx 3.403 \cdot 10^{38}$	$(2 - 2^{-52})2^{1023} \approx 1.798 \cdot 10^{308}$

1.6 Algorithmen, Kondition und Stabilität

Es ist wichtig zwischen folgenden Begriffen zu unterscheiden:

- **(Mathematisches) Problem**, das wir lösen möchten (Lösung einer linearen oder nichtlinearen Gleichung, Approximation einer Funktion, ...)

- **(Numerischer) Algorithmus** (oder **Prozedur**), der angewendet werden kann, um ein gegebenes Problem zu lösen (es kann verschiedene Algorithmen zur Lösung desselben Problems geben)

Ein **Problem** heißt

- **gut konditioniert**, falls kleine Störungen in den Problemdata zu kleinen Änderungen in den Lösungen führen.
- **schlecht konditioniert**, falls kleine Störungen in den Problemdata zu großen Änderungen in den Lösungen führen können.

Entsprechend heißt ein **(numerischer) Algorithmus**

- **vorwärtsstabil**, falls kleine Fehler in den Rechenschritten des Algorithmus zu kleinen Fehlern im Ergebnis führen.
- **instabil**, falls kleine Fehler in den Rechenschritten des Algorithmus zu großen Fehlern im Ergebnis führen können.

Natürlich ist man daran interessiert, stabile Algorithmen zu entwickeln.

Wir formalisieren die Begriffe.

Definition 1.6.1 (mathematisches Problem)

Unter einem **mathematischen Problem** verstehen wir die Aufgabe, zu einem Eingabedatum x eine Lösung $y = \mathcal{A}(x)$ zu ermitteln. Darin ist \mathcal{A} eine Abbildung

$$\mathcal{A} : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad x \mapsto y = \mathcal{A}(x).$$

Es stellt sich die Frage, wie sich Fehler Δx im Eingabedatum x auf das Ergebnis y auswirken.

Definition 1.6.2 (Kondition)

Wir nennen das mathematische Problem \mathcal{A} **gut konditioniert**, falls kleine Fehler in x zu kleinen Abweichungen in y führen. Andernfalls heißt es **schlecht konditioniert**.

Wir wollen den Begriff der Kondition genauer untersuchen und setzen voraus, daß die Abbildung \mathcal{A} zweimal stetig differenzierbar ist. Taylorentwicklung bis zum linearen Glied und Vernachlässigung der Terme höherer Ordnung liefert die Darstellung

$$\|\Delta y\| = \|\mathcal{A}(x + \Delta x) - \mathcal{A}(x)\| \approx \left\| \frac{\partial \mathcal{A}}{\partial x}(x) \cdot \Delta x \right\| \leq \left\| \frac{\partial \mathcal{A}}{\partial x}(x) \right\| \cdot \|\Delta x\| \quad (1.7)$$

für passende Normen.

Definition 1.6.3 (absolute und relative Kondition)

(a) Die Größe

$$\kappa_a(x) = \left\| \frac{\partial \mathcal{A}}{\partial x}(x) \right\| \quad (1.8)$$

heißt **absolute Kondition** des Problems \mathcal{A} in x .

(b) Für $0 \neq y = \mathcal{A}(x)$ heißt die Zahl

$$\kappa_r(x) = \kappa_a(x) \cdot \frac{\|x\|}{\|y\|} = \left\| \frac{\partial \mathcal{A}}{\partial x}(x) \right\| \cdot \frac{\|x\|}{\|y\|}$$

relative Kondition des Problems \mathcal{A} in x .

Für $y \neq 0$ hat der relative Fehler häufig eine höhere Aussagekraft als der absolute Fehler. Es gilt

$$\frac{\|\Delta y\|}{\|y\|} \leq \kappa_a(x) \cdot \frac{\|\Delta x\|}{\|y\|} = \kappa_a(x) \cdot \frac{\|x\|}{\|y\|} \cdot \frac{\|\Delta x\|}{\|x\|} = \kappa_r(x) \cdot \frac{\|\Delta x\|}{\|x\|}.$$

An Hand der absoluten und relativen Kondition des Problems läßt sich ablesen, wie stark sich eine Störung in x auf den absoluten bzw. relativen Fehler in y auswirkt. Damit läßt sich der Begriff Kondition auch folgendermaßen formulieren, abhängig davon, ob man an dem absoluten oder relativen Fehler interessiert ist:

Das mathematische Problem \mathcal{A} heißt **gut konditioniert**, falls κ_a (bzw. κ_r) klein ist (als Funktion von x).

Man beachte, daß die Kondition lediglich eine Aussage über die **Sensitivität** des mathematischen Problems macht. Rundungs- und Verfahrensfehler, die während der Lösung des Problems auf dem Rechner entstehen, sind nicht berücksichtigt. Um auch diese Fehler zu erfassen, müssen die einzelnen Rechenschritte, die zur Lösung des Problems durchgeführt werden, genauer spezifiziert werden.

Definition 1.6.4 (numerischer Algorithmus)

Eine konkrete Vorschrift, die die Abbildung \mathcal{A} auf dem Rechner realisiert, wird **numerischer Algorithmus** (kurz: *Algorithmus*) genannt. Das Endergebnis $y = \mathcal{A}(x)$ des numerischen Algorithmus wird darin über mehrere Rechenschritte r mit Zwischenergebnissen y_i , $i = 1, \dots, r$, gemäß

$$\begin{aligned} y_1 &= \mathcal{A}_1(x), \\ y_2 &= \mathcal{A}_2(y_1) = \mathcal{A}_2(\mathcal{A}_1(x)), \\ &\vdots \\ y &= y_r = \mathcal{A}_r(y_{r-1}) = \mathcal{A}_r(\mathcal{A}_{r-1}(\cdots(\mathcal{A}_1(x))\cdots)) \end{aligned}$$

erhalten. Der numerische Algorithmus \mathcal{A} läßt sich damit auch als Verkettung

$$\mathcal{A} = \mathcal{A}_r \circ \mathcal{A}_{r-1} \circ \cdots \circ \mathcal{A}_1.$$

der Rechenschritte schreiben.

Im Allgemeinen kann in jedem Schritt des numerischen Algorithmus ein Rechenfehler bedingt durch Rundung auftreten. Die Zwischenergebnisse und auch das Eingabedatum x sind also verfälscht, etwa gemäß

$$\begin{aligned}\tilde{y}_0 &= x \cdot (\mathbf{I} + \text{diag}(\varepsilon_0)), \\ \tilde{y}_1 &= \mathcal{A}_1(\tilde{y}_0) \cdot (\mathbf{I} + \text{diag}(\varepsilon_1)), \\ \tilde{y}_2 &= \mathcal{A}_2(\tilde{y}_1) \cdot (\mathbf{I} + \text{diag}(\varepsilon_2)), \\ &\vdots \\ \tilde{y} = \tilde{y}_r &= \mathcal{A}_r(\tilde{y}_{r-1}) \cdot (\mathbf{I} + \text{diag}(\varepsilon_r)).\end{aligned}$$

Die Vektoren ε_i , $i = 0, \dots, r$, beschreiben die in der Realisierung auftretenden relativen Rundungsfehler und sind komponentenweise beschränkt durch eps . Wir verwenden die Definition

$$\text{diag}(v) = \begin{pmatrix} v_1 & & \\ & \ddots & \\ & & v_n \end{pmatrix}$$

für den Vektor $v = (v_1, \dots, v_n)^\top$.

Das verfälschte Ergebnis \tilde{y} kann als Funktion der relativen Rundungsfehler aufgefaßt werden, d.h. es ist

$$\tilde{y} = \tilde{y}(x, \varepsilon_0, \varepsilon_1, \dots, \varepsilon_r).$$

Man beachte, daß das exakte Ergebnis y für $\varepsilon_i = 0$, $i = 0, \dots, r$, angenommen wird. Andererseits gelten auch die Interpretationen

$$\tilde{y}_i = \tilde{y}_i(\tilde{y}_{i-1}, \varepsilon_i), \quad i = 1, \dots, r, \quad \tilde{y}_0 = \tilde{y}_0(x, \varepsilon_0).$$

Mit Hilfe der Taylorentwicklung unter Vernachlässigung der Terme höherer Ordnung folgt die Darstellung

$$\begin{aligned}
\tilde{y}(x, \varepsilon_0, \dots, \varepsilon_r) - \tilde{y}(x, 0, \dots, 0) &\approx \sum_{i=0}^r \frac{\partial \tilde{y}_r}{\partial \varepsilon_i}(x, 0, \dots, 0) \cdot \varepsilon_i \\
&= \frac{\partial \tilde{y}_r}{\partial \varepsilon_r}(x, 0, \dots, 0) \cdot \varepsilon_r \\
&\quad + \sum_{i=1}^{r-1} \left(\frac{\partial \tilde{y}_r}{\partial \tilde{y}_{r-1}} \cdot \frac{\partial \tilde{y}_{r-1}}{\partial \tilde{y}_{r-2}} \cdots \frac{\partial \tilde{y}_{i+1}}{\partial \tilde{y}_i} \cdot \frac{\partial \tilde{y}_i}{\partial \varepsilon_i} \right) (x, 0, \dots, 0) \cdot \varepsilon_i \\
&\quad + \left(\frac{\partial \tilde{y}_r}{\partial \tilde{y}_{r-1}} \cdots \frac{\partial \tilde{y}_1}{\partial \tilde{y}_0} \cdot \frac{\partial \tilde{y}_0}{\partial \varepsilon_0} \right) (x, 0, \dots, 0) \cdot \varepsilon_0 \\
&= \text{diag}(\mathcal{A}_r(y_{r-1})) \cdot \varepsilon_r \\
&\quad + \sum_{i=1}^{r-1} \frac{\partial \mathcal{A}_r}{\partial y_{r-1}}(y_{r-1}) \cdots \frac{\partial \mathcal{A}_{i+1}}{\partial y_i}(y_i) \cdot \text{diag}(\mathcal{A}_i(y_{i-1})) \cdot \varepsilon_i \\
&\quad + \frac{\partial \mathcal{A}_r}{\partial y_{r-1}}(y_{r-1}) \cdots \frac{\partial \mathcal{A}_1}{\partial y_0}(y_0) \cdot \text{diag}(x) \cdot \varepsilon_0
\end{aligned}$$

für den absoluten Fehler. Darin bezeichnen y_i , $i = 0, \dots, r$, die exakten Zwischenergebnisse. Definiert man die absoluten Konditionszahlen für die Berechnung des Zwischenergebnisses y_i analog zu (1.8) durch

$$\kappa_a^i(y_{i-1}) = \left\| \frac{\partial \mathcal{A}_i}{\partial y_{i-1}}(y_{i-1}) \right\|,$$

erhält man die Fehlerabschätzung

$$\begin{aligned}
\|\tilde{y} - y\| &\leq \|\text{diag}(\mathcal{A}_r(y_{r-1})) \cdot \varepsilon_r\| \\
&\quad + \sum_{i=1}^{r-1} \kappa_a^r(y_{r-1}) \cdots \kappa_a^{i+1}(y_i) \cdot \|\text{diag}(\mathcal{A}_i(y_{i-1})) \cdot \varepsilon_i\| \\
&\quad + \kappa_a^r(y_{r-1}) \cdots \kappa_a^1(y_0) \cdot \|\text{diag}(x) \cdot \varepsilon_0\| \\
&\leq \text{eps} \left(\|\mathcal{A}_r(y_{r-1})\| \right. \\
&\quad + \sum_{i=1}^{r-1} \kappa_a^r(y_{r-1}) \cdots \kappa_a^{i+1}(y_i) \cdot \|\mathcal{A}_i(y_{i-1})\| \\
&\quad \left. + \kappa_a^r(y_{r-1}) \cdots \kappa_a^1(y_0) \cdot \|x\| \right).
\end{aligned}$$

Damit läßt sich der Begriff Vorwärtsstabilität auch folgendermaßen formulieren:

Der numerische Algorithmus \mathcal{A} heißt **vorwärtsstabil**, falls die Verstärkungsfaktoren $\kappa_a^r(y_{r-1}) \cdots \kappa_a^{i+1}(y_i)$, $i = 0, \dots, r$, klein sind.

1.6.1 Rückwärtsstabilität

Die Untersuchung der (relativen) Konditionszahlen in der Vorwärtsanalyse eines Problems führt häufig zu einer Überschätzung des Fehlers, da in jedem Schritt der schlechtestmögliche Fall betrachtet wird.

Neben der Vorwärtsstabilität, die eine Aussage über die Sensitivität eines Algorithmus (oder Problems) macht, gibt es noch den Begriff der Rückwärtsstabilität. Hierbei werden Störungen im Ergebnis des Algorithmus (oder Problems) betrachtet und dazu passende Argumente betrachtet. Genauer gilt folgende Definition.

Definition 1.6.5 (Rückwärtsstabilität)

Gegeben sei der numerische Algorithmus $y = \mathcal{A}(x)$. Der Algorithmus heißt **rückwärtsstabil in y_*** , wenn es (von y_* abhängige) Konstanten $R > 0$ und $S > 0$ gibt, so daß die folgende Beziehung gilt:

$$\forall x^1, x^2 : \|\mathcal{A}(x^i) - y_*\| \leq R, i = 1, 2 \quad \Rightarrow \quad \|x^1 - x^2\| \leq S \|\mathcal{A}(x^1) - \mathcal{A}(x^2)\|.$$

Beispiel 1.6.6

Die Funktionsauswertung

$$\mathcal{A}(x) = \begin{cases} (x+1)^2, & x < -1, \\ 0, & -1 \leq x \leq 1, \\ (x-1)^2, & x > 1 \end{cases}$$

ist vorwärtsstabil in $x = 0$ mit der absoluten Konditionszahl

$$\kappa_a = |\mathcal{A}'(0)| = 0.$$

Sie ist nicht rückwärtsstabil in $y_* = 0$, da in der Nähe des Funktionswerts $0 = \mathcal{A}(0)$ für kleine Differenzen in den Funktionswerten nicht auf den Abstand der Argumente geschlossen werden kann. Dies sieht man wie folgt. Sei $R > 0$ beliebig. Wähle $x^1 = -1 - \sqrt{R} < -1$ und $x^2 = 1 + \sqrt{R} > 1$. Dann gilt $|\mathcal{A}(x^i) - 0| = R$ für $i = 1, 2$, aber $|x^1 - x^2| \geq 2$.

Kapitel 2

Numerische Lösung linearer Gleichungssysteme

Das Folgende gilt sowohl für reellwertige als auch für komplexwertige Vektoren und Matrizen. Daher verwenden wir das Symbol \mathbb{K} , welches für \mathbb{R} oder \mathbb{C} steht.

Gesucht ist die Lösung $x = (x_1, \dots, x_n)^\top \in \mathbb{K}^n$ des linearen Gleichungssystems

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2, \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n, \end{aligned} \quad (2.1)$$

wobei a_{ij} , $i, j = 1, \dots, n$, und b_i , $i = 1, \dots, n$, gegebene Zahlen sind. Mit

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad (2.2)$$

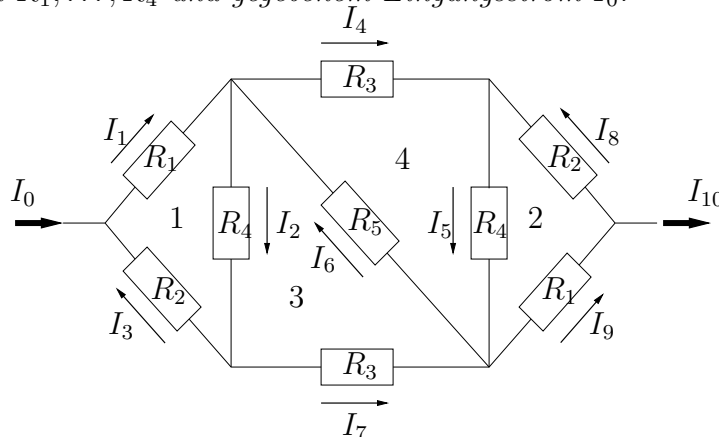
lautet das lineare Gleichungssystem in Matrixschreibweise

$$Ax = b \quad (A \in \mathbb{K}^{n \times n}, \quad x, b \in \mathbb{K}^n). \quad (2.3)$$

Lineare Gleichungssysteme treten in nahezu allen Anwendungen und Verfahren auf.

Beispiel 2.0.1 (Elektrische Schaltkreise, Kirchhoffsches Gesetz und lineare Gleichungssysteme)

Gegeben sei das folgende stationäre elektronische Netzwerk mit gegebenen ohmschen Widerständen R_1, \dots, R_4 und gegebenem Eingangsstrom I_0 .



Das erste Kirchhoffsche Gesetz (Knotensatz)

An jedem Knotenpunkt ist die Summe aller zu- (positiven) und abfließenden (negativen) Ströme unter Beachtung der durch die Pfeile angegebenen Richtungen in jedem Zeitpunkt gleich Null

liefert die Beziehungen

$$\begin{aligned} -I_1 + I_3 &= -I_0, \\ I_1 - I_2 - I_4 + I_6 &= 0, \\ I_2 - I_3 - I_7 &= 0, \\ I_4 - I_5 + I_8 &= 0, \\ I_5 - I_6 + I_7 - I_9 &= 0, \\ -I_8 + I_9 - I_{10} &= 0. \end{aligned}$$

Aus dem Ohmschen Gesetz

$$U = R \cdot I$$

und dem zweiten Kirchhoffschen Gesetz (Maschensatz)

In einer Masche ist die Summe aller Teilspannungen in jedem Zeitpunkt gleich Null

folgen die Gleichungen

$$\begin{aligned} R_1 \cdot I_1 + R_4 \cdot I_2 + R_2 \cdot I_3 &= 0, \\ R_2 \cdot I_8 + R_4 \cdot I_5 + R_1 \cdot I_9 &= 0, \\ R_4 \cdot I_2 + R_5 \cdot I_6 + R_3 \cdot I_7 &= 0, \\ R_3 \cdot I_4 + R_4 \cdot I_5 + R_5 \cdot I_6 &= 0. \end{aligned}$$

Diese 10 Gleichungen für die Ströme I_1, \dots, I_{10} können als lineares Gleichungssystem geschrieben werden:

$$\begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 \\ R_1 & R_4 & R_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_4 & 0 & 0 & R_2 & R_1 & 0 \\ 0 & R_4 & 0 & 0 & 0 & R_5 & R_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & R_3 & R_4 & R_5 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \\ I_8 \\ I_9 \\ I_{10} \end{pmatrix} = \begin{pmatrix} -I_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Unser Ziel ist es, Verfahren zur Bestimmung einer Lösung x von (2.3) zu entwickeln und zu analysieren.

2.1 Grundlagen und Lösbarkeit

Im folgenden werden wir verschiedene Typen von Matrizen und einige Eigenschaften verwenden. Sei $A \in \mathbb{K}^{n \times n}$ eine Matrix. Sie heißt

- **Diagonalmatrix**, falls $a_{ij} = 0$ für alle $i \neq j$ gilt, d.h.

$$A = \begin{pmatrix} * & & \\ & \ddots & \\ & & * \end{pmatrix}$$

- **Tridiagonalmatrix**, falls $a_{ij} = 0$ für alle $i, j \in \{1, \dots, n\}$ mit $|i - j| > 1$ gilt, d.h.

$$A = \begin{pmatrix} * & * & & \\ * & * & * & \\ & * & * & \ddots \\ & & \ddots & \ddots & * \\ & & & * & * \end{pmatrix}$$

- **untere Dreiecksmatrix**, falls $a_{ij} = 0$ für alle $i < j$ gilt, d.h.

$$A = \begin{pmatrix} * & & & \\ * & * & & \\ \vdots & \vdots & \ddots & \\ * & * & \dots & * \end{pmatrix}$$

- **obere Dreiecksmatrix**, falls $a_{ij} = 0$ für alle $i > j$ gilt, d.h.

$$A = \begin{pmatrix} * & * & \dots & * \\ & * & \dots & * \\ & & \ddots & \vdots \\ & & & * \end{pmatrix}$$

- **normierte untere (obere) Dreiecksmatrix**, falls A eine untere (obere) Dreiecksmatrix ist, deren Diagonaleinträge alle gleich Eins sind.

Desweiteren heißt $A \in K^{n \times n}$

- **hermitesch**, wenn $A^* = A$ mit $A^* = \bar{A}^\top$ gilt, wobei A^\top die transponierte Matrix von A bezeichnet und \bar{A} die konjugiert komplexe Matrix von A ist.
- **symmetrisch**, wenn $\mathbb{K} = \mathbb{R}$ und $A^\top = A$ gilt.
- **unitär**, falls $A^*A = I$ gilt.
- **orthogonal**, falls $\mathbb{K} = \mathbb{R}$ und $A^\top A = I$ gilt.
- **positiv semidefinit**, falls $x^*Ax \geq 0$ für alle $x \in \mathbb{K}^n$ gilt.
- **positiv definit**, falls $x^*Ax > 0$ für alle $x \in \mathbb{K}^n$, $x \neq 0$, gilt.

Definition 2.1.1

Sei $A \in \mathbb{K}^{m \times n}$ eine Matrix. Dann heißen

$$\begin{aligned} \text{Kern}(A) &:= \{x \in \mathbb{K}^n \mid Ax = 0\} && \text{Kern von } A, \\ \text{Bild}(A) &:= \{y \in \mathbb{K}^m \mid \exists x \in \mathbb{K}^n : y = Ax\} && \text{Bild von } A, \\ \text{Rang}(A) &:= \dim(\text{Bild}(A)) && \text{Rang von } A. \end{aligned}$$

$\text{Kern}(A)$ ist ein Unterraum des \mathbb{K}^n und $\text{Bild}(A)$ ist ein Unterraum von \mathbb{K}^m . $\text{Rang}(A)$ ist die maximale Anzahl von linear unabhängigen Spalten von A , was gleich der Anzahl der linear unabhängigen Zeilen von A ist.

Es stellt sich die Frage, welche Voraussetzungen an A und b gestellt werden müssen, damit das lineare Gleichungssystem (2.3) überhaupt eine Lösung x besitzt. Zunächst ist klar, daß das lineare Gleichungssystem **genau eine Lösung** besitzt, falls die Matrix A invertierbar (regulär) ist. Die eindeutige Lösung des Gleichungssystems ist dann durch

$$x = A^{-1}b \tag{2.4}$$

gegeben. Es kann aber auch der Fall eintreten, daß (2.3) mehrere Lösungen oder gar keine Lösung besitzt. Offensichtlich kann die Matrix A dann nicht regulär sein.

Beispiel 2.1.2

Das folgende lineare Gleichungssystem besitzt keine Lösung:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Das folgende lineare Gleichungssystem besitzt genau eine Lösung:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Das folgende lineare Gleichungssystem besitzt unendlich viele Lösungen:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Es gilt das folgende Kriterium, welches die Frage der Lösbarkeit von (2.3) abschließend beantwortet.

Satz 2.1.3

Das lineare Gleichungssystem (2.3) besitzt genau dann **mindestens eine Lösung**, falls gilt:

$$\text{Rang}(A) = \text{Rang}(A|b).$$

Unter $(A|b)$ wird die um die rechte Seite b erweiterte Matrix

$$(A|b) = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array} \right)$$

verstanden.

Beweis: Offenbar besitzt das lineare Gleichungssystem genau dann eine Lösung, wenn $b \in \text{Bild}(A)$ gilt.

Sei nun $b \in \text{Bild}(A)$. Dann gilt auch $\text{Bild}(A) = \text{Bild}(A|b)$ und somit $\text{Rang}(A) = \text{Rang}(A|b)$. Gilt nun $\text{Rang}(A) = \text{Rang}(A|b)$, so läßt sich b als Linearkombination der Spalten von A schreiben. Also gilt $b \in \text{Bild}(A)$ und somit besitzt das Gleichungssystem eine Lösung. \square

Die Struktur der Lösungsmenge eines linearen Gleichungssystems wird im folgenden Satz behandelt.

Satz 2.1.4

Seien $A \in \mathbb{K}^{n \times n}$ und $b \in \mathbb{K}^n$ gegeben. Sei \hat{x} eine Lösung von $Ax = b$. Die gesamte Lösungsmenge von $Ax = b$ ist gegeben durch $\hat{x} + \text{Kern}(A)$.

Beweis: Sei $y \in \text{Kern}(A)$, d.h. $Ay = 0$. Dann gilt $A(\hat{x} + y) = A\hat{x} + Ay = b$ und $\hat{x} + y$ ist Lösung.

Sei x Lösung. Dann gilt $A(x - \hat{x}) = b - b = 0$ und somit $x - \hat{x} \in \text{Kern}(A)$ bzw. $x \in \hat{x} + \text{Kern}(A)$. \square

Wir fassen für $A \in \mathbb{K}^{m \times n}$ noch einige aus der linearen Algebra bekannte Resultate zusammen:

- Für $A \in \mathbb{K}^{m \times n}$ gelten

$$n = \dim(\text{Kern}(A)) + \dim(\text{Bild}(A))$$

$$m = \dim(\text{Kern}(A^\top)) + \text{Rang}(A)$$

- Sind $A \in \mathbb{K}^{n \times n}$ und $B \in \mathbb{K}^{n \times n}$ invertierbar, so gelten

$$(A \cdot B)^{-1} = B^{-1} \cdot A^{-1} \quad \text{und} \quad (A^*)^{-1} = (A^{-1})^*.$$

- Sei $A \in \mathbb{K}^{n \times n}$. Die folgenden Aussagen sind äquivalent:

- A ist regulär (invertierbar)
- $\text{Kern}(A) = \{0\}$
- $\text{Bild}(A) = \mathbb{K}^n$
- $\text{Rang}(A) = n$
- $Ax = 0 \Leftrightarrow x = 0$
- $Ax = b$ hat für jedes b genau eine Lösung
- $\det(A) \neq 0$
- alle Eigenwerte von A sind von Null verschieden

- Für $A \in \mathbb{K}^{m \times n}$ gelten die folgenden Beziehungen

- $\text{Kern}(A) = \text{Bild}(A^*)^\perp$ und $\text{Bild}(A^*) = \text{Kern}(A)^\perp$
- $\text{Kern}(A) = \text{Kern}(A^*A)$
- $\text{Bild}(A^*) = \text{Bild}(A^*A)$

Hierin bezeichnet $V^\perp = \{w \in \mathbb{K}^n \mid w^*v = 0 \, \forall v \in V\}$ das orthogonale Komplement des Unterraums $V \in \mathbb{K}^n$.

Um das lineare Gleichungssystem numerisch lösen zu können, gehen wir im folgenden davon aus, daß die Matrix A regulär ist und somit genau eine Lösung existiert. Die numerische Berechnung einer nicht eindeutigen Lösung im Falle einer singulären Matrix A ist komplizierter und wird hier nicht betrachtet.

2.2 Gauss'sches Eliminationsverfahren und LR-Zerlegung

Es werden Algorithmen zur numerischen Lösung des linearen Gleichungssystems (2.3) entwickelt. Es wird vorausgesetzt, daß die Matrix A regulär ist. Formal kennen wir die Lösung dann schon, sie ist durch (2.4) gegeben. Dort wird aber die Inverse von A benötigt, die i.a. nicht bekannt ist und für große Matrizen auch nicht in einfacher Weise berechnet werden kann.

2.2.1 Einfach zu lösende Gleichungssysteme

Zunächst widmen wir uns linearen Gleichungssystemen, die „einfach“ lösbar sind.

Wir betrachten insbesondere lineare Gleichungssysteme, in denen die Matrix A Dreiecksgestalt hat. Diese Gleichungssysteme werden später im Gauss'schen Eliminationsverfahren bzw. der LR-Zerlegung eine wichtige Rolle spielen.

Zunächst betrachten wir das lineare Gleichungssystem

$$Ly = b \quad (2.5)$$

für $y \in \mathbb{K}^n$ mit einem Vektor $b \in \mathbb{K}^n$ und einer linken unteren normierten Dreiecksmatrix

$$L = \begin{pmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ \ell_{31} & \ell_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{n,n-1} & 1 \end{pmatrix} \in \mathbb{K}^{n \times n}.$$

In Komponenten lautet es

$$\begin{aligned} y_1 &= b_1, \\ \ell_{21}y_1 + y_2 &= b_2, \\ \ell_{31}y_1 + \ell_{32}y_2 + y_3 &= b_3, \\ &\vdots \\ \ell_{n1}y_1 + \ell_{n2}y_2 + \cdots + \ell_{n,n-1}y_{n-1} + y_n &= b_n. \end{aligned}$$

Dieses Gleichungssystem können wir direkt lösen, da die Gleichungen von oben beginnend sukzessive nach den Variablen y_i , $i = 1, \dots, n$, aufgelöst werden können:

$$\begin{aligned} y_1 &= b_1, \\ y_2 &= b_2 - \ell_{21}y_1, \\ y_3 &= b_3 - \ell_{31}y_1 - \ell_{32}y_2, \\ &\vdots \\ y_n &= b_n - \ell_{n1}y_1 - \ell_{n2}y_2 - \cdots - \ell_{n,n-1}y_{n-1} \end{aligned}$$

bzw.

$$y_i = b_i - \sum_{j=1}^{i-1} \ell_{ij}y_j, \quad i = 1, \dots, n. \quad (2.6)$$

Bei der Berechnung von y_i werden nur die bereits bekannten Werte y_j , $j = 1, \dots, i-1$, verwendet. Diesen Vorgang nennt man **Vorwärtssubstitution**.

Algorithmus 2.2.1 (Vorwärtssubstitution)

Input: untere Dreiecksmatrix L , rechte Seite b

Output: Lösung y von $Ly = b$.

$y(1) = b(1);$

for $i = 2:n$,

$y(i) = b(i);$

for $j = 1:(i-1)$,

$y(i) = y(i) - L(i,j) * y(j);$

end

end

Der Aufwand der Vorwärtssubstitution beträgt offenbar $\mathcal{O}(n^2)$.

Entsprechend kann man auch bei der Lösung des linearen Gleichungssystems

$$Rx = c \tag{2.7}$$

für x mit rechter oberer Dreiecksmatrix

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ & r_{22} & r_{23} & \cdots & r_{2n} \\ & & r_{33} & \ddots & \vdots \\ & & & \ddots & r_{n-1,n} \\ & & & & r_{nn} \end{pmatrix}, \quad r_{ii} \neq 0, \quad i = 1, \dots, n,$$

vorgehen. In diesem Fall kann man die Gleichungen

$$\begin{aligned} r_{11}x_1 + r_{12}x_2 + \dots + r_{1n}x_n &= c_1, \\ r_{22}x_2 + \dots + r_{2n}x_n &= c_2, \\ &\vdots \\ r_{n-1,n-1}x_{n-1} + r_{n-1,n}x_n &= c_{n-1}, \\ r_{nn}x_n &= c_n \end{aligned}$$

beginnend mit der letzten Zeile nach x_i , $i = n, n-1, \dots, 1$, auflösen:

$$\begin{aligned} x_1 &= \frac{1}{r_{11}} (c_1 - r_{12}x_2 - r_{13}x_3 - \dots - r_{1n}x_n), \\ x_2 &= \frac{1}{r_{22}} (c_2 - r_{23}x_3 - \dots - r_{2n}x_n), \\ &\vdots \\ x_{n-1} &= \frac{1}{r_{n-1,n-1}} (c_{n-1} - r_{n-1,n}x_n), \\ x_n &= \frac{c_n}{r_{nn}} \end{aligned}$$

bzw.

$$x_i = \frac{1}{r_{ii}} \left(c_i - \sum_{j=i+1}^n r_{ij}x_j \right), \quad i = n, n-1, \dots, 1.$$

Hier werden die Gleichungen „rückwärts“ aufgelöst, daher heißt der Vorgang auch **Rückwärtssubstitution**. Zu beachten ist, daß die Diagonalelemente von R nicht 0 sein dürfen, da in diesem Fall bei der Rückwärtssubstitution durch 0 geteilt werden würde.

Algorithmus 2.2.2 (Rückwärtssubstitution)

Input: obere Dreiecksmatrix R , rechte Seite c

Output: Lösung x von $Rx = c$.

$x(n) = c(n) / R(n,n);$

for $i = (n-1):-1:1,$

$x(i) = c(i);$

for $j = (i+1):n,$

$x(i) = x(i) - R(i,j) * x(j);$

end

$x(i) = x(i) / R(i,i);$

end

Der Aufwand der Rückwärtssubstitution beträgt wiederum $\mathcal{O}(n^2)$.

2.2.2 Das Gauss'sche Eliminationsverfahren

Wir haben im vorhergehenden Abschnitt gesehen, daß lineare Gleichungssysteme mit Dreiecksstruktur direkt gelöst werden können. Die Idee des Gauss'schen Eliminationsverfahrens besteht darin, die Matrix A durch elementare Zeilenumformungen schrittweise in eine rechte obere Dreiecksmatrix R und die rechte Seite b in einen Vektor c zu überführen, so daß ein Gleichungssystem der Form (2.7) entsteht. Dieses kann dann mittels Rückwärtssubstitution gelöst werden. Wichtig ist hierbei, daß die Lösung des Ausgangsproblems mit der des transformierten Problems übereinstimmt. Dies ist bei der Verwendung elementarer Zeilenumformungen gewährleistet. Elementare Zeilenumformungen sind

- die Multiplikation einer Zeile mit einem Wert ungleich Null,
- die Addition bzw. Subtraktion zweier Zeilen,
- sowie das Vertauschen zweier Zeilen.

Schematisch läuft der Gauss'sche Eliminationsalgorithmus wie folgt ab.

Gauss'scher Eliminationsalgorithmus:

(i) Setze $A^{(1)} := A$ und $b^{(1)} := b$.

(ii) Beginnend mit $A^{(1)}x = b^{(1)}$ führe $n - 1$ Transformationsschritte durch bis ein äquivalentes lineares Gleichungssystem mit oberer Dreiecksstruktur erreicht ist:

$$A^{(1)}x = b^{(1)} \quad \Leftrightarrow \quad A^{(2)}x = b^{(2)} \quad \Leftrightarrow \dots \Leftrightarrow \quad A^{(n)}x = b^{(n)}$$

(iii) Löse das lineare Gleichungssystem $A^{(n)}x = b^{(n)}$ durch Rückwärtssubstitution.

Beispiel 2.2.3

$$\underbrace{\begin{pmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{pmatrix}}_{=A^{(1)}} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \underbrace{\begin{pmatrix} 12 \\ 34 \\ 27 \\ -38 \end{pmatrix}}_{=b^{(1)}}$$

Gauss'sche Elimination liefert folgendes Ergebnis:

Start:

$$\left[A^{(1)} \mid b^{(1)} \right] = \left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 12 \\ 12 & -8 & 6 & 10 & 34 \\ 3 & -13 & 9 & 3 & 27 \\ -6 & 4 & 1 & -18 & -38 \end{array} \right]$$

Subtraktion des 2-fachen der 1. Zeile von der 2. Zeile und des 0.5-fachen der 1. Zeile von der 3. Zeile und des -1-fachen der 1. Zeile von der 4. Zeile liefert:

$$\left[A^{(2)} \mid b^{(2)} \right] = \left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 12 \\ 0 & -4 & 2 & 2 & 10 \\ 0 & -12 & 8 & 1 & 21 \\ 0 & 2 & 3 & -14 & -26 \end{array} \right]$$

Subtraktion des 3-fachen der 2. Zeile von der 3. Zeile und des -0.5-fachen der 2. Zeile von der 4. Zeile liefert:

$$\left[A^{(3)} \mid b^{(3)} \right] = \left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 12 \\ 0 & -4 & 2 & 2 & 10 \\ 0 & 0 & 2 & -5 & -9 \\ 0 & 0 & 4 & -13 & -21 \end{array} \right]$$

Subtraktion des 2-fachen der 3. Zeile von der 4. Zeile liefert:

$$\left[A^{(4)} \mid b^{(4)} \right] = \left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 12 \\ 0 & -4 & 2 & 2 & 10 \\ 0 & 0 & 2 & -5 & -9 \\ 0 & 0 & 0 & -3 & -3 \end{array} \right]$$

Rückwärtssubstitution:

$$x = (1, -3, -2, 1)^\top.$$

Im folgenden wird der Schritt von $i \mapsto i + 1$ im Detail beschrieben. Der Algorithmus sei bis zum i -ten Schritt fortgeschritten mit

$$A^{(i)} = \begin{pmatrix} a_{11}^{(i)} & \cdots & a_{1i}^{(i)} & \cdots & a_{1n}^{(i)} \\ & \ddots & \vdots & \ddots & \vdots \\ & & a_{ii}^{(i)} & \cdots & a_{in}^{(i)} \\ & & a_{i+1,i}^{(i)} & \cdots & a_{i+1,n}^{(i)} \\ & & \vdots & \ddots & \vdots \\ & & a_{ni}^{(i)} & \cdots & a_{nn}^{(i)} \end{pmatrix}, \quad b^{(i)} = \begin{pmatrix} b_1^{(i)} \\ \vdots \\ b_i^{(i)} \\ b_{i+1}^{(i)} \\ \vdots \\ b_n^{(i)} \end{pmatrix}.$$

Ziel ist es, die Elemente $a_{ji}^{(i)}$, $j = i + 1, \dots, n$, durch elementare Zeilenumformungen zu eliminieren. Es gelte $a_{ii}^{(i)} \neq 0$ für das sogenannte Pivotelement. Die Matrix $A^{(i+1)}$ erhält

man, indem das $\frac{a_{ji}^{(i)}}{a_{ii}^{(i)}}$ -fache der i -ten Zeile von den Zeilen $j = i + 1, \dots, n$ subtrahiert wird:

$$A^{(i+1)} = \begin{pmatrix} a_{11}^{(i+1)} & \cdots & a_{1,i+1}^{(i+1)} & \cdots & a_{1n}^{(i+1)} \\ & \ddots & \vdots & \ddots & \vdots \\ & & a_{i+1,i+1}^{(i+1)} & \cdots & a_{i+1,n}^{(i+1)} \\ & & a_{i+2,i+1}^{(i+1)} & \cdots & a_{i+2,n}^{(i+1)} \\ & & \vdots & \ddots & \vdots \\ & & a_{n,i+1}^{(i+1)} & \cdots & a_{nn}^{(i+1)} \end{pmatrix}$$

$$:= \begin{pmatrix} a_{11}^{(i)} & \cdots & a_{1i}^{(i)} & \cdots & a_{1n}^{(i)} \\ & \ddots & \vdots & \ddots & \vdots \\ & & a_{ii}^{(i)} & \cdots & a_{in}^{(i)} \\ & & a_{i+1,i}^{(i)} - \frac{a_{i+1,i}^{(i)}}{a_{ii}^{(i)}} a_{ii}^{(i)} & \cdots & a_{i+1,n}^{(i)} - \frac{a_{i+1,i}^{(i)}}{a_{ii}^{(i)}} a_{in}^{(i)} \\ & & \vdots & \ddots & \vdots \\ & & a_{ni}^{(i)} - \frac{a_{ni}^{(i)}}{a_{ii}^{(i)}} a_{ii}^{(i)} & \cdots & a_{nn}^{(i)} - \frac{a_{ni}^{(i)}}{a_{ii}^{(i)}} a_{in}^{(i)} \end{pmatrix}.$$

Die Matrix $A^{(i+1)}$ hat formal dieselbe Struktur wie $A^{(i)}$ mit dem Unterschied, daß in $A^{(i+1)}$ unterhalb der Hauptdiagonalen der i -ten Spalte, die in $A^{(i)}$ noch voll besetzt war, Nullen erzeugt wurden. Im nächsten Schritt würden dann die Elemente $a_{j,i+1}^{(i+1)}$, $j = i + 2, \dots, n$ eliminiert werden.

Entsprechend muß der Vektor $b^{(i)}$ transformiert werden:

$$b^{(i+1)} = \begin{pmatrix} b_1^{(i+1)} \\ \vdots \\ b_i^{(i+1)} \\ b_{i+1}^{(i+1)} \\ \vdots \\ b_n^{(i+1)} \end{pmatrix} := \begin{pmatrix} b_1^{(i)} \\ \vdots \\ b_i^{(i)} \\ b_{i+1}^{(i)} - \frac{a_{i+1,i}^{(i)}}{a_{ii}^{(i)}} b_i^{(i)} \\ \vdots \\ b_n^{(i)} - \frac{a_{ni}^{(i)}}{a_{ii}^{(i)}} b_i^{(i)} \end{pmatrix}.$$

Zusammenfassend lautet das Gauss'sche Eliminationsverfahren wie folgt (der Schleifendurchlauf für $j = k$ erzeugt nur Nulleinträge in A und kann ggf. weggelassen werden):

Algorithmus 2.2.4 (Gauss'scher Eliminationsalgorithmus)

Input: Matrix $A \in \mathbb{K}^{n \times n}$, rechte Seite $b \in \mathbb{K}^n$

Output: Lösung x mit $Ax = b$, A und b werden überschrieben.

%----- Elimination -----

For $k = 1, \dots, n-1$:

For $i = k + 1, \dots, n$:

$piv = a_{ik}/a_{kk}$

For $j = k, \dots, n$:

$a_{ij} = a_{ij} - piv \cdot a_{kj}$

end

$b_i = b_i - piv \cdot b_k$

end

end

%----- Backward substitution -----

Führe Rückwärtssubstitution für A und b durch.

2.2.3 Pivoting

Der Gauss'sche Eliminationsalgorithmus ist durchführbar, solange die Pivotelemente $a_{ii}^{(i)} \neq 0$, $i = 1, \dots, n-1$, erfüllen. Es kann gezeigt werden, daß dies für strikt diagonaldominante Matrizen und symmetrische, positiv definite Matrizen der Fall ist. Für andere Matrizen kann der Fall $a_{ii}^{(i)} = 0$ allerdings sehr schnell eintreten, wie das folgende Beispiel zeigt:

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Darüber hinaus entstehen numerische Probleme bedingt durch Rundungsfehler und Fehlerfortpflanzung bereits für $a_{ii}^{(i)} \approx 0$ wie das folgende Beispiel zeigt.

Beispiel 2.2.5

Betrachte

$$\begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

wobei $\varepsilon \approx 0$. Die exakte Lösung ist gegeben durch

$$x_1 = \frac{1}{1 - \varepsilon} \approx 1, \quad x_2 = \frac{1 - 2\varepsilon}{1 - \varepsilon} \approx 1.$$

Der Gauss'sche Eliminationsalgorithmus liefert

$$\begin{pmatrix} \varepsilon & 1 \\ 0 & 1 - \varepsilon^{-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 - \varepsilon^{-1} \end{pmatrix}.$$

Die Lösung dieses Systems lautet

$$x_2 = \frac{2 - \varepsilon^{-1}}{1 - \varepsilon^{-1}}, \quad x_1 = (1 - x_2)\varepsilon^{-1}.$$

Für betragsmäßig kleine Werte von ε erhält man auf Grund von Rundungsfehlern $x_2 \approx 1$ und anschließend $x_1 \approx 0$, also ein völlig falsches Ergebnis.

Falls wir die Gleichungen vertauschen, d.h.

$$\begin{pmatrix} 1 & 1 \\ \varepsilon & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix},$$

und anschließend den Gauss'schen Algorithmus anwenden, erhalten wir

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 - \varepsilon \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 - 2\varepsilon \end{pmatrix}$$

und die (beinahe) exakte Lösung

$$x_2 = \frac{1 - 2\varepsilon}{1 - \varepsilon} \approx 1, \quad x_1 = 2 - x_2 \approx 1.$$

Zeilenvertauschungen haben im obigen Beispiel zum Erfolg geführt. Dies bedeutet, daß die natürliche Reihenfolge bei der Wahl der Pivotelemente geändert wurde. Ziel dabei ist es, im Schritt i durch Zeilenvertauschungen in $A^{(i)}$ **und** $b^{(i)}$ unter den Elementen $a_{ji}^{(i)}$, $j = i, \dots, n$, eines mit $a_{ki}^{(i)} \neq 0$ zu finden und anschließend Zeile i mit Zeile k zu vertauschen. Ein solches Element existiert stets; andernfalls wäre A singulär. Um die Division durch ein betragsmäßig sehr kleines Pivotelement zu vermeiden (numerische Probleme!), wird bei der sogenannten **Spaltenpivotsuche** unter den in Frage kommenden Elementen $a_{ji}^{(i)} \neq 0$ mit $i \leq j \leq n$ das betragsmäßig größte Element

$$|a_{ki}^{(i)}| = \max\{|a_{ji}^{(i)}| \mid i \leq j \leq n, a_{ji}^{(i)} \neq 0\} \quad (2.8)$$

als neues Pivotelement verwendet, d.h. die Zeilen i und k werden vertauscht. Der Gauss'sche Algorithmus zusammen mit der Spaltenpivotsuche ist für reguläre Matrizen A bei exakter Rechnung stets durchführbar.

Anstatt der Spaltenpivotsuche kann auch die **Totalpivotsuche** verwendet werden. Im Gegensatz zur Spaltenpivotsuche, die sich nur auf die i -te Spalte von $A^{(i)}$ beschränkt, ermittelt die Totalpivotsuche das Pivotelement $a_{pq}^{(i)}$ mit $i \leq p, q \leq n$ in der kompletten Teilmatrix

$$\begin{pmatrix} a_{ii}^{(i)} & \cdots & a_{in}^{(i)} \\ \vdots & \ddots & \vdots \\ a_{ni}^{(i)} & \cdots & a_{nn}^{(i)} \end{pmatrix}$$

von $A^{(i)}$. Anschließend werden Zeile p mit Zeile i **und** Spalte q mit Spalte i vertauscht. In der Praxis wird die Totalpivotsuche selten verwendet, da der Aufwand mit $\mathcal{O}(n^2)$ pro Eliminationsschritt in derselben Größenordnung liegt wie der Gauss-Algorithmus selbst.

Natürlich müssen Zeilenvertauschungen auch in $b^{(i)}$ und Spaltenvertauschungen in x berücksichtigt werden.

In einer effizienten Implementierung werden Zeilenvertauschungen nicht explizit durchgeführt, um zeitaufwändiges (und unnötiges) Umspeichern zu vermeiden. Anstattdessen werden die Zeilenvertauschungen in einem Indexvektor $l = [l_1, l_2, \dots, l_n]$ protokolliert.

Die folgende Variante des Gauss'schen Eliminationsverfahrens verwendet eine skalierte Spaltenpivotsuche zur Bestimmung des Pivotelements.

Algorithmus 2.2.6 (Gauss'sche Elimination mit skaliertem Spaltenpivotsuche)

(i) Setze $A^{(1)} := A$, $b^{(1)} := b$, $l := [l_1, l_2, \dots, l_n] = [1, 2, \dots, n]$ und $k = 1$.

(ii) Berechne die **Skalierungsfaktoren**

$$s_{l_i} := \max_{1 \leq j \leq n} |a_{l_i, j}^{(k)}| \quad (k \leq i \leq n).$$

(iii) Bestimme das Pivotelement $a_{l_j, k}^{(k)}$ mit $k \leq j \leq n$ gemäß

$$\frac{a_{l_j, k}^{(k)}}{s_{l_j}} := \max_{k \leq i \leq n} \left| \frac{a_{l_i, k}^{(k)}}{s_{l_i}} \right|$$

(iv) Vertausche Werte l_k und l_j in l .

(v) Verwende Zeile l_k von $A^{(k)}$ als Pivotzeile und das Element $a_{l_k, k}^{(k)}$ als Pivotelement und berechne $A^{(k+1)}$ und $b^{(k+1)}$.

(vi) Falls $k < n - 1$, ersetze k durch $k + 1$ und gehe zu (iii).

(vii) Löse $A^{(n)}x = b^{(n)}$ durch Rückwärtssubstitution in der Reihenfolge l_n, l_{n-1}, \dots, l_1 .

Bemerkung 2.2.7

- Es existiert eine Variante des Algorithmus, bei dem die Skalierungsfaktoren s_i in jedem Durchlauf neu berechnet werden.
- Es ist wichtig zu bemerken, daß lediglich die Einträge in dem Indexvektor l vertauscht werden. Dies vermeidet das kostenintensive Umspeichern von Zeilen.
- Die Berechnung von $A^{(k+1)}$ und $b^{(k+1)}$ in Schritt (v) erfolgt wie in (2.2.4), wobei der Zeilenindex i durch l_i ersetzt wird, d.h. $a_{l_i, k}^{(k)}/a_{l_k, k}^{(k)}$ multipliziert mit Zeile l_k wird von den Gleichungen l_i , $i = k + 1, \dots, n$, subtrahiert, wohingegen die Zeilen l_1, \dots, l_k unverändert bleiben.

Beispiel 2.2.8*Betrachte*

$$\underbrace{\begin{pmatrix} 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \\ 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \end{pmatrix}}_{=A^{(1)}} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \underbrace{\begin{pmatrix} -19 \\ -34 \\ 16 \\ 26 \end{pmatrix}}_{=b^{(1)}}.$$

*Anwendung der Gauss'schen Elimination mit skaliertem Spaltenpivotsuche liefert:***Init:** $l = [1, 2, 3, 4]$, $s = [13, 18, 6, 12]$.**Schritt 1:** $k = 1$. *Pivotelement (nicht eindeutig):*

$$\max_{1 \leq i \leq 4} \left| \frac{a_{i,1}^{(1)}}{s_{l_i}} \right| = \max \left\{ \frac{3}{13}, \frac{6}{18}, \frac{6}{6}, \frac{12}{12} \right\} = \frac{6}{6} = \left| \frac{a_{3,1}^{(1)}}{s_3} \right|.$$

Zeile 3 ist Pivotzeile. Vertausche Zeilenindizes: $l = [3, 2, 1, 4]$. Elimination:

$$A^{(2)} = \begin{pmatrix} 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \\ 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \end{pmatrix}, \quad b^{(2)} = \begin{pmatrix} -27 \\ -18 \\ 16 \\ -6 \end{pmatrix}.$$

Schritt 2: $k = 2$. *Pivotelement:*

$$\max_{2 \leq i \leq 4} \left| \frac{a_{i,2}^{(2)}}{s_{l_i}} \right| = \max \left\{ \frac{2}{18}, \frac{12}{13}, \frac{4}{12} \right\} = \frac{12}{13} = \left| \frac{a_{1,2}^{(2)}}{s_1} \right|.$$

Zeile 1 ist Pivotzeile. Vertausche Zeilenindizes: $l = [3, 1, 2, 4]$. Elimination:

$$A^{(3)} = \begin{pmatrix} 0 & -12 & 8 & 1 \\ 0 & 0 & 13/3 & -83/6 \\ 6 & -2 & 2 & 4 \\ 0 & 0 & -2/3 & 5/3 \end{pmatrix}, \quad b^{(3)} = \begin{pmatrix} -27 \\ -45/2 \\ 16 \\ 3 \end{pmatrix}.$$

Schritt 3: $k = 3$. *Pivotelement:*

$$\max_{3 \leq i \leq 4} \left| \frac{a_{i,3}^{(3)}}{s_{l_i}} \right| = \max \left\{ \frac{13/3}{18}, \frac{2/3}{12} \right\} = \frac{13/3}{18} = \left| \frac{a_{2,3}^{(3)}}{s_2} \right|.$$

Zeile 2 ist Pivotzeile und $l = [3, 1, 2, 4]$. Elimination:

$$A^{(4)} = \begin{pmatrix} 0 & -12 & 8 & 1 \\ 0 & 0 & 13/3 & -83/6 \\ 6 & -2 & 2 & 4 \\ 0 & 0 & 0 & -6/13 \end{pmatrix}, \quad b^{(4)} = \begin{pmatrix} -27 \\ -45/2 \\ 16 \\ -6/13 \end{pmatrix}.$$

Rückwärtssubstitution: $x_4 = 1, x_2 = 1, x_1 = 3, x_3 = -2$.

Der Aufwand berechnet sich zu

Satz 2.2.9 (Aufwand)

Der Gauss'sche Eliminationsalgorithmus mit skalierter Spaltenpivotsuche zur Lösung von $Ax = b$ mit $A \in \mathbb{K}^{n \times n}$, $x, b \in \mathbb{K}^n$, benötigt

$$\frac{1}{3}n^3 + \frac{3}{2}n^2 + \frac{1}{6}n - 1$$

wesentliche Rechenoperationen (Multiplikationen oder Divisionen).

Beweis: Der Aufwand zur Bestimmung der Skalierungsfaktoren enthält keine Multiplikationen oder Divisionen, welches die teureren Rechenoperationen sind, und wird daher nicht gerechnet.

Die Anzahl der Multiplikationen und Divisionen pro Hauptiteration $k = 1, \dots, n-1$ beträgt $n-k+1$ für die Spaltenpivotsuche und $2(n-k) + (n-k)^2$ für die Elimination, insgesamt also

$$\begin{aligned} & \sum_{k=1}^{n-1} (n-k+1 + 2(n-k) + (n-k)^2) \\ &= \sum_{k=1}^{n-1} (3n+1+n^2 - (2n+3)k + k^2) \\ &= 3(n-1)n + n-1 + (n-1)n^2 - (2n+3)\frac{(n-1)n}{2} + \frac{(n-1)n(2n-1)}{6} \\ &= \frac{1}{3}n^3 + n^2 - \frac{1}{3}n - 1. \end{aligned}$$

Der Aufwand für die Rückwärtssubstitution beträgt

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

Insgesamt beträgt der Aufwand $\frac{1}{3}n^3 + \frac{3}{2}n^2 + \frac{1}{6}n - 1$ Multiplikationen und Divisionen. \square

2.2.4 Die LR-Zerlegung einer Matrix

In diesem Abschnitt werden wir sehen, daß das Gauss'sche Eliminationsverfahren nicht nur ein oberes Dreieckssystem produziert, sondern tatsächlich eine Faktorisierung

$$A = L \cdot R \tag{2.9}$$

der Matrix A erzeugt, wobei L eine untere Dreiecksmatrix und R eine obere Dreiecksmatrix ist.

Das lineare Gleichungssystem (2.3) kann dann gelöst werden, indem zunächst $y := Rx$ definiert wird und mit Hilfe der Vorwärtssubstitution

$$Ly = b$$

gelöst wird. Damit erhält man y . Im Anschluß wird mit der Rückwärtssubstitution

$$Rx = y$$

gelöst (y ist darin bereits bekannt). Dies liefert schließlich die Lösung x von (2.3). Dieser Vorgang wird **Vorwärts-Rückwärtssubstitution** genannt.

Diese Vorgehensweise ist besonders dann von Vorteil, falls das lineare Gleichungssystem (2.3) für viele verschiedene rechte Seiten b gelöst werden muß. Die LR -Zerlegung muß nur einmalig berechnet werden.

Beispiel 2.2.10 (Berechnung der Inversen A^{-1})

Die Inverse $X = A^{-1}$ einer invertierbaren Matrix A ist gegeben durch

$$A \cdot X = I.$$

Falls $x^{(i)}$ die i -te Spalte von X und $e^{(i)}$ den i -ten Einheitsvektor bezeichnen, müssen wir zur Bestimmung von X die folgenden n linearen Gleichungssysteme lösen:

$$Ax^{(i)} = e^{(i)}, \quad i = 1, 2, \dots, n.$$

Falls die LR -Zerlegung von A bekannt ist, müssen wir lediglich n Vorwärts-Rückwärts-Substitutionen durchführen. Der Gesamtaufwand zur Invertierung der Matrix beträgt damit $\mathcal{O}(n^3)$ (LR : $\mathcal{O}(n^3)$; FRS : $\mathcal{O}(n \cdot n^2)$). Würde das Gauss'sche Eliminationsverfahren für jedes der n Gleichungssysteme von Neuem angewendet werden, würde der Aufwand $\mathcal{O}(n^4)$ betragen.

Bemerkung 2.2.11

Es ist **niemals** notwendig, die Inverse A^{-1} einer Matrix explizit zu berechnen, wenn wir die Lösung $x = A^{-1}b$ eines linearen Gleichungssystems berechnen wollen. Es genügt, die LR -Zerlegung von A zu berechnen und Vorwärts-Rückwärtssubstitution durchzuführen.

Beispiel 2.2.12

$$A = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{pmatrix}.$$

Jeder Schritt des Gauss'schen Eliminationsverfahrens entspricht der Multiplikation von links mit einer Matrix L_k , so daß

$$\underbrace{\underbrace{L_3 \cdot L_2 \cdot \underbrace{L_1 \cdot A}_{=A^{(2)}}}_{=A^{(3)}}}_{=A^{(4)}} = R \quad \Rightarrow \quad A = \underbrace{L_1^{-1} \cdot L_2^{-1} \cdot L_3^{-1}}_{=:L} \cdot R$$

wobei

$$L_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, L_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{pmatrix}, L_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -2 & 1 \end{pmatrix}.$$

Zusammenfassend erzeugt das Gauss'sche Eliminationsverfahren die LR-Zerlegung von A :

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 3 & 1 & 0 \\ -1 & -\frac{1}{2} & 2 & 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}}_{=L_1^{-1}} \cdot \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & -\frac{1}{2} & 0 & 1 \end{pmatrix}}_{=L_2^{-1}} \cdot \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{pmatrix}}_{=L_3^{-1}}$$

und

$$R = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{pmatrix}.$$

(Verifiziere, daß tatsächlich $L \cdot R = A$ gilt!)

Der Transformationsschritt $i \mapsto i + 1$ im Gauss'schen Eliminationsverfahren kann mit Hilfe von linksseitigen Matrixmultiplikationen beschrieben werden. Es gilt (nachprüfen!)

$$A^{(i+1)} = L_i \cdot A^{(i)}$$

mit

$$L_i = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & l_{i+1,i} & 1 & \\ & & \vdots & & \ddots \\ & & l_{ni} & & & 1 \end{pmatrix}, \quad l_{ji} = -\frac{a_{ji}^{(i)}}{a_{ii}^{(i)}}, \quad j = i+1, \dots, n. \quad (2.10)$$

Die Inverse L_i^{-1} zu L_i läßt sich leicht angeben (nachrechnen!):

$$L_i^{-1} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{i+1,i} & 1 & \\ & & \vdots & & \ddots \\ & & -l_{ni} & & & 1 \end{pmatrix}.$$

Das Gauss'sche Eliminationsverfahren liefert also nach $n - 1$ Schritten die Beziehung

$$\begin{aligned} R &= A^{(n)} \\ &= L_{n-1} \cdot A^{(n-1)} \\ &= L_{n-1} \cdot L_{n-2} \cdot A^{(n-2)} \\ &\vdots \\ &= L_{n-1} \cdot L_{n-2} \cdots L_1 \cdot A^{(1)}. \end{aligned} \tag{2.11}$$

Mit $L := L_1^{-1} \cdots L_{n-2}^{-1} \cdot L_{n-1}^{-1}$ und $A^{(1)} = A$ folgt

$$A = L \cdot R.$$

Da das Produkt von zwei normierten linken unteren Dreiecksmatrizen wieder eine normierte linke untere Dreiecksmatrix ergibt, ist L normierte linke untere Dreiecksmatrix mit

$$L = \begin{pmatrix} 1 & & & & \\ -l_{21} & 1 & & & \\ \vdots & & \ddots & & \\ -l_{n1} & \cdots & -l_{n,n-1} & 1 \end{pmatrix}. \tag{2.12}$$

wobei die l_{ij} 's durch (2.10) gegeben sind (verifiziere dies!).

Um Speicherplatz zu sparen, speichern effiziente Computerimplementierungen L und R in A , d.h. die Elemente l_{ij} werden unterhalb der Diagonalen gespeichert und die Elemente von R werden auf und oberhalb der Diagonalen gespeichert.

Algorithmus 2.2.13 (LR-Zerlegung ohne Pivoting)

Input: Matrix $A \in \mathbb{K}^{n \times n}$, rechte Seite $b \in \mathbb{K}^n$

Output: Lösung $x \in \mathbb{K}^n$ des LGS $Ax = b$, A wird überschrieben und enthält unterhalb der Diagonalen die Matrix L und auf und oberhalb der Diagonalen die Matrix R

%----- LR Zerlegung -----

For $k = 1, \dots, n-1$:

```

For  $i = k + 1, \dots, n$ :
     $piv = a_{ik} / a_{kk}$ 
    For  $j = k + 1, \dots, n$ :
         $a_{ij} = a_{ij} - piv \cdot a_{kj}$ 
    end
     $a_{ik} = piv$ 
end
end
%———— Vorwaertssubstitution  $Ly=b$  (Ergebnis  $y$  steht in  $x$ ) ———-
 $x_1 = b_1$ 
For  $i=2, \dots, n$ :
     $x_i = b_i$ 
    For  $j = 1, \dots, i - 1$ :
         $x_i = x_i - a_{ij} \cdot x_j$ 
    end
end
%———— Rueckwaertssubstitution  $Rx=y$  ———-
 $x_n = x_n / a_{nn}$ 
For  $i=n-1, \dots, 1$ :
    For  $j = i + 1, \dots, n$ :
         $x_i = x_i - a_{i,j} \cdot x_j$ 
    end
     $x_i = x_i / a_{ii}$ 
end

```

2.2.5 Pivoting in Matrixnotation

Die Spaltenpivotsuche läßt sich ebenfalls durch Matrixmultiplikationen darstellen. Die Multiplikation einer Matrix von links mit der Permutationsmatrix

$$P = \left(\begin{array}{cccc|cccc} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ \hline & & & 0 & & & 1 & \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \\ \hline & & & 1 & & & 0 & \\ & & & & & & & 1 \\ & & & & & & & \ddots \\ & & & & & & & & 1 \end{array} \right) \begin{array}{l} \\ \\ \\ \text{Zeile } i \\ \\ \\ \text{Zeile } j \\ \\ \\ \end{array}$$

Spalte i Spalte j

vertauscht die **Zeilen** i und j , während die Multiplikation von rechts die **Spalten** i und j vertauscht. Zusätzlich gilt $P^\top P = I$ und $P = P^\top$. Die Spaltenpivotsuche im Schritt i entspricht daher der Multiplikation von $A^{(i)}$ von links mit einer Permutationsmatrix P_i bevor die Subdiagonalelemente eliminiert werden: $A^{(i+1)} = L_i \cdot P_i \cdot A^{(i)}$. Analog zu (2.11) erhält man

$$R = L_{n-1} \cdot P_{n-1} \cdot L_{n-2} \cdot P_{n-2} \cdots L_1 \cdot P_1 \cdot A. \quad (2.13)$$

Man kann zeigen daß für $i < k$ die folgende Beziehung gilt:

$$P_k \cdot L_i = \tilde{L}_i \cdot P_k.$$

Darin entsteht \tilde{L}_i aus L_i durch Vertauschung der Elemente l_{ki} und l_{ji} , wobei j den Zeilenindex bezeichnet, der durch P_k mit k vertauscht wird. Damit läßt sich (2.13) zu

$$P \cdot A = \tilde{L} \cdot R$$

mit $P = P_{n-1} \cdot P_{n-2} \cdots P_1$ und $\tilde{L} = \tilde{L}_1^{-1} \cdots \tilde{L}_{n-2}^{-1} \cdot L_{n-1}^{-1}$ umformen. Beachte, daß $\tilde{L}_1, \dots, \tilde{L}_{n-3}$ mehrfache Elementvertauschungen enthalten können.

Bei der Totalpivotsuche werden zusätzlich noch Spaltenvertauschungen notwendig: $A^{(i+1)} = L_i \cdot P_i \cdot A^{(i)} \cdot Q_i$. Analoges Vorgehen liefert

$$P \cdot A \cdot Q = \tilde{L} \cdot R.$$

P und \tilde{L} sind wie oben erklärt. Q ergibt sich zu $Q = Q_1 \cdot Q_{n-2} \cdots Q_{n-1}$.

Satz 2.2.14 (LR-Zerlegung mit Pivoting)

Sei $A \in \mathbb{K}^{n \times n}$ invertierbar. Dann definiert das Gauss'sche Eliminationsverfahren mit Spaltenpivotsuche eine Zerlegung $PA = LR$ mit einer rechten oberen Dreiecksmatrix R , einer normierten linken unteren Dreiecksmatrix L und einer Permutationsmatrix P .

Beweis: Bricht das Gauss'sche Eliminationsverfahren mit Spaltenpivotsuche nicht zusammen, d.h. sind alle Pivotelemente $a_{ki}^{(i)} \neq 0$ für alle $i = 1, \dots, n$, so liefern die vorangegangenen Betrachtungen die Behauptung.

Zu klären ist, daß das Verfahren für invertierbare Matrizen tatsächlich nicht zusammenbricht, d.h. daß alle Pivotelemente von Null verschieden sind. Angenommen, das Pivotelement im i -ten Teilschritt wäre Null, dann gilt gemäß der Auswahlregel (2.8) des Pivotelements $a_{ji}^{(i)} = 0$ für alle $i \leq j \leq n$ und $A^{(i)}$ hätte folgende Gestalt:

$$A^{(i)} = \left(\begin{array}{ccc|ccc} a_{11}^{(i)} & \cdots & & a_{1i}^{(i)} & \cdots & a_{1n}^{(i)} \\ & & \ddots & \vdots & \ddots & \vdots \\ & & & 0 & \cdots & a_{in}^{(i)} \\ \hline & & & 0 & \cdots & a_{i+1,n}^{(i)} \\ & & & \vdots & \ddots & \vdots \\ & & & 0 & \cdots & a_{nn}^{(i)} \end{array} \right).$$

Die Determinante des rechten unteren Teilblocks ist folglich 0 und somit ist $\det(A^{(i)}) = 0$. Wegen

$$\det(A^{(i)}) = \det(L_{i-1} \cdot P_{i-1} \cdots L_1 \cdot P_1 \cdot A) = \left(\prod_{j=1}^{i-1} \det(L_j) \det(P_j) \right) \det(A)$$

und $\det(L_j) = 1$, $\det(P_j) = \pm 1$, $j = 1, \dots, i-1$, folgt dann $\det(A) = 0$ im Widerspruch zur Invertierbarkeit von A . \square

Ohne Beweis sei erwähnt, daß

- die LR-Zerlegung ohne Pivoting für allgemeine invertierbare Matrizen **nicht** vorwärtsstabil ist.
- die LR-Zerlegung mit Spaltenpivoting für allgemeine invertierbare Matrizen **vorwärtsstabil** ist.
- die LR-Zerlegung mit totaler Pivotsuche für allgemeine invertierbare Matrizen **rückwärtsstabil** ist.

Beispiel 2.2.15

Wir lösen das lineare Gleichungssystem

$$Ax = b \quad \text{mit} \quad A = \begin{pmatrix} 2 & 18 & 9 \\ 4 & 4 & 4 \\ 1 & 33 & 9 \end{pmatrix}, \quad b = \begin{pmatrix} 21 \\ 4 \\ 9 \end{pmatrix},$$

indem wir die LR-Zerlegung mit Spaltenpivotsuche berechnen.

1.Schritt: Pivotelement $a_{21} = 4$.

$$P_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ -\frac{1}{4} & 0 & 1 \end{pmatrix}, \quad A^{(2)} = \begin{pmatrix} 4 & 4 & 4 \\ 0 & 16 & 7 \\ 0 & 32 & 8 \end{pmatrix}$$

2.Schritt: Pivotelement $a_{32}^{(2)} = 32$.

$$P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{pmatrix}, \quad A^{(3)} = \begin{pmatrix} 4 & 4 & 4 \\ 0 & 32 & 8 \\ 0 & 0 & 3 \end{pmatrix}$$

Insgesamt:

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \tilde{L} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix}, \quad R = \begin{pmatrix} 4 & 4 & 4 \\ 0 & 32 & 8 \\ 0 & 0 & 3 \end{pmatrix}$$

Lösung: Es ist $\tilde{b} := P \cdot b = P \cdot Ax = \tilde{L} \cdot Rx$. Es wird also $\tilde{L} \cdot Rx = \tilde{b} = (4, 9, 21)^\top$ mit Vorwärts-Rückwärtssubstitution gelöst.

$$\begin{aligned} \text{Vorwärtssubstitution:} \quad \tilde{L}y = \tilde{b} &\Rightarrow y = (4, 8, 15)^\top \\ \text{Rückwärtssubstitution:} \quad Rx = y &\Rightarrow x = (-3, -1, 5)^\top \end{aligned}$$

2.3 Cholesky-Zerlegung

Treten bei einer Anwendung ausschließlich lineare Gleichungssysteme mit hermitescher, positiv definiter Matrix $A \in \mathbb{K}^{n \times n}$ auf, sollte die Cholesky-Zerlegung an Stelle der LR-Zerlegung verwendet werden, da diese nur etwa den halben Aufwand erfordert.

Definition 2.3.1 (hermitesche Matrix)

Eine Matrix $A \in \mathbb{K}^{n \times n}$ heißt **hermitesch**, wenn $A^* = A$ gilt, wobei $A^* = \bar{A}^\top$ gilt.

Definition 2.3.2 (positive Definitheit)

Eine Matrix $A \in \mathbb{K}^{n \times n}$ heißt **positiv definit**, wenn $x^*Ax > 0$ für alle $x \neq 0$, $x \in \mathbb{K}^n$,

gilt.

Definition 2.3.3 (Cholesky-Zerlegung)

Sei $A \in \mathbb{K}^{n \times n}$. Eine Zerlegung der Form

$$A = L \cdot L^*$$

mit linker unterer (nicht normierter!) Dreiecksmatrix L und positiven Diagonalelementen heißt **Cholesky-Zerlegung von A** .

Wir benötigen folgendes Hilfsresultat.

Hilfsatz 2.3.4 (Schur Komplement)

Sei $A \in \mathbb{K}^{n \times n}$ hermitesch und positiv definit. Für $1 \leq p \leq n$ sei A partitioniert gemäß

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad A_{11} \in \mathbb{K}^{p \times p}.$$

Dann sind A_{11} und $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ hermitesch und positiv definit. S heißt **Schur-Komplement von A bzgl. A_{11}** .

Beweis: Wegen $A = A^*$ sind auch A_{11} und A_{22} hermitesch und es gilt $A_{12} = A_{21}^*$. Da A positiv definit ist, gilt für beliebiges $0 \neq x \in \mathbb{K}^p$

$$0 < \begin{pmatrix} x^* & 0 \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x \\ 0 \end{pmatrix} = x^* A_{11} x,$$

d.h. A_{11} ist positiv definit und somit invertierbar. Damit ist S wohldefiniert und hermitesch, da

$$S^* = A_{22}^* - A_{12}^*(A_{11}^{-1})^* A_{21}^* = A_{22} - A_{21} A_{11}^{-1} A_{12} = S.$$

Sei $0 \neq y \in \mathbb{K}^{n-p}$ beliebig und $x = -A_{11}^{-1} A_{12} y$. Dann gilt, wie man leicht nachrechnet,

$$0 < \begin{pmatrix} x^* & y^* \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = y^* S y,$$

d.h. S ist positiv definit. □

Es gilt folgender Satz.

Satz 2.3.5

Sei $A \in \mathbb{K}^{n \times n}$ hermitesch. A besitzt eine Cholesky-Zerlegung genau dann, wenn A positiv definit ist.

Beweis: Sei A hermitesch.

\Rightarrow : Besitzt A eine Cholesky-Zerlegung $A = L \cdot L^*$ mit einer unteren Dreiecksmatrix L , welche positive Diagonalelemente hat, so sind L und L^* invertierbar. Es folgt

$$x^*Ax = x^*L \cdot L^*x = (L^*x)^* \cdot L^*x \geq 0.$$

Der Fall $x^*Ax = 0$ tritt genau dann auf, wenn $L^*x = 0$ gilt, was wegen der Invertierbarkeit von L^* genau dann der Fall ist, wenn $x = 0$ gilt. Damit gilt für $x \neq 0$ stets $x^*Ax > 0$, d.h. A ist positiv definit.

\Leftarrow : Sei A positiv definit. Induktion über die Dimension n der Matrix.

$n = 1$:

Besteht A nur aus dem Element a_{11} , so ist a_{11} reellwertig, da A hermitesch ist, und es muß $a_{11} > 0$ gelten, da A als positiv definit vorausgesetzt war. Mit $L := (\sqrt{a_{11}})$ ist die Existenz der Cholesky-Zerlegung gezeigt.

$n - 1 \mapsto n$:

Sei die Behauptung für alle Matrizen der Dimension $n - 1$ gezeigt. Sei $A \in \mathbb{K}^{n \times n}$ hermitesch und positiv definit. Partitioniere A gemäß

$$A = \begin{pmatrix} a_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad A_{22} \in \mathbb{K}^{(n-1) \times (n-1)}, \quad A_{12} = A_{21}^*.$$

Wir machen den Ansatz

$$A = L \cdot L^*, \quad L = \begin{pmatrix} \ell_{11} & 0 \\ A_{21}/\bar{\ell}_{11} & L_S \end{pmatrix}$$

mit einer linken unteren Dreiecksmatrix L . Ausmultiplizieren und Vergleichen mit den Einträgen von A liefert

$$|\ell_{11}|^2 = a_{11}, \quad L_S \cdot L_S^* + \frac{1}{|\ell_{11}|^2} A_{21} \cdot A_{12} = A_{22}$$

bzw.

$$L_S \cdot L_S^* = A_{22} - \frac{1}{|\ell_{11}|^2} A_{21} \cdot A_{12} =: S.$$

Da A positiv definit ist, ist $a_{11} > 0$ und $\ell_{11} = \sqrt{a_{11}}$ ist wohldefiniert.

Gemäß Hilfsatz 2.3.4 ist die hermitesche $(n - 1) \times (n - 1)$ -Matrix S (Schur-Komplement von A bzgl. a_{11}) positiv definit. Nach Induktionsvoraussetzung besitzt sie somit eine Cholesky-Zerlegung der Form $L_S \cdot L_S^* = S$.

Dies zeigt die Existenz der Cholesky-Zerlegung.

□

Diese Äquivalenz von Cholesky-Zerlegung und positiver Definitheit einer hermiteschen Matrix liefert einen praktisch durchführbaren Algorithmus, um eine gegebene hermitesche Matrix auf positive Definitheit zu testen. Man muß lediglich die Cholesky-Zerlegung berechnen. Falls sie existiert (d.h. es gilt $\ell_{kk} > 0$ für alle k), ist A positiv definit, andernfalls nicht.

Die konkrete Berechnung von L erfolgt durch zeilenweisen Koeffizientenvergleich im Ansatz

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} \ell_{11} & & & \\ \ell_{21} & \ell_{22} & & \\ \vdots & \vdots & \ddots & \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{pmatrix} \cdot \begin{pmatrix} \bar{\ell}_{11} & \bar{\ell}_{21} & \cdots & \bar{\ell}_{n1} \\ & \bar{\ell}_{22} & \cdots & \bar{\ell}_{n2} \\ & & \ddots & \vdots \\ & & & \bar{\ell}_{nn} \end{pmatrix}.$$

Es folgt (beachte: für reelle Matrizen, soll eine reelle Zerlegung entstehen):

$$\begin{aligned} a_{11} &= \ell_{11} \bar{\ell}_{11} = |\ell_{11}|^2 \Rightarrow \ell_{11} = \sqrt{a_{11}}, \\ a_{21} &= \ell_{21} \bar{\ell}_{11} \Rightarrow \ell_{21} = a_{21} / \bar{\ell}_{11}, \\ a_{22} &= \ell_{21} \bar{\ell}_{21} + \ell_{22} \bar{\ell}_{22} \Rightarrow \ell_{22} = \sqrt{a_{22} - |\ell_{21}|^2}, \\ a_{31} &= \ell_{31} \bar{\ell}_{11} \Rightarrow \ell_{31} = a_{31} / \bar{\ell}_{11}, \\ a_{32} &= \ell_{31} \bar{\ell}_{21} + \ell_{32} \bar{\ell}_{22} \Rightarrow \ell_{32} = (a_{32} - \ell_{31} \bar{\ell}_{21}) / \bar{\ell}_{22}, \\ a_{33} &= \ell_{31} \bar{\ell}_{31} + \ell_{32} \bar{\ell}_{32} + \ell_{33} \bar{\ell}_{33} \Rightarrow \ell_{33} = \sqrt{a_{33} - |\ell_{31}|^2 - |\ell_{32}|^2}, \\ &\vdots \quad \vdots \quad \vdots \\ a_{ij} &= \sum_{k=1}^{j-1} \ell_{ik} \bar{\ell}_{jk} + \ell_{ij} \bar{\ell}_{jj} \Rightarrow \ell_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} \ell_{ik} \bar{\ell}_{jk} \right) / \bar{\ell}_{jj}, \quad j = 1, \dots, i-1, \\ a_{ii} &= \sum_{k=1}^{i-1} \ell_{ik} \bar{\ell}_{ik} + \ell_{ii} \bar{\ell}_{ii} \Rightarrow \ell_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} \ell_{ik} \bar{\ell}_{ik}} \end{aligned}$$

Insgesamt entstehen die Formeln

$$\begin{aligned} \ell_{ii} &= \sqrt{a_{ii} - \sum_{k=1}^{i-1} |\ell_{ik}|^2}, \quad i = 1, \dots, n, \\ \ell_{ij} &= \left(a_{ij} - \sum_{k=1}^{j-1} \ell_{ik} \bar{\ell}_{jk} \right) / \bar{\ell}_{jj}, \quad j = 1, \dots, i-1. \end{aligned}$$

Diese Formeln sind für positiv definite Matrizen wohldefiniert, da die Existenz einer Cholesky-Zerlegung bereits nachgewiesen wurde. Die obigen Formeln sind auch eindeutig unter der Forderung, daß für reelle Matrizen eine reelle Zerlegung entstehen soll, und wir erhalten

Satz 2.3.6

Die Cholesky-Zerlegung einer hermiteschen, positiv definiten Matrix ist eindeutig bestimmt.

Beispiel 2.3.7

Cholesky-Zerlegung:

$$A = \begin{pmatrix} 10 & 2 & -1 \\ 2 & 5 & 0 \\ -1 & 0 & 4 \end{pmatrix} \approx \begin{pmatrix} 3.1623 & 0 & 0 \\ 0.6325 & 2.1448 & 0 \\ -0.3162 & 0.0933 & 1.9726 \end{pmatrix} \begin{pmatrix} 3.1623 & 0.6325 & -0.3162 \\ 0 & 2.1448 & 0.0933 \\ 0 & 0 & 1.9726 \end{pmatrix}$$

Da A symmetrisch ist und die Cholesky-Zerlegung existiert ($l_{kk} > 0$ für alle k), folgt, daß A positiv definit ist.

Der Aufwand der Cholesky-Zerlegung beträgt

$$\sum_{i=1}^n \left(i - 1 + \sum_{j=1}^{i-1} (j - 1) + 1 \right) = \frac{1}{6}n^3 + \frac{1}{2}n^2 - \frac{2}{3}n = \frac{1}{6}n^3 + \mathcal{O}(n^2)$$

Multiplikationen und Divisionen und ist damit in etwa halb so groß wie bei der LR-Zerlegung.

2.4 Iterative Nachbesserung

Hat man die LR-Zerlegung der Matrix A berechnet und durch Vorwärts-Rückwärtssubstitution eine Näherung x (Rundungsfehler!) der Lösung \hat{x} berechnet, dann kann man x durch iterative Nachbesserung verbessern:

Algorithmus 2.4.1 (Iterative Nachbesserung)

Gegeben: LR-Zerlegung von A , Lösung x mit $Ax \approx b$.

Output: verbesserte Lösung \tilde{x}

- (1) Berechne das Residuum $r = b - Ax$ (mit doppelter Genauigkeit).
- (2) Löse $Az = r$ mittels Vorwärts-Rückwärtssubstitution.
- (3) Ersetze x durch $x + z$.

In der Praxis werden häufig einige wenige iterative Nachbesserungsschritte durchgeführt. Jeder Nachbesserungsschritt ist relativ günstig, da nur eine Matrix-Vektor Multiplikation mit Aufwand $\mathcal{O}(n^2)$ und eine Vorwärts-Rückwärtssubstitution mit Aufwand $\mathcal{O}(n^2)$ durchgeführt werden müssen.

2.5 Fehlereinfluß und Kondition

Gegenstand dieses Abschnitts ist die Untersuchung der Kondition eines linearen Gleichungssystems. Dazu benötigen wir zusätzlich zum Normbegriff für Vektoren auch den Normbegriff für Matrizen.

Beispiel 2.5.1 (Vandermonde-Matrix)

Wir versuchen, das lineare Gleichungssystem

$$\begin{pmatrix} 1 & 2 & 4 & 8 & \cdots & 2^{n-1} \\ 1 & 3 & 9 & 27 & \cdots & 3^{n-1} \\ 1 & 4 & 16 & 64 & \cdots & 4^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & n+1 & (n+1)^2 & (n+1)^3 & \cdots & (n+1)^{n-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} (2^n - 1)/1 \\ (3^n - 1)/2 \\ (4^n - 1)/3 \\ \vdots \\ ((n+1)^n - 1)/n \end{pmatrix}$$

für verschiedene Werte von n mittels Gauss'scher Elimination zu lösen. Die auftretende Matrix heißt **Vandermonde-Matrix**. Die exakte Lösung ist durch $x_i = 1$ für $i = 1, 2, \dots, n$ gegeben. Auf einem Computer mit ca. 16 Stellen Genauigkeit erhalten wir das exakte Resultat für $n \leq 14$. Für $n = 15$ jedoch erhalten wir plötzlich folgendes Ergebnis (gerundet auf 2 Stellen):

$$x_1 \approx -894.97, x_2 \approx 2074.34, x_3 \approx -2138.14, x_4 \approx 1309.97, x_5 \approx -531.35, x_6 \approx 153.62, \\ x_7 \approx -30.88, x_8 \approx 5.94, x_9 \approx 0.43, x_{10} \approx 1.05, x_{11} \approx \dots \approx x_{15} \approx 1.00$$

Dieses Ergebnis ist völlig wertlos und hat nichts mit der Lösung zu tun. Für $n > 15$ wird es noch schlimmer.

Was ist der Grund für dieses seltsame Verhalten in Beispiel 2.5.1? Offenbar spielen Rundungsfehler eine Rolle, da ansonsten bei exakter Rechnung die exakte Lösung gefunden werden müßte.

Wir interessieren uns nun für die Frage, wie Fehler in A und b (etwa Rundungsfehler, die beim Gauss'schen Eliminationsverfahren auftreten) die Lösung x von $Ax = b$ beeinflussen. Wir werden sehen, daß die sogenannte Konditionszahl einer Matrix eine entscheidende Rolle spielt.

Definition 2.5.2 (Vektornorm)

Sei V ein Vektorraum über \mathbb{K} . Die Abbildung $\|\cdot\| : V \rightarrow \mathbb{R}$ heißt Norm, wenn sie die folgenden Eigenschaften besitzt:

1. $\|x\| = 0$ gilt genau dann, falls $x = 0$ ist,
2. $\|cx\| = |c| \|x\|$ für alle $x \in V$ und $c \in \mathbb{K}$,
3. $\|x + y\| \leq \|x\| + \|y\|$ für alle $x, y \in V$.

Beispiel 2.5.3 (Vektornormen)

Häufig verwendete Vektornormen auf \mathbb{K}^n mit $x = (x_1, \dots, x_n)^\top$ sind

(i) die Maximumnorm: $\|x\|_\infty := \max_{1 \leq i \leq n} |x_i|$;

(ii) die Betragssummennorm: $\|x\|_1 := \sum_{i=1}^n |x_i|$;

(iii) die Euklidische Norm: $\|x\|_2 := \sqrt{\sum_{i=1}^n |x_i|^2}$.

(iv) p -Norm: Für $p \in \mathbb{N}$ ist $\|x\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$ eine Norm.

Bemerkung 2.5.4

In endlichdimensionalen Vektorräumen sind alle Normen äquivalent, d.h. für beliebige Normen $\|\cdot\|_{(1)}$ und $\|\cdot\|_{(2)}$ über demselben Vektorraum V gibt es Konstanten c_1, c_2 , so daß für alle $x \in V$ gilt

$$c_1 \|x\|_{(2)} \leq \|x\|_{(1)} \leq c_2 \|x\|_{(2)}.$$

Für unendlichdimensionale Vektorräume gilt dies nicht!

Formal kann man einer Vektornorm $\|\cdot\|_V$ eine „passende“ Matrixnorm $\|\cdot\|_M$ zuordnen.

Definition 2.5.5 (Matrixnorm)

Sei $\|\cdot\|_V$ eine Vektornorm. Die durch $\|\cdot\|_V$ induzierte Matrixnorm $\|\cdot\|_M$ auf dem Raum aller linearen Abbildungen $A: \mathbb{K}^n \rightarrow \mathbb{K}^m$ ist definiert als

$$\|A\|_M := \sup_{\|x\|_V \neq 0} \frac{\|Ax\|_V}{\|x\|_V} \left(= \sup_{\|x\|_V=1} \|Ax\|_V = \sup_{\|x\|_V \leq 1} \|Ax\|_V \right), \quad (2.14)$$

wobei $x \in \mathbb{K}^n$ und $A \in \mathbb{K}^{m \times n}$ gilt

Beachte: In Definition 2.5.5 wird in \mathbb{K}^n und \mathbb{K}^m dieselbe Vektornorm verwendet! Allgemeiner könnte man auch unterschiedliche Vektornormen verwenden, was wir aus Gründen der Übersichtlichkeit hier nicht tun. Die Wohldefiniertheit der Matrixnorm ergibt sich aus der Stetigkeit der Abbildung $x \mapsto \|Ax\|_V$ und der Kompaktheit der Menge $\{x \in \mathbb{K}^n \mid \|x\|_V = 1\}$. Daß $\|\cdot\|_M$ tatsächlich eine Norm ist, ergibt sich direkt aus den Normeigenschaften von $\|\cdot\|_V$ und kann leicht nachgerechnet werden.

Die so definierte Matrixnorm hat zwei schöne Eigenschaften:

Satz 2.5.6

Sei $\|\cdot\|_V$ eine Vektornorm und $\|\cdot\|_M$ die induzierte Matrixnorm. Dann gelten:

- (a) Verträglichkeit, d.h. es gilt $\|Ax\|_V \leq \|A\|_M \cdot \|x\|_V$ für alle $x \in \mathbb{K}^n$, $A \in \mathbb{K}^{m \times n}$.
- (b) Submultiplikativität, d.h. es gilt $\|A \cdot B\|_M \leq \|A\|_M \cdot \|B\|_M$ für alle $A \in \mathbb{K}^{m \times n}$, $B \in \mathbb{K}^{n \times q}$.

Beweis:

- (a) Für $x = 0$ ist die Behauptung klar. Für $\|x\|_V \neq 0$ gilt $\|Ax\|_V = \|A \frac{x}{\|x\|_V}\| \cdot \|x\|_V$ und somit

$$\|Ax\|_V \leq \left(\sup_{\|x\|_V \neq 0} \|A \frac{x}{\|x\|_V}\| \right) \cdot \|x\|_V = \left(\sup_{\|x\|_V \neq 0} \frac{\|Ax\|_V}{\|x\|_V} \right) \cdot \|x\|_V = \|A\|_M \cdot \|x\|_V.$$

- (b) Nach (a) gilt $\|A \cdot Bx\|_V \leq \|A\|_M \cdot \|Bx\|_V$ und somit

$$\|A \cdot B\|_M = \sup_{\|x\|_V \neq 0} \frac{\|A \cdot Bx\|_V}{\|x\|_V} \leq \|A\|_M \cdot \sup_{\|x\|_V \neq 0} \frac{\|Bx\|_V}{\|x\|_V} = \|A\|_M \cdot \|B\|_M.$$

□

Die konkrete Berechnung einer induzierten Matrixnorm für eine gegebene Vektornorm ist mitunter kompliziert.

Satz 2.5.7 (Zeilensummennorm)

Die durch die Vektornorm $\|\cdot\|_\infty$ induzierte Matrixnorm ist die sogenannte **Zeilensummennorm** $\|A\|_\infty := \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|$.

Beweis: Es bezeichne $\|\cdot\|_M$ die durch die Maximumnorm induzierte Matrixnorm. Zu zeigen ist $\|A\|_M = \|A\|_\infty$ für alle $A \in \mathbb{K}^{m \times n}$.

Es gilt

$$\begin{aligned} \|Ax\|_\infty &= \max_{1 \leq i \leq m} \left| \sum_{j=1}^n a_{ij} x_j \right| \\ &\leq \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}| \cdot |x_j| \\ &\leq \left(\max_{1 \leq j \leq n} |x_j| \right) \cdot \left(\max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}| \right) \\ &= \|x\|_\infty \cdot \|A\|_\infty. \end{aligned}$$

Somit gilt nach Definition der Matrixnorm $\|A\|_M \leq \|A\|_\infty$.

Wähle i_0 mit

$$\sum_{j=1}^n |a_{i_0,j}| = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|$$

und definiere $\hat{x} \in \mathbb{K}^n$ mit

$$\hat{x}_j = \begin{cases} 0, & \text{falls } a_{i_0,j} = 0, \\ \frac{\bar{a}_{i_0,j}}{|a_{i_0,j}|}, & \text{falls } a_{i_0,j} \neq 0. \end{cases}$$

Dann folgt

$$\begin{aligned} (A\hat{x})_{i_0} &= \sum_{j=1}^n a_{i_0,j} \hat{x}_j \\ &= \sum_{j=1}^n a_{i_0,j} \bar{a}_{i_0,j} \frac{1}{|a_{i_0,j}|} \\ &= \sum_{j=1}^n |a_{i_0,j}| \cdot 1 \\ &= \left(\max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}| \right) \cdot \|\hat{x}\|_\infty \\ &= \|A\|_\infty \cdot \|\hat{x}\|_\infty. \end{aligned}$$

Wegen $(A\hat{x})_{i_0} = \|A\hat{x}\|_\infty$ folgt somit $\|A\|_M \geq \|A\|_\infty$. □

In ähnlicher Weise zeigt man

Satz 2.5.8 (Spaltensummennorm)

Die durch die Betragssummennorm $\|\cdot\|_1$ induzierte Matrixnorm ist die sogenannte **Spaltensummennorm** $\|A\|_1 := \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|$.

Beweis: Übung. □

Schließlich betrachten wir noch die euklidische Norm:

Satz 2.5.9 (Spektralnrm)

Die durch die euklidische Vektornorm $\|\cdot\|_2$ induzierte Matrixnorm ist die **Spektralnrm**

$$\|A\|_2 = \sqrt{\rho(A^*A)},$$

wobei $\rho(A^*A)$ den Spektralradius von A^*A bezeichnet, d.h.

$$\rho(A^*A) = \max\{|\lambda| \mid \lambda \text{ ist Eigenwert von } A^*A\}.$$

Beweis: $B = A^*A$ ist hermitesch und somit normal, d.h. es gilt $B^*B = BB^*$. Dann gibt es eine unitäre Matrix T ($T^{-1} = T^*$; orthonormale Basis aus Eigenvektoren), so daß

$D = T^* A^* A T = (AT)^*(AT) \geq 0$ eine Diagonalmatrix ist, welche die reellen Eigenwerte von $A^* A$ auf der Diagonalen enthält.

Damit folgt

$$\begin{aligned} \|Ax\|_2 &= (x^* A^* A x)^{1/2} \\ &= (x^* T T^* A^* A T T^* x)^{1/2} \\ &= (x^* T D T^* x)^{1/2} \\ &\leq \left(\max_{1 \leq i \leq m} d_{ii} \right)^{1/2} \cdot (x^* T T^* x)^{1/2} \\ &= \sqrt{\rho(A^* A)} \cdot \|x\|_2. \end{aligned}$$

Also: $\|A\|_2 \leq \sqrt{\rho(A^* A)}$.

Wähle j_0 mit $d_{j_0, j_0} = \max_{1 \leq i \leq m} d_{ii} = \rho(A^* A)$. Dann folgt für die j_0 -Spalte von T :

$$\begin{aligned} \|AT^{j_0}\|_2 &= ((T^{j_0})^* A^* A T^{j_0})^{1/2} \\ &= ((T^{j_0})^* d_{j_0, j_0} T^{j_0})^{1/2} \\ &= \sqrt{\rho(A^* A)} \cdot \|T^{j_0}\|_2. \end{aligned}$$

Folglich ist $\|A\|_2 \geq \sqrt{\rho(A^* A)}$. □

Bemerkung 2.5.10

Es gibt weitere Matrixnormen, die aber i.a. nicht durch Vektornormen induziert sind:

- Die Matrixnorm $\|A\| = \max_{1 \leq i \leq m, 1 \leq j \leq n} |a_{ij}|$ ist durch keine Vektornorm induziert, da sie nicht submultiplikativ ist wie folgendes Beispiel zeigt:

$$A = B = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad A \cdot B = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}, \quad 2 = \|A \cdot B\| > \|A\| \cdot \|B\| = 1.$$

- Die **Frobenius-Schur-Norm**

$$\|A\|_F := \sqrt{\sum_{i=1, \dots, m} \sum_{j=1, \dots, n} |a_{ij}|^2},$$

ist verträglich und submultiplikativ bezüglich der euklidischen Vektornorm, aber sie wird nicht durch diese induziert.

- Ist A normal, so gilt $\|A\|_2 = \rho(A)$.

Mit diesen Vorbereitungen untersuchen wir den Fehlereinfluß in linearen Gleichungssystemen. Hierbei sei x die Lösung von $Ax = b$.

Störungen in der rechten Seite b :

Wie wirken sich Störungen Δb in der rechten Seite b in (2.3) auf die Lösung $x + \Delta x$ des gestörten Gleichungssystems

$$A(x + \Delta x) = b + \Delta b$$

aus?

Für die Abweichung der gestörten Lösung von der exakten Lösung gilt bezüglich einer Vektornorm $\|\cdot\|_V$ und der zugeordneten Matrixnorm $\|\cdot\|_M$ die Beziehung

$$\begin{aligned} \|\Delta x\|_V &= \|x + \Delta x - x\|_V \\ &= \|A^{-1}(b + \Delta b) - A^{-1}b\|_V \\ &= \|A^{-1}(b + \Delta b - b)\|_V \\ &\leq \|A^{-1}\|_M \cdot \|\Delta b\|_V. \end{aligned}$$

Mit Hilfe der **absoluten Konditionszahl** $\kappa_a = \|A^{-1}\|_M$ (vgl. Abschnitt 1.6) ergibt sich für den absoluten Fehler die Abschätzung

$$\|\Delta x\|_V \leq \kappa_a \cdot \|\Delta b\|_V. \quad (2.15)$$

Mit (2.15) und der Verträglichkeit folgt für $\|x\|_V, \|b\|_V \neq 0$ die Abschätzung

$$\|A^{-1}\|_M \cdot \frac{\|\Delta b\|_V}{\|b\|_V} = \|A^{-1}\|_M \cdot \frac{\|\Delta b\|_V}{\|Ax\|_V} \geq \frac{\|\Delta x\|_V}{\|Ax\|_V} \geq \frac{\|\Delta x\|_V}{\|A\|_M \cdot \|x\|_V}.$$

Umformen ergibt eine Abschätzung des relativen Fehlers

$$\frac{\|\Delta x\|_V}{\|x\|_V} \leq \|A\|_M \cdot \|A^{-1}\|_M \cdot \frac{\|\Delta b\|_V}{\|b\|_V}. \quad (2.16)$$

Definition 2.5.11 (Kondition einer Matrix)

Die Zahl

$$\kappa_M(A) := \|A\|_M \cdot \|A^{-1}\|_M$$

heißt **Konditionszahl** der Matrix A bzgl. der Matrixnorm $\|\cdot\|_M$. A heißt **schlecht konditioniert**, falls $\kappa_M(A)$ sehr groß ist.

Die Kondition erfüllt wegen

$$1 = \|I\|_M = \|A^{-1}A\|_M \leq \|A^{-1}\|_M \cdot \|A\|_M = \kappa_M(A)$$

stets $\kappa_M(A) \geq 1$.

Mit dieser Definition läßt sich (2.16) schreiben als

$$\frac{\|\Delta x\|_V}{\|x\|_V} \leq \kappa_M(A) \cdot \frac{\|\Delta b\|_V}{\|b\|_V}.$$

Die Kondition $\kappa_M(A)$ einer Matrix A macht also eine Aussage darüber, wie stark sich Störungen in der rechten Seite auf das Ergebnis auswirken können. Insbesondere bei schlecht konditionierten Matrizen mit $\kappa_M(A) \gg 1$ können sich bereits kleine Störungen in b sehr stark auf das Ergebnis auswirken.

Vergleich mit Abschnitt 1.6: dort ist $\kappa_r = \kappa_a \cdot \|b\|_V / \|x\|_V$; hier ist $\tilde{\kappa}_r = \|A\|_M \cdot \kappa_a$ mit $\tilde{\kappa}_r \geq \kappa_r$ gewählt, denn $\|b\|_V / \|x\|_V = \|Ax\|_V / \|x\|_V \leq \|A\|_M$.

Störungen in der Matrix A und der rechten Seite b :

Wir untersuchen nun die Auswirkungen von Störungen ΔA in A und Δb in b in (2.3) auf die Lösung $x + \Delta x$ des gestörten Gleichungssystems

$$(A + \Delta A)(x + \Delta x) = b + \Delta b.$$

Hilfsatz 2.5.12

Sei S eine Matrix mit $\|S\|_M < 1$. Dann existiert $(I - S)^{-1}$ und es gilt

$$\frac{1}{1 + \|S\|_M} \leq \|(I - S)^{-1}\|_M \leq \frac{1}{1 - \|S\|_M}.$$

Beweis: Sei λ Eigenwert von S , d.h. $Sx = \lambda x$, $x \neq 0$. Dann folgt

$$|\lambda| \cdot \|x\|_V = \|\lambda x\|_V = \|Sx\|_V \leq \|S\|_M \cdot \|x\|_V < \|x\|_V$$

und somit $|\lambda| < 1$. Die Eigenwerte von $I - S$ sind durch $1 - \lambda$ gegeben, wobei λ Eigenwert von S ist. Für die Eigenwerte von $I - S$ gilt dann $|1 - \lambda| \geq 1 - |\lambda| > 0$. Somit ist $I - S$ invertierbar.

Es gilt

$$I = (I - S)(I - S)^{-1} = (I - S)^{-1} - S(I - S)^{-1}$$

und somit

$$1 = \|I\|_M \leq \|(I - S)^{-1}\|_M + \|S\|_M \cdot \|(I - S)^{-1}\|_M.$$

Umformen liefert

$$\frac{1}{1 + \|S\|_M} \leq \|(I - S)^{-1}\|_M.$$

Ebenso gilt

$$S(I - S)^{-1} = (I - S)^{-1} - I$$

und daraus

$$\|S\|_M \cdot \|(I - S)^{-1}\|_M \geq \|(I - S)^{-1}\|_M - \|I\|_M.$$

Umformen liefert

$$\frac{1}{1 - \|S\|_M} \geq \|(I - S)^{-1}\|_M.$$

□

Es gilt folgender Satz.

Satz 2.5.13

Es gelte $Ax = b$ und $(A + \Delta A)(x + \Delta x) = b + \Delta b$. Dann gilt

$$\frac{\|\Delta x\|_V}{\|x\|_V} \leq \frac{\kappa_M(A)}{1 - \kappa_M(A) \cdot \frac{\|\Delta A\|_M}{\|A\|_M}} \cdot \left(\frac{\|\Delta A\|_M}{\|A\|_M} + \frac{\|\Delta b\|_V}{\|b\|_V} \right),$$

falls $\kappa_M(A) \cdot \frac{\|\Delta A\|_M}{\|A\|_M} < 1$ und $\|x\|_V \neq 0$ ist.

Beweis: Es gilt

$$\begin{aligned} \Delta x &= (A + \Delta A)^{-1}(b + \Delta b) - A^{-1}b \\ &= ((A + \Delta A)^{-1} - A^{-1})b + (A + \Delta A)^{-1}\Delta b \\ &= (A + \Delta A)^{-1}(I - (A + \Delta A)A^{-1})b + (A + \Delta A)^{-1}\Delta b \\ &= -(A + \Delta A)^{-1}\Delta A A^{-1}b + (A + \Delta A)^{-1}\Delta b \\ &= -(A + \Delta A)^{-1}\Delta A x + (A + \Delta A)^{-1}\Delta b. \end{aligned}$$

Gemäß Hilfsatz 2.5.12 gilt

$$\begin{aligned} \|(A + \Delta A)^{-1}\|_M &= \|(I + A^{-1}\Delta A)^{-1}A^{-1}\|_M \\ &\leq \frac{1}{1 - \|A^{-1}\|_M \cdot \|\Delta A\|_M} \|A^{-1}\|_M \\ &= \frac{\kappa_M(A)}{1 - \kappa_M(A) \frac{\|\Delta A\|_M}{\|A\|_M}} \cdot \frac{1}{\|A\|_M}. \end{aligned}$$

Zusammen mit $\|b\|_V = \|Ax\|_V \leq \|A\|_M \cdot \|x\|_V$ folgt

$$\begin{aligned} \frac{\|\Delta x\|_V}{\|x\|_V} &\leq \|(A + \Delta A)^{-1}\|_M \cdot (\|\Delta A\|_M \cdot \|x\|_V + \|\Delta b\|_V) \cdot \frac{1}{\|x\|_V} \\ &\leq \frac{\kappa_M(A)}{1 - \kappa_M(A) \cdot \frac{\|\Delta A\|_M}{\|A\|_M}} \cdot \frac{1}{\|A\|_M} \cdot \left(\|\Delta A\|_M + \frac{\|\Delta b\|_V}{\|b\|_V / \|A\|_M} \right), \end{aligned}$$

woraus die Behauptung folgt. □

Beispiel 2.5.14 (Vandermonde-Matrix, Revisited)

Nun können wir den Grund für das seltsame Verhalten in Beispiel 2.5.1 angeben. Die Vandermonde-Matrix ist ein Beispiel für eine extrem schlecht konditionierte Matrix.

Faustregel: Falls $\kappa_M(A) = 10^k$, dann muß man mit dem Verlust von mindestens k Stellen Genauigkeit bei der Lösung von $Ax = b$ rechnen. Für die Vandermonde-Matrix A mit $n = 15$ gilt

$$\kappa_M(A) = 1024224393669250854080 > 10^{21}.$$

Wir untersuchen noch einige Eigenschaften der Kondition bzgl. der Spektralnorm $\|\cdot\|_2$. Diese Norm bietet einige Vorteile, da für unitäre Matrizen $Q \in \mathbb{K}^{n \times n}$ für beliebige $x \in \mathbb{K}^n$ die Beziehung

$$\|Qx\|_2^2 = x^* Q^* Q x = x^* x = \|x\|_2^2 \quad \text{bzw.} \quad \|Qx\|_2 = \|x\|_2$$

gilt. Da mit Q auch Q^{-1} unitär ist, folgt ebenso $\|Q^{-1}x\|_2 = \|x\|_2$. Aus der Definition einer Matrixnorm folgt damit $\|Q\|_2 = \|Q^{-1}\|_2 = 1$ und somit

$$\kappa_2(Q) = 1.$$

Unitäre Matrizen haben also eine bzgl. der Spektralnorm optimale Kondition. Dies ist der Grund, warum wir später mit orthogonalen Householder-Matrizen arbeiten werden, vgl. Abschnitt 3.2.4. Im Vorgriff auf diesen Abschnitt notieren wir das folgende Resultat.

Satz 2.5.15

Sei $A \in \mathbb{K}^{n \times n}$ invertierbar. Dann gilt

$$\kappa_2(A^* A) = \kappa_2(A)^2$$

Beweis: Zunächst bemerken wir, daß $A^* A$ und AA^* dieselben Eigenwerte besitzen, denn:

$$\begin{array}{ll} \lambda \text{ EW von } A^* A \text{ mit Eigenvektor } v \neq 0 & \\ \Leftrightarrow A^* A v = \lambda v & \\ \begin{array}{l} A \text{ invertierbar} \\ \Leftrightarrow \end{array} & AA^* A v = \lambda A v \\ \begin{array}{l} w = Av \\ \Leftrightarrow \end{array} & AA^* w = \lambda w \\ \Leftrightarrow & \lambda \text{ EW von } AA^* \text{ mit Eigenvektor } w \neq 0 \end{array}$$

Für den Spektralradius folgt damit $\rho(A^* A) = \rho(AA^*)$ und damit

$$\|A\|_2 = \sqrt{\rho(A^* A)} = \sqrt{\rho(AA^*)} = \|A^*\|_2.$$

Weiter gilt

$$\|A^* A\|_2 = \sqrt{\rho((A^* A)^* A^* A)} = \sqrt{\rho((A^* A)^2)} = \rho(A^* A) = \|A\|_2^2.$$

Für die Kondition ergibt sich dann

$$\begin{aligned}
 \kappa_2(A^*A) &= \|A^*A\|_2 \cdot \|(A^*A)^{-1}\|_2 \\
 &= \|A^*A\|_2 \cdot \|A^{-1}(A^{-1})^*\|_2 \\
 &= \|A^*A\|_2 \cdot \|((A^{-1})^*)^*(A^{-1})^*\|_2 \\
 &= \|A\|_2^2 \cdot \|(A^{-1})^*\|_2^2 \\
 &= \|A\|_2^2 \cdot \|(A^{-1})\|_2^2 \\
 &= \kappa_2(A)^2.
 \end{aligned}$$

□

2.6 Iterative Lösung

Der Aufwand zur Berechnung der LR-Zerlegung und der Cholesky-Zerlegung (und später der QR-Zerlegung) für quadratische Matrizen ist jeweils von der Ordnung $\mathcal{O}(n^3)$. Der Aufwand wächst also kubisch mit der Dimension n des Gleichungssystems. Für sehr große Gleichungssysteme mit teilweise mehr als einer Milliarde Gleichungen, wie sie in praktischen Anwendungen durchaus entstehen können, ist der Aufwand damit zu groß. Es werden alternative Methoden benötigt, die weniger aufwendig sind.

Beispiel 2.6.1 (Diskretisierung einer partiellen Differentialgleichung)

Große, dünn besetzte Gleichungssysteme entstehen bei Diskretisierungsverfahren zur approximativen Lösung partieller Differentialgleichungen. Als Beispiel sei die 2-dimensionale partielle Differentialgleichung

$$\begin{aligned}
 -u_{xx}(x, y) - u_{yy}(x, y) &= f(x, y), & (x, y) \in \Omega := (0, 1) \times (0, 1), \\
 u(x, y) &= 0, & (x, y) \in \partial\Omega
 \end{aligned}$$

genannt. Hierbei ist eine Funktion $u : \Omega \rightarrow \mathbb{R}$ gesucht, die die obige partielle Differentialgleichung erfüllt. Eine gängige Vorgehensweise besteht in der Diskretisierung der Differentialgleichung auf dem äquidistanten Gitter

$$G := \{(x_i, y_j) \mid x_i = ih, y_j = jh, 0 \leq i, j \leq N\}, \quad h = 1/N.$$

Approximation der zweiten partiellen Ableitungen durch

$$\begin{aligned}
 u_{xx}(x_i, y_j) &\approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}, & 1 \leq i, j \leq N-1, \\
 u_{yy}(x_i, y_j) &\approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}, & 1 \leq i, j \leq N-1,
 \end{aligned}$$

mit $u_{i,j} \approx u(x_i, y_j)$ liefert das lineare Gleichungssystem $Au = b$ für den Vektor

$$u = (u_{1,1}, u_{1,2}, \dots, u_{1,N-1}, u_{2,1}, u_{2,2}, \dots, u_{2,N-1}, \dots, u_{N-1,1}, u_{N-1,2}, \dots, u_{N-1,N-1})^\top \in \mathbb{R}^{(N-1)^2},$$

mit

$$b = h^2(f_{1,1}, f_{1,2}, \dots, f_{1,N-1}, f_{2,1}, f_{2,2}, \dots, f_{2,N-1}, \dots, f_{N-1,1}, f_{N-1,2}, \dots, f_{N-1,N-1})^\top \in \mathbb{R}^{(N-1)^2},$$

$$f_{i,j} := f(x_i, y_j),$$

$$A = \begin{pmatrix} M_1 & D_2 & & & \\ D_2 & M_2 & D_3 & & \\ & \ddots & \ddots & \ddots & \\ & & D_{N-2} & M_{N-2} & D_{N-1} \\ & & & D_{N-1} & M_{N-1} \end{pmatrix}, \quad M_i = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix} \in \mathbb{R}^{(N-1)^2}$$

und $D_i = -I \in \mathbb{R}^{(N-1) \times (N-1)}$. Die Matrix A ist sehr groß (Dimension wächst quadratisch mit N), aber sehr dünn besetzt (pro Zeile sind maximal 5 Einträge $\neq 0$). Zudem ist sie positiv definit und symmetrisch (Beweis?).

In der Praxis werden nur die von Null verschiedenen Einträge der Matrix A und die zugehörigen Spalten- und Zeilenpositionen gespeichert. Damit kann man Multiplikationen der Form $A \cdot d$ mit einem Vektor d sehr effizient berechnen (der Aufwand hängt linear von der Anzahl der nichtverschwindenden Einträge in A ab). Wir werden im Folgenden sehen, daß die sogenannten iterativen Verfahren lediglich Matrix-Vektor Multiplikationen benötigen.

Die Idee der **iterativen Verfahren** zur Lösung von linearen Gleichungssystemen (2.3) basiert auf der geschickten Zerlegung der Matrix A in

$$A = M - N \tag{2.17}$$

mit Matrizen M und N , die noch genauer spezifiziert werden müssen. Es wird angenommen, daß M regulär ist und „leicht“ invertiert werden kann. Dann läßt sich das lineare Gleichungssystem

$$b = Ax = (M - N)x = Mx - Nx$$

umformen zu

$$x = M^{-1}Nx + M^{-1}b. \tag{2.18}$$

Mit $T := M^{-1}N$ und $c := M^{-1}b$ ist dies eine **Fixpunktgleichung** für x :

$$x = Tx + c. \tag{2.19}$$

Damit ist jede Lösung der Fixpunktgleichung (2.19) auch Lösung des linearen Gleichungssystems (2.3) und umgekehrt.

Definition 2.6.2 (Fixpunkt)

\hat{x} heißt **Fixpunkt** der Abbildung $g : \mathbb{K}^n \rightarrow \mathbb{K}^n$, $x \mapsto g(x)$, wenn

$$\hat{x} = g(\hat{x})$$

gilt.

Eine einfache Idee zur Berechnung eines Fixpunkts \hat{x} von g besteht darin, ausgehend von einem Startwert $x^{(0)}$ eine **Fixpunktiteration**

$$x^{(i+1)} = g(x^{(i)}), \quad i = 0, 1, 2, \dots$$

durchzuführen, indem das Ergebnis immer wieder in die Funktion eingesetzt wird. Auf diese Weise erhält man eine Folge $\{x^{(i)}\}$. Es stellt sich natürlich die Frage, ob die Folge auch tatsächlich gegen den gesuchten Fixpunkt \hat{x} konvergiert.

Beispiel 2.6.3

- (i) Wir wollen einen Fixpunkt \hat{x} mit $\hat{x} = \cos(\hat{x})$ bestimmen. Wir führen die Fixpunktiteration ausgehend von $x^{(0)} = 1$ für die Funktion $g(x) = \cos(x)$ durch. Die Fixpunktiteration lässt sich ganz einfach mit dem Taschenrechner durchführen, indem 1 eingetippt wird und anschließend immer wieder auf cos gedrückt wird (im Bogenmaß rechnen!). Nach mehrmaligem Drücken der cos-Taste erhält man mit fünf Nachkommastellen $\hat{x} = 0.739085$. Weiteres Drücken der cos-Taste ändert das Ergebnis nicht mehr wesentlich. Die Folge konvergiert offensichtlich.
- (ii) Jetzt wollen wir einen Fixpunkt von $g(x) = \exp(x)$ mit Startwert $x^{(0)} = 1$ berechnen. Hier das Ergebnis:

$$\begin{aligned} x^{(1)} &= 2.71828183 \\ x^{(2)} &= 15.15426223 \\ x^{(3)} &= 0.38142791 \cdot 10^7 \\ x^{(4)} &= 0.22316774 \cdot 10^{1656521} \end{aligned}$$

Der nächste Iterationsschritt liefert einen Exponentenüberlauf. Die Folge konvergiert offensichtlich nicht.

Beispiel 2.6.3 zeigt, daß die Fixpunktiteration nicht immer sinnvolle Resultate liefert. Offensichtlich hängt die Konvergenz von Eigenschaften der Funktion g ab. Ein hinreichendes Kriterium für die Konvergenz der Fixpunktfolge liefert

Satz 2.6.4 (Banachscher Fixpunktsatz)

Sei eine **Selbstabbildung** $g : D \rightarrow D$ mit **abgeschlossener Menge** $D \subseteq \mathbb{K}^n$ gegeben.
Für alle $x, y \in D$ sei die **Kontraktionsbedingung**

$$\|g(x) - g(y)\| \leq q\|x - y\| \quad \text{mit} \quad q < 1 \quad (2.20)$$

für eine geeignete Norm $\|\cdot\|$ erfüllt. Dann gilt:

- (a) g besitzt genau einen Fixpunkt \hat{x} in D .
- (b) Die Fixpunktiteration $x^{(i+1)} = g(x^{(i)})$, $i = 0, 1, 2, \dots$ konvergiert für jeden Startwert $x^{(0)}$ aus D gegen den Fixpunkt \hat{x} .
- (c) Es gilt die a-priori Fehlerabschätzung

$$\|x^{(i)} - \hat{x}\| \leq \frac{q^i}{1 - q} \|x^{(1)} - x^{(0)}\|.$$

Beweis: Auf Grund der Selbstabbildungseigenschaft ist die Fixpunktiteration wohldefiniert.

- (i) Existenz eines Fixpunkts:

Sei $x^{(0)} \in D$ beliebig. Wir zeigen, daß die Folge eine Cauchyfolge ist.

Die Kontraktionseigenschaft von g liefert für jedes $i \in \mathbb{N}$ die Beziehung

$$\|x^{(i+1)} - x^{(i)}\| \leq q\|x^{(i)} - x^{(i-1)}\| \leq \dots \leq q^i \|x^{(1)} - x^{(0)}\|.$$

Mit Hilfe der Dreiecksungleichung folgt dann für jedes $k \in \mathbb{N}$ die Abschätzung

$$\begin{aligned} \|x^{(i+k)} - x^{(i)}\| &\leq \|x^{(i+k)} - x^{(i+k-1)}\| + \|x^{(i+k-1)} - x^{(i+k-2)}\| + \dots + \|x^{(i+1)} - x^{(i)}\| \\ &\leq (q^{i+k-1} + q^{i+k-2} + \dots + q^i) \|x^{(1)} - x^{(0)}\| \\ &\leq \frac{q^i}{1 - q} \|x^{(1)} - x^{(0)}\|. \end{aligned}$$

Wegen $q < 1$ ist $\{x^{(i)}\}$ eine Cauchyfolge in D . Da D abgeschlossen ist, konvergiert die Folge dann gegen ein $\hat{x} \in D$.

Gemäß der Kontraktionsbedingung ist g lipschitz-stetig und insbesondere stetig. Damit gilt

$$\hat{x} = \lim_{i \rightarrow \infty} x^{(i+1)} = \lim_{i \rightarrow \infty} g(x^{(i)}) = g(\lim_{i \rightarrow \infty} x^{(i)}) = g(\hat{x}),$$

d.h. \hat{x} ist Fixpunkt.

(ii) Eindeutigkeit:

Angenommen, es gäbe einen weiteren Fixpunkt $\tilde{x} \neq \hat{x}$ in D . Dann gilt

$$\|\tilde{x} - \hat{x}\| = \|g(\tilde{x}) - g(\hat{x})\| \leq q\|\tilde{x} - \hat{x}\|.$$

Wegen $q < 1$ kann dies nur für $\|\tilde{x} - \hat{x}\| = 0$ gelten im Widerspruch zu $\tilde{x} \neq \hat{x}$.

(iii) Die Fehlerabschätzung folgt mit (i) aus

$$\|x^{(i)} - \hat{x}\| \leq \|x^{(i+1)} - \hat{x}\| + \|x^{(i+1)} - x^{(i)}\| \leq q\|x^{(i)} - \hat{x}\| + q^i\|x^{(1)} - x^{(0)}\|.$$

□

Anwendung der Fixpunktiteration auf die Fixpunktgleichung (2.19) liefert das folgende iterative Verfahren zur Lösung des linearen Gleichungssystems (2.3):

Algorithmus 2.6.5 (Iteratives Lösungsverfahren für (2.3))

(i) Zerlege die Matrix A gemäß (2.17) mit einer invertierbaren Matrix M , wähle einen beliebigen Startvektor $x^{(0)}$ und setze $k = 0$.

(ii) Falls $r_k = \|b - Ax^{(k)}\|_2$ hinreichend klein, STOP.

(iii) Berechne

$$Mx^{(k+1)} = Nx^{(k)} + b \quad (2.21)$$

(iv) Setze $k := k + 1$ und gehe zu (ii).

Bemerkung 2.6.6

- (2.21) ist unter den bisherigen Annahmen äquivalent mit (2.19).
- Konvergiert die Folge $\{x^{(k)}\}$ gegen \hat{x} , dann gilt $M\hat{x} = N\hat{x} + b = (M - A)\hat{x} + b = M\hat{x} - A\hat{x} + b$ und somit löst \hat{x} das lineare Gleichungssystem $A\hat{x} = b$.

Unter Anwendung des Banachschen Fixpunktsatzes lautet ein hinreichendes Konvergenzkriterium wie folgt.

Satz 2.6.7 (Hinreichendes Konvergenzkriterium)

Es gelte

$$\|T\|_M < 1 \quad (2.22)$$

bezüglich einer mit der Vektornorm $\|\cdot\|_V$ verträglichen Matrixnorm $\|\cdot\|_M$. Dann existiert genau ein Fixpunkt \hat{x} der Funktion $g(x) = Tx + c$ und die Fixpunktiteration

$$x^{(i+1)} = Tx^{(i)} + c, \quad i = 0, 1, 2, \dots \quad (2.23)$$

konvergiert für jeden Startwert $x^{(0)} \in \mathbb{K}^n$ gegen \hat{x} . Insbesondere ist \hat{x} Lösung des Gleichungssystems (2.3).

Beweis: Wegen $\|T\|_M < 1$ gilt

$$\|Tx + c - (Ty + c)\|_V = \|T(x - y)\|_V \leq \|T\|_M \cdot \|x - y\|_V$$

und die Kontraktionseigenschaft (2.20) ist nachgewiesen. Da die Abbildung $x \mapsto Tx + c$ affin linear ist, liegt eine Selbstabbildung mit $D = \mathbb{K}^n$ vor, so daß die Voraussetzungen des Banachschen Fixpunktsatzes erfüllt sind. Die Behauptungen ergeben sich aus demselben.

□

Die bisherige Konvergenzaussage hängt noch von den gewählten Normen $\|\cdot\|_V$ und $\|\cdot\|_M$ ab und setzt voraus, daß solche Normen mit (2.22) gefunden werden können. Ein notwendiges und hinreichendes Kriterium für die Konvergenz von (2.23) liefert

Satz 2.6.8

Sei $A = M - N$ invertierbar und $T = M^{-1} \cdot N$. Dann konvergiert (2.23) genau dann für jeden Startwert $x^{(0)} \in \mathbb{R}^n$ gegen die Lösung \hat{x} von (2.3), wenn für den Spektralradius $\rho(T)$ von T die Ungleichung $\rho(T) < 1$ gilt.

Beweis:

\Leftarrow Es gelte $\rho(T) < 1$. Man kann zeigen (Übung!), daß es zu jeder Matrix T und zu jedem $\varepsilon > 0$ eine Vektornorm $\|\cdot\|_V$ (im \mathbb{K}^n) existiert, so daß die induzierte Matrixnorm $\|\cdot\|_M$ die folgende Ungleichung erfüllt:

$$\|T\|_M \leq \rho(T) + \varepsilon.$$

Damit folgt die Konvergenz sofort aus Satz 2.6.7, wenn $\varepsilon > 0$ hinreichend klein gewählt wird.

\Rightarrow Wir zeigen die Behauptung unter der vereinfachenden Annahme, daß T diagonalisierbar ist, d.h. es gilt

$$T = S \cdot \Lambda \cdot S^{-1}, \quad \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

mit einer regulären Matrix S und einer Diagonalmatrix Λ , welche die Eigenwerte von T enthält. Ein vollständiger Beweis kann in analoger Weise mit einigen zusätzlichen technischen Schwierigkeiten mittels Jordan-Zerlegung von T geführt werden.

Die gemäß (2.23) konstruierte Folge $\{x^{(i)}\}$ sei für jeden Startwert $x^{(0)}$ konvergent gegen einen Fixpunkt \hat{x} . Es gilt

$$x^{(i)} - \hat{x} = T(x^{(i-1)} - \hat{x}) = \dots = T^i(x^{(0)} - \hat{x}) = S\Lambda^i S^{-1}(x^{(0)} - \hat{x}).$$

Mit $z^{(i)} := S^{-1}(x^{(i)} - \hat{x})$ folgt für jede Komponente $z_j^{(i)}$ von $z^{(i)}$,

$$z_j^{(i)} = \lambda_j^i z_j^{(0)}, \quad \forall i \in \mathbb{N}, j = 1, \dots, n.$$

Da $x^{(i)}$ gegen \hat{x} konvergiert, konvergiert $z^{(i)}$ gegen Null, so daß

$$\lim_{i \rightarrow \infty} \lambda_j^i = 0 \quad \forall j = 1, \dots, n$$

gilt. Dies ist genau dann der Fall, wenn $|\lambda_j| < 1$ für $j = 1, \dots, n$ gilt. Da T und Λ dieselben Eigenwerte besitzen, gilt somit $\rho(T) < 1$.

□

Es stellt sich noch die Frage, wie schnell die so konstruierten Näherungen $x^{(i)}$ gegen den Fixpunkt \hat{x} konvergieren.

Wegen

$$\|x^{(i+1)} - \hat{x}\|_V = \|Tx^{(i)} + c - (T\hat{x} + c)\|_V = \|T(x^{(i)} - \hat{x})\|_V \leq \|T\|_M \cdot \|x^{(i)} - \hat{x}\|_V$$

mit $\|T\|_M < 1$ erhalten wir i.a. nur eine sogenannte **lineare Konvergenzgeschwindigkeit** für die Fixpunktiteration, da sich der Fehler im schlechtesten Fall in jeder Iteration nur um den Faktor $\|T\|_M$ verringert. Insbesondere kann die Konvergenz für $\|T\|_M \approx 1$ sehr langsam sein.

2.6.1 Spezielle Verfahren

Es werden spezielle Verfahren durch geeignete Wahl von M und N konstruiert. Die Fixpunktiteration (2.23) ist besonders dann günstig, falls M leicht invertiert werden kann und die auftretenden Multiplikationen $M^{-1}N$ und $M^{-1}b$ billig sind, wie es häufig bei dünn besetzten Matrizen der Fall ist.

Wir zerlegen A in

$$A = D - L - R$$

wobei D die Diagonalelemente von A , L die Elemente unterhalb der Diagonalen von $-A$ und R die Elemente oberhalb der Diagonalen von $-A$ enthalten.

2.6.1.1 Gesamtschrittverfahren (Jacobi-Verfahren)

Wir wählen in (2.17)

$$M = D, \quad N = L + R$$

und erhalten den **Gesamtschrittoperator**

$$T = D^{-1}(L + R).$$

Die Fixpunktiteration (2.23) für das Gesamtschrittverfahren lautet somit

$$x^{(i+1)} = D^{-1}((L + R)x^{(i)} + b). \quad (2.24)$$

Die Inverse von $D = \text{diag}(a_{11}, a_{22}, \dots, a_{nn})$ ist sehr leicht zu berechnen, da D eine Diagonalmatrix ist. Es ist $D^{-1} = \text{diag}(1/a_{11}, 1/a_{22}, \dots, 1/a_{nn})$. Um das Verfahren durchführen zu können, müssen die Diagonalelemente ungleich Null sein.

Algorithmus 2.6.9 (Jacobi-Verfahren)

(i) Wähle $M = \text{diag}(a_{11}, a_{22}, \dots, a_{nn})$.

(ii) Löse (2.21) bzgl. $x^{(i+1)}$:

$$x_j^{(i+1)} = \frac{1}{a_{jj}} \left(b_j - \sum_{k \neq j} a_{jk} x_k^{(i)} \right), \quad j = 1, \dots, n.$$

Ein einfach überprüfbares Kriterium zur Konvergenz des Verfahrens liefert der folgende Satz.

Satz 2.6.10 (Konvergenz des Gesamtschrittverfahrens)

Die Matrix $A \in \mathbb{K}^{n \times n}$ sei **strikt diagonaldominant**, d.h. es gelte

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, \dots, n.$$

Dann konvergiert das Gesamtschrittverfahren für jeden Startvektor $x^{(0)} \in \mathbb{R}^n$ gegen die Lösung \hat{x} von $Ax = b$ und es gilt die Fehlerabschätzung

$$\|x^{(i)} - \hat{x}\|_\infty \leq \frac{\|T\|_\infty^i}{1 - \|T\|_\infty} \|x^{(1)} - x^{(0)}\|_\infty, \quad i = 1, 2, \dots,$$

wobei $T = D^{-1}(L + R)$ den Gesamtschrittoperator bezeichnet.

Beweis: Die strikte Diagonaldominanz impliziert $\|T\|_\infty < 1$, wobei $\|\cdot\|_\infty$ die Zeilensummennorm bezeichnet. Die Konvergenz folgt dann aus Satz 2.6.7. Die Fehlerabschätzung folgt aus dem Banach'schen Fixpunktsatz. \square

Ein analoger Satz gilt für die Spaltensummennorm $\|\cdot\|_1$, wenn gefordert wird, daß

$$|a_{jj}| > \sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}|, \quad j = 1, \dots, n.$$

2.6.1.2 Einzelschrittverfahren (Gauss-Seidel-Verfahren)

Wir wählen in (2.17)

$$M = D - L, \quad N = R$$

und erhalten den **Einzelschrittoperator**

$$T = (D - L)^{-1}R.$$

Die Fixpunktiteration (2.23) für das Einzelschrittverfahren lautet somit

$$x^{(i+1)} = (D - L)^{-1} (Rx^{(i)} + b). \quad (2.25)$$

Algorithmus 2.6.11 (Gauss-Seidel-Verfahren)

(i) Wähle M als den unteren Dreiecksanteil von A , einschließlich Diagonale.

(ii) Löse (2.21) bzgl. $x^{(i+1)}$ mittels Vorwärtssubstitution:

$$x_j^{(i+1)} = \frac{1}{a_{jj}} \left(b_j - \sum_{k=1}^{j-1} a_{jk} x_k^{(i+1)} - \sum_{k=j+1}^n a_{jk} x_k^{(i)} \right), \quad j = 1, \dots, n.$$

Man beachte, daß die auf der rechten Seite auftretenden Komponenten $x_k^{(i+1)}$, $k < j$ der neuen Iterierten bereits zur Berechnung von $x_j^{(i+1)}$ verwendet werden.

Ein einfach überprüfbares Kriterium zur Konvergenz des Verfahrens liefert der folgende Satz.

Satz 2.6.12 (Konvergenz des Einzelschrittverfahrens)

Die Matrix $A \in \mathbb{K}^{n \times n}$ erfülle das **Sassenfeld-Kriterium**:

(i) Sämtliche Diagonalelemente von A sind $\neq 0$.

(ii) Es gilt $\beta = \max_{i=1, \dots, n} \beta_i < 1$ mit

$$\begin{aligned} \beta_1 &= \frac{1}{|a_{11}|} \sum_{j=2}^n |a_{1j}|, \\ \beta_i &= \frac{1}{|a_{ii}|} \left(\sum_{j=1}^{i-1} |a_{ij}| \beta_j + \sum_{j=i+1}^n |a_{ij}| \right), \quad i = 2, \dots, n. \end{aligned}$$

Dann konvergiert das Einzelschrittverfahren für jeden Startvektor $x^{(0)} \in \mathbb{K}^n$ gegen die Lösung \hat{x} von $Ax = b$ und es gilt die Fehlerabschätzung

$$\|x^{(i)} - \hat{x}\|_\infty \leq \frac{\beta^i}{1 - \beta} \|x^{(1)} - x^{(0)}\|_\infty, \quad i = 1, 2, \dots$$

Beweis: Wir zeigen, daß $\|T\|_\infty \leq \beta$ für den Einzelschrittoperator $T = (D - L)^{-1}R$ gilt.

Sei $x \in \mathbb{K}^n$ mit $\|x\|_\infty \leq 1$ und $y = Tx$, d.h. $(D - L)y = Rx$. Dann folgt

$$a_{11}y_1 = - \sum_{j=2}^n a_{1j}x_j \quad \Rightarrow \quad |y_1| \leq \frac{1}{|a_{11}|} \sum_{j=2}^n |a_{1j}| = \beta_1.$$

Die Behauptung folgt nun per Induktion. Sei also $|y_i| \leq \beta_i$ für $k = 1, 2, \dots, j-1$ bewiesen. Dann folgt

$$\sum_{k=1}^{j-1} a_{jk}y_k + a_{jj}y_j = - \sum_{k=j+1}^n a_{jk}x_k \quad \Rightarrow \quad |y_j| \leq \frac{1}{|a_{jj}|} \left(\sum_{k=1}^{j-1} |a_{jk}| \beta_k + \sum_{k=j+1}^n |a_{jk}| \right) = \beta_j.$$

Somit gilt also $|y_i| \leq \beta_i$ für $i = 1, 2, \dots, n$. Damit gilt $\|T\|_\infty = \sup_{\|x\|_\infty \leq 1} \|Tx\|_\infty \leq \beta$. Die Konvergenz folgt dann aus Satz 2.6.7. Die Fehlerabschätzung folgt aus dem Banach'schen Fixpunktsatz. \square

Beispiel 2.6.13 (Diskretisierung einer partiellen Differentialgleichung, Teil II)

Wir betrachten wiederum Beispiel 2.6.1 mit $N = 50$ und

$$f(x, y) = 1000 \cdot \sin((x - 0.5) \cdot (y - 0.5)).$$

Für $N = 50$ hat A im Gleichungssystem $Au = b$ die Dimension $49^2 = 2401$ und von den $2401^2 = 5764801$ Einträgen sind lediglich 11809 Einträge von Null verschieden.

Wir lösen das Gleichungssystem mit dem Gauss-Seidel-Verfahren, wählen $u^{(0)} = 0$ als Startwert und brechen die Iteration ab, wenn das relative Residuum

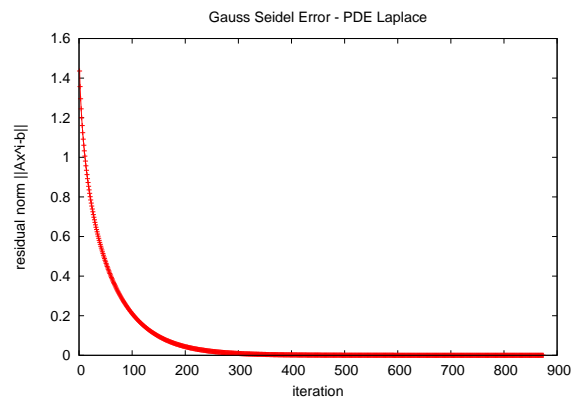
$$\frac{r_k}{r_0} = \frac{\|b - Au^{(k)}\|_2}{\|b - Au^{(0)}\|_2}, \quad k = 0, 1, 2, \dots$$

kleiner oder gleich der Toleranz 10^{-5} ist.

ITER RESIDUUM

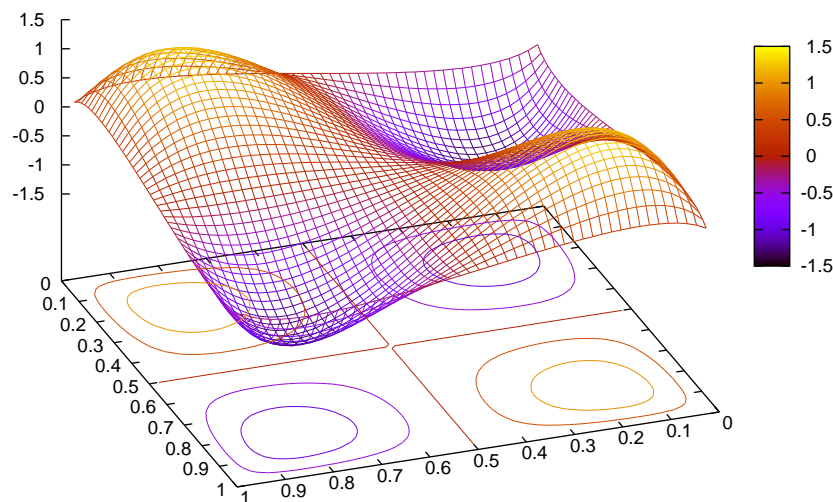
```
-----
1  1.43604533
2  1.35716494
3  1.29587804
4  1.24465554
5  1.20013794
6  1.16048560
7  1.12456585
```

```
...
870 1.61266564E-05
871 1.60113515E-05
872 1.58971717E-05
873 1.57841049E-05
874 1.56721393E-05
ITERATION SUCCESSFUL:
INITIAL ERROR: 1.56259882
FINAL ERROR: 1.55612629E-05
ITERATIONS: 875
```



Obige Abbildung zeigt das Residuum $\|b - Ax\|_2$ der Iterierten.

Die folgende Abbildung zeigt die mit dem Gauss-Seidel-Verfahren berechnete numerische Lösung u der partiellen Differentialgleichung:



Beispiel 2.6.14

Wir lösen das lineare Gleichungssystem

$$\begin{pmatrix} 2 & -1 & 0 \\ 1 & 6 & -2 \\ 4 & -3 & 8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ -4 \\ 5 \end{pmatrix}$$

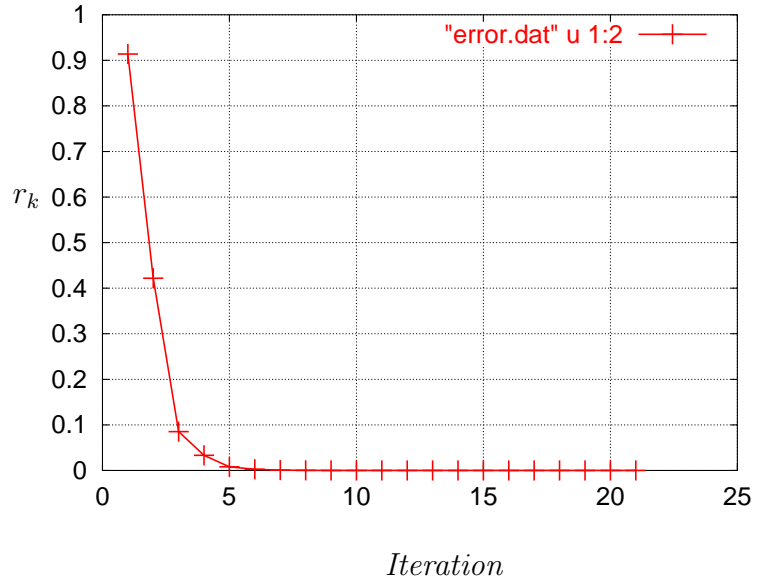
mit dem Gauss-Seidel-Verfahren und erhalten den folgenden Output:

IT NORM RESIDUUM

```

1  0.913821342
2  0.421932152
3  0.0851292437
4  0.033203378
5  0.00789585657
6  0.00256967289
7  0.000723273381
8  0.000195688235
9  6.52208342E-05
10 1.46979182E-05
11 5.78429145E-06
12 1.09568644E-06
13 5.04543055E-07
14 8.21821862E-08
15 4.32969302E-08
16 6.34648404E-09
17 3.65635959E-09
18 5.17553689E-10
19 3.03904484E-10
20 4.49659492E-11
21 2.48600765E-11
ITERATION SUCCESSFUL:
INITIAL ERROR: 6.70820393
FINAL ERROR: 4.09426575E-12
ITERATIONS: 22
APPROXIMATE SOLUTION: 0.62 -0.76 0.03

```



Die Abbildung zeigt das Residuum

$$r_k = \|b - Ax^{(k)}\|_2^2, \quad k = 0, 1, 2, \dots$$

Man erkennt sehr gut, dass der Fortschritt in den ersten Iterationen sehr groß ist, die Konvergenz des Verfahrens am Ende allerdings sehr langsam ist.

Häufig beobachtet man in der Praxis, daß das Einzelschrittverfahren schneller konvergiert als das Gesamtschrittverfahren.

2.6.1.3 SOR-Verfahren

Das SOR-Verfahren ist ein Relaxierungsverfahren, welches mit Hilfe eines Relaxationsparameters w eine gewichtete Lösung zwischen der Lösung des Einzelschrittverfahrens und der vorigen Iterierten $x^{(i)}$ berechnet.

Algorithmus 2.6.15 (Successive Overrelaxation (SOR))

(i) Wähle einen **Relaxationsparameter** $0 < w < 2$.

(ii) Die neue Iterierte $x^{(i+1)}$ ist gegeben durch eine gewichtete Summe der Iterierten des Gauss-Seidel-Verfahrens und der vorigen Iterierten $x^{(i)}$:

$$x_j^{(i+1)} = (1 - w)x_j^{(i)} + \frac{w}{a_{jj}} \left(b_j - \sum_{k=1}^{j-1} a_{jk}x_k^{(i+1)} - \sum_{k=j+1}^n a_{jk}x_k^{(i)} \right), \quad j = 1, \dots, n$$

Um den Zusammenhang mit dem Einzelschrittverfahren deutlicher hervorzuheben, kann die Iterationsvorschrift auch geschrieben werden als

$$\begin{aligned} z_j^{(i+1)} &= \frac{1}{a_{jj}} \left(b_j - \sum_{k=1}^{j-1} a_{jk} x_k^{(i+1)} - \sum_{k=j+1}^n a_{jk} x_k^{(i)} \right), \\ x_j^{(i+1)} &= w z_j^{(i+1)} + (1-w) x_j^{(i)}, \quad j = 1, \dots, n. \end{aligned}$$

In Matrixschreibweise entspricht dies mit dem Operator

$$T = (D - wL)^{-1} ((1-w)D + wR)$$

der Gleichung

$$x^{(i+1)} = (D - wL)^{-1} ((1-w)Dx^{(i)} + w(Rx^{(i)} + b)).$$

Für $w = 1$ ergibt sich das Einzelschrittverfahren.

Durch eine geeignete Wahl des Parameters w erhofft man sich eine höhere Konvergenzgeschwindigkeit im Vergleich zum Einzelschrittverfahren.

Ist A symmetrisch mit positiven Diagonalelementen, so konvergiert das SOR-Verfahren genau dann, wenn A positiv definit ist und $0 < w < 2$ gilt, vgl. Hämmerlin und Hoffmann [HH94, S. 386].

2.7 Das konjugierte Gradientenverfahren

Ein gänzlich anderer Ansatz zur Lösung des linearen Gleichungssystems (2.3) wird beim **konjugierten Gradientenverfahren** verfolgt. Wir beschränken uns hier auf den reellen Fall $\mathbb{K} = \mathbb{R}$. An Stelle des Gleichungssystems $Ax = b$ wird das Minimierungsproblem

$$\text{Minimiere } f(x) = \frac{1}{2} x^\top A x - b^\top x \tag{2.26}$$

betrachtet. Eine Lösung des Minimierungsproblems ist stets auch Lösung von (2.3), denn für ein Minimum von f ist die Bedingung

$$0 = \nabla f(x) = Ax - b$$

notwendig. Ist A positiv definit und symmetrisch, so sind beide Problemstellungen äquivalent. Für symmetrische, positiv definite Matrizen gilt das konjugierte Gradientenverfahren als eines der effizientesten Verfahren.

Das Verfahren arbeitet ebenfalls iterativ, läßt sich aber nicht als Fixpunktiteration darstellen. Im folgenden wird vorausgesetzt, daß A symmetrisch und positiv definit ist. Ausgehend von einer Startschätzung $x^{(0)}$ wird die nächste Iterierte

$$x^{(i+1)} = x^{(i)} + \alpha_i d^{(i)}, \quad i = 0, 1, 2, \dots$$

mit Hilfe einer **Suchrichtung** $d^{(i)}$ und einer **Schrittweite** α_i berechnet. Dabei werden Suchrichtung und Schrittweite so bestimmt, daß die Zielfunktion in jeder Iteration abnimmt.

2.7.1 Bestimmung der Schrittweite

Für eine gegebene Suchrichtung $d^{(i)}$ wird die Schrittweite α_i bestimmt, indem f in Richtung $d^{(i)}$ minimiert wird, vgl. Abbildung 2.1.

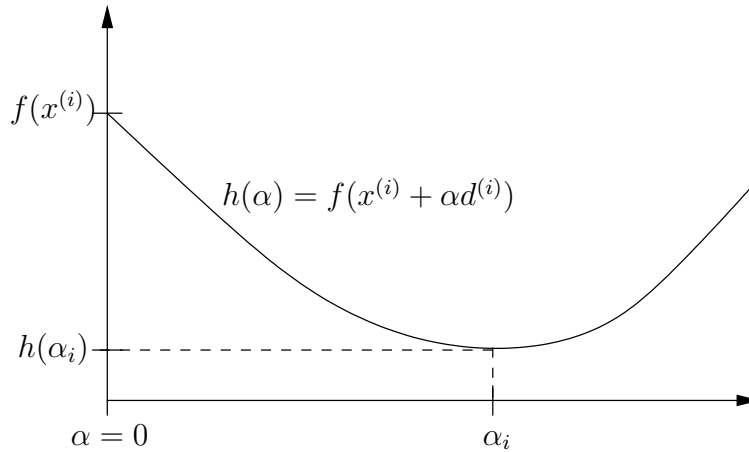


Abbildung 2.1: Liniensuche zur Bestimmung der Schrittweite α_i .

Dieser Vorgang wird **Liniensuche** genannt. Die Aufgabe besteht also darin, die Funktion

$$h(\alpha) := f(x^{(i)} + \alpha d^{(i)})$$

bezüglich α zu minimieren. Dies ist eine eindimensionale Optimierungsaufgabe mit eindeutiger Lösung, da A positiv definit ist. Berücksichtigt man die Struktur von f , kann man die optimale Schrittweite

$$\alpha_i = \frac{(b - Ax^{(i)})^\top d^{(i)}}{(d^{(i)})^\top A d^{(i)}} \quad (2.27)$$

ausrechnen, wobei der Nenner wegen der positiven Definitheit von A ungleich Null ist.

2.7.2 Bestimmung der Suchrichtung

Im ersten Schritt des Verfahrens wird die Richtung des steilsten Abstiegs verwendet. Bekanntlich zeigt der Gradient einer Funktion in Richtung des steilsten Anstiegs der Funktion. Der negative Gradient zeigt somit in Richtung des steilsten Abstiegs. Daher wird

$$d^{(0)} = -\nabla f(x^{(0)}) = -(Ax^{(0)} - b) = b - Ax^{(0)}$$

als erste Suchrichtung gewählt. Die weiteren Suchrichtungen werden so konstruiert, daß sie bzgl. des Skalarprodukts $\langle x, y \rangle_A := x^\top A y$ orthogonal zu den bisherigen Suchrichtungen sind.

Definition 2.7.1

Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Die Vektoren $s_1, \dots, s_m \in \mathbb{R}^n$, $s_i \neq 0$, $i = 1, \dots, m$, heißen **A-orthogonal** (oder **A-konjugiert**), falls gilt

$$\langle s_i, s_j \rangle_A = 0 \quad \text{für } i \neq j.$$

Beachte, daß A-orthogonale Richtungen linear unabhängig sind, denn

$$\sum_{i=1}^m \alpha_i s_i = 0 \quad \Rightarrow \quad 0 = s_k^\top A^\top \left(\sum_{i=1}^m \alpha_i s_i \right) = \alpha_k \underbrace{s_k^\top A^\top s_k}_{>0} \quad \Rightarrow \quad \alpha_k = 0.$$

2.7.3 Konvergenz

Damit ist das Verfahren vollständig beschrieben. Insbesondere bilden n A-orthogonale Richtungen eine Basis des \mathbb{R}^n und jeder Punkt des \mathbb{R}^n kann als Linearkombination von (maximal) n A-orthogonalen Richtungen dargestellt werden. Insbesondere besitzt auch das Minimum \hat{x} der Funktion f eine solche Darstellung. Der folgende Satz zeigt, daß nach spätestens n Schritten das Minimum gefunden wird.

Satz 2.7.2

Sei f gegeben durch (2.26) und $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Die Vektoren $d^{(0)}, \dots, d^{(n-1)}$ seien A-orthogonal und $x^{(0)} \in \mathbb{R}^n$ sei beliebig. Desweiteren gelte

$$x^{(i+1)} = x^{(i)} + \alpha_i d^{(i)}, \quad i = 0, \dots, n-1$$

mit exakter Schrittweite gemäß (2.27). Dann gilt

$$f(x^{(n)}) = \min_{x \in \mathbb{R}^n} f(x).$$

Beweis: (vgl. Jarre und Stoer [JS04])

Jeder Vektor $v \in \mathbb{R}^n$ läßt sich darstellen als

$$v = \sum_{j=0}^{n-1} \gamma_j d^{(j)} \quad \Rightarrow \quad (d^{(i)})^\top A v = \gamma_i \underbrace{(d^{(i)})^\top A d^{(i)}}_{>0} \quad \Rightarrow \quad \gamma_i = \frac{(d^{(i)})^\top A v}{(d^{(i)})^\top A d^{(i)}}.$$

Damit folgt

$$v = \sum_{j=0}^{n-1} \frac{(d^{(j)})^\top A v}{(d^{(j)})^\top A d^{(j)}} d^{(j)}. \quad (2.28)$$

Nach Wahl von α_i gemäß (2.27) folgt

$$\begin{aligned}
 \alpha_i &= \frac{(b - Ax^{(i)})^\top d^{(i)}}{(d^{(i)})^\top Ad^{(i)}} \\
 &= \frac{\left(b - A\left(x^{(0)} + \sum_{j=0}^{i-1} \alpha_j d^{(j)}\right)\right)^\top d^{(i)}}{(d^{(i)})^\top Ad^{(i)}} \\
 &= \frac{(b - Ax^{(0)})^\top d^{(i)} - \sum_{j=0}^{i-1} \alpha_j (d^{(j)})^\top Ad^{(i)}}{(d^{(i)})^\top Ad^{(i)}} \\
 &\stackrel{\langle d^{(j)}, d^{(i)} \rangle_A = 0}{=} - \frac{(d^{(i)})^\top (Ax^{(0)} - b)}{(d^{(i)})^\top Ad^{(i)}}.
 \end{aligned}$$

Wegen $x^{(n)} = x^{(0)} + \sum_{i=0}^{n-1} \alpha_i d^{(i)}$ folgt dann

$$x^{(n)} = x^{(0)} - \sum_{i=0}^{n-1} \frac{(d^{(i)})^\top (Ax^{(0)} - b)}{(d^{(i)})^\top Ad^{(i)}} d^{(i)} = x^{(0)} - \sum_{i=0}^{n-1} \frac{(d^{(i)})^\top \overbrace{A(x^{(0)} - A^{-1}b)}^{=:v}}{(d^{(i)})^\top Ad^{(i)}} d^{(i)}.$$

Definition von $v := x^{(0)} - A^{-1}b$ zusammen mit der Darstellung von v in (2.28) liefert

$$x^{(n)} = x^{(0)} - (x^{(0)} - A^{-1}b) = A^{-1}b = \arg \min_{x \in \mathbb{R}^n} f(x).$$

□

Der Satz liefert mit anderen Worten die Darstellung

$$\hat{x} = x^{(0)} + \sum_{k=0}^{n-1} \alpha_k d^{(k)}$$

für eine Lösung \hat{x} von $Ax = b$.

Es bleibt die Frage, wie A -orthogonale Richtungen berechnet werden können. Für eine beliebige Basis $\{v_1, \dots, v_n\}$ des \mathbb{R}^n kann dies beispielsweise durch das **Gram-Schmidtsche Orthogonalisierungsverfahren** bzgl. des Skalarprodukts $\langle \cdot, \cdot \rangle_A$ erfolgen. Die Aufgabe besteht hier darin, für linear unabhängige Vektoren v_1, \dots, v_n A -orthogonale Vektoren s_1, \dots, s_n zu berechnen. Zur Konstruktion geht man iterativ wie folgt vor. Zunächst wird $s_1 = v_1$ gesetzt. Sind bereits A -orthogonale Vektoren s_1, \dots, s_{k-1} bestimmt, so wird s_k über den Ansatz

$$s_k = -v_k + \sum_{j=1}^{k-1} \lambda_j s_j$$

mit noch zu bestimmenden Koeffizienten λ_j , $j = 1, \dots, k-1$, gewählt. Die Koeffizienten λ_j , $j = 1, \dots, k-1$, sind so zu bestimmen, daß $\langle s_\ell, s_k \rangle_A = 0$ für alle $\ell = 1, \dots, k-1$ gilt.

Daraus ergibt sich für $\ell = 1, \dots, k-1$ auf Grund der A-Orthogonalität von s_1, \dots, s_{k-1} die Bedingung

$$0 = \langle s_\ell, s_k \rangle_A = -\langle s_\ell, v_k \rangle_A + \sum_{j=1}^{k-1} \lambda_j \langle s_\ell, s_j \rangle_A = -\langle s_\ell, v_k \rangle_A + \lambda_\ell \langle s_\ell, s_\ell \rangle_A$$

und somit

$$\lambda_\ell = \frac{\langle s_\ell, v_k \rangle_A}{\langle s_\ell, s_\ell \rangle_A}, \quad \ell = 1, \dots, k-1.$$

Zusammenfassend entsteht

Algorithmus 2.7.3 (Gram-Schmidtsches Orthogonalisierungsverfahren)

Gegeben: linear unabhängige Vektoren v_1, \dots, v_n .

Ausgabe: paarweise A-orthogonale Vektoren s_1, \dots, s_n .

$s_1 := v_1$

for $k = 2, \dots, n$

$$s_k := -v_k + \sum_{j=1}^{k-1} \frac{\langle s_j, v_k \rangle_A}{\langle s_j, s_j \rangle_A} s_j$$

end

Allerdings legt dieses Verfahren die Suchrichtungen von vornherein fest und liefert i.a. auch keine Abstiegsrichtungen.

Besser ist es, die Richtungen schrittweise zu erzeugen und darauf zu achten, daß Abstiegsrichtungen entstehen (was insbesondere im Hinblick auf nicht-quadratische Funktionen von Bedeutung ist). Zur Abkürzung sei ab jetzt

$$g^{(i+1)} := \nabla f(x^{(i+1)}) = Ax^{(i+1)} - b = A(x^{(i)} + \alpha_i d^{(i)}) - b. \quad (2.29)$$

Wir starten mit $d^{(0)} = -\nabla f(x^{(0)}) = -g^{(0)}$ und gehen davon aus, daß bereits A-orthogonale Richtungen $d^{(0)}, \dots, d^{(i)}$ vorliegen. Zusätzlich sei $g^{(j)} \neq 0$, $0 \leq j \leq i+1$ (ansonsten wären wir fertig). Die Schrittweite α_i sei durch (2.27) gegeben und kann geschrieben werden als

$$\alpha_i = -\frac{(g^{(i)})^\top d^{(i)}}{(d^{(i)})^\top A d^{(i)}}. \quad (2.30)$$

Durch Einsetzen von α_i und $g^{(i+1)}$ zeigt man leicht, daß

$$(g^{(i+1)})^\top d^{(i)} = (Ax^{(i)} + \alpha_i A d^{(i)} - b)^\top d^{(i)} = 0. \quad (2.31)$$

(i) Aus $g^{(i+1)} - g^{(i)} = \alpha_i A d^{(i)}$ folgt für $j < i$ die Beziehung

$$(g^{(i+1)} - g^{(i)})^\top d^{(j)} = \alpha_i (d^{(i)})^\top A d^{(j)} = 0 \quad (2.32)$$

und damit auch

$$\begin{aligned} (g^{(i+1)})^\top d^{(j)} &= (g^{(j+1)})^\top d^{(j)} + \sum_{k=j+1}^i (g^{(k+1)} - g^{(k)})^\top d^{(j)} \\ &\stackrel{(2.31), (2.32)}{=} 0. \end{aligned}$$

Insgesamt folgt also

$$(g^{(i+1)})^\top d^{(j)} = 0 \quad \forall j \leq i. \quad (2.33)$$

(ii) Gilt $g^{(i+1)} \neq 0$, so bedeutet (2.33), daß

$$g^{(i+1)} \perp \text{span}(d^{(0)}, \dots, d^{(i)}). \quad (2.34)$$

Analog zur Gram-Schmidt-Orthogonalisierung machen wir den Ansatz

$$d^{(i+1)} = -g^{(i+1)} + \sum_{j=0}^i \frac{\langle d^{(j)}, g^{(i+1)} \rangle_A}{\langle d^{(j)}, d^{(j)} \rangle_A} d^{(j)}. \quad (2.35)$$

In der Tat folgt aus der A -Orthogonalität von $\{d^{(0)}, \dots, d^{(i)}\}$ die Beziehung $\langle d^{(i+1)}, d^{(j)} \rangle_A = 0$ für $j = 0, \dots, i$. Somit ist auch $\{d^{(0)}, \dots, d^{(i+1)}\}$ A -orthogonal.

Zudem ist $d^{(i+1)}$ auch Abstiegsrichtung, denn Multiplikation von links mit $(g^{(i+1)})^\top$ liefert zusammen mit (2.33) die Beziehung

$$(g^{(i+1)})^\top d^{(i+1)} = -\|g^{(i+1)}\|^2 < 0. \quad (2.36)$$

Beachte, daß (2.30) $\alpha_{i+1} > 0$ liefert.

(iii) Wir zeigen, daß in (2.35) alle Summanden mit $j < i$ verschwinden. Wegen (2.35) ist

$$g^{(i+1)} \in \text{span}(d^{(0)}, \dots, d^{(i+1)}) \quad \text{bzw.} \quad g^{(j)} \in \text{span}(d^{(0)}, \dots, d^{(j)}), \quad 0 \leq j \leq i+1.$$

Wegen (2.34) folgt

$$(g^{(i+1)})^\top g^{(j)} = 0 \quad \forall j \leq i. \quad (2.37)$$

Für $j < i$ folgt dann

$$(g^{(i+1)})^\top A d^{(j)} = \frac{1}{\alpha_j} (g^{(i+1)})^\top (g^{(j+1)} - g^{(j)}) = 0.$$

(2.35) reduziert sich somit zu

$$d^{(i+1)} = -g^{(i+1)} + \frac{\langle d^{(i)}, g^{(i+1)} \rangle_A}{\langle d^{(i)}, d^{(i)} \rangle_A} d^{(i)}. \quad (2.38)$$

Schließlich folgt mit $Ad^{(i)} = \frac{1}{\alpha_i} (g^{(i+1)} - g^{(i)})$:

$$\begin{aligned}
 (g^{(i+1)})^\top Ad^{(i)} &= \frac{1}{\alpha_i} (\|g^{(i+1)}\|^2 - (g^{(i+1)})^\top g^{(i)}) \\
 &\stackrel{(2.37)}{=} \frac{1}{\alpha_i} \|g^{(i+1)}\|^2, \\
 (d^{(i)})^\top Ad^{(i)} &\stackrel{(2.30)}{=} -\frac{1}{\alpha_i} (g^{(i)})^\top d^{(i)} \\
 &\stackrel{(2.36)}{=} \frac{1}{\alpha_i} \|g^{(i)}\|^2.
 \end{aligned}$$

Einsetzen in (2.38) liefert

$$d^{(i+1)} = -g^{(i+1)} + \frac{\|g^{(i+1)}\|^2}{\|g^{(i)}\|^2} d^{(i)}. \quad (2.39)$$

Die obigen Betrachtungen zusammen mit Satz 2.7.2 und der Tatsache, daß maximal n A -orthogonale Richtungen existieren, beweisen

Satz 2.7.4

Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch und positiv definit. Dann endet das obige Verfahren für einen beliebigen Startwert $x^{(0)} \in \mathbb{R}^n$ nach $m \leq n$ Schritten mit $Ax^{(m)} = b$.

Für das kleinste $m \leq n$ mit $Ax^{(m)} = b$ gelten für alle $l \leq m$ die Orthogonalitätsrelationen:

$$\begin{aligned}
 (d^{(i)})^\top Ad^{(j)} &= 0, & 0 \leq j < i \leq l, \\
 (g^{(i)})^\top g^{(j)} &= 0, & 0 \leq j < i \leq l, \text{ vgl. (2.37)} \\
 (g^{(i)})^\top d^{(j)} &= 0, & 0 \leq j < i \leq l, \text{ vgl. (2.33)} \\
 (g^{(i)})^\top d^{(i)} &= -\|g^{(i)}\|^2, & 0 \leq i \leq l, \text{ vgl. (2.36)}.
 \end{aligned}$$

Die obigen Betrachtungen liefern ein konstruktives Verfahren zur Berechnung der A -orthogonalen Suchrichtungen. Zusammen mit Satz 2.7.2 folgt daraus die Endlichkeit des folgenden Algorithmus:

Algorithmus 2.7.5 (CG-Verfahren)

(i) Wähle einen Startvektor $x^{(0)} \in \mathbb{R}^n$, $\varepsilon \geq 0$, berechne $g^{(0)} = Ax^{(0)} - b$ und setze $d^{(0)} = -g^{(0)}$, $i = 0$.

(ii) Falls $\|g^{(i)}\| \leq \varepsilon$, STOP.

(iii) Berechne

$$\begin{aligned}\alpha_i &= \frac{\|g^{(i)}\|^2}{(d^{(i)})^\top Ad^{(i)}}, \\ x^{(i+1)} &= x^{(i)} + \alpha_i d^{(i)}, \\ g^{(i+1)} &= g^{(i)} + \alpha_i Ad^{(i)}, \\ \beta_i &= \frac{\|g^{(i+1)}\|^2}{\|g^{(i)}\|^2}, \\ d^{(i+1)} &= -g^{(i+1)} + \beta_i d^{(i)}.\end{aligned}$$

(iv) Setze $i := i + 1$, und gehe zu (ii).

In der Praxis führt man häufig aufgrund von auftretenden Rundungsfehlern nach einer gewissen Anzahl von Iterationen einen Restart des Verfahrens durch.

Beispiel 2.7.6 (Diskretisierung einer partiellen Differentialgleichung, Teil III)

Wir betrachten wiederum Beispiel 2.6.1 mit $N = 100$ und

$$f(x, y) = 1000 \cdot \sin((x - 0.5) \cdot (y - 0.5)).$$

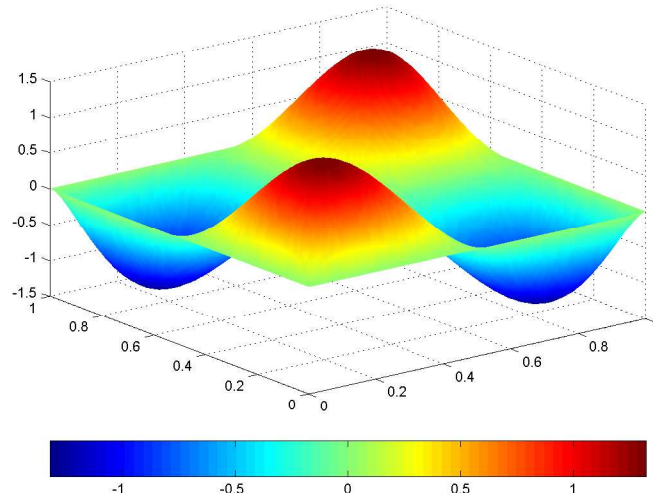
Die partielle Differentialgleichung wird wie zuvor diskretisiert und führt auf das lineare Gleichungssystem $Au = b$. Anschließend wird das äquivalente Problem

$$\frac{1}{2}u^\top Au - b^\top u \rightarrow \min$$

mit dem CG-Verfahren gelöst. Dabei werden nur die von Null verschiedenen Einträge von A mit einer entsprechenden Indizierung gespeichert, so daß Multiplikationen Ad sehr effizient durchgeführt werden können. Für $N = 100$ hat A die Dimension $99^2 = 9801$ und von den $9801^2 = 96059601$ Einträgen sind lediglich 48609 Einträge von Null verschieden. Als Startwert wählen wir $u^{(0)} = 0$. Für $N = 100$ und $\varepsilon = 10^{-8}$ ergibt sich folgende Ausgabe:

ITER	GRAD	DX	ALPHA	BETA
0	0.8055956E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
1	0.2271664E+01	0.6796640E+01	0.8287775E+01	0.7951592E+01
3	0.2274125E+01	0.7564077E+01	0.7990598E+00	0.1092265E+01
5	0.2100902E+01	0.7055373E+01	0.5045940E+00	0.8358480E+00
7	0.1932038E+01	0.6517664E+01	0.4768494E+00	0.8424729E+00
9	0.1767712E+01	0.5963891E+01	0.4706184E+00	0.8557311E+00
11	0.1611152E+01	0.5421028E+01	0.4690171E+00	0.8656038E+00
13	0.1463812E+01	0.4902665E+01	0.4686974E+00	0.8722568E+00
15	0.1326138E+01	0.4414901E+01	0.4687291E+00	0.8765485E+00
17	0.1198085E+01	0.3960077E+01	0.4688082E+00	0.8791410E+00
19	0.1079372E+01	0.3538615E+01	0.4688279E+00	0.8804796E+00
21	0.9696041E+00	0.3149930E+01	0.4687504E+00	0.8808559E+00
23	0.8683385E+00	0.2792889E+01	0.4685635E+00	0.8804600E+00
25	0.7751144E+00	0.2466078E+01	0.4682637E+00	0.8794144E+00
...				
135	0.3018280E-06	0.5503303E-06	0.4188414E+00	0.6769555E+00
137	0.2309873E-06	0.4445599E-06	0.4462299E+00	0.7570287E+00
139	0.1747984E-06	0.3426776E-06	0.4463246E+00	0.7220239E+00
141	0.1093353E-06	0.1928130E-06	0.4093333E+00	0.6062993E+00
143	0.6981008E-07	0.1201987E-06	0.4214496E+00	0.6967370E+00
145	0.5219784E-07	0.9674677E-07	0.4486677E+00	0.7779052E+00
147	0.3285968E-07	0.5685132E-07	0.3985365E+00	0.6319190E+00
149	0.2182741E-07	0.3802979E-07	0.4169753E+00	0.7148059E+00
151	0.1323888E-07	0.2172648E-07	0.3884909E+00	0.5765313E+00
153	0.8588212E-08	0.1449120E-07	0.4170621E+00	0.7137157E+00

Die folgende Abbildung zeigt die numerische Lösung u der partiellen Differentialgleichung:



Bemerkung 2.7.7

Kelley [Kel95], Th. 2.2.3, S. 15, zeigt, daß das CG-Verfahren das Minimum in höchstens k Schritten liefert, wobei k die Anzahl der verschiedenen Eigenwerte von A bezeichnet.

Obwohl das CG-Verfahren (zumindest theoretisch) nach maximal n Schritten terminiert, ist es in der Praxis aus Aufwandsgründen oft nicht möglich, n Iterationen tatsächlich durchzuführen (n kann sehr groß sein, etwa 1 Milliarde).

Daher ist es sinnvoll, das CG-Verfahren als iteratives Verfahren zu betrachten. Es stellt sich dann die Frage nach der Konvergenzgeschwindigkeit. Es läßt sich folgende Abschätzung beweisen (vgl. Kelley [Kel95, S. 17]):

$$\|x^{(i)} - \hat{x}\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^i \|x^{(0)} - \hat{x}\|_A, \quad (2.40)$$

wobei $\kappa(A) = \lambda_{\max}/\lambda_{\min}$ die Kondition der Matrix A bzgl. der Norm $\|\cdot\|_2$ bezeichnet.

Die Fehlerabschätzung zeigt, daß das Verfahren umso schneller konvergiert, je näher die Kondition $\kappa(A)$ bei Eins liegt. Umgekehrt ist die Konvergenz umso langsamer, je größer die Kondition von A ist. Es ist daher erstrebenswert, eine möglichst gut konditionierte Matrix A zu haben, was durch die sogenannte **Vorkonditionierung** (oder **Präkonditionierung**) erreicht werden kann.

Hierbei wird eine Variablentransformation durchgeführt:

$$\begin{aligned} x &:= Sz, \quad S \in \mathbb{R}^{n \times n} \text{ regulär,} \\ \hat{f}(z) &:= f(Sz) = \frac{1}{2} z^\top S^\top A S z - (S^\top b)^\top z. \end{aligned}$$

Natürlich sollte S so gewählt werden, daß die Kondition von $S^\top A S$ kleiner ist als die von

A. Nun wenden wir das übliche CG-Verfahren auf \hat{f} an und erhalten die Größen

$$\hat{x}^{(i)}, \quad \hat{g}^{(i)}, \quad \hat{d}^{(i)}, \quad \hat{\alpha}_i, \quad \hat{\beta}_i.$$

Rücksubstitution liefert

$$x^{(i)} = S\hat{x}^{(i)}, \quad S^\top g^{(i)} = \hat{g}^{(i)}, \quad d^{(i)} = S\hat{d}^{(i)}.$$

Unter Berücksichtigung der Rücksubstitution ergibt sich:

Algorithmus 2.7.8 (Präkonditioniertes CG-Verfahren)

(i) Wähle eine symmetrische und positiv definite Matrix $B \in \mathbb{R}^{n \times n}$, einen Startvektor $x^{(0)} \in \mathbb{R}^n$, $\varepsilon \geq 0$, berechne $g^{(0)} = Ax^{(0)} - b$ und setze $d^{(0)} = -Bg^{(0)}$, $i = 0$.

(ii) Falls $\|g^{(i)}\| \leq \varepsilon$, STOP.

(iii) Berechne

$$\begin{aligned} \hat{\alpha}_i &= \frac{(g^{(i)})^\top Bg^{(i)}}{(d^{(i)})^\top Ad^{(i)}}, \\ x^{(i+1)} &= x^{(i)} + \hat{\alpha}_i d^{(i)}, \\ g^{(i+1)} &= g^{(i)} + \hat{\alpha}_i Ad^{(i)}, \\ \hat{\beta}_i &= \frac{(g^{(i+1)})^\top Bg^{(i+1)}}{(g^{(i)})^\top Bg^{(i)}}, \\ d^{(i+1)} &= -Bg^{(i+1)} + \hat{\beta}_i d^{(i)}. \end{aligned}$$

(iv) Setze $i := i + 1$, und gehe zu (ii).

Im Algorithmus tritt nur noch die symmetrische und positiv definite Matrix $B = SS^\top$ auf. Wir wählen B als Approximation an A^{-1} . Dann ist $BA \approx I$. Die Eigenwerte von BA liegen dann nahe bei 1. Wegen

$$S^{-1}BAS = S^{-1}SS^\top AS = S^\top AS$$

ist BA ähnlich zu $S^\top AS$ und folglich besitzen BA und $S^\top AS$ dieselben Eigenwerte. Da $S^\top AS$ symmetrisch ist, gilt die Formel $\kappa_2(S^\top AS) = \lambda_{\max}/\lambda_{\min}$, wobei λ_{\max} und λ_{\min} den betragsmäßig größten bzw. kleinsten Eigenwert von $S^\top AS$ bezeichnen. Da die Eigenwerte nahe bei 1 liegen ist auch die Kondition nahe bei 1.

Desweiteren sollte B so gewählt werden, daß die Multiplikationen $g \mapsto Bg$ kostengünstig ausgewertet werden können.

Mögliche Ansätze für B sind:

- (a) $B = D^{-1}$, wobei D die Diagonale von A bezeichnet.
- (b) Berechne die Cholesky-Zerlegung $A = L \cdot L^\top$, approximiere L durch Weglassen kleiner Elemente durch \hat{L} und setze $B = \hat{L}^{-\top} \hat{L}^{-1}$. Dann gilt $BA = \hat{L}^{-\top} \hat{L}^{-1} L L^\top \approx I$. Dieses Verfahren ist brauchbar für dünn besetzte Matrizen.

Kapitel 3

Lineare Ausgleichsprobleme

Häufig gilt es, über- oder unterbestimmte Gleichungssysteme zu lösen.

Beispiel 3.0.1 (Lineares Ausgleichsproblem)

Die Messung eines physikalischen Vorgangs hat an den Ortspunkten $p_i = (p_{1,i}, p_{2,i})$, $i = 1, \dots, 9$, die folgenden Meßwerte y_i , $i = 1, \dots, 9$, ergeben:

Meßpunkt $i =$	1	2	3	4	5	6	7	8	9
$p_{1,i}$	6	7	9	11	13	15	17	18	19
$p_{2,i}$	0	1	2	3	4	5	6	7	8
y_i	96	189	283	373	467	553	647	733	832

Es wird ein affin-linearer Zusammenhang zwischen den Meßpunkten und den Meßwerten vermutet und folgender Ansatz gewählt

$$y(p_1, p_2) = x_1 + x_2 p_1 + x_3 p_2,$$

wobei $x = (x_1, x_2, x_3)^\top \in \mathbb{R}^3$ unbekannt ist. Ein naheliegender Versuch, x zu bestimmen, besteht darin, das lineare Gleichungssystem

$$y_i = y(p_{1,i}, p_{2,i}) = x_1 + x_2 p_{1,i} + x_3 p_{2,i}, \quad i = 1, \dots, 9,$$

zu lösen. In Matrixschreibweise lautet dies $Ax = b$ mit

$$b = \begin{pmatrix} 96 \\ 189 \\ 283 \\ 373 \\ 467 \\ 553 \\ 647 \\ 733 \\ 832 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 6 & 0 \\ 1 & 7 & 1 \\ 1 & 9 & 2 \\ 1 & 11 & 3 \\ 1 & 13 & 4 \\ 1 & 15 & 5 \\ 1 & 17 & 6 \\ 1 & 18 & 7 \\ 1 & 19 & 8 \end{pmatrix}.$$

Offenbar ist dieses lineare Gleichungssystem überbestimmt, da die Anzahl der Gleichungen die Anzahl der Unbekannten übersteigt. Leider stellt sich bei näherer Betrachtung heraus, daß das Gleichungssystem **keine Lösung** besitzt. Der Versuch, den Vektor x über die Lösung des Gleichungssystems $Ax = b$ zu bestimmen, schlägt also fehl.

Ein sinnvoller Zugang ist es daher, den Vektor x so zu wählen, daß der Fehler im Gleichungssystem $\|Ax - b\|_2^2$ möglichst klein wird. Dies führt auf das **Minimierungsproblem**

$$\|Ax - b\|_2^2 \rightarrow \min.$$

Dieses Problem ist auch als **lineares Ausgleichsproblem** oder **Least-Squares-Problem** bekannt, welches bereits auf Gauss zurück geht.



JOHANN CARL FRIEDRICH GAUSS
 Born: 30.4.1777 in Braunschweig (Germany)
 Died: 23.2.1855 in Göttingen (Germany)

In diesem Kapitel beschäftigen wir uns mit dem folgenden Problem.

Problem 3.0.2 (Lineares Ausgleichsproblem)

Gegeben seien eine Matrix $A \in \mathbb{R}^{m \times n}$ und ein Vektor $b \in \mathbb{R}^m$ mit $m, n \in \mathbb{N}$. Bestimme die Lösung $x = (x_1, \dots, x_n)^\top \in \mathbb{R}^n$ des Minimierungsproblems

$$\text{Minimiere} \quad f(x) := \frac{1}{2} \|Ax - b\|_2^2 \quad (3.1)$$

Bemerkung 3.0.3

- Im Fall $m = n$ und A invertierbar, ist die Lösung eindeutig durch das lineare Gleichungssystem $Ax = b$ bestimmt.
- Interessant ist insbesondere der Fall $m > n$, der in praktischen Anwendungen besonders häufig auftritt, zumeist ist m dort wesentlich grösser als n .
- Der Fall $m < n$ kann ebenfalls auftreten. In diesem Fall ist das Gleichungssystem $Ax = b$ unterbestimmt, es kann jedoch unlösbar sein, wenn $\text{Rang}(A) \neq \text{Rang}(A|b)$.
- Anstatt der Norm $\|\cdot\|_2$ können auch andere Normen verwendet werden, etwa $\|\cdot\|_\infty$ oder $\|\cdot\|_1$, jedoch führt dies i.a. auf andere Lösungen und die Berechnung von

Lösungen gestaltet sich mitunter schwierig, da die Funktion f i.a. nicht mehr differenzierbar ist.

3.1 Gauss'sche Normalengleichungen

Wir betrachten das lineare Ausgleichsproblem von einem geometrischen Standpunkt. Dazu definieren wir den linearen Vektorraum

$$V := \{y \in \mathbb{R}^m \mid y = Ax, x \in \mathbb{R}^n\} = \text{Bild}(A).$$

V ist ein Teilraum des \mathbb{R}^m mit Dimension $\dim(V) = \text{Rang}(A)$. Das lineare Ausgleichsproblem lautet somit:

Finde $\hat{y} \in V$, so daß

$$\|b - \hat{y}\|_2 \leq \|b - y\|_2 \quad \forall y \in V. \quad (3.2)$$

Definition 3.1.1

Sei $(X, \|\cdot\|)$ ein normierter Vektorraum, $V \subset X$ ein Teilraum und $b \in X$. $\hat{y} \in V$ heißt **Bestapproximierende an b in V** , wenn (3.2) gilt.

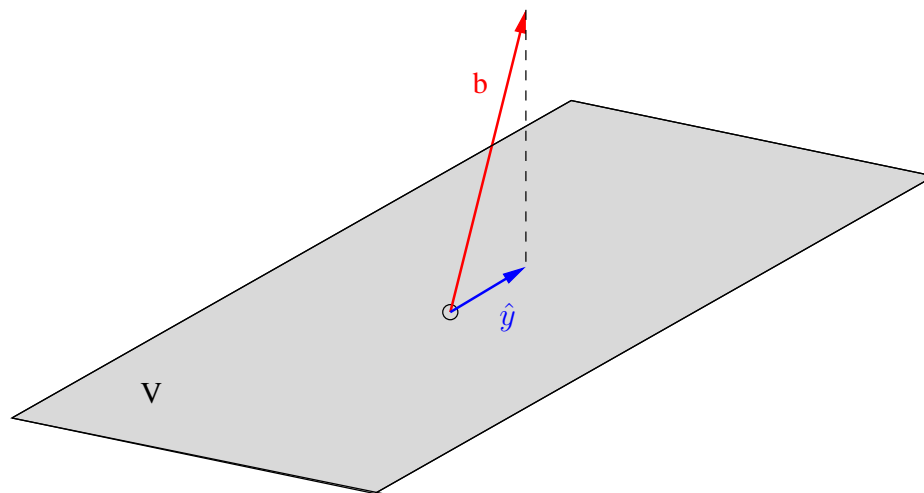


Abbildung 3.1: Bestapproximierende \hat{y} an b in V .

Aus der geometrischen Anschauung ist klar, daß \hat{y} genau dann Bestapproximierende an b in V ist, wenn $b - \hat{y}$ senkrecht auf V steht, vgl. Abbildung 3.1. Dies besagt der folgende Satz, der in recht allgemeiner Form gilt.

Satz 3.1.2 (Projektionssatz)

Sei X ein Vektorraum über $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$ und $\langle \cdot, \cdot \rangle$ ein Skalarprodukt auf $X \times X$. Sei $b \in X$ und $V \subset X$ ein Teilraum. Dann gelten:

(i) Es existiert höchstens eine Bestapproximierende \hat{y} an b in V .

(ii) $\hat{y} \in V$ ist Bestapproximierende an b in V genau dann, wenn

$$\langle b - \hat{y}, v \rangle = 0 \quad \forall v \in V. \quad (3.3)$$

Beweis: Wir zeigen zunächst (ii). Teil (i) ergibt sich unterwegs.

\Rightarrow : Sei \hat{y} Bestapproximierende an b in V . Angenommen, (3.3) gilt nicht. Dann gibt es ein $v \in V$ mit $\langle b - \hat{y}, v \rangle = a \neq 0$. Insbesondere ist $v \neq 0$. Setze $w = \hat{y} + a \frac{v}{\|v\|^2} \in V$. Es folgt

$$\|b - w\|^2 = \langle b - \hat{y} - \frac{av}{\|v\|^2}, b - \hat{y} - \frac{av}{\|v\|^2} \rangle = \|b - \hat{y}\|^2 - \frac{|a|^2}{\|v\|^2} < \|b - \hat{y}\|^2.$$

Dies ist ein Widerspruch dazu, daß \hat{y} Bestapproximierende ist.

\Leftarrow : Es gelte (3.3). Dann gilt für alle $v \in V$ mit $v \neq \hat{y}$,

$$\begin{aligned} \|b - v\|^2 &= \langle b - v, b - v \rangle \\ &= \langle b - \hat{y} + \hat{y} - v, b - \hat{y} + \hat{y} - v \rangle \\ &= \|b - \hat{y}\|^2 + \|\hat{y} - v\|^2 + \langle b - \hat{y}, \hat{y} - v \rangle + \langle \hat{y} - v, b - \hat{y} \rangle \\ &= \|b - \hat{y}\|^2 + \|\hat{y} - v\|^2 \\ &> \|b - \hat{y}\|^2. \end{aligned}$$

Damit ist \hat{y} Bestapproximierende. (i) ist hiermit ebenfalls gezeigt, da obige Abschätzung auf einen Widerspruch führt, wenn man annimmt, daß es zwei unterschiedliche Bestapproximierende gibt. □

Anwendung des Projektionssatzes 3.1.2 auf das lineare Ausgleichsproblem liefert folgenden Satz:

Satz 3.1.3 (Gauss'sche Normalengleichungen)

$\hat{x} \in \mathbb{R}^n$ löst das lineare Ausgleichsproblem genau dann, wenn die **Gauss'schen Normalengleichungen**

$$A^\top A \hat{x} = A^\top b$$

gelten.

Beweis: Mit $\hat{y} = A\hat{x}$ und $v = Ax$ lautet (3.3):

$$0 = \langle b - A\hat{x}, Ax \rangle = (b - A\hat{x})^\top Ax = (A^\top b - A^\top A\hat{x})^\top x.$$

Da diese Gleichung für alle $x \in \mathbb{R}^n$ gelten muß, muß zwangsläufig $A^\top A \hat{x} - A^\top b = 0$ gelten. \square

Die Gauss'schen Normalengleichungen charakterisieren also die Lösung des linearen Ausgleichsproblems. Es gilt zu klären, wann die Normalengleichungen (bzw. das lineare Ausgleichsproblem) eine eindeutige Lösung besitzen. Offenbar ist $A^\top A$ symmetrisch und positiv semidefinit.

Satz 3.1.4

Die Gauss'schen Normalengleichungen besitzen genau dann eine eindeutige Lösung, wenn $\text{Rang}(A) = n$ ist.

Beweis: Zu zeigen ist, daß $A^\top A \in \mathbb{R}^{n \times n}$ positiv definit und somit invertierbar ist. Es gilt $\text{Rang}(A) = \text{Rang}(A^\top A)$. Beweis: Wegen $Ax = 0 \Rightarrow A^\top Ax = 0$ gilt $\text{Kern}(A) \subseteq \text{Kern}(A^\top A)$. Sei $x \in \text{Kern}(A^\top A)$, also $A^\top Ax = 0$. Dann ist $\langle A^\top Ax, x \rangle = \langle Ax, Ax \rangle = \|Ax\|^2 = 0$, also $x \in \text{Kern}(A)$. Insgesamt: $\text{Kern}(A) = \text{Kern}(A^\top A)$. Aus der linearen Algebra weiß man: $\text{Rang}(A^\top A) = n - \dim(\text{Kern}(A^\top A)) = n - \dim(\text{Kern}(A)) = n - (n - \text{Rang}(A)) = \text{Rang}(A)$. Damit gilt $\text{Rang}(A^\top A) = n$ genau dann, wenn $\text{Rang}(A) = n$ gilt.

Alternativer Beweis ($m \geq n$): $0 = x^\top A^\top Ax = \|Ax\|_2^2$ gilt genau dann, wenn $Ax = 0$ gilt. Gilt $\text{Rang}(A) = n$, sind die Spalten von A linear unabhängig. Damit gilt $Ax = 0$ genau dann, wenn $x = 0$ gilt. Gilt andererseits $Ax = 0$ für ein $x \neq 0$, so sind die Spalten von A linear abhängig und $\text{Rang}(A) < n$. \square

Beispiel 3.1.5 (vgl. Beispiel 3.0.1)

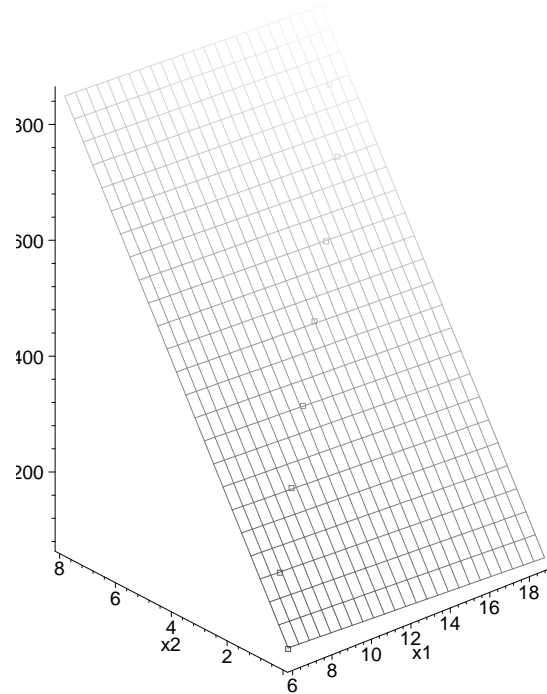
Wir betrachten wiederum Beispiel 3.0.1 und lösen das lineare Ausgleichsproblem (3.1) mit Hilfe der Normalengleichungen $A^\top Ax = A^\top b$. Es gilt

$$A^\top A = \begin{pmatrix} 9 & 115 & 36 \\ 115 & 1655 & 565 \\ 36 & 565 & 204 \end{pmatrix}, \quad A^\top b = \begin{pmatrix} 4173 \\ 62916 \\ 22176 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

Die Lösung der Normalengleichungen lautet

$$x_1 = \frac{533}{5} = 106.6, \quad x_2 = -\frac{96}{65} \approx -1.476923077, \quad x_3 = \frac{6109}{65} \approx 93.98461538.$$

Grafisch ergibt sich folgendes:



Ein Nachteil der Gauss'schen Normalgleichungen liegt in der möglicherweise schlechten Kondition der Matrix $A^T A$ im Vergleich zur Matrix A , da für quadratische Matrizen gemäß Satz 2.5.15 $\kappa_2(A^T A) = \kappa_2(A)^2$ gilt. Aus diesem Grund erfolgt die numerische Lösung des Ausgleichsproblems in der Regel nicht über die Normalgleichungen, sondern wird mit Hilfe einer QR-Zerlegung der Matrix A durchgeführt. Eigenschaften und Verfahren zur Berechnung einer QR-Zerlegung werden in den folgenden Abschnitten untersucht.

3.2 QR-Zerlegung einer Matrix

Ähnlich zur LR-Zerlegung beim Gauss'schen Eliminationsverfahren ist es unser Ziel, eine nicht notwendig quadratische Matrix $A \in \mathbb{R}^{m \times n}$ zu faktorisieren in $A = Q \cdot R$.

Definition 3.2.1 (QR-Zerlegung)

Sei $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ gegeben. Eine Zerlegung der Form $A = Q \cdot R$ mit $Q \in \mathbb{R}^{m \times m}$ und $R \in \mathbb{R}^{m \times n}$, wobei Q orthogonal ist und R die Gestalt

$$R = \begin{pmatrix} \bar{R} \\ 0 \end{pmatrix}$$

mit einer invertierbaren rechten oberen Dreiecksmatrix $\bar{R} \in \mathbb{R}^{n \times n}$ hat, heißt **QR-Zerlegung von A** .

Wir werden sehen, daß eine solche QR-Zerlegung stets möglich ist, wenn $\text{Rang}(A) = n$ gilt. Eine QR-Zerlegung ist sehr nützlich für

- die Lösung von linearen Gleichungssystemen;
- die Lösung von linearen Ausgleichsproblemen;
- die numerische Rangbestimmung einer Matrix.

Wir beschränken die Darstellung auf den Fall $\mathbb{K} = \mathbb{R}$. Im komplexen Fall gelten die Beziehungen analog.

3.2.1 QR-Zerlegung zur Lösung von linearen Gleichungssystemen

Ist eine QR-Zerlegung von A bekannt, so kann das lineare Gleichungssystem (2.3) im Fall $n = m$ gelöst werden, indem die Orthogonalität von Q ausgenutzt wird:

$$b = Ax = Q \cdot Rx \quad \Leftrightarrow \quad c := Q^\top b = Rx.$$

Das lineare Gleichungssystem $Rx = c$ kann wiederum mit Rückwärtssubstitution gelöst werden. Der Hauptvorteil der QR-Zerlegung im Vergleich zur LR-Zerlegung liegt in der Tatsache begründet, daß orthogonale Transformationen invariant bzgl. der Spektralnorm sind. Für einen beliebigen Vektor x gilt

$$\|Qx\|_2 = \|x\|_2, \quad \|Q\|_2 = 1.$$

Speziell gilt $\|Q^\top b\|_2 = \|b\|_2$ und somit werden durch Multiplikation mit Q^\top möglicherweise vorhandene Fehler in b nicht verstärkt. Für den relativen Fehler in (2.16) erhält man für die LR-Zerlegung wegen $\kappa_2(A) = \kappa_2(L \cdot R) \leq \kappa_2(L) \cdot \kappa_2(R)$ die im Vergleich zu (2.16) „schlechtere“ Abschätzung

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \kappa_2(L) \cdot \kappa_2(R) \cdot \frac{\|\Delta b\|_2}{\|b\|_2}.$$

Bei Verwendung der QR-Zerlegung gilt $\kappa_2(A) = \kappa_2(Q \cdot R) = \kappa_2(R)$ und somit folgt

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \kappa_2(R) \cdot \frac{\|\Delta b\|_2}{\|b\|_2}.$$

Fazit: Die QR-Zerlegung ist numerisch stabiler als die LR-Zerlegung. Der Nachteil der QR-Zerlegung liegt im etwa doppelt so hohen Aufwand im Vergleich zur LR-Zerlegung.

Bemerkung 3.2.2

Die rechte obere Dreiecksmatrix R in der LR-Zerlegung von A ist nicht zu verwechseln mit der rechten oberen Dreiecksmatrix R , die bei der QR-Zerlegung berechnet wird. Die jeweiligen Matrizen sind im allgemeinen verschieden.

3.2.2 Lösung des linearen Ausgleichsproblems mittels QR-Zerlegung

Wir wenden uns wieder dem linearen Ausgleichsproblem (3.1) zu und möchten die Funktion $f(x) = \frac{1}{2}\|Ax - b\|_2^2$ minimieren, wobei $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ und $b \in \mathbb{R}^m$ gegeben sind.

Wir setzen voraus, daß eine QR-Zerlegung von A gegeben ist gemäß

$$A = Q \cdot R, \quad R = \begin{pmatrix} \bar{R} \\ 0 \end{pmatrix}, \quad Q \in \mathbb{R}^{m \times m}, R \in \mathbb{R}^{m \times n}.$$

Hierin sei $\bar{R} \in \mathbb{R}^{n \times n}$ eine invertierbare rechte obere Dreiecksmatrix und Q eine orthogonale Matrix. Wegen der Invarianz der Euklidnorm gilt

$$\|Ax - b\|_2^2 = \|Q^\top(Ax - b)\|_2^2 = \|Rx - Q^\top b\|_2^2.$$

Definiere

$$c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} := Q^\top b, \quad c_1 \in \mathbb{R}^n, c_2 \in \mathbb{R}^{m-n}.$$

Damit folgt

$$\|Rx - Q^\top b\|_2^2 = \|\bar{R}x - c_1\|_2^2 + \|c_2\|_2^2.$$

Dieser Ausdruck wird genau dann minimal, wenn $\bar{R}x = c_1$ gilt, welches ein lineares Gleichungssystem für x darstellt und mittels Rückwärtssubstitution gelöst werden kann. Die Lösung dieses Gleichungssystems

$$\hat{x} = \bar{R}^{-1}c_1$$

ist auch die Lösung der Minimierungsaufgabe und somit des linearen Ausgleichsproblems. Der Wert des Zielfunktional kann ebenfalls direkt abgelesen werden und lautet

$$f(\hat{x}) = \frac{1}{2}\|c_2\|_2^2.$$

Diese Vorgehensweise zur Lösung des Ausgleichsproblems ist dem Weg über die Gauss'schen Normalengleichungen aus Stabilitätsgründen vorzuziehen, da die Kondition der Matrix $A^\top A$ häufig sehr schlecht ist, während bei Verwendung der QR-Zerlegung bzgl. der Euklidnorm keine zusätzliche Fehlerverstärkung durch den Faktor Q auftritt. In den folgenden Abschnitten werden Algorithmen zur Berechnung einer QR-Zerlegung diskutiert.

3.2.3 QR-Zerlegung durch Gram-Schmidt'sche Orthogonalisierung

Zunächst widmen wir uns der Frage nach der Eindeutigkeit einer QR-Zerlegung. Das Beispiel

$$\begin{pmatrix} 5 & 7 \\ -5 & -7 \\ -1 & 1 \end{pmatrix} \approx \begin{pmatrix} -0.700 & 0.099 & 0.707 \\ 0.700 & -0.099 & 0.707 \\ 0.140 & 0.990 & 0 \end{pmatrix} \begin{pmatrix} -7.140 & -9.660 \\ 0 & 2.380 \\ 0 & 0 \end{pmatrix} \\ \approx \begin{pmatrix} 0.700 & 0.099 & 0.707 \\ -0.700 & -0.099 & 0.707 \\ -0.140 & 0.990 & 0 \end{pmatrix} \begin{pmatrix} 7.140 & 9.660 \\ 0 & 2.380 \\ 0 & 0 \end{pmatrix}$$

zeigt, daß es i.a. mehrere QR-Zerlegungen derselben Matrix mit unterschiedlichen Matrizen Q und R geben kann. Im folgenden werden wir jedoch sehen, daß stets eine eindeutige sogenannte reduzierte QR-Zerlegung existiert.

Definition 3.2.3 (Reduzierte QR-Zerlegung)

Sei $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$. Eine Zerlegung der Form $A = \hat{Q} \cdot \hat{R}$ mit $\hat{Q} \in \mathbb{R}^{m \times n}$ und $\hat{R} \in \mathbb{R}^{n \times n}$, bei der die Diagonalelemente von \hat{R} gleich eins sind und $\hat{Q}^\top \hat{Q} = \text{diag}(d_1, \dots, d_n)$ mit $d_j \neq 0$, $j = 1, \dots, n$, gilt, heißt **reduzierte QR-Zerlegung von A** . (Man beachte die Dimensionen von \hat{Q} und \hat{R} !).

Der folgende Satz besagt, daß die reduzierte QR-Zerlegung stets eindeutig ist.

Satz 3.2.4 (Eindeutigkeit)

Sei $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ und $\text{Rang}(A) = n$ gegeben. Dann besitzt A höchstens eine reduzierte QR-Zerlegung $A = \hat{Q} \cdot \hat{R}$.

Beweis: A besitze eine weitere Zerlegung $A = \tilde{Q} \cdot \tilde{R}$ mit den angegebenen Eigenschaften. Die Spalten von A seien mit a^j , $j = 1, \dots, n$, bezeichnet; entsprechende Bezeichnungen gelten für \hat{Q} , \tilde{Q} , \hat{R} und \tilde{R} .

Wir zeigen die Eindeutigkeit durch Induktion über die Anzahl n der Spalten von A .

Für $n = 1$ gilt $\hat{Q} = \tilde{Q} = a^1$ und $\hat{R} = \tilde{R} = 1$.

Sei die Behauptung bewiesen für $1 \leq n \leq k$. Nach Induktionsvoraussetzung erzeugen die Faktorisierungen für die Spalten (a^1, \dots, a^k) von A identische Faktorisierungen. Für $n = k + 1$ folgt dann

$$\begin{pmatrix} a^1 & \dots & a^k & a^{k+1} \end{pmatrix} \\ = \left(\begin{pmatrix} \hat{q}^1 & \dots & \hat{q}^k \end{pmatrix} \begin{pmatrix} \hat{r}^1 & \dots & \hat{r}^k \end{pmatrix} \begin{pmatrix} \hat{q}^1 & \dots & \hat{q}^k & \hat{q}^{k+1} \end{pmatrix} \hat{r}^{k+1} \right) \\ = \left(\begin{pmatrix} \hat{q}^1 & \dots & \hat{q}^k \end{pmatrix} \begin{pmatrix} \hat{r}^1 & \dots & \hat{r}^k \end{pmatrix} \begin{pmatrix} \hat{q}^1 & \dots & \hat{q}^k & \tilde{q}^{k+1} \end{pmatrix} \tilde{r}^{k+1} \right).$$

Also:

$$\sum_{j=1}^k \hat{r}_j^{k+1} \hat{q}^j + \hat{q}^{k+1} = \sum_{j=1}^k \tilde{r}_j^{k+1} \hat{q}^j + \tilde{q}^{k+1}.$$

Multiplikation mit $(\hat{q}^\ell)^\top$ mit $1 \leq \ell \leq k$ liefert

$$\hat{r}_\ell^{k+1} d_\ell = \tilde{r}_\ell^{k+1} d_\ell.$$

Wegen $d_\ell \neq 0$ folgt $\hat{r}_\ell^{k+1} = \tilde{r}_\ell^{k+1}$ und damit $\hat{q}^{k+1} = \tilde{q}^{k+1}$. \square

Existenz einer reduzierten QR-Zerlegung

Es stellt sich die Frage, ob eine reduzierte QR-Zerlegung existiert und wie sie im Fall der Existenz berechnet werden kann. Wir gehen konstruktiv vor und entwickeln einen Algorithmus.

Sei $A = (a^1, \dots, a^n) \in \mathbb{R}^{m \times n}$ mit $m \geq n$ und $\text{Rang}(A) = n$ gegeben. Die Spalten von \hat{Q} seien mit $\hat{q}^1, \dots, \hat{q}^n$ bezeichnet. Die Einträge von \hat{R} seien mit \hat{r}_{ij} bezeichnet, wobei $\hat{r}_{ii} = 1$ für $i = 1, \dots, n$ gelte. Die Gleichung $A = \hat{Q} \cdot \hat{R}$ lautet dann in Matrixschreibweise

$$\begin{pmatrix} a^1 & \dots & a^n \end{pmatrix} = \begin{pmatrix} \hat{q}^1 & \dots & \hat{q}^n \end{pmatrix} \begin{pmatrix} 1 & \hat{r}_{12} & \dots & \hat{r}_{1n} \\ & 1 & \dots & \hat{r}_{2n} \\ & & \ddots & \vdots \\ & & & 1 \end{pmatrix}$$

Spaltenweise ist dies äquivalent mit

$$\begin{aligned} a^1 &= \hat{q}^1 \\ a^2 &= \hat{r}_{12} \hat{q}^1 + \hat{q}^2 \\ &\vdots \\ a^n &= \hat{r}_{1n} \hat{q}^1 + \dots + \hat{r}_{n-1,n} \hat{q}^{n-1} + \hat{q}^n \end{aligned}$$

bzw.

$$a^j = \hat{q}^j + \sum_{i=1}^{j-1} \hat{r}_{ij} \hat{q}^i, \quad j = 1, \dots, n.$$

Hieraus ergibt sich durch sukzessives Anwenden beginnend mit $j = 1$ die Vorschrift

$$\hat{q}^j = a^j - \sum_{i=1}^{j-1} \hat{r}_{ij} \hat{q}^i, \quad j = 1, \dots, n,$$

wobei die Koeffizienten \hat{r}_{ij} noch bestimmt werden müssen. Aus der Forderung $(\hat{q}^i)^\top \hat{q}^j = 0$ für $i = 1, \dots, j-1$ folgt

$$0 = (\hat{q}^i)^\top \hat{q}^j = (\hat{q}^i)^\top a^j - \sum_{k=1}^{j-1} \hat{r}_{kj} (\hat{q}^i)^\top \hat{q}^k = (\hat{q}^i)^\top a^j - \hat{r}_{ij} (\hat{q}^i)^\top \hat{q}^i$$

und damit

$$\hat{r}_{ij} = \frac{(\hat{q}^i)^\top a^j}{(\hat{q}^i)^\top \hat{q}^i}, \quad i = 1, \dots, j-1.$$

Dies ist gerade das Gram-Schmidt'sche Orthogonalisierungsverfahren, welches wir schon im Rahmen des konjugierten Gradientenverfahrens kennen gelernt haben.

Algorithmus 3.2.5 (Gram-Schmidtsches Orthogonalisierungsverfahren)

Gegeben: Matrix $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ und $\text{Rang}(A) = n$.

Ausgabe: reduzierte QR-Zerlegung $A = \hat{Q} \cdot \hat{R}$.

```

 $\hat{q}^1 := a^1$ 
for  $j = 2, \dots, n$ :
    for  $i = 1, \dots, j-1$ :
         $\hat{r}_{ij} := \frac{(\hat{q}^i)^\top a^j}{(\hat{q}^i)^\top \hat{q}^i}$ 
    end
     $\hat{q}^j := a^j - \sum_{i=1}^{j-1} \hat{r}_{ij} \hat{q}^i$ 
end

```

Das Gram-Schmidt'sche Orthogonalisierungsverfahren zeigt konstruktiv die Existenz einer reduzierten QR-Zerlegung und es gilt

Satz 3.2.6 (Existenz einer reduzierten QR-Zerlegung)

Sei $A \in \mathbb{R}^{m \times n}$ mit $m \geq n$ und $\text{Rang}(A) = n$ gegeben. Dann existiert eine reduzierte QR-Zerlegung $A = \hat{Q} \cdot \hat{R}$.

Beweis: Es bleibt noch zu zeigen, daß das Verfahren wohldefiniert ist. Hierzu müssen wir zeigen, daß $(\hat{q}^j)^\top \hat{q}^j \neq 0$ für $j = 1, \dots, n$ gilt. Wäre $\hat{q}^j = 0$ für ein minimales $j \in \{1, \dots, n\}$, so folgte

$$a^j \in \text{span}\{\hat{q}^1, \dots, \hat{q}^{j-1}\} = \text{span}\{a^1, \dots, a^{j-1}\}$$

im Widerspruch zur linearen Unabhängigkeit der Spalten von A . □

Ist eine reduzierte QR-Zerlegung $A = \hat{Q} \cdot \hat{R}$ bekannt, so kann sie zu einer vollen QR-Zerlegung $A = Q \cdot R$ gemäß Definition 3.2.1 mit $Q \in \mathbb{R}^{m \times m}$ und $R \in \mathbb{R}^{m \times n}$ erweitert werden, indem man zu \hat{Q} weitere $m - n$ orthonormale Spalten hinzufügt (hier hat man gewisse Freiheiten) und \hat{R} unten durch eine Nullmatrix der Dimension $(m - n) \times n$ ergänzt, also

$$A = Q \cdot R := \begin{pmatrix} \hat{Q} & \tilde{Q} \end{pmatrix} \cdot \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}.$$

Beachte, daß Q hierin noch nicht orthogonal ist, denn es gilt

$$Q^\top Q = \begin{pmatrix} \hat{Q}^\top \\ \tilde{Q}^\top \end{pmatrix} \begin{pmatrix} \hat{Q} & \tilde{Q} \end{pmatrix} = \begin{pmatrix} \hat{Q}^\top \hat{Q} & \hat{Q}^\top \tilde{Q} \\ \tilde{Q}^\top \hat{Q} & \tilde{Q}^\top \tilde{Q} \end{pmatrix} = \begin{pmatrix} D & 0 \\ 0 & I \end{pmatrix}, \quad D = \text{diag}(d_1, \dots, d_n).$$

Die Spalten von \hat{Q} müssen daher noch orthonormiert werden, was entsprechende Anpassungen in \hat{R} nach sich zieht.

Umgekehrt kann jede volle QR-Zerlegung $A = Q \cdot R$ in die reduzierte QR-Zerlegung überführt werden vermöge

$$A = Q \cdot R = \underbrace{Q \cdot \text{diag}(r_{11}, \dots, r_{nm}, 0, \dots, 0)}_{=:\begin{pmatrix} \hat{Q} & \tilde{Q} \end{pmatrix}} \cdot \underbrace{\text{diag}(1/r_{11}, \dots, 1/r_{nm}, 0, \dots, 0)}_{=:\begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}} \cdot R.$$

Leider gilt die QR-Zerlegung nach Gram-Schmidt als numerisch instabil, wie das folgende Beispiel zeigt. Es sei jedoch erwähnt, daß es stabile Varianten des Verfahrens gibt, auf die wir hier nicht eingehen möchten, da in der Regel die im nächsten Abschnitt beschriebene QR-Zerlegung nach Householder für numerische Berechnungen verwendet wird.

Beispiel 3.2.7

Betrachte die Matrix

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 10^{-8} & 0 & 0 \\ 0 & 10^{-8} & 0 \\ 0 & 0 & 10^{-8} \end{pmatrix}.$$

Das Gram-Schmidt-Verfahren liefert bei doppelt genauer Rechnung (16 Nachkommastellen) die folgende Matrix

$$\hat{Q} = \begin{pmatrix} 1 & 0 & 0 \\ 10^{-8} & -10^{-8} & -10^{-8} \\ 0 & 10^{-8} & 0 \\ 0 & 0 & 10^{-8} \end{pmatrix}, \quad \hat{R} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Es folgt

$$\hat{Q}^\top \hat{Q} = \begin{pmatrix} 1 & -10^{-16} & -10^{-16} \\ -10^{-16} & 2 \cdot 10^{-16} & 10^{-16} \\ -10^{-16} & 10^{-16} & 2 \cdot 10^{-16} \end{pmatrix},$$

und die Spalten 2 und 3 von \hat{Q} sind nahezu linear abhängig.

3.2.4 QR-Zerlegung nach Householder

In diesem Abschnitt wird die QR-Zerlegung nach Householder besprochen, die eine numerisch stabile Methode zur Berechnung einer QR-Zerlegung $A = Q \cdot R$ gemäß Definition 3.2.1 darstellt.

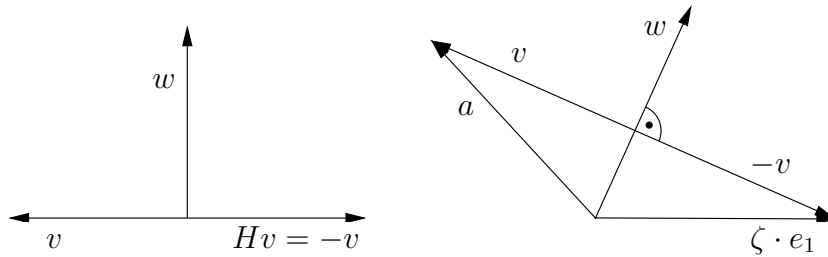


Abbildung 3.2: Householder Transformationen: Spiegelung (links) und Abbildung auf Vielfaches des Einheitsvektors bei spezieller Wahl des Vektors v (rechts).

Das Verfahren arbeitet mit Spiegelungen, welche durch Multiplikation mit der sogenannten Householder-Matrix H realisiert werden.

Für einen Vektor v ist die Householder-Matrix H definiert als

$$H = I - \frac{2}{v^\top v} v v^\top. \quad (3.4)$$

Sie hat folgende Eigenschaften, vgl. auch Abbildung 3.2, die man leicht nachrechnet:

Satz 3.2.8

Für H gemäß (3.4) gelten folgende Aussagen:

- (i) H ist symmetrisch ($H^\top = H$) und orthogonal ($H^\top \cdot H = I$).
- (ii) H beschreibt eine Spiegelung des Vektors v gemäß $Hv = -v$.
- (iii) $Hw = w$ für jeden Vektor w mit $w^\top v = 0$ (w und v stehen senkrecht aufeinander).

Durch geschickte Wahl des Vektors v ist es möglich, einen gegebenen Vektor a (z.B. die Spalte einer Matrix) auf ein Vielfaches des Einheitsvektors abzubilden. Auf dieser Idee basiert die QR-Zerlegung nach Householder.

Analog zur Berechnung der LR-Zerlegung wird ausgehend von A eine Folge von Matrizen erzeugt bis eine rechte obere Dreiecksmatrix entsteht:

$$A = A^{(1)} \rightarrow A^{(2)} \rightarrow A^{(3)} \rightarrow \dots \rightarrow A^{(\ell+1)} = R \quad (\ell = \min\{m-1, n\})$$

In jedem Schritt des Verfahrens wird eine speziell konstruierte Householder-Matrix $H^{(i)}$ von links an $A^{(i)}$ heranmultipliziert, um die nächste Iterierte $A^{(i+1)} = H^{(i)} \cdot A^{(i)}$ zu erhalten.

Insgesamt erhält man so

$$\begin{aligned}
 R &= A^{(\ell+1)} \\
 &= H^{(\ell)} \cdot A^{(\ell)} \\
 &= H^{(\ell)} \cdot H^{(\ell-1)} \cdot A^{(\ell-1)} \\
 &\vdots \\
 &= H^{(\ell)} \cdot H^{(\ell-1)} \dots H^{(1)} \cdot A^{(1)}.
 \end{aligned}$$

Mit

$$Q := (H^{(1)})^\top \dots (H^{(\ell-1)})^\top \cdot (H^{(\ell)})^\top$$

(Q ist als Produkt von orthogonalen Matrizen ebenfalls orthogonal!) folgt

$$A = Q \cdot R.$$

Im folgenden wird die Konstruktion der Matrix $H^{(i)}$ beschrieben. Das Verfahren sei bis zur Matrix

$$A^{(i)} = \left(\begin{array}{cc|ccc} a_{11}^{(i)} & \dots & a_{1i}^{(i)} & \dots & a_{1n}^{(i)} \\ & \ddots & \vdots & \ddots & \vdots \\ \hline & & a_{ii}^{(i)} & \dots & a_{in}^{(i)} \\ & & a_{i+1,i}^{(i)} & \dots & a_{i+1,n}^{(i)} \\ & & \vdots & \ddots & \vdots \\ & & a_{mi}^{(i)} & \dots & a_{mn}^{(i)} \end{array} \right) =: \left(\begin{array}{c|c} R_{11}^{(i)} & A_{12}^{(i)} \\ \hline & A_{22}^{(i)} \end{array} \right) \quad (3.5)$$

fortgeschritten, so daß die ersten $i - 1$ Spalten bereits in Dreiecksgestalt vorliegen. Da wir eine rechte obere Dreiecksmatrix konstruieren wollen, müssen die Elemente $a_{ji}^{(i)}$, $j = i + 1, \dots, m$, eliminiert werden. Dazu werden wir nun eine Householdertransformation derart konstruieren, daß der Vektor

$$a^{(i)} := \begin{pmatrix} a_{ii}^{(i)} \\ a_{i+1,i}^{(i)} \\ \vdots \\ a_{mi}^{(i)} \end{pmatrix} \in \mathbb{R}^{m-i+1} \quad (3.6)$$

auf ein Vielfaches $\zeta \cdot e_i$ des Einheitsvektors $e_i = (1, 0, \dots, 0)^\top \in \mathbb{R}^{m-i+1}$ abgebildet wird, siehe Abbildung 3.2. Dies geht natürlich nur dann, wenn $\|a^{(i)}\|_2 \neq 0$ gilt. Ist $\|a^{(i)}\|_2 \neq 0$, so kann $H_i = I$ gewählt werden, allerdings ist dann $r_{ii} = 0$ in der resultierenden Zerlegung und \bar{R} in Definition 3.2.1 ist nicht invertierbar.

In der folgenden Herleitung gelte $\|a^{(i)}\|_2 \neq 0$. Es gilt, einen Vektor v_i zu konstruieren, so daß

$$H_i \cdot a^{(i)} = \left(I - \frac{2}{v_i^\top v_i} v_i v_i^\top \right) \cdot a^{(i)} = \zeta \cdot e_i \quad (3.7)$$

mit einer Konstanten $\zeta \neq 0$ gilt. Mit $H_i \in \mathbb{R}^{(m-i+1) \times (m-i+1)}$ ist dann auch

$$H^{(i)} := \left(\begin{array}{c|c} I & \\ \hline & H_i \end{array} \right) \in \mathbb{R}^{m \times m}$$

orthogonal und wir erhalten

$$A^{(i+1)} = H^{(i)} \cdot A^{(i)} = \left(\begin{array}{c|c} I & \\ \hline & H_i \end{array} \right) \cdot \left(\begin{array}{c|c} R_{11}^{(i)} & A_{12}^{(i)} \\ \hline & A_{22}^{(i)} \end{array} \right) = \left(\begin{array}{c|c} R_{11}^{(i)} & A_{12}^{(i)} \\ \hline & H_i \cdot A_{22}^{(i)} \end{array} \right),$$

wobei die ersten $i - 1$ Zeilen von $A^{(i)}$ durch die Multiplikation mit $H^{(i)}$ unverändert bleiben, da diese bereits in Dreiecksgestalt vorliegen.

Da H_i orthogonal ist, folgt aus (3.7) die Bedingung

$$\|H_i \cdot a^{(i)}\|_2 = \|a^{(i)}\|_2 = |\zeta|.$$

Der Ansatz

$$v_i = \frac{1}{\|a^{(i)}\|_2} (a^{(i)} + \alpha e_i) \quad (3.8)$$

liefert nach kurzer Rechnung

$$\begin{aligned} v_i^\top a^{(i)} &= \|a^{(i)}\|_2 + \frac{\alpha a_{ii}^{(i)}}{\|a^{(i)}\|_2}, \\ v_i^\top v_i &= \frac{\|a^{(i)} + \alpha e_i\|_2^2}{\|a^{(i)}\|_2^2} = 1 + \frac{2\alpha a_{ii}^{(i)}}{\|a^{(i)}\|_2^2} + \frac{\alpha^2}{\|a^{(i)}\|_2^2}, \\ H_i \cdot a^{(i)} &= a^{(i)} - \frac{2 \left(\|a^{(i)}\|_2^2 + \alpha a_{ii}^{(i)} \right)}{\|a^{(i)}\|_2^2 + 2\alpha a_{ii}^{(i)} + \alpha^2} a^{(i)} - \frac{2\alpha \left(\|a^{(i)}\|_2^2 + \alpha a_{ii}^{(i)} \right)}{\|a^{(i)}\|_2^2 + 2\alpha a_{ii}^{(i)} + \alpha^2} e_i. \end{aligned} \quad (3.9)$$

Um $H_i a^{(i)} = \zeta e_i$ zu erfüllen, wählen wir α so, daß die ersten beiden Terme in (3.9) wegfallen, also

$$1 - \frac{2 \left(\|a^{(i)}\|_2^2 + \alpha a_{ii}^{(i)} \right)}{\|a^{(i)}\|_2^2 + 2\alpha a_{ii}^{(i)} + \alpha^2} = 0.$$

Auflösen nach α liefert

$$\alpha^2 = \|a^{(i)}\|_2^2 \quad \Rightarrow \quad \alpha = \pm \|a^{(i)}\|_2.$$

Einsetzen von α in (3.9) liefert $H_i \cdot a^{(i)} = -\alpha e_i$. Durch die Wahl des Vorzeichens von α haben wir nun zwei Möglichkeiten, v_i in (3.8) zu wählen. Um Auslöschung in der ersten Komponente von v_i zu vermeiden, wählen wir α so, daß

$$\text{sign}(\alpha) = \text{sign}(a_{ii}^{(i)}) \quad \text{bzw.} \quad \alpha = \text{sign}(a_{ii}^{(i)}) \|a^{(i)}\|_2$$

gilt und erhalten somit aus (3.8)

$$v_i = \frac{1}{\|a^{(i)}\|_2} \cdot a^{(i)} + \text{sign}(a_{ii}^{(i)}) \cdot e_i.$$

Diese Wahl von v_i leistet das Gewünschte mit

$$H_i \cdot a^{(i)} = -\text{sign}(a_{ii}^{(i)}) \|a^{(i)}\|_2 \cdot e_i.$$

Bei der Implementierung ist darauf zu achten, daß die Matrixmultiplikation $H_i \cdot A_{22}^{(i)}$ niemals explizit ausgeführt wird. Anstattdessen wird ausgenutzt, daß für eine Matrix B die Darstellung

$$H \cdot B = B - \frac{2}{v^\top v} v w^\top$$

mit $w = B^\top v$ gilt. Die Berechnung von w und $v w^\top$ benötigt jeweils nur $m \cdot n$ Operationen. Die explizite Berechnung von $H \cdot B$ benötigt $\mathcal{O}(m^2 n)$ Operationen. Zusammenfassend erhalten wir folgenden Algorithmus.

Algorithmus 3.2.9 (QR-Zerlegung nach Householder)

Gegeben: Matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$. Setze $A^{(1)} := A$.

Für $i = 1, 2, \dots, \min\{m-1, n\}$:

Seien $a^{(i)}$ und $A_{22}^{(i)}$ definiert gemäß (3.6) und (3.5).

Falls $\|a^{(i)}\|_2 = 0$, setze $H_i = I$.

Andernfalls setze

$$\begin{aligned} v &= \frac{a^{(i)}}{\|a^{(i)}\|_2} + \text{sign}(a_{ii}^{(i)}) e_i, \\ \beta &= \frac{2}{v^\top v} = \left(1 + \frac{|a_{ii}^{(i)}|}{\|a^{(i)}\|_2} \right)^{-1}, \\ w &= \beta (A_{22}^{(i)})^\top v \end{aligned}$$

und aktualisiere $A_{22}^{(i)} := A_{22}^{(i)} - v \cdot w^\top$.

Der Aufwand (Multiplikationen und Divisionen) im i -ten Schleifendurchlauf beträgt

$$\begin{aligned} v &: 2(m-i+1) \\ \beta &: 2 \\ w &: (n-i+1)(m-i+1) + (n-i+1) \\ A_{22}^{(i)} &: (n-i+1)(m-i+1) \end{aligned}$$

Insgesamt ergibt sich ein Aufwand von

$$\sum_{i=1}^n ((m-i+1)(2+n-i+1+n-i+1) + (n-i+1) + 2) = mn^2 - \frac{1}{3}n^3 + \mathcal{O}(mn+n^2).$$

Im Fall $m = n$ ist daher der Aufwand mit $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ in etwa doppelt so teuer wie bei der Gauss'schen LR-Zerlegung.

Beispiel 3.2.10

Gesucht ist die QR-Zerlegung der Matrix

$$A = \begin{pmatrix} 5 & 7 \\ -5 & -7 \\ -1 & 1 \end{pmatrix}.$$

Es gilt $A = Q \cdot R$ mit (gerundete Werte)

$$R \approx \begin{pmatrix} -7.140 & -9.660 \\ 0 & 2.380 \\ 0 & 0 \end{pmatrix}, \quad Q \approx \begin{pmatrix} -0.700 & 0.099 & 0.707 \\ 0.700 & -0.099 & 0.707 \\ 0.140 & 0.990 & 0 \end{pmatrix}$$

Bemerkung 3.2.11

Die QR-Zerlegung von $A \in \mathbb{R}^{m \times n}$, $m \geq n$, $\text{Rang}(A) = n$ liefert eine Zerlegung der Form

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix}, \quad Q = (Y, Z) \in \mathbb{R}^{m \times m}, \quad Y \in \mathbb{R}^{m \times n}, \quad Z \in \mathbb{R}^{m \times (m-n)}, \quad R \in \mathbb{R}^{n \times n}.$$

Hierin ist Y eine Orthonormalbasis des Bildes von A , Z ist eine Orthonormalbasis des orthogonalen Komplements von $\text{Bild}(A)$, d.h. des Kerns von A^\top . Jedes $x \in \mathbb{R}^n$ kann als $x = Yx_Y + Zx_Z$ mit eindeutig bestimmten Vektoren $x_Y \in \mathbb{R}^n$ und $x_Z \in \mathbb{R}^{(m-n)}$ dargestellt werden.

Bemerkung 3.2.12

Eine QR-Zerlegung von A kann auch mittels sogenannter Givens-Rotationen erzeugt wer-

den. Hierbei werden orthogonale Rotationsmatrizen der Form

$$T_{ij}(c, s) = \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & c & & & -s & \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \\ & & s & & & & c & \\ & & & & & & & 1 & \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ \leftarrow i \\ \\ \\ \leftarrow j \\ \\ \\ \\ \end{matrix}$$

$\begin{matrix} \uparrow & \uparrow \\ i & j \end{matrix}$

mit geeignet gewählten Werten c und s mit $c^2 + s^2 = 1$ dazu verwendet, um den j -ten Eintrag in einem Vektor a durch $T_{ij}(c, s) \cdot a$ zu Null zu rotieren. $T_{ij}(c, s)$ beschreibt eine Drehung in der durch die Einheitsvektoren e_j und e_i aufgespannten Ebene um den Winkel ϕ , der durch $c = \cos(\phi)$ und $s = \sin(\phi)$ festgelegt ist.

Givens-Rotationen kommen insbesondere bei Hessenbergmatrizen zur Anwendung, welche bei Eigenwertaufgaben auftreten.

3.2.5 Rangbestimmung einer Matrix mittels QR-Zerlegung und Pivoting

Die QR-Zerlegung kann auch zur numerischen Bestimmung des Rangs einer Matrix verwendet werden, wenn eine zusätzliche Spaltenpivotsuche verwendet wird. Besitzt die Matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$, keinen vollen Rang n , so hat auch die Matrix R in der QR-Zerlegung keinen vollen Rang. In vielen Anwendungen ist es nun interessant zu wissen, welche Spalten von A linear abhängig sind. Dies kann durch Spaltenvertauschungen ermittelt werden.

Das Householder-Verfahrens sei fortgeschritten bis zur Matrix

$$A^{(i)} = \left(\begin{array}{c|c} R_{11}^{(i)} & A_{12}^{(i)} \\ \hline & A_{22}^{(i)} \end{array} \right)$$

mit einer rechten oberen Dreiecksmatrix $R_{11}^{(i)}$. Als nächstes wird für die Matrix $A_{22}^{(i)}$ eine Spalte ermittelt, die die größte Norm unter allen Spalten von $A_{22}^{(i)}$ besitzt. Diese normmaximale Spalte wird nun mit der ersten Spalte in $A_{22}^{(i)}$ und entsprechend auch in $A_{12}^{(i)}$ vertauscht. Anschließend wird eine Householdertransformation angewendet und man erhält

$$A^{(i+1)} = H^{(i)} \cdot A^{(i)} \cdot P_i$$

mit einer geeigneten Permutationsmatrix P_i . Wiederholte Anwendung dieser Vorschrift führt letztendlich auf eine Zerlegung der Form

$$A \cdot P = Q \cdot R, \quad P = P_1 \cdot P_2 \cdots P_n,$$

und

$$R = \begin{pmatrix} \bar{R} & S \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \times n}, \quad \bar{R} \in \mathbb{R}^{k \times k}, \quad S \in \mathbb{R}^{k \times (n-k)},$$

wobei \bar{R} invertierbar ist und $\text{Rang}(A) = k(\leq n)$ gilt.

Die letzten $n - k$ Spalten in R bzw. in $A \cdot P$ sind somit linear abhängig. Durch Multiplikation von rechts mit P^\top lassen sich daraus die linear abhängigen Spalten von A ermitteln.

Angewendet auf das Ausgleichsproblem ergibt sich wegen der Invarianz der Euklidnorm und der Orthogonalität von P

$$\|Ax - b\|_2^2 = \|Q^\top(APP^\top x - b)\|_2^2 = \|RP^\top x - Q^\top b\|_2^2.$$

Definiere

$$c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} := Q^\top b, \quad c_1 \in \mathbb{R}^k, \quad c_2 \in \mathbb{R}^{m-k}, \quad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} := P^\top x, \quad y_1 \in \mathbb{R}^k, \quad y_2 \in \mathbb{R}^{n-k}.$$

Damit folgt

$$\|RP^\top x - Q^\top b\|_2^2 = \|\bar{R}y_1 + Sy_2 - c_1\|_2^2 + \|c_2\|_2^2.$$

Dieser Ausdruck wird minimal für $\bar{R}y_1 = c_1$, $y_2 = 0$, was ein lineares Gleichungssystem für y_1 darstellt. Die Lösung dieses Gleichungssystems ist auch **eine** Lösung der Minimierungsaufgabe. Der Wert des Zielfunktional kann ebenfalls direkt abgelesen werden:

$$\|c_2\|_2^2.$$

Offenbar ist $y_1 = \bar{R}^{-1}c_1$ und $y_2 = 0$ aber nicht die einzige Lösung, da das Gleichungssystem $\bar{R}y_1 + Sy_2 - c_1 = 0$ für beliebiges $y_2 \in \mathbb{R}^{n-k}$ durch

$$y_1 = \bar{R}^{-1}(c_1 - Sy_2)$$

gelöst wird. Häufig ist man nur an einer normminimalen Lösung (y_1, y_2) interessiert. Dies führt auf den Begriff der Pseudoinversen bzw. Moore-Penrose-Inversen, worauf wir hier nicht weiter eingehen wollen.

Kapitel 4

Nichtlineare Gleichungen

In vielen Problemstellungen ist es erforderlich, eine

nichtlineare Gleichung

$$F(x) = 0, \quad F : \mathbb{R} \rightarrow \mathbb{R}, \quad x \in \mathbb{R}, \quad (4.1)$$

oder ein

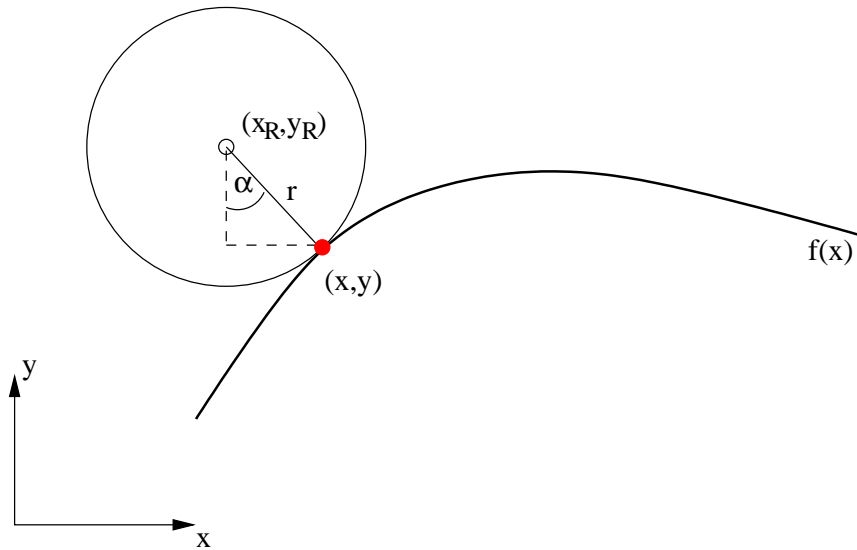
nichtlineares Gleichungssystem

$$F(x) = \begin{pmatrix} F_1(x_1, \dots, x_n) \\ \vdots \\ F_n(x_1, \dots, x_n) \end{pmatrix} = 0, \quad F : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad x = (x_1, \dots, x_n)^\top \in \mathbb{R}^n \quad (4.2)$$

zu lösen. Offenbar ist (4.1) als Spezialfall mit $n = 1$ in (4.2) enthalten. Es gibt jedoch spezielle Verfahren für den eindimensionalen Fall $n = 1$. Nichtlineare Gleichungssysteme können meist nicht analytisch gelöst werden, sondern erfordern numerische Verfahren, wie das Beispiel $F(x) = x - \cos(x)$ zeigt. In industriellen Anwendungen treten häufig Probleme mit mehreren tausend Variablen auf.

Beispiel 4.0.1

Bei der Simulation eines Autos tritt das Problem auf, den Kontaktpunkt (x, y) eines Autoreifens mit Mittelpunkt (x_R, y_R) und Radius r auf einer unebenen Fahrbahn zu bestimmen, vgl. Abbildung. Das Fahrbahnprofil sei durch die nichtlineare Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ beschrieben.



Der Kontaktpunkt ist implizit gegeben durch die nichtlineare Gleichung

$$f(x_R + r \sin \alpha) = y_R - r \cos \alpha,$$

die den Winkel α festlegt.

Beispiel 4.0.2 (Reaktionsgeschwindigkeiten)

Wir betrachten ein Beispiel aus der Biochemie. Gegeben sind n Proteine und ein Peptid P . Die n Proteine $1, \dots, n$ reagieren mit dem Peptid P und bilden feste Komplexe $1P, \dots, nP$ mit Konzentrationen $[1P], \dots, [nP]$. Die totalen Konzentrationen $[P]_t, [1]_t, \dots, [n]_t$ und die Reaktionsgeschwindigkeiten k_1, \dots, k_n für die Reaktionen $i + P \rightarrow iP$, $i = 1, \dots, n$, sind bekannt.

Es bestehen die folgenden Zusammenhänge für die Konzentrationen $[P]_f$ und $[i]_f$, $i = 1, \dots, n$, der noch nicht gebundenen (freien) Peptide und Proteine:

$$\begin{aligned} [P]_f &= [P]_t - ([1P] + \dots + [nP]), \\ [i]_f &= [i]_t - [iP], \quad i = 1, \dots, n, \\ [iP] \cdot k_i &= [i]_f \cdot [P]_f, \quad i = 1, \dots, n \end{aligned}$$

Gesucht sind die Konzentrationen der Komplexe $[iP]$, $i = 1, \dots, n$, und die Konzentration $[P]_f$.

Einsetzen der obigen Beziehungen führt auf das nichtlineare Gleichungssystem

$$[iP] \cdot \left(k_i + [P]_t - \sum_{j=1}^n [jP] \right) - [i]_t \cdot \left([P]_t - \sum_{j=1}^n [jP] \right) = 0, \quad i = 1, \dots, n.$$

Dieses nichtlineare System ist für Konzentrationen $[iP]$, $i = 1, \dots, n$, zu lösen. Im allgemeinen besitzt das System mehrere Lösungen. Interessant sind dabei nur nicht-negative

Konzentrationen.

Weitere Anwendungen für nichtlineare Gleichungssysteme:

- Diskretisierung von Anfangswertproblemen mittels impliziter Integrationsverfahren
- Diskretisierung von nichtlinearen partiellen Differentialgleichungen und Randwertproblemen
- Technische Anwendungen, z.B. Bestimmung des Kontaktpunkts eines Autoreifens auf einer unebenen Fahrbahn, Bestimmung von Gleichgewichtslösungen einer Differentialgleichung, ...
- Lagrange-Newton-Verfahren in der restringierten Optimierung
- Newtonverfahren in der unrestringierten Optimierung
- Bestimmung von Polynomnullstellen (Eigenwerte sind Nullstellen des charakteristischen Polynoms)

Im folgenden werden Algorithmen zur approximativen Bestimmung von zumindest einer Nullstelle von F untersucht, wobei vorausgesetzt wird, daß eine Nullstelle existiert.

Definition 4.0.3 (Nullstelle)

$\hat{x} \in \mathbb{R}^n$ heißt Nullstelle von (4.2), wenn $F(\hat{x}) = 0$ gilt.

Die Verfahren in diesem Kapitel haben die Gemeinsamkeit, daß sie **iterativ** arbeiten, d.h. es wird eine Folge $\{x^{(k)}\}$ mit

$$x^{(k+1)} = g(x^{(k)}), \quad k = 0, 1, 2, \dots$$

berechnet, wobei g eine geeignete Funktion ist. In jedem Iterationsschritt muß dabei ein einfach zu lösendes Problem gelöst werden. Das Hauptaugenmerk wird auf folgenden Fragestellungen liegen:

- Unter welchen Voraussetzungen konvergierte die Folge $\{x^{(k)}\}$ gegen \hat{x} ?
- Wie schnell konvergiert $\{x^{(k)}\}$ gegen \hat{x} ?

Die Konvergenzgeschwindigkeit einer Folge ist wie folgt definiert, wobei eine Folge umso schneller konvergiert, je höher die Konvergenzordnung ist.

Definition 4.0.4 (lineare Konvergenz, quadratische Konvergenz, superlineare Konvergenz, Konvergenzordnung)

Die Folge $\{x^{(i)}\}$ konvergiere gegen \hat{x} .

- (a) Die Folge $\{x^{(i)}\}$ heißt **linear konvergent**, falls es eine Konstante $0 \leq C < 1$ und eine Zahl i_0 gibt mit

$$\|x^{(i+1)} - \hat{x}\| \leq C \cdot \|x^{(i)} - \hat{x}\| \quad \forall i \geq i_0.$$

- (b) Die Folge $\{x^{(i)}\}$ heißt **quadratisch konvergent**, falls es eine Konstante C und eine Zahl i_0 gibt mit

$$\|x^{(i+1)} - \hat{x}\| \leq C \cdot \|x^{(i)} - \hat{x}\|^2 \quad \forall i \geq i_0.$$

- (c) Die Folge $\{x^{(i)}\}$ heißt **konvergent mit Ordnung** $p > 1$, falls es eine Konstante C und eine Zahl i_0 gibt mit

$$\|x^{(i+1)} - \hat{x}\| \leq C \cdot \|x^{(i)} - \hat{x}\|^p \quad \forall i \geq i_0.$$

- (d) Die Folge $\{x^{(i)}\}$ heißt **superlinear konvergent**, falls es eine Folge $\{C_i\}$ mit $\lim_{i \rightarrow \infty} C_i = 0$ und eine Zahl i_0 gibt mit

$$\|x^{(i+1)} - \hat{x}\| \leq C_i \cdot \|x^{(i)} - \hat{x}\| \quad \forall i \geq i_0.$$

Beachte, daß die Konvergenz der Folge $\{x^{(i)}\}$ in der Definition explizit vorausgesetzt wurde. Im Fall der linearen und superlinearen Konvergenz ist dies nicht nötig, da lineare und superlineare Konvergenz automatisch die Konvergenz der Folge nach sich ziehen. Für die übrigen Konvergenzbegriffe gilt dies nicht, so daß beim Beweis der Konvergenz mit Ordnung $p > 1$ stets zunächst die Konvergenz der Folge gezeigt werden muss.

4.1 Bisektion (Intervallschachtelungsverfahren)

Im eindimensionalen Fall $n = 1$ wird unter der Voraussetzung, daß F stetig ist, häufig das Bisektionsverfahren zur Bestimmung **einer** Nullstelle verwendet. Ausgehend von einem Startintervall $[a, b]$, $a < b$, mit

$$F(a) \cdot F(b) < 0 \tag{4.3}$$

berechnet das Bisektionsverfahren in jedem Schritt ein neues Intervall mit halber Länge, in dem eine Nullstelle von F enthalten ist. Man erhält so eine beliebig genaue Einschachtelung einer Nullstelle. Das Bisektionsverfahren kann u.a. angewendet werden, um Eigenwerte zu berechnen, indem eine Nullstelle des charakteristischen Polynoms gesucht wird.

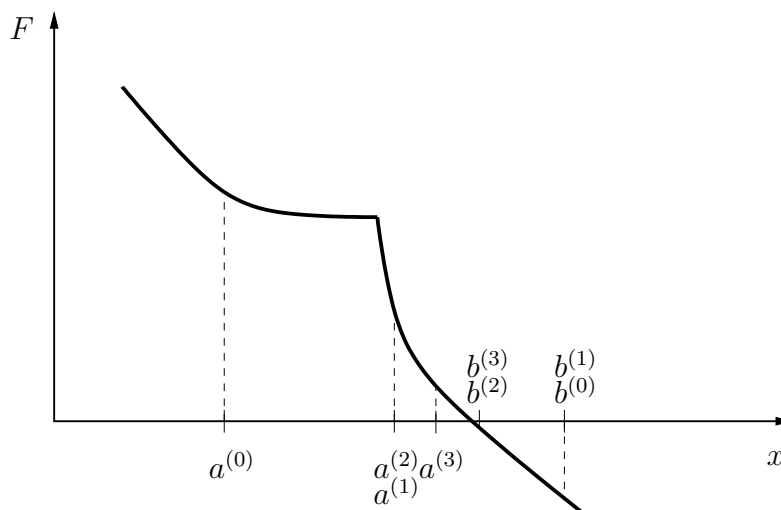


Abbildung 4.1: Bisektion: Iterierte Intervallhalbierung.

Bedingung (4.3) besagt, daß $F(a)$ und $F(b)$ unterschiedliche Vorzeichen besitzen. In diesem Fall garantiert der **Zwischenwertsatz** 1.3.1, daß im Intervall (a, b) (mindestens) eine Nullstelle von F existiert.

Durch iterierte Intervallschachtelung, vgl. Abbildung 4.1, erlaubt das Bisektionsverfahren die beliebig genaue Approximation einer Nullstelle in $[a, b]$ (es ist aber nicht gesagt, welche Nullstelle man bekommt).

Algorithmus 4.1.1 (Bisektionsverfahren)

(0) *Input:*

- a, b mit $a < b$ und $F(a) \cdot F(b) < 0$
- Toleranz $tol > 0$
- maximale Iterationszahl n_{max}

(1) Setze $i = 0$, $a^{(0)} = a$, $b^{(0)} = b$.

(2) Setze $x^{(i)} = (a^{(i)} + b^{(i)})/2$ und $w = F(x^{(i)})$.

Falls $|b^{(i)} - a^{(i)}| \leq tol$ oder $i > n_{max}$, STOP mit Ausgabe $x^{(i)}$ und w .

Falls $w = 0$, dann ist $x^{(i)}$ eine Nullstelle, STOP mit Ausgabe $x^{(i)}$ und w .

Falls $w \cdot F(a^{(i)}) < 0$, setze $a^{(i+1)} = a^{(i)}$, $b^{(i+1)} = x^{(i)}$.

Falls $w \cdot F(a^{(i)}) > 0$, setze $a^{(i+1)} = x^{(i)}$, $b^{(i+1)} = b^{(i)}$.

(3) Setze $i = i + 1$ und gehe zu (2).

In jeder Iteration $i = 0, 1, 2, \dots$ des Bisektionsverfahrens wird ein Intervall $[a^{(i)}, b^{(i)}]$ konstruiert. Per Konstruktion des Intervalls wissen wir nach dem Zwischenwertsatz, daß mindestens eine Nullstelle von F im Intervall $[a^{(i)}, b^{(i)}]$ für alle $i = 0, 1, 2, \dots$ liegen muß. Entsprechend ist die beste Approximation für alle Nullstellen von F im Intervall $[a^{(i)}, b^{(i)}]$ in der i -ten Iteration gegeben durch

$$x^{(i)} = \frac{a^{(i)} + b^{(i)}}{2}.$$

Seit $\hat{x}^{(i)}$ eine beliebige Nullstelle von F im Intervall $[a^{(i)}, b^{(i)}]$. Dann erhalten wir die Fehlerabschätzung

$$|x^{(i)} - \hat{x}^{(i)}| \leq \frac{b^{(i)} - a^{(i)}}{2} \quad \forall i = 0, 1, 2, \dots$$

Da $b^{(i)} - a^{(i)} = (b^{(i-1)} - a^{(i-1)})/2$ gilt, folgt durch Induktion, daß

$$|x^{(i)} - \hat{x}^{(i)}| \leq \frac{b^{(0)} - a^{(0)}}{2^{i+1}} = \frac{b - a}{2^{i+1}} \quad \forall i = 0, 1, 2, \dots$$

Dies ist gerade die Konvergenz des Verfahren gegen eine Nullstelle von F , da $(b-a)/2^{i+1} \rightarrow 0$ für $i \rightarrow \infty$. Zusammenfassend erhalten wir

Satz 4.1.2

Sei F stetig in $[a, b]$, $a < b$, mit $F(a) \cdot F(b) < 0$.

Nach i Iterationen des Bisektionsverfahrens gilt für die Approximation $x^{(i)}$ einer beliebigen Nullstelle $\hat{x}^{(i)} \in [a^{(i)}, b^{(i)}]$ die Fehlerabschätzung

$$|x^{(i)} - \hat{x}^{(i)}| \leq \frac{b - a}{2^{i+1}}.$$

Häufig ist man daran interessiert, eine Nullstelle mit einer vorgegebenen Toleranz $\varepsilon > 0$ zu approximieren. Die dafür notwendige Anzahl an Iterationen bei exakter Rechnung ergibt sich aus der Fehlerabschätzung

$$|x^{(i)} - \hat{x}^{(i)}| \leq \frac{b - a}{2^{i+1}} < \varepsilon$$

und berechnet sich zu

$$i > \log_2 \left(\frac{b - a}{2\varepsilon} \right) = \frac{\log \left(\frac{b-a}{2\varepsilon} \right)}{\log 2}.$$

Beachte, daß die Folge $\{x^{(i)}\}$ i.a. **nicht** linear konvergent im Sinne von Definition 4.0.4 ist. Aber da $b^{(i)} - a^{(i)} = (b^{(i-1)} - a^{(i-1)})/2$ gilt, ist die Folge $\{(b^{(i)} - a^{(i)})/2\}$ der Fehlerschranken linear konvergent mit Konstante $C = 1/2$. Daher betrachten wir das Bisektionsverfahren ebenfalls als linear konvergent.

Beispiel 4.1.3

Berechnung von $\sqrt{2}$ als (positive) Nullstelle von $F(x) = x^2 - 2$.

Ausgabe des Bisektionsverfahrens mit Startintervall $[1, 1.5]$:

```
Iteration 1 : x= 1.5000000000 in [1.0000000000, 1.5000000000], error: 0.2500000000
Iteration 2 : x= 1.2500000000 in [1.2500000000, 1.5000000000], error: 0.1250000000
Iteration 3 : x= 1.3750000000 in [1.3750000000, 1.5000000000], error: 0.0625000000
Iteration 4 : x= 1.4375000000 in [1.3750000000, 1.4375000000], error: 0.0312500000
Iteration 5 : x= 1.4062500000 in [1.4062500000, 1.4375000000], error: 0.0156250000
Iteration 6 : x= 1.4218750000 in [1.4062500000, 1.4218750000], error: 0.0078125000
Iteration 7 : x= 1.4140625000 in [1.4140625000, 1.4218750000], error: 0.0039062500
Iteration 8 : x= 1.4179687500 in [1.4140625000, 1.4179687500], error: 0.0019531250
Iteration 9 : x= 1.4160156250 in [1.4140625000, 1.4160156250], error: 0.0009765625
Iteration 10: x= 1.4150390625 in [1.4140625000, 1.4150390625], error: 0.0004882812
Iteration 11: x= 1.4145507812 in [1.4140625000, 1.4145507812], error: 0.0002441406
Iteration 12: x= 1.4143066406 in [1.4140625000, 1.4143066406], error: 0.0001220703
Iteration 13: x= 1.4141845703 in [1.4141845703, 1.4143066406], error: 0.0000610352
Iteration 14: x= 1.4142456055 in [1.4141845703, 1.4142456055], error: 0.0000305176
Iteration 15: x= 1.4142150879 in [1.4141845703, 1.4142150879], error: 0.0000152588
Iteration 16: x= 1.4141998291 in [1.4141998291, 1.4142150879], error: 0.0000076294
Iteration 17: x= 1.4142074585 in [1.4142074585, 1.4142150879], error: 0.0000038147
```

x = 1.4142 f = -1.7264e-05 a = 1.4142 b = 1.4142

Das Bisektionsverfahren bietet folgende **Vorteile**:

- Es werden nur Funktionsauswertungen benötigt und das Verfahren ist sehr einfach zu implementieren.
- Die Funktion muß lediglich stetig sein. Differenzierbarkeit wird nicht benötigt.
- Es ist sehr genau und konvergiert stets, wenn die Anfangswerte richtig gewählt wurden.

Leider hat das Bisektionsverfahren auch einige **Nachteile**:

- Das Verfahren läßt sich nur für $n = 1$ anwenden. Eine Erweiterung auf $n > 1$ ist schwierig.
- Das Verfahren liefert nur eine lineare Konvergenzordnung mit Konstante $C = 0.5$.
- Es muss ein Startintervall $[a, b]$ mit $F(a) \cdot F(b) < 0$ bestimmt werden.

4.2 Fixpunktiteration

Wir haben im Zusammenhang mit iterativen Verfahren zur Lösung von linearen Gleichungssystemen bereits den Banachschen Fixpunktsatz 2.6.4 zur iterativen Berechnung eines Fixpunkts der Gleichung

$$x = g(x)$$

kennengelernt. Die Aufgabenstellung (4.2) läßt sich nun leicht auf „Fixpunktgestalt“ transformieren, etwa in der Form

$$F(x) = 0 \quad \Leftrightarrow \quad x = x + \rho(x) \cdot F(x) =: g(x),$$

wobei $\rho(x) \in \mathbb{R}^{n \times n}$ eine reguläre, problemabhängige Matrix ist, die dazu dienen soll, die Voraussetzungen des Banachschen Fixpunktsatzes zu erfüllen. Der naive Ansatz $\rho(x) = \pm I$ für alle x führt nur selten zum Erfolg. Anstelle des Nullstellenproblems (4.2) wird also ein Fixpunkt \hat{x} der Funktion g gesucht. Unter der Annahme, daß $\rho(\hat{x})$ regulär ist, ist der Fixpunkt \hat{x} wegen

$$\hat{x} = \hat{x} + \rho(\hat{x}) \cdot F(\hat{x}) \quad \Leftrightarrow \quad \rho(\hat{x}) \cdot F(\hat{x}) = 0 \quad \Leftrightarrow \quad F(\hat{x}) = 0$$

auch Nullstelle von F .

Für einen geeigneten Startwert $x^{(0)}$ wird die Fixpunktiteration

$$x^{(i+1)} = g(x^{(i)}) = x^{(i)} + \rho(x^{(i)}) \cdot F(x^{(i)}), \quad i = 0, 1, \dots$$

zur Approximation von \hat{x} durchgeführt, vgl. Abbildung 4.2.

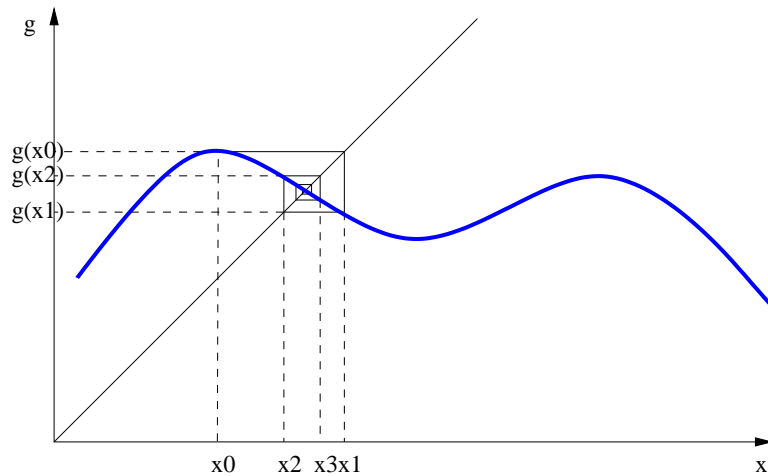


Abbildung 4.2: Schematische Darstellung der Fixpunktiteration.

Diese Methode ist auch dann anwendbar, wenn F nicht differenzierbar ist. Um das Konvergenzresultat im Banachschen Fixpunktsatz anwenden zu können, müssen folgende Voraussetzungen erfüllt sein, vgl. Satz 2.6.4:

- **Selbstabbildungseigenschaft**, d.h. es muß eine abgeschlossene Menge $D \subseteq \mathbb{R}^n$ existieren, so daß $g(D) \subseteq D$ gilt.
- g muß die **Kontraktionsbedingung**

$$\|g(x) - g(y)\| \leq q \|x - y\|$$

für ein $q < 1$ und alle $x, y \in D$ erfüllen.

Kennt man die Kontraktionszahl $q < 1$, so folgt aus dem Banachschen Fixpunktsatz die Fehlerabschätzung

$$\|x^{(i)} - \hat{x}\| \leq \frac{q^i}{1 - q} \|x^{(1)} - x^{(0)}\|.$$

Der Nachteil der Fixpunktiteration liegt in der i.a. nur linearen Konvergenzgeschwindigkeit.

Ist g stetig differenzierbar, so kann die Kontraktionsbedingung (2.20) mit Hilfe der Ableitung von g überprüft werden.

Satz 4.2.1

Es sei $D \subseteq \mathbb{R}^n$ abgeschlossen und konvex und $g : D \rightarrow D$ stetig differenzierbar. Gibt es eine Konstante $q < 1$ mit $\|g'(x)\| \leq q$ für alle $x \in D$, so erfüllt g in D die Kontraktionsbedingung (2.20).

Beweis: Seien $x, y \in D$ gegeben. Anwendung des Mittelwertsatzes in Integralform (vgl. Satz 1.3.4) liefert

$$g(x) - g(y) = \int_0^1 g'(x + t(y - x))(y - x) dt.$$

Wir schätzen mit einer geeigneten Vektornorm und einer verträglichen Matrixnorm ab:

$$\begin{aligned} \|g(x) - g(y)\| &\leq \int_0^1 \|g'(x + t(y - x))(y - x)\| dt \\ &\leq \int_0^1 \|g'(x + t(y - x))\| \cdot \|y - x\| dt \\ &\leq \max_{z \in D} \|g'(z)\| \cdot \int_0^1 \|y - x\| dt \\ &\leq q \|x - y\|. \end{aligned}$$

□

Bemerkung 4.2.2

Es ist möglich, die Kontraktionszahl q numerisch zu schätzen, sobald einige Iterierte zur Verfügung stehen. Man betrachte die Quotienten

$$q \approx \frac{|x^{(i+1)} - x^{(i)}|}{|x^{(i)} - x^{(i-1)}|}, \quad i = 1, 2, \dots$$

Hintergrund: Aus der Kontraktivität von g folgt

$$|x^{(i+1)} - x^{(i)}| = |g(x^{(i)}) - g(x^{(i-1)})| \leq q |x^{(i)} - x^{(i-1)}|.$$

Das folgende Beispiel zeigt einerseits, daß die Fixpunktiteration nicht immer konvergiert, andererseits wird der Einfluß von ρ verdeutlicht.

Beispiel 4.2.3

Berechnung von $\sqrt{2}$ als (positive) Nullstelle von $F(x) = x^2 - 2$. Ausgehend vom Startwert $x^{(0)} = 2$ wird zunächst mit $\rho(x) = 1$ für alle x gerechnet. Ausgabe

Iteration 1 : 4.0000000000, Fehler: 2.5857864376
 Iteration 2 : 18.0000000000, Fehler: 16.5857864376
 Iteration 3 : 340.0000000000, Fehler: 338.5857864376
 Iteration 4 : 115938.0000000000, Fehler: 115936.5857864376

Die Fixpunktiteration divergiert. Hier ist $g(x) = x + x^2 - 2$ und $g'(x) = 1 + 2x$. Damit ist $|g'(x)| \geq 1$ für $x \notin (-1, 0)$. In einer Umgebung des Startwerts $x^{(0)} = 2$ ist die Kontraktionsbedingung nicht erfüllt. Wird bei gleichem Startwert hingegen $\rho(x) = -1/4$ gewählt, erhält man die Ausgabe

Iteration 1 : 1.5000000000, Fehler: 0.0857864376, Rate: 0.1464466094
 Iteration 2 : 1.4375000000, Fehler: 0.0232864376, Rate: 0.2714466094
 Iteration 3 : 1.4208984375, Fehler: 0.0066848751, Rate: 0.2870716094
 Iteration 4 : 1.4161603451, Fehler: 0.0019467827, Rate: 0.2912220000
 Iteration 5 : 1.4147828143, Fehler: 0.0005692520, Rate: 0.2924065231
 Iteration 6 : 1.4143802114, Fehler: 0.0001666490, Rate: 0.2927509058
 Iteration 7 : 1.4142623658, Fehler: 0.0000488034, Rate: 0.2928515566
 Iteration 8 : 1.4142278560, Fehler: 0.0000142936, Rate: 0.2928810180
 Iteration 9 : 1.4142177488, Fehler: 0.0000041864, Rate: 0.2928896454
 Iteration 10 : 1.4142147886, Fehler: 0.0000012262, Rate: 0.2928921722
 Iteration 11 : 1.4142139215, Fehler: 0.0000003591, Rate: 0.2928929122
 Iteration 12 : 1.4142136676, Fehler: 0.0000001052, Rate: 0.2928931292
 Iteration 13 : 1.4142135932, Fehler: 0.0000000308, Rate: 0.2928931913
 Iteration 14 : 1.4142135714, Fehler: 0.0000000090, Rate: 0.2928932135
 Iteration 15 : 1.4142135650, Fehler: 0.0000000026, Rate: 0.2928932004

Die Fixpunktiteration konvergiert offensichtlich. Es ist $g(x) = x - (x^2 - 2)/4$ und $g'(x) = 1 - x/2$. Damit ist $|g'(x)| < 1$ für $x \in (0, 4)$. Damit ist g kontrahierend in einer Umgebung des Startwerts $x^{(0)} = 2$. Zu beachten ist noch, daß die Iterationsfolge im Intervall $(0, 4)$ verbleibt. Somit sind die Voraussetzungen des Banachschen Fixpunktsatzes erfüllt und es folgt die Konvergenz. Die Konvergenzkonstante beträgt hier ca. 0.29289.

Die Wahl von ρ ist entscheidend und leider problemabhängig.

Bemerkung 4.2.4 (Wahl von ρ)

Wir werden später sehen, daß die Wahl $\rho(x) = -F'(x)^{-1}$ sehr geeignet ist und auf das sogenannte Newtonverfahren

$$x^{(i+1)} = x^{(i)} - F'(x^{(i)})^{-1}F(x^{(i)}), \quad i = 0, 1, 2, \dots$$

führt. Insbesondere kann das Newtonverfahren also als Fixpunktiteration mit

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

interpretiert werden.

4.3 Newtonverfahren

Die grundsätzliche Idee für das Newtonverfahren wird zunächst im eindimensionalen Fall (4.1) erläutert und später auf den mehrdimensionalen Fall (4.2) übertragen. In jedem Fall wird vorausgesetzt, daß die Funktion F mindestens **stetig differenzierbar** ist.

4.3.1 Der eindimensionale Fall

Die Idee des Newtonverfahrens ist es, die Funktion F lokal im Punkt $x^{(i)}$ durch ihre **Tangente** $T(x)$ zu approximieren, d.h.

$$F(x) \approx T(x) := F(x^{(i)}) + F'(x^{(i)})(x - x^{(i)}),$$

vgl. Abbildung 4.3.

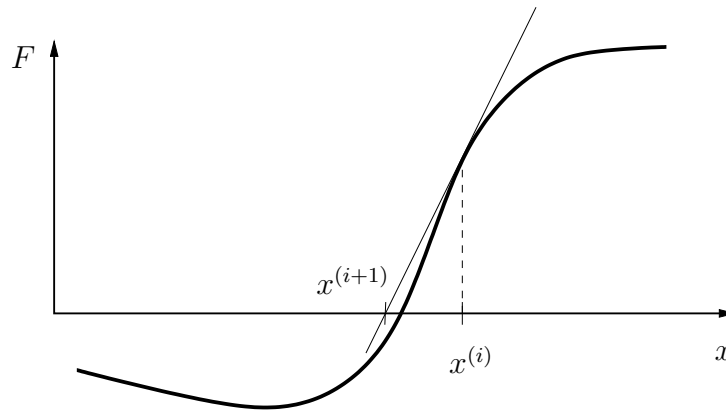


Abbildung 4.3: Newtonverfahren: Lokale Approximation von F durch Tangente im Punkt $x^{(i)}$. Die Nullstelle der Tangente definiert die neue Iterierte $x^{(i+1)}$.

Natürlich ist die Tangente i.a. lediglich eine Approximation an F . Ist F zweimal stetig differenzierbar, so gilt nach dem Taylor'schen Satz die Beziehung

$$F(x) = \underbrace{F(x^{(i)}) + F'(x^{(i)})(x - x^{(i)})}_{=T(x)} + \frac{1}{2}F''(\xi)(x - x^{(i)})^2,$$

wobei ξ zwischen x und $x^{(i)}$ liegt. Der große Vorteil der Approximation durch T besteht darin, daß die Nullstelle von T leicht berechnet werden kann, während dies für F nicht möglich ist. Auf Grund der Taylorentwicklung kann man auch erwarten, daß die Nullstelle von T eine hinreichend gute Approximation für eine Nullstelle \hat{x} von F darstellt, falls die Linearisierung in der Nähe von \hat{x} erfolgt.

Es sei $F'(x^{(i)}) \neq 0$. Dann besitzt T eine Nullstelle, die mit $x^{(i+1)}$ bezeichnet wird und sich berechnet zu

$$T(x^{(i+1)}) = 0 \quad \Leftrightarrow \quad x^{(i+1)} = x^{(i)} - \frac{F(x^{(i)})}{F'(x^{(i)})}.$$

$x^{(i+1)}$ wird als neue Approximation der Nullstelle gewählt. Iterative Anwendung dieser Regel liefert das lokale Newtonverfahren.

Algorithmus 4.3.1 ((Lokales) Newtonverfahren)

(i) Wähle $x^{(0)} \in \mathbb{R}$, $tol > 0$ und setze $i = 0$.

(ii) Falls $|F(x^{(i)})| \leq tol$, STOP.

(iii) Berechne

$$x^{(i+1)} = x^{(i)} - \frac{F(x^{(i)})}{F'(x^{(i)})}$$

(iv) Setze $i = i + 1$ und gehe zu (ii).

Bemerkung 4.3.2

Weitere sinnvolle Abbruchbedingungen mit geeignet gewählten Konstanten $\delta, \varepsilon, \gamma > 0$ lauten

$$\begin{aligned} |F'(x^{(i)})| &\leq \delta \\ |F(x^{(i)})| &\leq \varepsilon |F'(x^{(i)})| \\ |x^{(i+1)} - x^{(i)}| &\leq \gamma (1 + |F(x^{(i)})|) \end{aligned}$$

Es stellt sich die Frage, inwiefern das oben konstruierte Verfahren sinnvoll ist und wann es konvergiert.

Satz 4.3.3

Sei F zweimal stetig differenzierbar auf $[a, b]$. F habe eine Nullstelle $\hat{x} \in (a, b)$ mit $F'(\hat{x}) \neq 0$. Dann existiert ein $r > 0$, so daß das Newtonverfahren für alle Startwerte $x^{(0)}$ mit $|x^{(0)} - \hat{x}| \leq r$ mindestens quadratisch gegen \hat{x} konvergiert.

Beweis:

(i) Konvergenz:

Das Newtonverfahren wird als Fixpunktiteration mit

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

aufgefaßt, d.h. $\rho(x) = -1/F'(x)$. Wegen $F(\hat{x}) = 0$ folgt in \hat{x}

$$g'(\hat{x}) = 1 - \frac{F'(\hat{x})^2 - F(\hat{x})F''(\hat{x})}{F'(\hat{x})^2} = \frac{F(\hat{x})F''(\hat{x})}{F'(\hat{x})^2} = 0.$$

Da $F'(\hat{x}) \neq 0$ vorausgesetzt ist und da $g'(\hat{x}) = 0$ gilt, folgt aus der Stetigkeit von F' und g' , daß es ein $r > 0$ mit

$$|F'(x)| \geq \mu > 0, \quad |g'(x)| \leq q < 1$$

für alle $x \in [\hat{x} - r, \hat{x} + r] \subseteq [a, b]$ gibt. Also ist g eine kontrahierende Abbildung auf $[\hat{x} - r, \hat{x} + r]$. Es bleibt noch zu zeigen, daß die Fixpunktiteration nicht aus dem Intervall $[\hat{x} - r, \hat{x} + r]$ hinausführt. Zunächst sei bemerkt, daß \hat{x} Fixpunkt von g ist, denn wegen $F(\hat{x}) = 0$ folgt $g(\hat{x}) = \hat{x} - F(\hat{x})/F'(\hat{x}) = \hat{x}$. Sei nun $x \in [\hat{x} - r, \hat{x} + r]$ beliebig, d.h. es gilt $|x - \hat{x}| \leq r$. Zu zeigen ist, daß dann auch $g(x) \in [\hat{x} - r, \hat{x} + r]$ bzw. $|g(x) - \hat{x}| \leq r$ gilt. Dies folgt aus

$$|g(x) - \hat{x}| = |g(x) - \underbrace{g(\hat{x})}_{=\hat{x}}| \leq \underbrace{q}_{<1} \underbrace{|x - \hat{x}|}_{\leq r} < r.$$

Damit sind die Voraussetzungen des Fixpunktsatzes bewiesen und es folgt die Konvergenz des Newtonverfahrens für jeden Startwert $x^{(0)} \in [\hat{x} - r, \hat{x} + r]$.

(ii) Konvergenzordnung:

Taylorentwicklung um die Iterierte $x^{(i)}$ liefert

$$0 = F(\hat{x}) = F(x^{(i)}) + F'(x^{(i)})(\hat{x} - x^{(i)}) + \frac{1}{2}F''(\xi)(\hat{x} - x^{(i)})^2, \quad \xi \in [\hat{x} - r, \hat{x} + r].$$

Damit folgt

$$x^{(i+1)} - \hat{x} = x^{(i)} - \frac{F'(x^{(i)})}{F'(x^{(i)})} - \hat{x} = \frac{1}{2} \frac{F''(\xi)}{F'(x^{(i)})} (\hat{x} - x^{(i)})^2.$$

Da F'' beschränkt ist auf der kompakten Menge $[\hat{x} - r, \hat{x} + r]$ (folgt aus der Stetigkeit von F'') folgt für alle Startwerte $x^{(0)} \in [\hat{x} - r, \hat{x} + r]$

$$|x^{(i+1)} - \hat{x}| \leq \underbrace{\frac{1}{2\mu} \max_{\xi \in [a,b]} |F''(\xi)|}_{=:C} \cdot |\hat{x} - x^{(i)}|^2, \quad i = 0, 1, 2, \dots$$

Dies ist gerade die quadratische Konvergenz.

□

Bemerkung 4.3.4

Beachte, daß das Newtonverfahren nur für solche Anfangswerte $x^{(0)}$ mit $|\hat{x} - x^{(0)}| \leq r$ konvergiert. Für Anfangswerte außerhalb dieser Umgebung kann das Verfahren divergieren. Man spricht hier von **lokaler Konvergenz**.

Beispiel 4.3.5

Wir betrachten erneut das Problem, $\sqrt{2}$ als (positive) Nullstelle von $F(x) = x^2 - 2$ zu berechnen. Das Newtonverfahren liefert die folgende Ausgabe:

Iteration 1 : 1.5000000000, Fehler: 0.0857864376
 Iteration 2 : 1.4166666667, Fehler: 0.0024531043
 Iteration 3 : 1.4142156863, Fehler: 0.0000021239
 Iteration 4 : 1.4142135624, Fehler: 0.0000000000

Hier wird die Stärke des Newtonverfahrens im Vergleich zum Bisektionsverfahren deutlich. Während das Bisektionsverfahren hier 17 Iterationen benötigte, konvergiert das Newtonverfahren in nur 4 Iterationen, wobei die Anzahl der führenden Nullen im Fehler sich von Iteration zu Iteration in etwa verdoppeln. Dies ist die quadratische Konvergenz des Newtonverfahrens.

4.3.2 Der n -dimensionale Fall

Die Idee des Newtonverfahrens läßt sich problemlos auf den n -dimensionalen Fall übertragen. Das Newtonverfahren basiert auf lokaler Taylorentwicklung der vektorwertigen Funktion $F = (F_1, \dots, F_n)^\top$ um die Iterierte $x^{(i)}$:

$$F(x) = F(x^{(i)}) + F'(x^{(i)}) \cdot (x - x^{(i)}) + o(\|x - x^{(i)}\|).$$

Darin bezeichnet

$$F'(x) = \begin{pmatrix} \frac{\partial F_1}{\partial x_1}(x) & \cdots & \frac{\partial F_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1}(x) & \cdots & \frac{\partial F_n}{\partial x_n}(x) \end{pmatrix}$$

die **Jacobimatrix** von F and der Stelle $x = (x_1, \dots, x_n)^\top$. Vernachlässigt man den Term $o(\|x - x^{(i)}\|)$ in der Taylorentwicklung von F , so wird F lokal durch ihre Richtungstangente T im Punkt $x^{(i)}$ in Richtung $x - x^{(i)}$ mit

$$T(x) = F(x^{(i)}) + F'(x^{(i)}) \cdot (x - x^{(i)})$$

approximiert (F wird in $x^{(i)}$ linearisiert.). Die neue Iterierte $x^{(i+1)}$ ergibt aus der Nullstelle der Richtungstangente und führt auf die Darstellung

$$x^{(i+1)} = x^{(i)} - F'(x^{(i)})^{-1} \cdot F(x^{(i)}). \quad (4.4)$$

In der Praxis wird die Inverse der Jacobimatrix $F'(x)^{-1}$ **niemals** explizit berechnet, da hierfür n lineare Gleichungssysteme der Dimension $n \times n$ gelöst werden müßten. Stattdessen wird (4.4) in der Form

$$\begin{aligned} F'(x^{(i)}) \cdot d^{(i)} &= -F(x^{(i)}), \\ x^{(i+1)} &= x^{(i)} + d^{(i)}, \quad i = 0, 1, 2, \dots \end{aligned}$$

gelöst. Diese Variante benötigt nicht mehr die Inverse der Jacobimatrix und erfordert nur die Lösung **eines** linearen Gleichungssystems für $d^{(i)}$. Zusammenfassend erhalten wir

Algorithmus 4.3.6 ((Lokales) Newtonverfahren in \mathbb{R}^n)

(i) Wähle $x^{(0)} \in \mathbb{R}^n$, $tol > 0$ und setze $i = 0$.

(ii) Falls $\|F(x^{(i)})\| \leq tol$, STOP.

(iii) Löse (z.B. mit der LR-Zerlegung) das lineare Gleichungssystem

$$F'(x^{(i)}) \cdot d^{(i)} = -F(x^{(i)})$$

und setze

$$x^{(i+1)} = x^{(i)} + d^{(i)}.$$

(iv) Setze $i = i + 1$ und gehe zu (ii).

Wir wollen die Konvergenz des Verfahrens untersuchen und benötigen einige Hilfssätze. Ziel ist es, die lokal superlineare Konvergenz bzw. sogar die lokal quadratische Konvergenz zu zeigen. Hilfreich ist wiederum die Interpretation des Newtonverfahrens als spezielle Fixpunktiteration

$$x^{(i+1)} = g(x^{(i)}), \quad i = 0, 1, 2, \dots \quad (4.5)$$

mit der Fixpunktfunktion

$$g(x) := x - F'(x)^{-1} \cdot F(x). \quad (4.6)$$

Zunächst zeigen wir, daß die Jacobimatrix $F'(x)$ für x hinreichend nahe an \hat{x} invertierbar bleibt, sofern $F'(\hat{x})$ invertierbar war.

Hilfsatz 4.3.7 (Banach-Lemma)

Seien $A, B \in \mathbb{R}^{n \times n}$ mit $\|I - B \cdot A\| < 1$ und $\|\cdot\|$ eine induzierte Matrixnorm. Dann sind A und B invertierbar, und es gilt die Abschätzung

$$\|A^{-1}\| \leq \frac{\|B\|}{1 - \|I - B \cdot A\|}.$$

Eine analoge Ungleichung gilt für B .

Beweis: Gemäß Hilfsatz 2.5.12 gilt für eine beliebige Matrix $S \in \mathbb{R}^{n \times n}$ mit $\|S\| < 1$ die Ungleichung

$$\|(I - S)^{-1}\| \leq \frac{1}{1 - \|S\|},$$

wobei $I - S$ invertierbar ist. Wähle $S = I - B \cdot A$. Nach Voraussetzung gilt $\|S\| = \|I - B \cdot A\| < 1$ und somit ist $I - S = B \cdot A$ nach Hilfsatz 2.5.12 invertierbar. Folglich sind A und B invertierbar. Aus $A^{-1} = (I - S)^{-1}B$ folgt mit obiger Ungleichung

$$\|A^{-1}\| \leq \|(I - S)^{-1}\| \cdot \|B\| \leq \frac{\|B\|}{1 - \|I - B \cdot A\|}.$$

□

Hilfsatz 4.3.8

Seien $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ stetig differenzierbar, $\hat{x} \in \mathbb{R}^n$ und $F'(\hat{x})$ invertierbar. Dann existiert ein $r > 0$, so daß auch $F'(x)$ für alle $x \in U_r(\hat{x}) = \{x \in \mathbb{R}^n \mid \|x - \hat{x}\| < r\}$ invertierbar ist. Weiter existiert eine Konstante $C > 0$ mit

$$\|F'(x)^{-1}\| \leq C \quad \forall x \in U_r(\hat{x}).$$

Beweis: Da F' in \hat{x} stetig ist, existiert $r > 0$ mit

$$\|F'(\hat{x}) - F'(x)\| \leq \frac{1}{2\|F'(\hat{x})^{-1}\|} \quad \forall x \in U_r(\hat{x}).$$

Also ist

$$\|I - F'(\hat{x})^{-1}F'(x)\| \leq \|F'(\hat{x})^{-1}\| \cdot \|F'(\hat{x}) - F'(x)\| \leq \frac{1}{2}$$

für alle $x \in U_r(\hat{x})$. Das Banach-Lemma 4.3.7 mit $A := F'(x)$ und $B := F'(\hat{x})^{-1}$ liefert

$$\|F'(x)^{-1}\| \leq \frac{\|F'(\hat{x})^{-1}\|}{1 - \underbrace{\|I - F'(\hat{x})^{-1}F'(x)\|}_{\leq 1/2}} \leq 2\|F'(\hat{x})^{-1}\| =: C.$$

□

Der folgende Hilfssatz stellt fest, daß $g'(\hat{x}) = 0$ in einer Nullstelle \hat{x} von F gilt, wobei g durch (4.6) gegeben ist.

Hilfsatz 4.3.9

Seien \hat{x} Nullstelle von F in (4.2), $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ stetig differenzierbar in \hat{x} und $F'(\hat{x})$ invertierbar. Dann existiert ein $r > 0$, so daß g in (4.6) für alle $x \in U_r(\hat{x})$ wohldefiniert ist. Darüber hinaus ist g differenzierbar in \hat{x} mit

$$g'(\hat{x}) = I - F'(\hat{x})^{-1} \cdot F'(\hat{x}) = 0.$$

Beweis: Übung

□

Der folgende Satz zeigt die schnelle lokale Konvergenz, falls $g'(\hat{x}) = 0$ in einem Fixpunkt von g gilt.

Hilfsatz 4.3.10

Sei $D \subseteq \mathbb{R}^n$ offen und $g : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ besitze einen Fixpunkt \hat{x} in D . Weiter sei g differenzierbar in \hat{x} mit

$$g'(\hat{x}) = 0.$$

Dann existiert ein $r > 0$ mit $U_r(\hat{x}) \subseteq D$, so daß die Iteration (4.5) für alle Startwerte $x^{(0)} \in U_r(\hat{x})$ lokal superlinear gegen \hat{x} konvergiert. Darüber hinaus ist \hat{x} der einzige Fixpunkt in $U_r(\hat{x})$.

Gilt zusätzlich für Konstanten $\alpha \geq 0$ und $p > 1$ noch

$$\|g(x) - g(\hat{x})\| \leq \alpha \|x - \hat{x}\|^p \quad \forall x \in U_r(\hat{x}),$$

so ist die Konvergenzordnung lokal um \hat{x} (mindestens) gleich p .

Beweis:

(i) Konvergenz:

Da g differenzierbar ist in \hat{x} , existiert zu jedem $\varepsilon > 0$ ein $r > 0$ mit

$$\frac{\|g(x) - g(\hat{x}) - g'(\hat{x})(x - \hat{x})\|}{\|x - \hat{x}\|} \leq \varepsilon \quad \forall x \in U_r(\hat{x}), x \neq \hat{x}.$$

Wegen $g'(\hat{x}) = 0$ folgt daraus

$$\|g(x) - g(\hat{x})\| = \|g(x) - \hat{x}\| \leq \varepsilon \|x - \hat{x}\| \quad \forall x \in U_r(\hat{x}).$$

Wähle nun $\varepsilon < 1$. Dann ist g eine Selbstabbildung auf $U_r(\hat{x})$ wegen

$$x \in U_r(\hat{x}) \Rightarrow \|g(x) - \hat{x}\| \leq \varepsilon \|x - \hat{x}\| < r \Rightarrow g(x) \in U_r(\hat{x}).$$

Insbesondere ist die Folge $\{x^{(i)}\}$ in (4.5) wohldefiniert und verbleibt für jeden Startwert $x^{(0)}$ in $U_r(\hat{x})$. Die Konvergenz der Folge folgt nun ähnlich wie im Banach'schen Fixpunktsatz: Wähle $x^{(0)} \in U_r(\hat{x})$ beliebig. Es folgt

$$\|x^{(i)} - \hat{x}\| = \|g(x^{(i-1)}) - \hat{x}\| \leq \varepsilon \|x^{(i-1)} - \hat{x}\| \leq \dots \leq \varepsilon^i \|x^{(0)} - \hat{x}\| \xrightarrow{i \rightarrow \infty} 0.$$

Zur Eindeutigkeit: Ist $y \in U_r(\hat{x})$ ein beliebiger Fixpunkt von g , dann ist $\|\hat{x} - y\| = \|g(\hat{x}) - g(y)\| \leq \varepsilon \|\hat{x} - y\|$ und daher $y = \hat{x}$.

(ii) Superlineare Konvergenz:

Sei nun $\{x^{(i)}\}$ eine Iterationsfolge mit

$$\lim_{i \rightarrow \infty} x^{(i)} = \hat{x}.$$

Ist $x^{(i_0)} = \hat{x}$ für ein $i_0 \in \mathbb{N}$, so gilt $x^{(i_0+1)} = g(x^{(i_0)}) = g(\hat{x}) = \hat{x}$ und somit $x^{(i)} = \hat{x}$ für alle $i \geq i_0$. Wir können also ohne Beschränkung der Allgemeinheit annehmen, daß $x^{(i)} \neq \hat{x}$ für alle $i \in \mathbb{N}$ gilt. Wegen $g'(\hat{x}) = 0$ und der Differenzierbarkeit von g in \hat{x} folgt dann

$$\lim_{i \rightarrow \infty} \frac{\|x^{(i+1)} - \hat{x}\|}{\|x^{(i)} - \hat{x}\|} = \lim_{i \rightarrow \infty} \frac{\|g(x^{(i)}) - g(\hat{x}) - g'(\hat{x})(x^{(i)} - \hat{x})\|}{\|x^{(i)} - \hat{x}\|} = 0,$$

d.h. die Folge ist superlinear konvergent.

(iii) **Konvergenzordnung p :**

Wähle $\tilde{r} = (\frac{1}{\alpha})^{1/(p-1)} > 0$. Für alle $x \in U_{\tilde{r}}(\hat{x})$ gilt $\|x - \hat{x}\| \leq (\frac{1}{\alpha})^{1/(p-1)}$. Dann gilt für $x^{(i)} \in U_{\tilde{r}}(\hat{x})$ die Beziehung

$$\|x^{(i+1)} - \hat{x}\| = \|g(x^{(i)}) - g(\hat{x})\| \leq \alpha \|x^{(i)} - \hat{x}\|^p \leq \|x^{(i)} - \hat{x}\| \leq \tilde{r},$$

d.h. die Folge $\{x^{(i)}\}$ bleibt in $U_{\tilde{r}}(\hat{x})$. Der Rest folgt aus der Voraussetzung. □

Mit diesen Vorarbeiten können wir die Konvergenz des Newtonverfahrens beweisen.

Satz 4.3.11 (Konvergenzsatz für das Newtonverfahren)

Es sei $D \subseteq \mathbb{R}^n$ offen und $F : D \rightarrow \mathbb{R}^n$ besitze eine Nullstelle \hat{x} in D . F sei stetig differenzierbar auf D und $F'(\hat{x})$ sei invertierbar. Dann gelten folgende Aussagen:

(i) Es gibt ein $r > 0$, so daß das lokale Newton-Verfahren für alle Startwerte $x^{(0)} \in U_r(\hat{x})$ wohldefiniert ist und die Folge $\{x^{(i)}\}$ konvergiert gegen \hat{x} .

(ii) Die Konvergenzrate ist superlinear.

(iii) \hat{x} ist die einzige Nullstelle in $U_r(\hat{x})$.

(iv) Erfüllt F' zusätzlich noch die Lipschitz-Bedingung

$$\|F'(x) - F'(y)\| \leq L \cdot \|x - y\| \quad \forall x, y \in D,$$

so ist die Folge $\{x^{(i)}\}$ sogar quadratisch konvergent.

Beweis: Die superlineare Konvergenz und die Eindeutigkeit folgen aus den Hilfssätzen 4.3.10 und 4.3.9. Zu zeigen bleibt die quadratische Konvergenz. Nach Hilfssatz 4.3.10 genügt es,

$$\|g(x) - g(\hat{x})\| \leq \alpha \|x - \hat{x}\|^2 \quad \forall x \in U_r(\hat{x})$$

mit einer Konstanten α zu zeigen. Anwendung des Mittelwertsatzes in Integralform 1.3.4 und Ausnutzung der Lipschitz-Bedingung liefert

$$\begin{aligned} \|g(x) - g(\hat{x})\| &= \|x - \hat{x} - F'(x)^{-1}F(x)\| \\ &= \|F'(x)^{-1}(F(\hat{x}) - F(x) - F'(x)(\hat{x} - x))\| \\ &\leq \|F'(x)^{-1}\| \cdot \left\| \int_0^1 F'(x + t(\hat{x} - x))(\hat{x} - x)dt - F'(x)(\hat{x} - x) \right\| \\ &= \|F'(x)^{-1}\| \cdot \left\| \int_0^1 (F'(x + t(\hat{x} - x)) - F'(x)) \cdot (\hat{x} - x)dt \right\| \\ &\leq \|F'(x)^{-1}\| \cdot \int_0^1 Lt \|\hat{x} - x\|^2 dt \\ &= \|F'(x)^{-1}\| \cdot L \cdot \|\hat{x} - x\|^2 \cdot \int_0^1 t dt \\ &= \|F'(x)^{-1}\| \cdot \frac{L}{2} \cdot \|\hat{x} - x\|^2. \end{aligned}$$

Da $F'(\hat{x})$ invertierbar ist, existiert nach Hilfssatz 4.3.8 auch $F'(x)^{-1}$ in einer Umgebung von \hat{x} und ist dort beschränkt. Damit ist die quadratische Konvergenz gezeigt. \square

Beispiel 4.3.12 (Funktion von Himmelblau)

Wir betrachten ein Beispiel aus der unrestringierten Optimierung. Gesucht ist ein lokales Minimum der Funktion

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2.$$

Aus der Theorie der nichtlinearen Optimierung ist bekannt, daß ein lokales Minimum (\hat{x}, \hat{y}) von f notwendig die Bedingung $\nabla f(\hat{x}, \hat{y}) = 0$ erfüllt, d.h. wir suchen eine Nullstelle der nichtlinearen Funktion

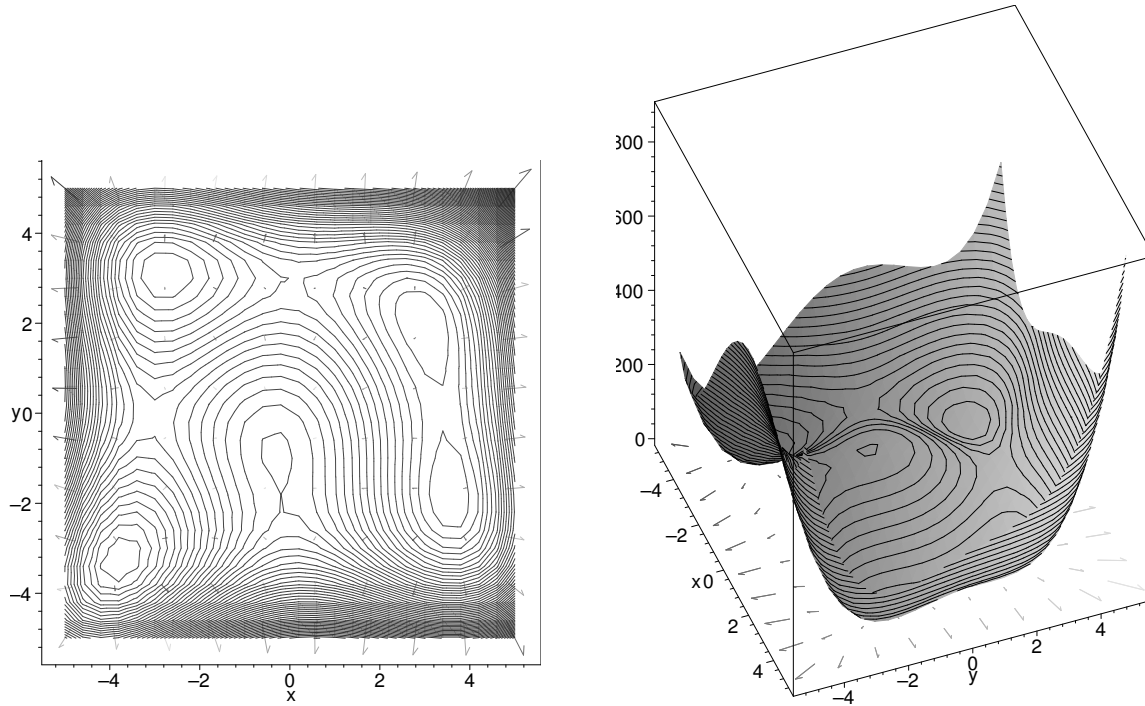
$$F(x, y) := \nabla f(x, y) = \begin{pmatrix} 4x(x^2 + y - 11) + 2(x + y^2 - 7) \\ 2(x^2 + y - 11) + 4y(x + y^2 - 7) \end{pmatrix},$$

welche die Jacobimatrix

$$F'(x, y) = \nabla^2 f(x, y) = \begin{pmatrix} 12x^2 + 4y - 42 & 4(x + y) \\ 4(x + y) & 4x + 12y^2 - 26 \end{pmatrix}$$

besitzt. Ein Blick auf den Graphen von f zeigt, daß es in diesem Beispiel mehrere Nullstellen gibt, nämlich

- 4 lokale Minimalstellen (zugleich global) mit Funktionswert 0; darunter der Punkt $(3, 2)^\top$.
- 4 Sattelpunkte
- ein lokales Maximum in $(-0.270845, -0.923039)^\top$



Als Abbruchkriterium des Newtonverfahrens verwenden wir $\|F(x, y)\| \leq 10^{-13}$ und als Startpunkt $(x^{(0)}, y^{(0)}) = (4, 2.5)$.

Das Newtonverfahren liefert das folgende Ergebnis:

ITER	X(1)	X(2)	F(X)	DX	P=1	P=2
0	0.4000000E+01	0.2500000E+01	0.1351240E+03	0.0000000E+00	0.0000000E+00	0.0000000E+00
1	0.3281417E+01	0.2056664E+01	0.2617493E+02	0.8443389E+00	0.2567589E+00	0.2296521E+00
2	0.3035131E+01	0.1988137E+01	0.2424694E+01	0.2556422E+00	0.1291689E+00	0.4499638E+00
3	0.3000634E+01	0.1999744E+01	0.4203618E-01	0.3639711E-01	0.1844451E-01	0.4974264E+00
4	0.3000000E+01	0.2000000E+01	0.1406811E-04	0.6837008E-03	0.3217815E-03	0.4704954E+00
5	0.3000000E+01	0.2000000E+01	0.1500787E-11	0.2200729E-06	0.1109167E-06	0.5039996E+00
6	0.3000000E+01	0.2000000E+01	0.0000000E+00	0.2440976E-13	0.0000000E+00	0.0000000E+00

Die letzten beiden Spalten testen die Folge $(x^{(i)}, y^{(i)})^\top$, $i = 0, 1, 2, \dots$, auf lineare ($P=1$ strebt gegen Wert $\neq 0$, $P=2$ explodiert), superlineare ($P=1$ strebt gegen 0, $P=2$ explodiert) und quadratische Konvergenz ($P=1$ strebt gegen 0, $P=2$ strebt gegen Wert $\neq 0$).

Die Rechnungen zeigen eine quadratische Konvergenz des Newtonverfahrens gegen das Minimum $(3, 2)^\top$.

Für andere Startwerte konvergiert das Newtonverfahren mitunter gegen andere Nullstellen von F .

In der Praxis stellt sich die Frage, wann das Newtonverfahren angehalten werden kann. Das Kriterium $\|F(x^{(i)})\| \leq \text{tol}$ ist zwar naheliegend, aber nicht skalierungsinvariant, d.h. durch Multiplikation von F mit einer Konstanten, läßt sich die Bedingung stets erfüllen und gibt mitunter keine Auskunft über die Güte der Iterierten $x^{(i)}$.

Als geeigneteres Konvergenzkriterium hat sich beispielsweise der **natürliche Monotonietest** bewährt, vgl. Deuffhard und Hohmann [DH91]. In jedem Iterationsschritt wird

dabei geprüft, ob

$$\|\bar{d}^{(i+1)}\| \leq \Theta \|d^{(i)}\|$$

gilt, wobei meist $\Theta = 0.5$ gewählt wird. Darin bezeichnet $d^{(i)}$ die Newtonrichtung und $\bar{d}^{(i+1)}$ ist die Lösung des linearen Gleichungssystems

$$F'(x^{(i)})\bar{d}^{(i+1)} = -F(x^{(i+1)}).$$

Die Lösung dieses Gleichungssystems erfordert nur eine Vorwärts-Rückwärtssubstitution, da die LR-Zerlegung von $F'(x^{(i)})$ bereits bei der Berechnung von $x^{(i+1)}$ berechnet wurde.

4.3.3 Varianten des Newtonverfahrens

In der Praxis werden häufig Varianten des Newtonverfahrens verwendet, die hauptsächlich versuchen, die Vorteile des Newtonverfahrens (superlineare bzw. quadratische Konvergenzgeschwindigkeit, Anwendbarkeit für $n > 1$) zu erhalten und die Nachteile des Newtonverfahrens zu umgehen. Die Hauptnachteile des Verfahrens bestehen in der i.a. nur lokalen Konvergenz des Verfahrens, d.h. sobald der Startwert nicht hinreichend nahe an der Nullstelle liegt, besteht die Gefahr der Divergenz des Verfahrens. Desweiteren wird die Jacobimatrix von F benötigt. Die Berechnung derselben kann sehr aufwändig sein und ist oft nur numerisch möglich, mitunter ist sie auch nicht einmal invertierbar.

Vereinfachtes Newtonverfahren

Beim vereinfachten Newtonverfahren wird die Jacobimatrix F' nicht in jedem Iterationsschritt neu berechnet, sondern es wird vereinfachend die konstante Matrix $A_0 := F'(x^{(0)})$ für alle Iterationen (bzw. für eine bestimmte Anzahl von Iterationen) verwendet. Durch diesen Ansatz reduziert sich zwar der Rechenaufwand drastisch, was insbesondere von Vorteil ist, wenn die Auswertung von F' sehr teuer ist. Leider geht auch die superlineare Konvergenz des Newtonverfahrens verloren. Immerhin kann für das vereinfachte Newtonverfahren noch die lineare Konvergenz gezeigt werden.

Finite-Differenzen-Approximation

In der Praxis kann die Jacobimatrix häufig nicht explizit angegeben werden. Stattdessen wird sie durch Differenzenquotienten approximiert, etwa durch den Vorwärtsdifferenzenquotienten

$$\frac{\partial F_j}{\partial x_i}(x) \approx \frac{F_j(x + he_i) - F_j(x)}{h}$$

mit dem i -ten Einheitsvektor $e_i \in \mathbb{R}^n$.

Unter gewissen Voraussetzungen kann lineare Konvergenz und sogar superlineare Konvergenz bewiesen werden.

Quasi-Newton-Verfahren:

Quasi-Newton-Verfahren ersetzen die Jacobimatrix $F'(x^{(i)})$ im Newtonverfahren durch eine Matrix B_i , die eine geeignete Approximation an $F'(x^{(i)})$ darstellen soll. Dieser Ansatz vermeidet die Berechnung der Jacobimatrix und reduziert den Rechenaufwand, vorausgesetzt, die Matrix B_i kann kostengünstig berechnet werden. Bei Quasi-Newton-Verfahren wird die Approximation B_{i+1} durch eine Update-Formel aus B_i berechnet. Dabei wird gefordert, daß B_{i+1} möglichst nahe an B_i liegt, wobei der Abstand durch die Frobeniusnorm $\|B_{i+1} - B_i\|_F$ gemessen wird. Es stellt sich die Frage, welche weiteren Eigenschaften die Matrizen B_i , $i = 0, 1, \dots$, besitzen müssen, damit das Quasi-Newton-Verfahren ähnlich schnell konvergiert wie das Newtonverfahren (also möglichst superlinear).

Der folgende Satz charakterisiert die superlineare Konvergenz. Hierbei verwenden wir die Abkürzungen

$$d^{(i)} := x^{(i+1)} - x^{(i)}, \quad y^{(i)} := F(x^{(i+1)}) - F(x^{(i)}).$$

Satz 4.3.13 (Dennis, Moré)

Es sei $D \subseteq \mathbb{R}^n$ konvex und offen und $F : D \rightarrow \mathbb{R}^n$ stetig differenzierbar. $\hat{x} \in D$ sei Nullstelle von F und $F'(\hat{x})$ sei invertierbar. Desweiteren erfülle $F'(\cdot)$ die Lipschitz-Bedingung

$$\|F'(x) - F'(y)\| \leq L \cdot \|x - y\| \quad \forall x, y \in D.$$

Die Folge $\{x^{(i)}\}$ sei durch

$$B_i(x^{(i+1)} - x^{(i)}) = -F(x^{(i)}), \quad i = 0, 1, 2, \dots \quad (4.7)$$

definiert und es gelte

$$\lim_{i \rightarrow \infty} x^{(i)} = \hat{x}, \quad x^{(i)} \neq \hat{x}, \quad x^{(i)} \in D \quad \forall i.$$

Schließlich seien die Matrizen B_i für alle i invertierbar.

Dann sind äquivalent:

(a) Die Folge $\{x^{(i)}\}$ ist superlinear konvergent.

(b) Es gilt

$$\lim_{i \rightarrow \infty} \frac{\|(B_i - F'(\hat{x}))d^{(i)}\|}{\|d^{(i)}\|} = 0. \quad (4.8)$$

(c) Es gilt

$$\lim_{i \rightarrow \infty} \frac{\|B_i d^{(i)} - y^{(i)}\|}{\|d^{(i)}\|} = 0. \quad (4.9)$$

Beweis: siehe Jarre und Stoer [JS04], S. 175 □

Welche Konsequenz hat der Satz für die Wahl von B_i ?

Bedingung (c) wäre sofort erfüllt, wenn $B_i d^{(i)} = y^{(i)} = F(x^{(i+1)}) - F(x^{(i)})$ gelten würde. Allerdings ist diese Bedingung i.a. nicht erfüllbar, da $d^{(i)}$ durch (4.7) gegeben ist.

Jedoch kann die neue Matrix B_{i+1} so bestimmt werden, daß wenigstens die sogenannte **Quasi-Newton-Gleichung** oder **Sekantenbedingung**

$$B_{i+1} d^{(i)} = y^{(i)} \quad (4.10)$$

erfüllt ist. Im eindimensionalen Fall $n = 1$ ist B_{i+1} hierdurch eindeutig festgelegt und das resultierende Verfahren entspricht genau dem Sekantenverfahren, welches in Abschnitt 4.4 behandelt wird.

Bemerkung 4.3.14

Die Quasi-Newton-Gleichung kann auch wie folgt motiviert werden. Der Mittelwertsatz in Integralform 1.3.4 liefert

$$y^{(i)} = F(x^{(i+1)}) - F(x^{(i)}) = \int_0^1 F'(x^{(i)} + t(x^{(i+1)} - x^{(i)})) dt (x^{(i+1)} - x^{(i)}).$$

Damit kann B_{i+1} als Approximation an die gemittelte Jacobimatrix

$$\int_0^1 F'(x^{(i)} + t(x^{(i+1)} - x^{(i)})) dt$$

interpretiert werden.

Ein Ansatz zur expliziten Berechnung von B_{i+1} ergibt sich, wenn gefordert wird, daß B_{i+1} und B_i sich nur entlang der Richtung $x^{(i+1)} - x^{(i)}$ unterscheiden und für alle $y \perp d^{(i)}$ identisch sind, d.h.

$$(B_{i+1} - B_i)y = 0 \quad \forall y \perp d^{(i)}.$$

Diese Forderung wird erfüllt durch die Matrix

$$B_{i+1} = B_i + \frac{u(d^{(i)})^\top}{(d^{(i)})^\top d^{(i)}}, \quad u \in \mathbb{R}^n.$$

Der Vektor $u \in \mathbb{R}^n$ wird durch die Quasi-Newton-Bedingung (4.10) festgelegt:

$$y^{(i)} = B_{i+1} d^{(i)} = B_i d^{(i)} + u \quad \Rightarrow \quad u = y^{(i)} - B_i d^{(i)}.$$

Wir erhalten den sogenannten **Broyden-Rang-1-Update**

$$B_{i+1} = B_i + \frac{(y^{(i)} - B_i d^{(i)})(d^{(i)})^\top}{(d^{(i)})^\top d^{(i)}}, \quad d^{(i)} = x^{(i+1)} - x^{(i)}, \quad y^{(i)} = F(x^{(i+1)}) - F(x^{(i)}). \quad (4.11)$$

Bemerkung 4.3.15

Einen interessanten Zusammenhang mit der Optimierung liefert die folgende Aussage:
 B_{i+1} ist die eindeutig bestimmte Lösung des Minimierungsproblems

$$\min_{B \in \mathbb{R}^{n \times n}} \|B - B_i\|_F \quad \text{unter} \quad Bd^{(i)} = y^{(i)}.$$

($\|\cdot\|_F$ bezeichnet die Frobenius-Norm).

Zum Start des Verfahrens wird noch eine geeignete Startmatrix B_0 benötigt, etwa $B_0 = F'(x^{(0)})$, falls $F'(x^{(0)})$ invertierbar ist, andernfalls $B_0 = \alpha I$ mit einem $\alpha > 0$.

Zusammenfassend lautet das Quasi-Newton-Verfahren wie folgt:

Algorithmus 4.3.16 (Quasi-Newton-Verfahren)

(i) Wähle $x^{(0)} \in \mathbb{R}^n$, $tol > 0$, $B_0 \in \mathbb{R}^{n \times n}$ invertierbar und setze $i = 0$.

(ii) Falls $\|F(x^{(i)})\| \leq tol$, STOP.

(iii) Löse (z.B. mit der LR-Zerlegung) das lineare Gleichungssystem

$$B_i \cdot d^{(i)} = -F(x^{(i)})$$

und setze

$$x^{(i+1)} = x^{(i)} + d^{(i)}.$$

(iv) Berechne B_{i+1} gemäß (4.11).

(v) Setze $i = i + 1$ und gehe zu (ii).

Globalisierung des Newtonverfahrens

Ein Hauptnachteil des Newtonverfahrens ist die nur lokale Konvergenz. Es gibt jedoch Ansätze, die eine Konvergenz des Newtonverfahrens für beliebige Startwerte garantieren (natürlich nur unter bestimmten Voraussetzungen). Die Globalisierung des Newtonverfahrens besteht in der Einführung einer Schrittweite $\alpha_i > 0$ bei der Berechnung von $x^{(i+1)}$ gemäß

$$x^{(i+1)} = x^{(i)} + \alpha_i d^{(i)}.$$

Die Schrittweite α_i wird hierbei so bestimmt, daß das Residuum $\|F\|$ beim Übergang von $x^{(i)}$ zu $x^{(i+1)}$ hinreichend stark abnimmt. Diese Technik ist aus der Theorie der nichtlinearen Optimierung bekannt und kann z.B. durch eine sogenannte Liniensuche für die

eindimensionale Funktion

$$\varphi(\alpha) := \|F(x^{(i)} + \alpha d^{(i)})\|_2^2$$

realisiert werden. Details finden sich z.B. in Geiger und Kanzow [GK99] und Kosmol [Kos93]. Globalisierte Newtonverfahren sind in der Regel so konstruiert, daß die Schrittweite $\alpha_i = 1$ akzeptiert wird, sobald $x^{(i)}$ den lokalen Konvergenzeinzugsbereich des lokalen Newtonverfahrens erreicht. Insofern erben die globalisierten Varianten die lokalen Konvergenzeigenschaften des Newtonverfahrens.

Wir illustrieren einige der oben beschriebenen Varianten numerisch.

Beispiel 4.3.17 (Funktion von Himmelblau, vgl. Beispiel 4.3.12)

Wir betrachten wieder Beispiel 4.3.12 und suchen ein lokales Minimum der Funktion

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2,$$

indem die für ein lokales Minimum notwendige Bedingung $\nabla f(\hat{x}, \hat{y}) = 0$ mit dem Newtonverfahren und Newtonvarianten gelöst wird. Gesucht ist also eine Nullstelle der nicht-linearen Funktion

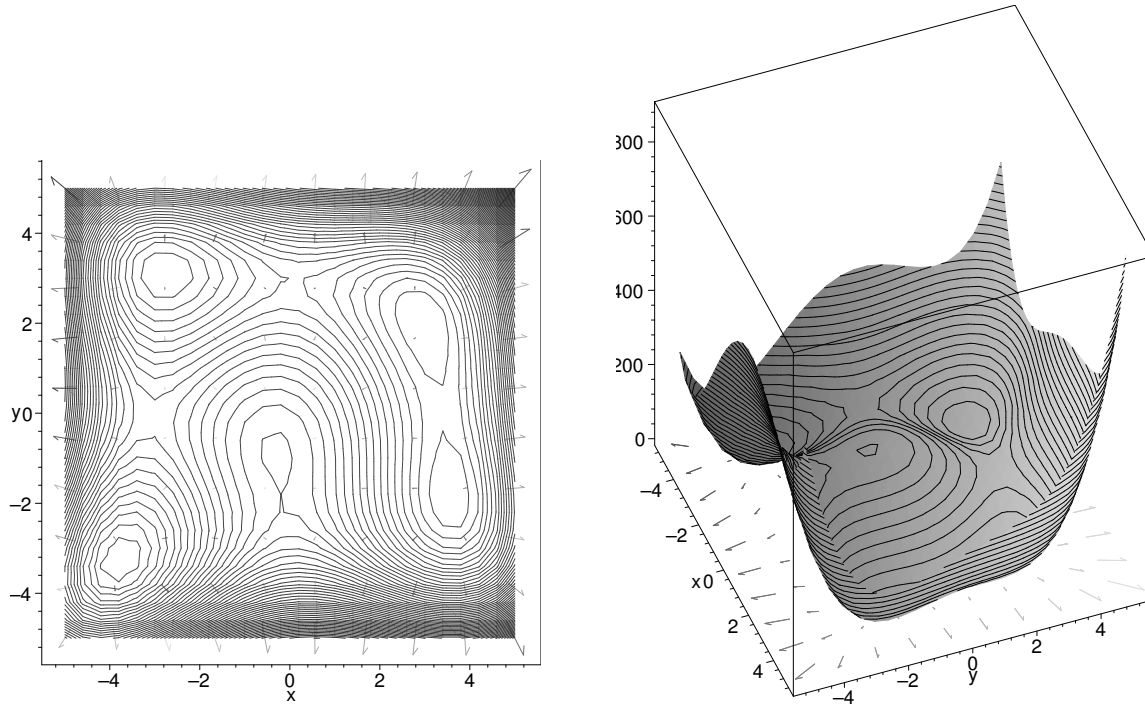
$$F(x, y) := \nabla f(x, y) = \begin{pmatrix} 4x(x^2 + y - 11) + 2(x + y^2 - 7) \\ 2(x^2 + y - 11) + 4y(x + y^2 - 7) \end{pmatrix},$$

welche die Jacobimatrix

$$F'(x, y) = \nabla^2 f(x, y) = \begin{pmatrix} 12x^2 + 4y - 42 & 4(x + y) \\ 4(x + y) & 4x + 12y^2 - 26 \end{pmatrix}$$

besitzt. Ein Blick auf den Graphen von f zeigt, daß es in diesem Beispiel mehrere Nullstellen gibt, nämlich

- 4 lokale Minimalstellen (zugleich global) mit Funktionswert 0; darunter der Punkt $(3, 2)^\top$.
- 4 Sattelpunkte
- ein lokales Maximum in $(-0.270845, -0.923039)^\top$



Wir diskutieren einige Varianten des Newtonverfahrens. Die Iterationsvorschrift für einen Startwert $z^{(0)} = (x^{(0)}, y^{(0)})^\top$ lautet

$$\begin{aligned} A(z^{(i)})d^{(i)} &= -F(z^{(i)}), \\ z^{(i+1)} &= z^{(i)} + d^{(i)}, \quad i = 0, 1, 2, \dots \end{aligned}$$

Die Matrix $A(\cdot) \in \mathbb{R}^{n \times n}$ wird dabei auf die folgenden Arten berechnet:

- (i) $A(x, y) := F'(x, y)$ (klassisches Newtonverfahren);
- (ii) finite Differenzen-Approximation der Matrix $F'(x, y)$, d.h. $A(x, y) = (A_{jk}(x, y))_{j,k=1,2}$ ist gegeben durch

$$A(x, y) := \begin{pmatrix} \frac{F_1(x+h, y) - F_1(x, y)}{h_i} & \frac{F_1(x, y+h) - F_1(x, y)}{h_i} \\ \frac{F_2(x+h, y) - F_2(x, y)}{h_i} & \frac{F_2(x, y+h) - F_2(x, y)}{h_i} \end{pmatrix}.$$

Die Schrittweite h_i sei durch $h_i = \frac{0.1}{i+1}$ gegeben, wobei i den Iterationsindex des Newtonverfahrens bezeichnet. Beachte, daß $h_i \rightarrow 0$ für $i \rightarrow \infty$ gilt.

- (iii) vereinfachtes Newtonverfahren mit $A(x^{(i)}, y^{(i)}) := F'(x^{(0)}, y^{(0)})$ für $i = 0, 1, 2, \dots$
- (iv) Quasi-Newton-Verfahren mit $B_0 = F'(x^{(0)}, y^{(0)})$ und $A(z^{(i)}) = B_i$.

Als Abbruchkriterium des Newtonverfahrens verwenden wir jeweils $\|F(x, y)\| \leq 10^{-13}$ und als Startpunkt $(x^{(0)}, y^{(0)}) = (4, 2.5)$.

- (i) Das klassische Newtonverfahren liefert das folgende Ergebnis:

ITER	X(1)	X(2)	F(X)	DX	P=1	P=2
0	0.4000000E+01	0.2500000E+01	0.1351240E+03	0.0000000E+00	0.0000000E+00	0.0000000E+00
1	0.3281417E+01	0.2056664E+01	0.2617493E+02	0.8443389E+00	0.2567589E+00	0.2296521E+00
2	0.3035131E+01	0.1988137E+01	0.2424694E+01	0.2556422E+00	0.1291689E+00	0.4499638E+00
3	0.3000634E+01	0.1999744E+01	0.4203618E-01	0.3639711E-01	0.1844451E-01	0.4974264E+00
4	0.3000000E+01	0.2000000E+01	0.1406811E-04	0.6837008E-03	0.3217815E-03	0.4704954E+00
5	0.3000000E+01	0.2000000E+01	0.1500787E-11	0.2200729E-06	0.1109167E-06	0.5039996E+00
6	0.3000000E+01	0.2000000E+01	0.0000000E+00	0.2440976E-13	0.0000000E+00	0.0000000E+00

Die letzten beiden Spalten testen die Folge $(x^{(i)}, y^{(i)})^\top$, $i = 0, 1, 2, \dots$ auf lineare ($P=1$ strebt gegen Wert $\neq 0$, $P=2$ explodiert), superlineare ($P=1$ strebt gegen 0, $P=2$ explodiert) und quadratische Konvergenz ($P=1$ strebt gegen 0, $P=2$ strebt gegen Wert $\neq 0$).

Die Rechnungen zeigen eine quadratische Konvergenz des klassischen Newtonverfahrens gegen das Minimum $(3, 2)^\top$.

(ii) Das Newtonverfahren mit finiten Differenzen liefert das folgende Ergebnis:

ITER	X(1)	X(2)	F(X)	DX	P=1	P=2
0	0.4000000E+01	0.2500000E+01	0.1351240E+03	0.0000000E+00	0.0000000E+00	0.0000000E+00
1	0.3300753E+01	0.2071139E+01	0.2853753E+02	0.8202854E+00	0.2764246E+00	0.2472417E+00
2	0.3044647E+01	0.1988635E+01	0.3188243E+01	0.2690673E+00	0.1490712E+00	0.4823498E+00
3	0.3001880E+01	0.1998866E+01	0.1165668E+00	0.4397377E-01	0.4765407E-01	0.1034366E+01
4	0.3000033E+01	0.1999965E+01	0.1813021E-02	0.2149084E-02	0.2198280E-01	0.1001284E+02
5	0.3000000E+01	0.1999999E+01	0.2684597E-04	0.4737814E-04	0.1856229E-01	0.3846122E+03
6	0.3000000E+01	0.2000000E+01	0.3878641E-06	0.8817947E-06	0.1574341E-01	0.1757351E+05
7	0.3000000E+01	0.2000000E+01	0.5091715E-08	0.1391334E-07	0.1351821E-01	0.9584736E+06
8	0.3000000E+01	0.2000000E+01	0.5957654E-10	0.1884045E-09	0.1182730E-01	0.6203365E+08
9	0.3000000E+01	0.2000000E+01	0.6269307E-12	0.2231274E-11	0.1056873E-01	0.4686828E+10
10	0.3000000E+01	0.2000000E+01	0.8758808E-14	0.2360669E-13	0.9316950E-02	0.3909375E+12

Die Rechnungen zeigen eine superlineare Konvergenz des Newtonverfahrens mit finiten Differenzen gegen das Minimum $(3, 2)^\top$.

(iii) Das vereinfachte Newtonverfahren liefert das folgende Ergebnis:

ITER	X(1)	X(2)	F(X)	DX	P=1	P=2
0	0.4000000E+01	0.2500000E+01	0.1351240E+03	0.0000000E+00	0.0000000E+00	0.0000000E+00
1	0.3281417E+01	0.2056664E+01	0.2617493E+02	0.8443389E+00	0.2567589E+00	0.2296521E+00
3	0.3071606E+01	0.1980562E+01	0.5151481E+01	0.6534609E-01	0.5470076E+00	0.4032719E+01
5	0.3022629E+01	0.1985541E+01	0.1403370E+01	0.1757236E-01	0.6120523E+00	0.1394972E+02
7	0.3007754E+01	0.1993240E+01	0.4467921E+00	0.6376896E-02	0.6191372E+00	0.3726386E+02
9	0.3002760E+01	0.1997232E+01	0.1540965E+00	0.2446909E-02	0.6154210E+00	0.9688894E+02
11	0.3001001E+01	0.1998926E+01	0.5514621E-01	0.9306565E-03	0.6121533E+00	0.2552127E+03
13	0.3000366E+01	0.1999593E+01	0.2006926E-01	0.3497360E-03	0.6102365E+00	0.6801452E+03
15	0.3000135E+01	0.1999848E+01	0.7361719E-02	0.1304451E-03	0.6092050E+00	0.1825138E+04
17	0.3000050E+01	0.1999943E+01	0.2710854E-02	0.4845351E-04	0.6086619E+00	0.4915933E+04
19	0.3000018E+01	0.1999979E+01	0.1000183E-02	0.1795865E-04	0.6083767E+00	0.1326686E+05
21	0.3000007E+01	0.1999992E+01	0.3693921E-03	0.6648483E-05	0.6082266E+00	0.3584082E+05
23	0.3000003E+01	0.1999997E+01	0.1364969E-03	0.2459848E-05	0.6081473E+00	0.9687759E+05
25	0.3000001E+01	0.1999999E+01	0.5045188E-04	0.9098193E-06	0.6081053E+00	0.2619347E+06
...						
61	0.3000000E+01	0.2000000E+01	0.8472333E-12	0.1511068E-13	0.6121321E+00	0.1587489E+14
63	0.3000000E+01	0.2000000E+01	0.3142520E-12	0.5625244E-14	0.6151352E+00	0.4292624E+14
65	0.3000000E+01	0.2000000E+01	0.1217558E-12	0.2090737E-14	0.6172799E+00	0.1154325E+15
66	0.3000000E+01	0.2000000E+01	0.7240833E-13	0.1287557E-14	0.6201737E+00	0.1878784E+15

Sogar das vereinfachte Newtonverfahren konvergiert, allerdings nur linear.

(iv) Das Quasi-Newton-Verfahren liefert folgende Ausgabe:

ITER	X(1)	X(2)	F(X)	DX	P=1	P=2
0	0.4000000E+01	0.2500000E+01	0.1351240E+03	0.0000000E+00	0.0000000E+00	0.0000000E+00
1	0.3281417E+01	0.2056664E+01	0.2617493E+02	0.8443389E+00	0.2567589E+00	0.2296521E+00
2	0.3102197E+01	0.1979909E+01	0.7659568E+01	0.1949641E+00	0.3628210E+00	0.1263898E+01
3	0.3025921E+01	0.1971163E+01	0.1434747E+01	0.7677560E-01	0.3722890E+00	0.3574434E+01

4	0.3010235E+01	0.1983499E+01	0.5553883E+00	0.1995640E-01	0.5007674E+00	0.1291466E+02
5	0.3002280E+01	0.1996181E+01	0.1249161E+00	0.1496997E-01	0.2290828E+00	0.1179786E+02
6	0.3000046E+01	0.1999958E+01	0.2604890E-02	0.4388274E-02	0.1405047E-01	0.3158708E+01
7	0.2999993E+01	0.2000013E+01	0.3905744E-03	0.7631605E-04	0.2309428E+00	0.3695149E+04
8	0.2999999E+01	0.2000002E+01	0.6782850E-04	0.1193996E-04	0.1727739E+00	0.1197019E+05
9	0.3000000E+01	0.2000000E+01	0.3561841E-06	0.2506850E-05	0.5248580E-02	0.2104684E+04
10	0.3000000E+01	0.2000000E+01	0.1481444E-09	0.1309415E-07	0.4156278E-03	0.3175468E+05
11	0.3000000E+01	0.2000000E+01	0.8364453E-13	0.5443132E-11	0.5476148E-03	0.1006639E+09

Die Rechnungen zeigen eine superlineare Konvergenz des Quasi-Newton-Verfahrens gegen das Minimum $(3, 2)^\top$.

4.3.4 Existenz einer Nullstelle

In der bisherigen Konvergenzanalyse hatten wir die Existenz einer Nullstelle vorausgesetzt. Der folgende Satz beweist die Existenz einer Nullstelle gleich mit.

Satz 4.3.18 (Newton-Kantorovich)

Sei $D \subseteq \mathbb{R}^n$ offen und konvex und $F : D \rightarrow \mathbb{R}^n$ stetig differenzierbar in D und stetig auf \bar{D} . Die Jacobimatrix F' erfülle folgende Annahmen:

(a) Es existiere $L > 0$ mit

$$\|F'(x) - F'(y)\| \leq L \cdot \|x - y\| \quad \forall x, y \in D$$

(b) Die Inverse $F'(x)^{-1}$ existiert für alle $x \in D$ und es gibt ein $C > 0$ mit $\|F'(x)^{-1}\| \leq C$ für alle $x \in D$.

(c) Für ein $x^{(0)} \in D$ gilt $0 < \|F'(x^{(0)})^{-1}F(x^{(0)})\| \leq \alpha$ und $h = \frac{\alpha \cdot L \cdot C}{2} < 1$.

(d) Die offene Kugel $K = \{x \in \mathbb{R}^n \mid \|x - x^{(0)}\| < r\}$ mit $r = \frac{\alpha}{1-h}$ liegt in D .

Dann gelten folgende Aussagen:

(i) Die Iterationsfolge des Newtonverfahrens ist wohldefiniert und liegt in K .

(ii) Der Grenzwert $\lim_{i \rightarrow \infty} x^{(i)} = \hat{x}$ existiert, und es gilt $\hat{x} \in K$, $F(\hat{x}) = 0$.

(iii) Es gilt die Fehlerabschätzung

$$\|x^{(i)} - \hat{x}\| \leq \alpha \frac{h^{2^i - 1}}{1 - h^{2^i}}.$$

Beweis: (vgl. Lempio [Lem98, S.132])

(i) Aus (c) folgt

$$\|x^{(1)} - x^{(0)}\| = \|F'(x^{(0)})^{-1}F(x^{(0)})\| \leq \alpha < \frac{\alpha}{1-h} = r,$$

d.h. $\|x^{(1)} - x^{(0)}\| \leq \alpha h^{2^0-1}$ und $x^{(1)} \in K$. Die Behauptung folgt nun induktiv: Sei bewiesen, daß für ein k ,

$$\|x^{(i)} - x^{(i-1)}\| \leq \alpha h^{2^{i-1}-1}, \quad x^{(i)} \in K, \quad i = 1, 2, \dots, k.$$

Dann folgt mit (b)

$$\begin{aligned} \|x^{(k+1)} - x^{(k)}\| &= \|F'(x^{(k)})^{-1}F(x^{(k)})\| \\ &\leq C \cdot \|F(x^{(k)})\| \\ &= C \cdot \|F(x^{(k)}) - F(x^{(k-1)}) - F'(x^{(k-1)})(x^{(k)} - x^{(k-1)})\|. \end{aligned}$$

Aus (a) und dem Mittelwertsatz in Integralform folgt

$$\begin{aligned} &\|F(x^{(k)}) - F(x^{(k-1)}) - F'(x^{(k-1)})(x^{(k)} - x^{(k-1)})\| \\ &= \left\| \int_0^1 (F'(x^{(k-1)} + t(x^{(k)} - x^{(k-1)})) - F'(x^{(k-1)})) (x^{(k)} - x^{(k-1)}) dt \right\| \\ &\leq L \int_0^1 t \|x^{(k)} - x^{(k-1)}\|^2 dt \\ &= \frac{L}{2} \|x^{(k)} - x^{(k-1)}\|^2. \end{aligned}$$

Insgesamt gilt

$$\|x^{(k+1)} - x^{(k)}\| \leq \frac{C \cdot L}{2} \|x^{(k)} - x^{(k-1)}\|^2.$$

Nach Induktionsannahme folgt weiter

$$\|x^{(k+1)} - x^{(k)}\| \leq \frac{C \cdot L}{2} \alpha^2 \left(h^{2^{k-1}-1}\right)^2 = \frac{C \cdot L}{2} \alpha^2 h^{2^k-2} = \alpha h^{2^k-1}.$$

Damit folgt durch Abschätzung mit der geometrischen Summe

$$\begin{aligned} \|x^{(k+1)} - x^{(0)}\| &\leq \|x^{(k+1)} - x^{(k)}\| + \|x^{(k)} - x^{(k-1)}\| + \dots + \|x^{(1)} - x^{(0)}\| \\ &\leq \alpha \left(h^{2^k-1} + h^{2^{k-1}-1} + \dots + h^7 + h^3 + h^1 + 1\right) \\ &< \frac{\alpha}{1-h} = r. \end{aligned}$$

Somit ist $x^{(k+1)} \in K$ und $\|x^{(k+1)} - x^{(k)}\| \leq \alpha h^{2^k-1}$.

(ii) Mit der letzten Abschätzung in (i) folgt für $j \in \mathbb{N}$

$$\begin{aligned} &\|x^{(k+j)} - x^{(k)}\| \\ &\leq \|x^{(k+j)} - x^{(k+j-1)}\| + \|x^{(k+j-1)} - x^{(k+j-2)}\| + \dots + \|x^{(k+1)} - x^{(k)}\| \\ &\leq \alpha h^{2^k-1} \left(\left(h^{2^k}\right)^{2^{j-1}-1} + \left(h^{2^k}\right)^{2^{j-2}-1} + \dots + \left(h^{2^k}\right)^1 + \left(h^{2^k}\right)^0 \right) \\ &< \frac{\alpha h^{2^k-1}}{1-h^{2^k}}. \end{aligned} \tag{4.12}$$

Wegen $h < 1$ ist $\{x^{(k)}\}$ Cauchy-Folge in K und der Grenzwert $\lim_{i \rightarrow \infty} x^{(i)} = \hat{x}$ existiert.

Wegen $F(x^{(k)}) = -F'(x^{(k)})(x^{(k+1)} - x^{(k)})$ und

$$\begin{aligned} \|F'(x^{(k)})\| &\leq \|F'(x^{(k)}) - F'(x^{(0)})\| + \|F'(x^{(0)})\| \\ &\leq L \cdot \|x^{(k)} - x^{(0)}\| + \|F'(x^{(0)})\| \\ &\leq L \cdot r + \|F'(x^{(0)})\| \end{aligned}$$

ist

$$\|F(x^{(k)})\| \leq (L \cdot r + \|F'(x^{(0)})\|) \|x^{(k+1)} - x^{(k)}\| \quad k = 0, 1, \dots$$

Aus der Stetigkeit von F und $\|\cdot\|$ auf \bar{D} folgt dann

$$0 = \lim_{k \rightarrow \infty} \|F(x^{(k)})\| = \|F(\lim_{k \rightarrow \infty} x^{(k)})\| = \|F(\hat{x})\|.$$

(iii) Folgt durch Grenzübergang $j \rightarrow \infty$ in (4.12).

□

4.4 Sekantenverfahren

Wiederum im eindimensionalen Fall $n = 1$ verwendet man häufig auch das Sekantenverfahren. Sind zwei Näherungen $x^{(i-1)} \neq x^{(i)}$ gegeben, approximiert man die Funktion F lokal durch die Sekante $S(x)$, die durch die Punkte $(x^{(i-1)}, F(x^{(i-1)}))$ und $(x^{(i)}, F(x^{(i)}))$ verläuft, vgl. Abbildung 4.4. Die Sekante ist durch

$$S(x) = F(x^{(i-1)}) + \frac{x - x^{(i-1)}}{x^{(i)} - x^{(i-1)}} (F(x^{(i)}) - F(x^{(i-1)}))$$

gegeben. Die nächste Näherung $x^{(i+1)}$ ist durch die Nullstelle der Sekante gegeben: $S(x^{(i+1)}) = 0$. Diese läßt sich leicht ausrechnen:

$$\begin{aligned} x^{(i+1)} &= x^{(i)} - \left(\frac{x^{(i)} - x^{(i-1)}}{F(x^{(i)}) - F(x^{(i-1)})} \right) F(x^{(i)}) \\ &= \frac{x^{(i-1)}F(x^{(i)}) - x^{(i)}F(x^{(i-1)})}{F(x^{(i)}) - F(x^{(i-1)})}, \quad i = 1, 2, 3, \dots \end{aligned}$$

Zum Start des Verfahrens werden **zwei** Anfangswerte $x^{(0)}$ und $x^{(1)}$ mit $F(x^{(0)}) \neq F(x^{(1)})$ benötigt. Ein Problem entsteht, wenn während der Iteration der Fall $F(x^{(i)}) \approx F(x^{(i-1)})$ auftritt. Deshalb sollte das Verfahren abgebrochen werden, wenn

$$|F(x^{(i)}) - F(x^{(i-1)})| \leq \varepsilon |F(x^{(i)})|$$

mit einer gegebenen Toleranz ε gilt.

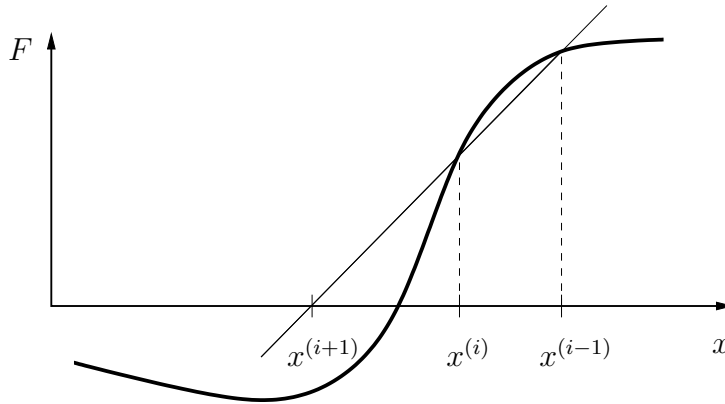


Abbildung 4.4: Sekantenverfahren: Ersetzung der Tangente durch Sekante.

Es gilt

Satz 4.4.1

Sei $\hat{x} \in (a, b)$ eine Nullstelle von F . Sei F zweimal stetig differenzierbar auf $[a, b]$ und $F'(\hat{x}) \neq 0$.

Dann gilt: Das Sekantenverfahren konvergiert **lokal** gegen \hat{x} . Die Konvergenzordnung ist $p = (1 + \sqrt{5})/2 \approx 1.618$.

Beweis: siehe Lempio [Lem98, S.140]

□

Gemäß Definition 4.0.4 ist das Sekantenverfahren also lokal superlinear konvergent.

Vorteile:

- Es werden nur Funktionsauswertungen benötigt (sehr einfach zu implementieren).
- Die Konvergenzordnung ist $p = (1 + \sqrt{5})/2 \approx 1.618 > 1$. Die Konvergenzordnung ist formal zwar kleiner als die des Newtonverfahrens, berücksichtigt man jedoch den Aufwand zur Durchführung eines Iterationsschrittes, zeigt sich, daß das Sekantenverfahren nur eine Funktionsauswertung benötigt, während das Newtonverfahren eine Funktionsauswertung und eine Ableitung benötigt, die i.a. teurer als eine Funktionsauswertung ist. Um den gleichen Aufwand zu erhalten, können zwei Schritte des Sekantenverfahrens bei nur einem Newtonschritt durchgeführt werden. Damit erhielte man die Konvergenzordnung $p^2 \approx 2.618$.

Nachteile:

- Nur für $n = 1$ anwendbar.
- Das Verfahren konvergiert nur lokal und es ist instabil, falls $F(x^{(i)}) \approx F(x^{(i-1)})$ gilt.

Beispiel 4.4.2

Wir betrachten erneut das Problem, $\sqrt{2}$ als (positive) Nullstelle von $F(x) = x^2 - 2$ zu berechnen. Ausgabe:

Iteration 1 :	1.3333333333, Fehler:	0.0808802290
Iteration 2 :	1.4000000000, Fehler:	0.0142135624
Iteration 3 :	1.4146341463, Fehler:	0.0004205840
Iteration 4 :	1.4142114385, Fehler:	0.0000021239
Iteration 5 :	1.4142135621, Fehler:	0.0000000003
Iteration 6 :	1.4142135624, Fehler:	0.0000000000

Kapitel 5

Interpolation

Interpolationsaufgaben beschäftigen sich mit der Konstruktion einer Funktion, die an endlich vielen Stellen mit einer i.a. unbekannten oder schwer zu berechnenden Funktion übereinstimmt.

Eine typische Aufgabenstellung ist die Messung eines physikalischen oder technischen Vorgangs, welche $n + 1 \in \mathbb{N}$ verschiedene reellwertige Stützwerte

$$\begin{array}{c|c|c|c|c} x & x_0 & x_1 & \cdots & x_n \\ \hline f(x) & f_0 & f_1 & \cdots & f_n \end{array} \quad (5.1)$$

ergebe. Die Funktion f , die dem Vorgang zu Grunde liegt, ist häufig nicht explizit bekannt und liegt somit nur an den Stützstellen x_0, \dots, x_n mit Funktionswerten $f_i = f(x_i), i = 0, \dots, n$, vor. Um trotzdem mit Funktionswerten **zwischen** den Stützstellen arbeiten zu können und nicht jedesmal ein mitunter aufwändiges Experiment durchführen zu müssen, wird eine Approximation an die unbekannte Funktion f konstruiert, die zumindest die folgende Interpolationsaufgabe löst und ggf. weitere sinnvolle Eigenschaften besitzt (z.B. hinreichend hohe Glattheit).

Problem 5.0.1 (Interpolationsaufgabe)

Bestimme eine geeignete Funktion $g : \mathbb{R} \rightarrow \mathbb{R}$, die die Interpolationsbedingungen

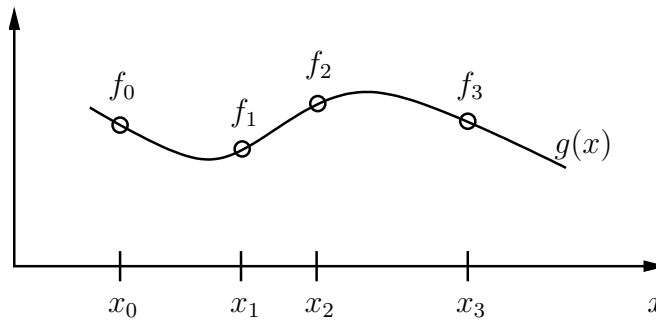
$$g(x_i) = f_i, \quad i = 0, \dots, n, \quad (5.2)$$

erfüllt, vgl. auch Abbildung 5.1.

Eine Funktion g , die die Interpolationsbedingungen (5.2) erfüllt, heißt **interpolierende Funktion**. Man sagt auch, g **interpoliert die Stützwerte (5.1)**.

Interpolationsaufgaben treten auch in der Computergrafik (zeichnen einer geeigneten Kurve durch vorgegebene Punkte), in technischen Geräten (z.B. Scanner oder Digitalkameras: echte Auflösung 600×1200 dpi, interpolierte Auflösung 9600×9600 dpi), in der Bildverarbeitung (Kompressionsalgorithmen) und in der Signalverarbeitung (Approximation von Signalen) auf.

Interpolierende Funktionen können auch zur Approximation von Funktionen dienen, deren Auswertung sehr kostenintensiv ist. In diesem Fall stellt sich die Frage, wie gut die Appro-

Abbildung 5.1: Interpolation: Stützwerte und interpolierende Funktion g .

ximation durch die interpolierende Funktion ist, d.h. wie groß der Approximationsfehler ist.

Beispiel 5.0.2

Die Funktion $f(x) = \exp(x)$ soll mittels Interpolation von Funktionswerten approximiert werden. Insbesondere soll f an der Stelle $x = 0.4$ approximiert werden.

- (a) Wir verwenden die Funktionswerte $1 = f(0) = \exp(0)$ und $e = f(1) = \exp(1)$ und berechnen aus den Interpolationsbedingungen

$$\begin{aligned} 1 &= p(0) = a_0, \\ \exp(1) &= p(1) = a_0 + a_1, \end{aligned}$$

das interpolierende Polynom ersten Grades $p(x) = a_0 + a_1x = 1 + (\exp(1) - 1)x$. Damit ergibt sich an der Stelle $x = 0.4$ der Approximationsfehler

$$|p(0.4) - f(0.4)| \approx |1.6873127 - 1.4918247| = 0.1954880.$$

Dieser Fehler ist verhältnismäßig groß.

- (b) Wir verwenden nun die Funktionswerte $1 = f(0) = \exp(0)$, $1.6487213 \approx f(0.5) = \exp(0.5)$ und $e = f(1) = \exp(1)$ und berechnen aus den Interpolationsbedingungen

$$\begin{aligned} 1 &= p(0) = a_0, \\ \exp(0.5) &= p(0.5) = a_0 + \frac{a_1}{2} + \frac{a_2}{4}, \\ \exp(1) &= p(1) = a_0 + a_1 + a_2, \end{aligned}$$

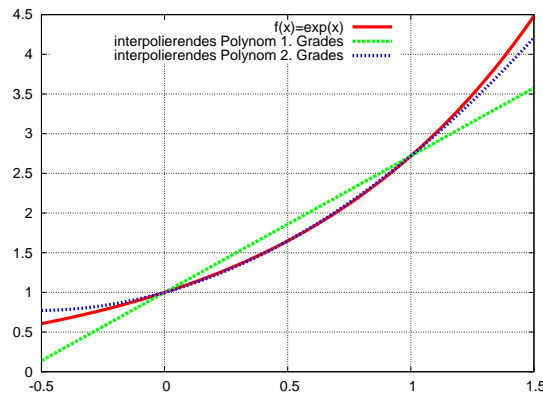
das interpolierende Polynom $p(x) = a_0 + a_1x + a_2x^2$ zweiten Grades mit

$$a_0 = 1, \quad a_1 = 4 \exp(0.5) - \exp(1) - 3, \quad a_2 = 2 \exp(1) + 2 - 4 \exp(0.5).$$

Damit ergibt sich an der Stelle $x = 0.4$ der Approximationsfehler

$$|p(0.4) - f(0.4)| \approx |1.4853099 - 1.4918247| = 0.0065148.$$

Dieser Fehler ist deutlich kleiner als in (a), vgl. nachstehende Abbildung.



5.1 Polynominterpolation

Bei der **Polynominterpolation** werden Polynome als interpolierende Funktion g gewählt. Ein Polynom p vom Höchstgrad r kann allgemein als

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_rx^r = \sum_{i=0}^r a_ix^i \quad (5.3)$$

mit den $r + 1$ Koeffizienten a_0, a_1, \dots, a_r dargestellt werden. r bezeichnet den Grad des Polynoms. Unser Ziel ist es, ein **interpolierendes Polynom** p mit

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

zu finden. Folgende Fragestellungen sind dabei relevant:

- Existenz und Eindeutigkeit eines interpolierenden Polynoms?
- Wie kann das (bzw. ein) Interpolationspolynom berechnet werden?
- Wie groß ist der Interpolationsfehler zwischen den Stützstellen?

Wir weisen die **Existenz eines interpolierenden Polynoms** konstruktiv nach, indem wir explizit ein interpolierendes Polynom konstruieren. Dabei setzen wir voraus, daß die Knoten x_i , $i = 0, \dots, n$, in (5.1) paarweise verschieden sind.

Definition 5.1.1 (Lagrange Polynom)

Für paarweise verschiedene Knoten x_i , $i = 0, \dots, n$, ist für $k = 0, \dots, n$ das k -te **Lagrange'sche Polynom** definiert durch

$$L_k(x) := \prod_{i=0, i \neq k}^n \frac{x - x_i}{x_k - x_i}.$$

Für paarweise verschiedene Knoten x_i , $i = 0, \dots, n$, sind die Lagrange'schen Polynome wohldefiniert, da der Nenner niemals Null wird. Man sieht ebenfalls sofort, daß L_k , $k = 0, \dots, n$, Polynome vom Grad n sind, da jedes L_k sich aus n Produkten von Polynomen ersten Grades zusammensetzt. Desweiteren gilt für $k = 0, \dots, n$,

$$L_k(x_j) = \begin{cases} 1 & \text{falls } j = k, \\ 0 & \text{falls } j \neq k. \end{cases}$$

Mit Hilfe dieser schönen Eigenschaft kann nun ein Interpolationspolynom angegeben werden:

$$p(x) = f_0 L_0(x) + f_1 L_1(x) + \dots + f_n L_n(x) = \sum_{j=0}^n f_j L_j(x). \quad (5.4)$$

(Machen Sie sich klar, daß $p(x_k) = f_k$ für $k = 0, 1, \dots, n$ gilt!) Mithin haben wir konstruktiv die Existenz einer Lösung nachgewiesen. Weiter gilt

Satz 5.1.2

Sind die Stützstellen x_i , $i = 0, \dots, n$, paarweise verschieden, so gibt es genau ein interpolierendes Polynom vom Höchstgrad n .

Beweis: Die Existenz eines interpolierenden Polynoms vom Höchstgrad n wurde bereits durch das Polynom p in (5.4) konstruktiv nachgewiesen. Es verbleibt, die Eindeutigkeit nachzuweisen. Es seien also p_1 und p_2 interpolierende Polynome vom Höchstgrad n . Dann ist das Polynom $p = p_1 - p_2$ ebenfalls ein Polynom mit Höchstgrad n und es hat mindestens $n + 1$ verschiedenen Nullstellen

$$p(x_i) = p_1(x_i) - p_2(x_i) = f_i - f_i = 0, \quad i = 0, \dots, n.$$

Da ein Polynom mit Höchstgrad n , welches verschieden vom Nullpolynom ist, maximal n reelle Nullstellen haben kann (Fundamentalsatz der Algebra), muß $p = p_1 - p_2$ das Nullpolynom sein. Also gilt $p_1 = p_2$ und die Eindeutigkeit ist gezeigt. \square

Zumindest theoretisch kann das interpolierende Polynom vom Höchstgrad n gemäß (5.4) berechnet werden. Es zeigt sich jedoch, daß die Berechnung gemäß (5.4) für gegebenes x sehr rundungsfehleranfällig und somit numerisch instabil ist. Deshalb sollte man den Wert eines Interpolationspolynoms an der Stelle x in der Praxis nicht über (5.4) berechnen, sondern auf die Newton'schen dividierten Differenzen des nächsten Abschnitts zurück greifen. Ein weiterer Nachteil von (5.4) ist, daß das Interpolationspolynom komplett neu berechnet werden muß, wenn nachträglich weitere Stützwerte hinzugefügt werden.

Bemerkung 5.1.3 (Interpolation und Vandermonde-Matrix)

Wertet man die Interpolationsbedingungen (5.2) mit dem Polynom (5.3) aus, folgt das lineare Gleichungssystem

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^r \\ 1 & x_1 & x_1^2 & \cdots & x_1^r \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^r \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_r \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix} \quad (5.5)$$

für die Koeffizienten a_0, \dots, a_r . Die Matrix nennt man auch **Vandermonde-Matrix**. Die Frage der Lösbarkeit der Interpolationsaufgabe ist also gleichbedeutend mit der Frage nach der Lösbarkeit des linearen Gleichungssystems. Falls der Grad r des Polynoms kleiner als n ist, besitzt das Gleichungssystem i.a. keine Lösung (\rightarrow Gegenbeispiel?). Für $r = n$ und verschiedene Stützstellen ist die Vandermonde-Matrix regulär, da wir die Existenz und Eindeutigkeit eines interpolierenden Polynoms bereits bewiesen haben. Es empfiehlt sich nicht, das lineare Gleichungssystem zu lösen, da die Vandermonde-Matrix extrem schlecht konditioniert ist, wie wir bereits in Beispiel 2.5.1 gesehen haben. Für $r > n$ ist die Interpolationsaufgabe ebenfalls stets lösbar, allerdings nicht eindeutig.

5.1.1 Newton'sche dividierte Differenzen

Wir leiten ein effizientes Verfahren zur Bestimmung des interpolierenden Polynoms her und stellen das Interpolationspolynom p hierfür in der Form

$$p(x) = \alpha_0 + \alpha_1(x - x_0) + \alpha_2(x - x_0)(x - x_1) + \dots + \alpha_n(x - x_0) \cdots (x - x_{n-1}) \quad (5.6)$$

dar. Beachte, daß die α_i , $i = 0, \dots, n$, i.a. verschieden von den a_i , $i = 0, \dots, n$, in der Darstellung (5.3) sind. Auswertung der Interpolationsbedingungen (5.2) für p führt auf das lineare Gleichungssystem

$$\begin{aligned} f_0 &= \alpha_0 \\ f_1 &= \alpha_0 + \alpha_1(x_1 - x_0) \\ f_2 &= \alpha_0 + \alpha_1(x_2 - x_0) + \alpha_2(x_2 - x_0)(x_2 - x_1) \\ &\vdots \\ f_n &= \alpha_0 + \alpha_1(x_n - x_0) + \dots + \alpha_n(x_n - x_0) \cdots (x_n - x_{n-1}). \end{aligned}$$

Dieses Gleichungssystem hat Dreiecksgestalt und besitzt, falls die Stützstellen verschieden sind, eine eindeutige Lösung für $\alpha_0, \dots, \alpha_n$. Die Lösung kann mit Hilfe der **Newton'schen dividierten Differenzen** $f[x_i, \dots, x_{i+k}]$ angegeben werden, welche wir im folgenden herleiten möchten.

Es bezeichne $p_{x_i x_{i+1} \dots x_{i+k}}$ das Polynom vom Höchstgrad k , welches die Stützwerte (x_j, f_j) , $j = i, \dots, i+k$, interpoliert. Nach Satz 5.1.2 ist dieses eindeutig bestimmt. Der folgende Satz gibt eine Rekursionsformel zur Berechnung an.

Satz 5.1.4 (Rekursionsformel von Neville)

Für die Interpolationspolynome $p_{x_i x_{i+1} \dots x_{i+k}}$ gilt für $k \geq 1$ die Rekursionsformel

$$p_{x_i x_{i+1} \dots x_{i+k}}(x) = \frac{(x - x_i)p_{x_{i+1} \dots x_{i+k}}(x) - (x - x_{i+k})p_{x_i x_{i+1} \dots x_{i+k-1}}(x)}{x_{i+k} - x_i}.$$

Beweis: Der Beweis wird induktiv nach dem Grad k der interpolierenden Polynome geführt. Für $k = 0$ ist $p_{x_i}(x) = f_i$ das interpolierende Polynom.

Sei die Behauptung richtig für Polynome vom Grad $k-1$ ($k \geq 1$). Definiere das Polynom

$$q(x) := \frac{(x - x_i)p_{x_{i+1} \dots x_{i+k}}(x) - (x - x_{i+k})p_{x_i x_{i+1} \dots x_{i+k-1}}(x)}{x_{i+k} - x_i},$$

welches offenbar den Grad k hat. Nach Induktionsvoraussetzung gilt:

$$\begin{aligned} q(x_i) &= p_{x_i x_{i+1} \dots x_{i+k-1}}(x_i) = f_i, \\ q(x_j) &= \frac{(x_j - x_i)p_{x_{i+1} \dots x_{i+k}}(x_j) - (x_j - x_{i+k})p_{x_i x_{i+1} \dots x_{i+k-1}}(x_j)}{x_{i+k} - x_i} \\ &= \frac{(x_j - x_i)f_j - (x_j - x_{i+k})f_j}{x_{i+k} - x_i} = f_j, \quad j = i+1, \dots, i+k-1, \\ q(x_{i+k}) &= p_{x_{i+1} \dots x_{i+k}}(x_{i+k}) = f_{i+k}. \end{aligned}$$

Also interpoliert q genauso wie $p_{x_i x_{i+1} \dots x_{i+k}}$ die Stützstellen (x_j, f_j) , $j = i, \dots, i+k$. Da das Interpolationspolynom nach Satz 5.1.2 eindeutig ist, gilt $q \equiv p_{x_i x_{i+1} \dots x_{i+k}}$. \square

Definition 5.1.5 (Newton'sche dividierte Differenz)

Der Koeffizient vor x^k in $p_{x_i x_{i+1} \dots x_{i+k}}$ heißt Newton'sche dividierte Differenz und wird mit $f[x_i, \dots, x_{i+k}]$ bezeichnet.

Mit Hilfe von Satz 5.1.4 erhalten wir eine Rekursionsformel für die Newton'schen dividierten Differenzen.

Satz 5.1.6 (Rekursionsformel für Newton'sche dividierte Differenzen)

Die Newtonschen dividierten Differenzen sind rekursiv durch

$$\begin{aligned} f[x_i] &:= f_i, \\ f[x_i, x_{i+1}, \dots, x_{i+k}] &:= \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i} \end{aligned}$$

gegeben und können nach folgendem Schema spaltenweise berechnet werden:

$$\begin{array}{ccccccc}
 f_0 & = & f[x_0] & & & & \\
 f_1 & = & f[x_1] & f[x_0, x_1] & & & \\
 f_2 & = & f[x_2] & f[x_1, x_2] & f[x_0, x_1, x_2] & & \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\
 f_n & = & f[x_n] & f[x_{n-1}, x_n] & f[x_{n-2}, x_{n-1}, x_n] & \cdots & f[x_0, \dots, x_n]
 \end{array}$$

Der Aufwand des Schemas beträgt $\frac{1}{2}n(n+1)$ Divisionen und $n(n+1)$ Subtraktionen.

Beweis: Folgt direkt aus Satz 5.1.4 durch Koeffizientenvergleich. \square

Ein wesentlicher Vorteil der dividierten Differenzen ist, daß zusätzliche Knoten leicht hinzugefügt werden können. Hierfür muß das Schema lediglich um eine weitere Zeile ergänzt werden.

Der folgende Satz klärt den Zusammenhang zwischen dem Interpolationspolynom in (5.6) und den Newton'schen dividierten Differenzen.

Satz 5.1.7

Das Interpolationspolynom p zu den Stützstellen (x_i, f_i) , $i = 0, \dots, n$, ist durch (5.6) mit

$$\alpha_i = f[x_0, \dots, x_i], \quad i = 0, 1, \dots, n$$

gegeben.

Beweis: Es gilt die Darstellung

$$\begin{aligned}
 p &= p_{x_0 \dots x_n} \\
 &= p_{x_0} + (p_{x_0 x_1} - p_{x_0}) + (p_{x_0 x_1 x_2} - p_{x_0 x_1}) + \dots + (p_{x_0 x_1 \dots x_n} - p_{x_0 x_1 \dots x_{n-1}}).
 \end{aligned}$$

Das Polynom $p_{x_0 x_1 \dots x_{k+1}} - p_{x_0 x_1 \dots x_k}$ hat den Höchstgrad $k+1$, die $k+1$ Nullstellen x_0, \dots, x_k , sowie per Definition den Koeffizienten $f[x_0, \dots, x_{k+1}]$ für x^{k+1} . Da $p_{x_0 x_1 \dots x_k}$ ein Polynom vom Grad k ist, gilt

$$p_{x_0 x_1 \dots x_{k+1}}(x) - p_{x_0 x_1 \dots x_k}(x) = f[x_0, \dots, x_{k+1}](x - x_0)(x - x_1) \cdots (x - x_k),$$

woraus die Behauptung folgt. \square

Das Polynom p in (5.6) kann mit Hilfe des **Horner-Algorithmus** effizient an einer Stelle x ausgewertet werden. Dazu wird p in geschachtelter Form

$$p(x) = \alpha_0 + (x - x_0) \{ \alpha_1 + (x - x_1) [\alpha_2 + (\dots + (\alpha_{n-1} + (x - x_{n-1}) \alpha_n) \dots)] \} \quad (5.7)$$

geschrieben. Diese geschachtelte Form kann für ein gegebenes x effizient durch den nachstehenden Algorithmus berechnet werden:

Algorithmus 5.1.8 (Horner Algorithmus)

Gegeben: $x, \alpha_i, i = 0, \dots, n$.

Resultat: $v = p(x)$.

Setze $v = \alpha_n$.

for $i = n - 1, \dots, 0$:

$v := \alpha_i + v(x - x_i)$

end

Beispiel 5.1.9

Wir betrachten zwei Interpolationsaufgaben

	i	1	2	3	4	5
(i)	x_i	-2	-1	0	1	2
	f_i	0	0	10	0	0

	i	1	2	3	4	5	6	7	8	9	10	11	12	13
(ii)	x_i	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
	f_i	0	0	0	0	0	0	10	0	0	0	0	0	0

Abbildung 5.2 zeigt die zugehörigen Interpolationspolynome. Beispiel (ii) verdeutlicht eine grundsätzliche Problematik bei der Interpolation mit Polynomen: Mit wachsender Anzahl der Stützpunkte wächst auch der Polynomgrad und das Interpolationspolynom weist starke Oszillationen auf, die zum Rand hin immer stärker werden. In der Praxis interpoliert man die gegebenen Stützpunkte daher häufig **abschnittsweise** durch Polynome mit nicht zu hohem Grad und „heftet“ die abschnittsweise definierten Polynome aneinander.

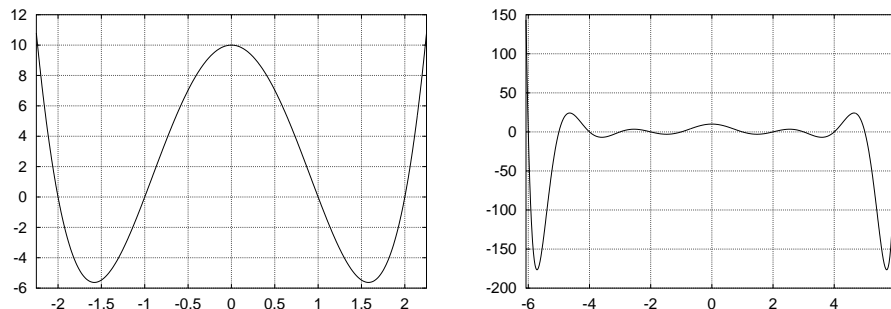


Abbildung 5.2: Interpolationspolynome für die Daten (i) (links) und (ii) (rechts).

Ist man nur am Wert des Interpolationspolynoms an einer festen Stelle x interessiert, kommt der **Algorithmus von Neville** zum Einsatz. Im Gegensatz zu den Newton'schen

dividierten Differenzen werden nicht die Koeffizienten des Interpolationspolynoms berechnet, sondern nur der Funktionswert an der Stelle x . Es wird wiederum vorausgesetzt, daß die Stützstellen $x_i, i = 0, \dots, n$, verschieden sind. Das Neville-Schema verwendet die Darstellung des Interpolationspolynoms aus Satz 5.1.4.

Algorithmus 5.1.10 (Algorithmus von Neville)

(0) Sei die feste Stelle $x \in \mathbb{R}$ gegeben. Setze

$$p_{x_i} := f_i, \quad i = 0, \dots, n.$$

(1) Berechne rekursiv die Werte

$$p_{x_i \dots x_{i+k}} := \frac{(x - x_i)p_{x_{i+1} \dots x_{i+k}} - (x - x_{i+k})p_{x_i \dots x_{i+k-1}}}{x_{i+k} - x_i}.$$

Die Berechnungsvorschrift kann in Tableauform („Neville-Schema“) zusammengefaßt werden

$$\begin{array}{ccccccc} f_0 & = & p_{x_0} & & & & \\ f_1 & = & p_{x_1} & p_{x_0 x_1} & & & \\ f_2 & = & p_{x_2} & p_{x_1 x_2} & p_{x_0 x_1 x_2} & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\ f_n & = & p_{x_n} & p_{x_{n-1} x_n} & p_{x_{n-2} x_{n-1} x_n} & \cdots & p_{x_0 x_1 \dots x_n} \end{array}$$

Der gesuchte Funktionswert des Interpolationspolynoms ist $p_{x_0 x_1 \dots x_n}$.

5.1.2 Fehlerdarstellung interpolierender Polynome

Der folgende Satz gibt Auskunft über die Approximationsgüte des interpolierenden Polynoms vom Höchstgrad n an eine Funktion f zwischen den Stützstellen in (5.1).

Satz 5.1.11

Seien $f : [a, b] \rightarrow \mathbb{R}$ $(n + 1)$ -mal stetig differenzierbar und die Stützstellen $x_i \in [a, b]$, $i = 0, \dots, n$, verschieden. Für das Interpolationspolynom p höchstens n -ten Grades mit $p(x_i) = f(x_i)$, $i = 0, \dots, n$, gilt

$$f(x) - p(x) = \frac{1}{(n + 1)!} f^{(n+1)}(\xi) \cdot \prod_{i=0}^n (x - x_i) \quad (5.8)$$

mit einer von x abhängigen Stelle ξ mit

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

Beweis: Für $x \in \{x_0, \dots, x_n\}$ ist die Behauptung offenbar wahr.

Sei nun $x \notin \{x_0, \dots, x_n\}$ fest gewählt. Definiere

$$g(t) := f(t) - p(t) - \alpha \prod_{i=0}^n (t - x_i),$$

wobei α so bestimmt sei, daß $g(x) = 0$ gilt. Dies ist wegen $x \notin \{x_0, \dots, x_n\}$ möglich. Dann besitzt g auf dem Intervall

$$I := [\min\{x_0, \dots, x_n, x\}, \max\{x_0, \dots, x_n, x\}]$$

die $n + 2$ Nullstellen x_0, x_1, \dots, x_n, x . Desweiteren ist g $(n + 1)$ -mal stetig differenzierbar. Nach dem Satz von Rolle¹ hat g' in I noch mindestens $n + 1$ Nullstellen, g'' mindestens n Nullstellen, usw. Schließlich hat $g^{(n+1)}$ noch mindestens eine Nullstelle $\xi \in I$. Als Polynom vom Höchstgrad n gilt $p^{(n+1)} \equiv 0$, also

$$0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - \alpha(n + 1)!$$

und somit

$$\alpha = \frac{f^{(n+1)}(\xi)}{(n + 1)!}.$$

Damit ergibt sich aus $g(x) = 0$ die Behauptung gemäß

$$f(x) - p(x) = \alpha \prod_{i=0}^n (x - x_i) = \frac{f^{(n+1)}(\xi)}{(n + 1)!} \cdot \prod_{i=0}^n (x - x_i).$$

□

Die Fehlerdarstellung in (5.8) hängt von der Funktion f und dem Polynom

$$\omega(x) = \prod_{i=0}^n (x - x_i)$$

ab, wobei letzteres nur von den Stützstellen x_i , $i = 0, \dots, n$, abhängt. Insbesondere folgt aus ihr die Abschätzung

$$\begin{aligned} |f(x) - p(x)| &\leq \frac{1}{(n + 1)!} \|f^{(n+1)}\|_{\infty} \cdot |\omega(x)| \\ &\leq \frac{1}{(n + 1)!} \|f^{(n+1)}\|_{\infty} \cdot \|\omega\|_{\infty} \end{aligned} \quad (5.9)$$

¹Sei g in $[a, b]$ stetig und in (a, b) differenzierbar und es gelte $g(a) = g(b)$. Dann gibt es $\xi \in (a, b)$ mit $g'(\xi) = 0$.

für alle $x \in I := [\min\{x_0, \dots, x_n\}, \max\{x_0, \dots, x_n\}]$. Hierin bezeichnet $\|\cdot\|_\infty$ die Supremumsnorm, die für eine stetige Funktion $g : [a, b] \rightarrow \mathbb{R}$ definiert ist durch

$$\|g\|_\infty := \max_{a \leq x \leq b} |g(x)|.$$

Falls die Stützstellen $x_i, i = 0, \dots, n$, frei wählbar sind, kann der Approximationsfehler in (5.9) für gegebenes f minimiert werden, indem die Stützstellen $x_i, i = 0, \dots, n$, optimal gewählt werden. Optimal bedeutet in diesem Zusammenhang, daß

$$\|\omega\|_\infty = \max_{\min\{x_0, \dots, x_n\} \leq x \leq \max\{x_0, \dots, x_n\}} |\omega(x)|$$

durch geeignete Wahl von paarweise verschiedenen Stützstellen $x_i, i = 0, \dots, n$, minimal wird. Es gilt also, paarweise verschiedene Stützstellen $x_i, i = 0, \dots, n$, zu bestimmen, die ω bzgl. der Supremumsnorm minimieren.

Zur Lösung dieser Fragestellung beschränken wir uns auf das Intervall $[-1, 1]$. Dies ist keine Einschränkung, da jedes Intervall $[a, b]$ mit $a < b$ durch eine lineare Transformation bijektiv auf $[-1, 1]$ abgebildet werden kann.

Wir werden sehen, daß die Tschbyscheff-Polynome eine entscheidende Rolle spielen werden.

Definition 5.1.12 (Tschebyscheff-Polynome)

Die durch

$$T_n(x) := \cos(n \cdot \arccos(x)), \quad n = 0, 1, 2, \dots$$

definierte Funktion $T_n : [-1, 1] \rightarrow \mathbb{R}$ heißt **n-tes Tschebyscheff-Polynom**.

Die Bezeichnung Tschebyscheff-Polynom bedarf einer genaueren Untersuchung, da durch die Definition nicht sofort ersichtlich ist, daß es sich bei T_n tatsächlich um ein Polynom handelt.

Satz 5.1.13 (Eigenschaften von Tschebyscheff-Polynomen)

Die Tschebyscheff-Polynome besitzen folgende Eigenschaften:

(i) Es gilt die 3-Term-Rekursion

$$T_{n+1}(x) = 2 \cdot x \cdot T_n(x) - T_{n-1}(x), \quad n = 1, 2, \dots,$$

mit $T_0(x) = 1$ und $T_1(x) = x$.

(ii) T_n ist ein Polynom vom Grad n .

(iii) T_{n+1} besitzt die $n + 1$ Nullstellen

$$x_i = \cos\left(\frac{2i+1}{2n+2}\pi\right), \quad i = 0, \dots, n.$$

(iv) T_{n+1} besitzt die Darstellung

$$T_{n+1}(x) = 2^n \cdot \prod_{i=0}^n (x - x_i), \quad n = 0, 1, 2, \dots$$

mit x_i aus (iii).

(v) Es gilt $\|T_{n+1}\|_\infty = 1$ mit $T_{n+1}(\bar{x}_i) = (-1)^i$ für die Extremstellen

$$\bar{x}_i = \cos\left(\frac{i}{n+1}\pi\right), \quad i = 0, \dots, n+1.$$

Beweis:

(i) Für $n = 0$ und $n = 1$ folgt aus der Definition der Tschebyscheff-Polynome $T_0(x) = 1$ und $T_1(x) = x$.

Zum Nachweis der Rekursion verwenden wir die Additionstheoreme

$$\begin{aligned} \cos(z_1 + z_2) &= \cos(z_1)\cos(z_2) - \sin(z_1)\sin(z_2), \\ \sin\left(\frac{z_1 + z_2}{2}\right)\sin\left(\frac{z_1 - z_2}{2}\right) &= \frac{1}{2}(\cos(z_2) - \cos(z_1)). \end{aligned}$$

Mit $z_1 := n \cdot \arccos(x)$ und $z_2 := \arccos(x)$ folgt aus dem ersten Additionstheorem

$$\begin{aligned} T_{n+1}(x) &= x \cdot \cos(n \cdot \arccos(x)) - \sin(n \cdot \arccos(x)) \sin(\arccos(x)) \\ &= x \cdot T_n(x) - \sin(n \cdot \arccos(x)) \sin(\arccos(x)). \end{aligned}$$

Mit $z_1 = (n+1) \arccos(x)$ und $z_2 = (n-1) \arccos(x)$ folgt für den zweiten Term mit dem zweiten Additionstheorem

$$\begin{aligned} \sin(n \cdot \arccos(x)) \sin(\arccos(x)) &= \frac{1}{2}(\cos((n-1) \arccos(x)) - \cos((n+1) \arccos(x))) \\ &= \frac{1}{2}(T_{n-1}(x) - T_{n+1}(x)). \end{aligned}$$

Zusammenfassen liefert $T_{n+1}(x) = x \cdot T_n(x) - \frac{1}{2}(T_{n-1}(x) - T_{n+1}(x))$, woraus die Behauptung folgt.

(ii) folgt aus der 3-Term-Rekursion in (i).

(iii) Einsetzen liefert für $i = 0, \dots, n$:

$$T_{n+1}(x_i) = \cos \left((n+1) \frac{2i+1}{2n+2} \pi \right) = \cos \left(\frac{2i+1}{2} \pi \right) = 0.$$

(iv) Nach (ii) ist T_{n+1} Polynom vom Grad $n+1$, welches nach (iii) die Nullstellen x_i , $i = 0, \dots, n$, besitzt. Daraus folgt die Darstellung

$$T_{n+1}(x) = \alpha \prod_{i=0}^n (x - x_i).$$

Die Konstante α ist dabei die Konstante vor dem Term x^{n+1} in T_{n+1} . Nach (i) ist dieser Koeffizient aber offenbar gerade 2^n .

(v) Auf Grund der Definition von T_{n+1} gilt offenbar $\|T_{n+1}\|_\infty \leq 1$. Einsetzen von \bar{x}_i liefert

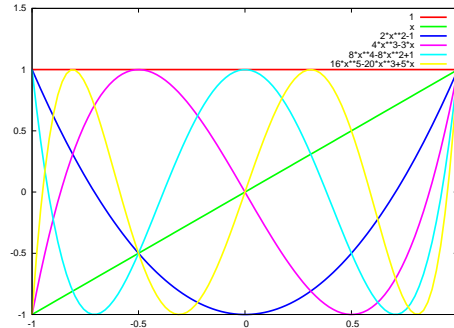
$$T_{n+1}(\bar{x}_i) = \cos \left((n+1) \arccos \left(\cos \left(\frac{i}{n+1} \pi \right) \right) \right) = \cos(i\pi) = (-1)^i$$

für $i = 0, \dots, n+1$, also auch $|T_{n+1}(\bar{x}_i)| = 1$.

□

Auswertung der 3-Term-Rekursion liefert die Tschebyscheff-Polynome bis zum Grad 5:

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



Der folgende Satz stellt den Zusammenhang zwischen den Nullstellen der Tschebyscheff-Polynome und dem Minimalpolynom ω her.

Satz 5.1.14

Seien x_i , $i = 0, \dots, n$, die Nullstellen des Tschebyscheff-Polynoms T_{n+1} aus Satz 5.1.13, (iii), sowie

$$\omega(x) = \prod_{i=0}^n (x - x_i).$$

Dann gilt

$$\|\omega\|_\infty = \max_{x \in [-1, 1]} |\omega(x)| \leq \max_{x \in [-1, 1]} |\tilde{\omega}(x)| = \|\tilde{\omega}\|_\infty$$

für jedes Polynom

$$\tilde{\omega}(x) = \prod_{i=0}^n (x - \tilde{x}_i),$$

welches auf beliebigen, paarweise verschiedenen Stützstellen $\tilde{x}_i \in [-1, 1]$, $i = 0, \dots, n$, basiert.

Beweis: Annahme: Es gebe paarweise verschiedene $\tilde{x}_i \in [-1, 1]$, $i = 0, \dots, n$, mit $\|\tilde{\omega}\|_{\infty} < \|\omega\|_{\infty}$.

Dann ist $p := \tilde{\omega} - \omega$ nicht das Nullpolynom. Desweiteren ist p ein Polynom vom Höchstgrad n , da sich die führenden Terme x^{n+1} von $\tilde{\omega}$ und ω auslöschen.

Gemäß Satz 5.1.13, (iv), stimmt ω bis auf einen konstanten Faktor mit T_{n+1} überein und besitzt die in Satz 5.1.13, (v), genannten Extremstellen \bar{x}_i , $i = 0, \dots, n+1$.

Für zwei aufeinander folgende Extremstellen \bar{x}_i und \bar{x}_{i+1} gilt wegen Satz 5.1.13, (v),

$$\omega(\bar{x}_i) + \omega(\bar{x}_{i+1}) = 0$$

und

$$|\omega(\bar{x}_i)| = |\omega(\bar{x}_{i+1})| = \|\omega\|_{\infty}.$$

Wegen $\|\tilde{\omega}\|_{\infty} < \|\omega\|_{\infty}$ haben $p(\bar{x}_i)$ und $p(\bar{x}_{i+1})$ verschiedene Vorzeichen. Nach dem Zwischenwertsatz besitzt p dann in $(\bar{x}_i, \bar{x}_{i+1})$ mindestens eine Nullstelle. Dies gilt für jedes Intervall $(\bar{x}_i, \bar{x}_{i+1})$, $i = 0, \dots, n$, so daß p mindestens $n+1$ Nullstellen besitzt. Da p aber ein Polynom vom Höchstgrad n ist, muß es das Nullpolynom sein, was im Widerspruch zu $p \neq 0$ steht. \square

Gemäß Satz 5.1.14 approximiert ein Polynom mit Tschbyscheff-Nullstellen als Stützstellen eine gegebene Funktion f besser, als ein Polynom mit anderer Wahl von Stützstellen.

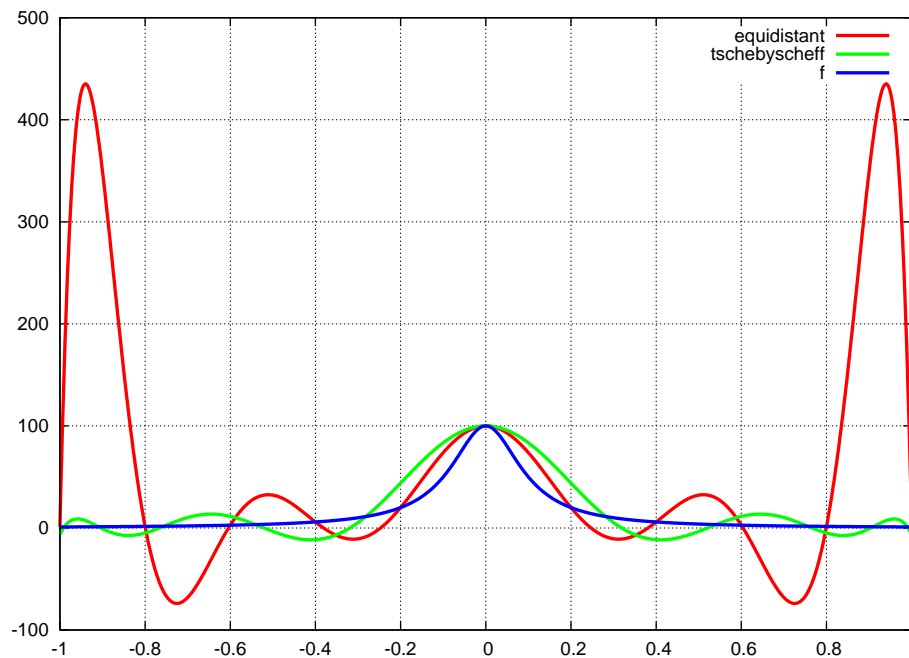
Beispiel 5.1.15

Betrachte

$$f(x) = \frac{1}{x^2 + 0.01}.$$

Die folgende Abbildung zeigt das interpolierende Polynom p_{eq} zu den äquidistanten Stützstellen $x_i = -1 + i \cdot h$, $i = 0, \dots, 10$, mit $h = 2/10$, sowie das interpolierende Polynom zu den Tschebyscheff-Stützstellen

$$x_i = \cos\left(\frac{2i+1}{2n+2}\pi\right), \quad i = 0, \dots, 10.$$



Es ist deutlich zu sehen, daß das auf Tschebyscheff-Stützstellen basierende Interpolationspolynom die Funktion f besser (im Sinne der Supremumsnorm) approximiert als das auf äquidistanten Stützstellen basierende Interpolationspolynom. Insbesondere am Rand oszilliert letzteres stark.

5.2 Splineinterpolation

Im Computer Aided Design (CAD) oder in anderen Interpolationsaufgaben sind vor allem „visuell ansprechende“ interpolierende Funktionen erwünscht. Wir haben in Beispiel 5.1.9 gesehen, daß Polynominterpolation hier nur bedingt geeignet ist, da mit wachsendem Polynomgrad starke Oszillationen auftreten können. Polynome mit niedrigem Polynomgrad hingegen sehen angenehm aus. Dies motiviert die Verwendung von sogenannten **Splines**. Ein Spline besteht dabei aus abschnittsweise definierten Polynomen, die an den Stützstellen unter Beachtung von Glattheitsanforderungen zusammengesetzt werden.

Definition 5.2.1 (Spline vom Grad k)

Eine Funktion s heißt **Spline vom Grad k auf $[a, b]$** , $k \in \mathbb{N}$, falls s folgende Bedingungen erfüllt:

- s ist $(k - 1)$ -mal stetig differenzierbar in $[a, b]$.
- Es gibt eine Partition $a = x_0 < x_1 < \dots < x_n = b$ von $[a, b]$, so daß s in jedem Teilintervall $[x_i, x_{i+1}]$, $0 \leq i \leq n - 1$, ein Polynom vom Höchstgrad k ist.

Für $\mathbb{G} = \{x_0 < x_1 < \dots < x_n\}$ bezeichnet

$$\mathcal{S}_k(\mathbb{G}) := \{s : [x_0, x_n] \rightarrow \mathbb{R} \mid s \text{ ist Spline vom Grad } k \text{ auf } [x_0, x_n]\}$$

den Raum der Splines vom Grad k mit Stützstellen in \mathbb{G} .

Ein **interpolierender Spline vom Grad k** für die Daten (x_i, f_i) , $i = 0, \dots, n$, ist ein Spline s vom Grad k , der die Interpolationsbedingungen $s(x_i) = f_i$, $i = 0, \dots, n$, erfüllt.

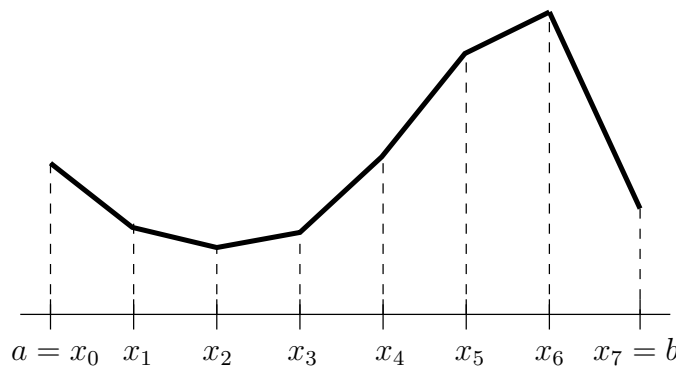
Beispiel 5.2.2 (Spline vom Grad 1)

Eine stetige, stückweise lineare Funktion auf $[a, b]$ ist ein Spline vom Grad 1. Offenbar ist der interpolierende Spline s vom Grad 1 für (x_i, f_i) , $i = 0, \dots, n$, eindeutig bestimmt durch

$$s(x) = \begin{cases} s_0(x), & x_0 \leq x \leq x_1, \\ s_1(x), & x_1 \leq x \leq x_2, \\ \vdots & \\ s_{n-1}(x), & x_{n-1} \leq x \leq x_n \end{cases}$$

mit

$$s_i(x) = s|_{[x_i, x_{i+1}]}(x) = f_i + \frac{x - x_i}{x_{i+1} - x_i}(f_{i+1} - f_i), \quad i = 0, \dots, n-1.$$



Im folgenden werden wir uns speziell mit Splines vom Grad 3, den sogenannten **kubischen Splines**, beschäftigen. Kubische Splines werden insbesondere in der Computergrafik verwendet, um Funktionsverläufe zu erzeugen, die für das menschliche Auge „glatt“ erscheinen. Der Hintergrund ist, daß das menschliche Auge Unstetigkeiten in der Krümmung (also i. W. in der zweiten Ableitung) noch wahrnehmen kann, während Funktionen mit stetiger Krümmung (also stetiger zweiter Ableitung) als glatt aufgefaßt werden.

Zur Motivation betrachten wir folgendes Problem:

Ein Balken soll an den Stellen (5.1) durch Lager so eingespannt werden, daß dort nur Kräfte senkrecht zur Biegelinie wirken. Er wird dann eine Lage annehmen, die durch

eine minimale Biegeenergie $E(s)$ charakterisiert ist, wobei die Funktion s die Biegekurve bezeichnet. Die Biegeenergie ist durch

$$E(s) = c \int_{x_0}^{x_n} \frac{s''(x)^2}{\sqrt{1 + s'(x)^2}} dx$$

mit einer Konstanten c gegeben. Die Funktion

$$\kappa(x) = \frac{s''(x)}{\sqrt{1 + s'(x)^2}}^3$$

beschreibt gerade die **Krümmung der Funktion** s . Für $|s'(x)| \ll 1$ gilt näherungsweise

$$E(s) \approx c \int_{x_0}^{x_n} s''(x)^2 dx. \quad (5.10)$$

Die Aufgabe besteht nun darin, eine Funktion s zu finden, die das Integral in (5.10) und damit auch (näherungsweise) die Biegeenergie unter den Nebenbedingungen $s(x_i) = f_i, i = 0, \dots, n$ minimiert.

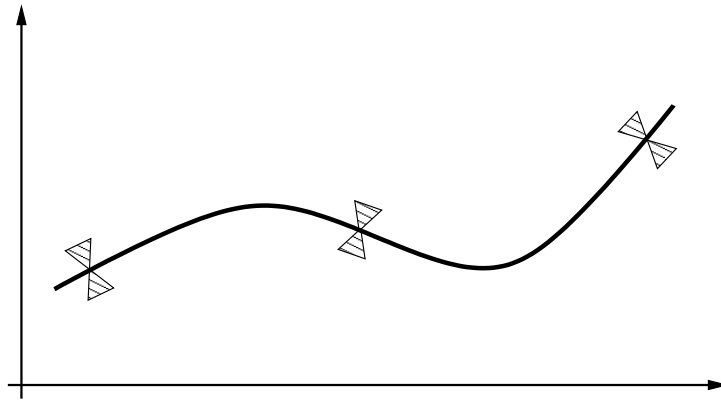


Abbildung 5.3: Spline vom Grad 3 (kubischer Spline): Kurve mit minimaler Krümmung durch gegebene Punkte.

Wir vergleichen nun die Biegeenergie einer beliebigen Funktion

$$g \in \mathcal{F} := \{g : [x_0, x_n] \rightarrow \mathbb{R} \mid g \text{ ist stetig differenzierbare interpolierende Funktion, deren zweite Ableitung in jedem Intervall } (x_i, x_{i+1}) \text{ stetig mit existierenden Grenzwerten } \lim_{x \rightarrow x_i^+} f''(x) \text{ und } \lim_{x \rightarrow x_{i+1}^-} f''(x) \text{ ist}\}$$

mit der Biegeenergie des interpolierenden kubischen Splines. Es gilt

$$\begin{aligned}
 \int_{x_0}^{x_n} g''(x)^2 dx &= \int_{x_0}^{x_n} (g''(x) - s''(x) + s''(x))^2 dx \\
 &= \int_{x_0}^{x_n} s''(x)^2 dx + \int_{x_0}^{x_n} (g''(x) - s''(x))^2 dx \\
 &\quad + 2 \int_{x_0}^{x_n} (g''(x) - s''(x))s''(x) dx.
 \end{aligned} \tag{5.11}$$

Weiter gilt

$$\begin{aligned}
 \int_{x_0}^{x_n} (g''(x) - s''(x))s''(x) dx &= \sum_{i=1}^n \int_{x_{i-1}}^{x_i} (g''(x) - s''(x))s''(x) dx \\
 &\stackrel{p.I.}{=} \sum_{i=1}^n (g'(x) - s'(x))s''(x) \Big|_{x_{i-1}}^{x_i} \\
 &\quad - \sum_{i=1}^n \int_{x_{i-1}}^{x_i} (g'(x) - s'(x))s'''(x) dx.
 \end{aligned}$$

Aus der Stetigkeit von g' , s' und s'' folgt

$$\sum_{i=1}^n (g'(x) - s'(x))s''(x) \Big|_{x_{i-1}}^{x_i} = (g'(x) - s'(x))s''(x) \Big|_{x_0}^{x_n}.$$

Da g und s stetig differenzierbare interpolierende Funktionen sind und s''' konstant ist in jedem Intervall $[x_{i-1}, x_i]$, folgt

$$\int_{x_{i-1}}^{x_i} (g'(x) - s'(x))s'''(x) dx = (g(x) - s(x))s'''(x) \Big|_{x_{i-1}^+}^{x_i^-} = 0.$$

Einsetzen der Erkenntnisse in (5.11) liefert

$$\begin{aligned}
 \int_{x_0}^{x_n} g''(x)^2 dx &= \int_{x_0}^{x_n} s''(x)^2 dx + \int_{x_0}^{x_n} (g''(x) - s''(x))^2 dx \\
 &\quad + 2(g'(x) - s'(x))s''(x) \Big|_{x_0}^{x_n}.
 \end{aligned} \tag{5.12}$$

Wegen

$$\int_{x_0}^{x_n} (g''(x) - s''(x))^2 dx \geq 0$$

folgt aus (5.12) sofort folgender

Satz 5.2.3 (Minimaleigenschaft interpolierender kubischer Splines)

Ein interpolierender kubischer Spline s erfüllt

$$\int_{x_0}^{x_n} s''(x)^2 dx \leq \int_{x_0}^{x_n} g''(x)^2 dx \quad \forall g \in \mathcal{F},$$

*wenn noch **eine** der folgenden Zusatzbedingungen gilt:*

(i) $s''(x_0) = s''(x_n) = 0$ (s heißt **natürlicher Spline**); (geometrisch: die Krümmung in a und b ist Null, daher kann der natürliche Spline außerhalb von $[a, b]$ linear fortgesetzt werden);

(ii) s und g sind periodisch mit Periode $x_n - x_0$ (s heißt **periodischer Spline**);

(iii) $f'_0 = g'(x_0) = s'(x_0)$, $f'_n = g'(x_n) = s'(x_n)$ für gegebenes f'_0 und f'_n (s heißt **Spline mit hermitescher Datenvorgabe**);

s ist eindeutig bestimmt.

Beweis: Unter den Zusatzbedingungen entfällt der letzte Term in (5.12), wobei aus der $x_n - x_0$ -Periodizität in (ii) folgt, daß $(g'(x_0) - s'(x_0))s''(x_0) = (g'(x_n) - s'(x_n))s''(x_n)$ gilt. Zur Eindeutigkeit: Seien s, \tilde{s} interpolierende kubische Splines. Dann gilt

$$\begin{aligned} \int_{x_0}^{x_n} \tilde{s}''(x)^2 dx &= \int_{x_0}^{x_n} s''(x)^2 dx + \int_{x_0}^{x_n} (\tilde{s}''(x) - s''(x))^2 dx, \\ \int_{x_0}^{x_n} s''(x)^2 dx &= \int_{x_0}^{x_n} \tilde{s}''(x)^2 dx + \int_{x_0}^{x_n} (s''(x) - \tilde{s}''(x))^2 dx. \end{aligned}$$

Subtraktion liefert

$$\int_{x_0}^{x_n} (\tilde{s}''(x) - s''(x))^2 dx = 0.$$

Da s und \tilde{s} zweimal stetig differenzierbar sind, folgt $\tilde{s}''(x) - s''(x) = 0$ für alle x . Zweifache Integration liefert $\tilde{s}(x) = s(x) + cx + d$ mit Konstanten c und d . Aus der Interpolationseigenschaft folgt $f_0 = \tilde{s}(x_0) = s(x_0) + cx_0 + d = f_0 + cx_0 + d$ und $f_n = \tilde{s}(x_n) = s(x_n) + cx_n + d = f_n + cx_n + d$. Dies ist nur für $c = d = 0$ erfüllt. \square

Konstruktion interpolierender kubischer Splines

Ziel ist es nun, einen die Stützstellen 5.1 interpolierenden kubischen Spline zu konstruieren. Nach Definition ist ein interpolierender kubischer Spline s für $i = 0, \dots, n-1$ im Intervall $[x_i, x_{i+1}]$ ein Polynom s_i vom Höchstgrad 3, etwa

$$s(x) = \begin{cases} s_0(x), & x_0 \leq x \leq x_1, \\ s_1(x), & x_1 \leq x \leq x_2, \\ \vdots & \\ s_{n-1}(x), & x_{n-1} \leq x \leq x_n \end{cases} \quad (5.13)$$

mit

$$s_i(x) = s|_{[x_i, x_{i+1}]}(x) = \alpha_i + \beta_i(x - x_i) + \frac{1}{2}\gamma_i(x - x_i)^2 + \frac{1}{6}\delta_i(x - x_i)^3. \quad (5.14)$$

Weiterhin müssen die folgenden Bedingungen erfüllt sein:

$$\begin{aligned} s_0(x_0) &= f_0, \\ s_i(x_{i+1}) = s_{i+1}(x_{i+1}) &= f_{i+1}, & i = 0, 1, \dots, n-2, \\ s_{n-1}(x_n) &= f_n, \\ s'_i(x_{i+1}) &= s'_{i+1}(x_{i+1}), & i = 0, 1, \dots, n-2, \\ s''_i(x_{i+1}) &= s''_{i+1}(x_{i+1}), & i = 0, 1, \dots, n-2. \end{aligned}$$

Aus der Darstellung (5.14) folgt

$$\gamma_i = s''_i(x_i) = s''(x_i), \quad i = 0, 1, \dots, n-1.$$

Zusätzlich definieren wir $\gamma_n := s''(x_n) = s''_{n-1}(x_n)$ und

$$h_{i+1} := x_{i+1} - x_i, \quad i = 0, 1, \dots, n-1.$$

Zunächst wird gezeigt, wie s_i (und damit auch s) berechnet werden kann, wenn die sogenannten **Momente** γ_i , $i = 0, \dots, n$, bekannt sind.

(a) Aus $s''_i(x_{i+1}) = s''_{i+1}(x_{i+1}) = \gamma_{i+1}$, $i = 0, \dots, n-1$, folgt

$$s''_i(x_{i+1}) = \gamma_i + \delta_i h_{i+1} = \gamma_{i+1} = s''_{i+1}(x_{i+1}).$$

Auflösen nach δ_i liefert die Darstellung

$$\delta_i = \frac{\gamma_{i+1} - \gamma_i}{h_{i+1}}, \quad i = 0, \dots, n-1. \quad (5.15)$$

(b) Aus $s_i(x_i) = f_i$, $i = 0, \dots, n-1$, folgt sofort

$$\alpha_i = f_i, \quad i = 0, \dots, n-1.$$

Aus $s_i(x_{i+1}) = f_{i+1}$, $i = 0, \dots, n-1$, folgt

$$s_i(x_{i+1}) = f_i + \beta_i h_{i+1} + \frac{\gamma_i}{2} h_{i+1}^2 + \frac{\gamma_{i+1} - \gamma_i}{6h_{i+1}} h_{i+1}^3 = f_{i+1}.$$

Auflösen nach β_i liefert die Darstellung

$$\beta_i = \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{2\gamma_i + \gamma_{i+1}}{6} h_{i+1}, \quad i = 0, \dots, n-1. \quad (5.16)$$

Zusammenfassend sind die Koeffizienten α_i , β_i und δ_i in (5.14) nach (5.15) und (5.16) gegeben durch

$$\begin{aligned} h_{i+1} &= x_{i+1} - x_i, & i = 0, \dots, n-1, \\ \alpha_i &= f_i, & i = 0, \dots, n-1, \\ \beta_i &= \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{2\gamma_i + \gamma_{i+1}}{6} h_{i+1}, & i = 0, \dots, n-1, \\ \delta_i &= \frac{\gamma_{i+1} - \gamma_i}{h_{i+1}}, & i = 0, \dots, n-1, \end{aligned}$$

falls die Momente γ_i , $i = 0, \dots, n$, bekannt sind. Es verbleibt, die Momente γ_i , $i = 0, \dots, n$, zu berechnen. Hierzu werden die Bedingungen $s'_i(x_i) = s'_{i-1}(x_i)$, $i = 1, \dots, n-1$, ausgenutzt. Diese führen auf

$$\beta_i = \beta_{i-1} + \gamma_{i-1}h_i + \frac{1}{2}\delta_{i-1}h_i^2, \quad i = 1, \dots, n-1.$$

Einsetzen von β_i, β_{i-1} und δ_{i-1} aus (5.15) und (5.16) liefert nach etwas Rechnung

$$\frac{h_i}{h_i + h_{i+1}}\gamma_{i-1} + 2\gamma_i + \frac{h_{i+1}}{h_i + h_{i+1}}\gamma_{i+1} = \frac{6}{h_i + h_{i+1}} \left(\frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \right) \quad (5.17)$$

für $i = 1, 2, \dots, n-1$. (5.17) definiert insgesamt $n-1$ lineare Gleichungen für die $n+1$ Momente γ_i , $i = 0, \dots, n$. Wir benötigen also noch zwei weitere Bedingungen, um eine eindeutige Lösung zu ermöglichen. Hier kommt **eine** der Bedingungen (i), (ii) oder (iii) aus Satz 5.2.3 ins Spiel und je nachdem, welche Bedingung gewählt wird, entsteht ein natürlicher Spline, ein periodischer Spline oder ein Spline mit hermitescher Datenvorgabe. In jedem Fall führt die Auswertung von (5.17) zusammen mit einer der Bedingungen (i), (ii) oder (iii) aus Satz 5.2.3 jeweils auf ein lineares Gleichungssystem der Form

$$A \cdot \gamma = d, \quad \gamma := (\gamma_0, \dots, \gamma_n)^\top, \quad d = (d_0, \dots, d_n)^\top \quad (5.18)$$

für die Momente γ_i , $i = 0, \dots, n$. Wir diskutieren die einzelnen Fälle und die daraus entstehenden linearen Gleichungssysteme.

Natürlicher Spline

Es wird $s''(x_0) = s''(x_n) = 0$ gefordert. Per Definition der Momente führt dies auf die beiden Gleichungen $\gamma_0 = 0$ und $\gamma_n = 0$. Zusammen mit (5.17) ist das resultierende Gleichungssystem (5.18) dann gegeben durch

$$A = \begin{pmatrix} 1 & & & & & \\ \mu_1 & 2 & \lambda_1 & & & \\ & \mu_2 & 2 & \lambda_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu_{n-2} & 2 & \lambda_{n-2} \\ & & & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & & & & 1 \end{pmatrix},$$

$$d_i = \begin{cases} 0, & \text{für } i = 0 \text{ und } i = n, \\ \frac{6}{h_i + h_{i+1}} \left[\frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \right], & \text{für } i = 1, \dots, n-1, \end{cases}$$

$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}}, \quad i = 1, \dots, n-1,$$

$$\mu_i = 1 - \lambda_i, \quad i = 1, \dots, n-1.$$

Periodischer Spline

Für einen periodischen Spline muß zunächst $f_0 = f_n$ gelten. Weiter wird gefordert, daß $s'(x_0) = s'(x_n)$ und $s''(x_0) = s''(x_n)$ gelten. Mit den obigen Bezeichnungen führt dies auf die beiden Gleichungen

$$0 = s''(x_0) - s''(x_n) = \gamma_0 - \gamma_n$$

und

$$0 = s'(x_0) - s'(x_n) = \beta_0 - (\beta_{n-1} + \gamma_{n-1}h_n + \frac{1}{2}\delta_{n-1}h_n^2).$$

Einsetzen von β_0, β_{n-1} und δ_{n-1} aus (5.15) und (5.16) liefert nach etwas Rechnung die Gleichung

$$2\frac{h_1}{h_1+h_n}\gamma_0 + \frac{h_1}{h_1+h_n}\gamma_1 + \frac{h_n}{h_1+h_n}\gamma_{n-1} + 2\frac{h_n}{h_1+h_n}\gamma_n = \frac{6}{h_1+h_n} \left(\frac{f_1-f_0}{h_1} - \frac{f_n-f_{n-1}}{h_n} \right).$$

Zusammen mit (5.17) ist das resultierende Gleichungssystem (5.18) dann gegeben durch

$$A = \begin{pmatrix} 1 & & & & & -1 \\ \mu_1 & 2 & \lambda_1 & & & \\ & \mu_2 & 2 & \lambda_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu_{n-2} & 2 & \lambda_{n-2} \\ & & & & \mu_{n-1} & 2 & \lambda_{n-1} \\ 2\lambda_n & \lambda_n & & & & \mu_n & 2\mu_n \end{pmatrix},$$

$$d_i = \begin{cases} 0, & \text{für } i = 0, \\ \frac{6}{h_i+h_{i+1}} \left[\frac{f_{i+1}-f_i}{h_{i+1}} - \frac{f_i-f_{i-1}}{h_i} \right], & \text{für } i = 1, \dots, n-1, \\ \frac{6}{h_1+h_n} \left[\frac{f_1-f_0}{h_1} - \frac{f_n-f_{n-1}}{h_n} \right], & \text{für } i = n, \end{cases}$$

$$\lambda_i = \frac{h_{i+1}}{h_i+h_{i+1}}, \quad i = 1, \dots, n-1, \quad \lambda_n = \frac{h_1}{h_1+h_n},$$

$$\mu_i = 1 - \lambda_i, \quad i = 1, \dots, n.$$

Spline mit hermitescher Datenvorgabe

In x_0 und x_n werden neben den Funktionswerten $s(x_0) = f_0$ und $s(x_n) = f_n$ auch Ableitungen $s'(x_0) = f'_0$ und $s'(x_n) = f'_n$ vorgegeben. Mit den obigen Bezeichnungen führt dies auf die beiden Gleichungen

$$f'_0 = s'(x_0) = \beta_0 = \frac{f_1-f_0}{h_1} - \frac{2\gamma_0+\gamma_1}{6}h_1$$

und

$$f'_n = s'(x_n) = \beta_{n-1} + \gamma_{n-1}h_n + \frac{1}{2}\delta_{n-1}h_n^2.$$

Einsetzen von β_0, β_{n-1} und δ_{n-1} aus (5.15) und (5.16) liefert nach etwas Rechnung die Gleichung

$$\gamma_{n-1} + 2\gamma_n = \frac{6}{h_n} \left(f'_n - \frac{f_n - f_{n-1}}{h_n} \right).$$

Zusammen mit (5.17) ist das resultierende Gleichungssystem (5.18) dann gegeben durch

$$A = \begin{pmatrix} 2 & 1 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & 1 & 2 \end{pmatrix},$$

$$d_i = \begin{cases} \frac{6}{h_1} \left[\frac{f_1 - f_0}{h_1} - f'_0 \right], & \text{für } i = 0, \\ \frac{6}{h_i + h_{i+1}} \left[\frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \right], & \text{für } i = 1, \dots, n-1, \\ \frac{6}{h_n} \left[f'_n - \frac{f_n - f_{n-1}}{h_n} \right], & \text{für } i = n, \end{cases}$$

$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}}, \quad i = 1, \dots, n-1,$$

$$\mu_i = 1 - \lambda_i, \quad i = 1, \dots, n-1.$$

Im Falle des natürlichen Splines und des Splines mit hermitescher Datenvorgabe hat die Matrix A eine **tridiagonale Struktur** und ist wegen $\lambda_i + \mu_i = 1 < 2$ **strikt diagonal-dominant**. Bei der numerischen Lösung wird diese tridiagonale Struktur ausgenutzt, um Rechenzeit zu sparen.

5.2.1 Fehlerdarstellung interpolierender kubischer Splines

Wie bei der Polynominterpolation interessieren wir uns für Approximationseigenschaften von kubischen Splines bezüglich einer gegebenen Funktion f . Zusätzlich zur Supremumsnorm $\|\cdot\|_\infty$ definieren wir die L_2 -Norm

$$\|g\|_2 := \left(\int_a^b |g(t)|^2 dt \right)^{1/2}$$

für eine quadratintegrierbare Funktion $g : [a, b] \rightarrow \mathbb{R}$.

Satz 5.2.4

Gegeben seien $a = x_0 < x_1 < \dots < x_n = b$ und eine zweimal stetig differenzierbare Funktion $f : [a, b] \rightarrow \mathbb{R}$.

Der interpolierende kubische Spline s mit

$$s(x_i) = f(x_i), \quad i = 0, \dots, n,$$

und einer der Bedingungen (i), (ii) oder (iii) aus Satz 5.2.3 erfüllt die Fehlerabschätzungen

$$\begin{aligned}\|f - s\|_2 &\leq \frac{1}{4} \|f''\|_2 h_{max}^2, \\ \|f - s\|_\infty &\leq \frac{1}{2} \|f''\|_2 h_{max}^{3/2},\end{aligned}$$

wobei $h_{max} = \max_{i=0, \dots, n-1} (x_{i+1} - x_i)$ sei.

Beweis: Sei $g := f - s$. Nach Definition von h_{max} liegt für jedes $x \in [a, b]$ mindestens eine Stützstelle $x_j = x_j(x)$ im Intervall $[x - h_{max}/2, x + h_{max}/2]$, welche dann auch eine Nullstelle von g ist, da f und s dort übereinstimmen.

Wegen $g(x_j) = 0$ gilt für jedes t zwischen x und x_j die Beziehung

$$g(t) = \int_{x_j}^t g'(\tau) d\tau.$$

Die Cauchy-Schwarzsche Ungleichung liefert für g und $t = x$ die Ungleichung

$$\begin{aligned}|g(x)| &\leq \left| \int_{x_j}^x 1^2 d\tau \right|^{1/2} \cdot \left| \int_{x_j}^x g'(\tau)^2 d\tau \right|^{1/2} \\ &= |x - x_j|^{1/2} \cdot \left| \int_{x_j}^x g'(\tau)^2 d\tau \right|^{1/2} \\ &\leq \left(\frac{h_{max}}{2} \right)^{1/2} \|g'\|_2.\end{aligned}$$

Übergang zur Supremumsnorm liefert

$$\|g\|_\infty \leq \left(\frac{h_{max}}{2} \right)^{1/2} \|g'\|_2. \quad (5.19)$$

Analog folgt für jedes t zwischen x und x_j ,

$$|g(t)|^2 \leq |t - x_j| \cdot \left| \int_{x_j}^t g'(\tau)^2 d\tau \right| \leq |t - x_j| \cdot \left| \int_{x_j}^x g'(\tau)^2 d\tau \right|.$$

Integration liefert

$$\left| \int_{x_j}^x |g(t)|^2 dt \right| \leq \frac{1}{2} |x - x_j|^2 \cdot \left| \int_{x_j}^x g'(\tau)^2 d\tau \right| \leq \frac{1}{2} \left(\frac{h_{max}}{2} \right)^2 \cdot \left| \int_{x_j}^x g'(\tau)^2 d\tau \right|.$$

Durch geeignete Wahl verschiedener Werte von x kann das Intervall $[a, b]$ als Vereinigung endlich vieler, sich nicht überschneidender Intervalle der Form $[x_j, x]$ bzw. $[x, x_j]$ dargestellt werden. Fasst man die obigen Integrale zusammen, so erhält man

$$\|g\|_2^2 \leq \frac{1}{2} \left(\frac{h_{max}}{2} \right)^2 \|g'\|_2^2$$

bzw.

$$\|g\|_2 \leq \frac{1}{2\sqrt{2}} h_{max} \|g'\|_2. \quad (5.20)$$

Wegen $g(x_j) = g(x_{j+1}) = 0$ besitzt g' nach dem Satz von Rolle in jedem Intervall (x_j, x_{j+1}) , $j = 0, \dots, n-1$, mindestens eine Nullstelle. Damit liegt für jedes $x \in [a, b]$ mindestens eine Nullstelle von g' im Intervall $[x - h_{max}, x + h_{max}]$. Eine zu oben analoge Vorgehensweise angewendet auf g' liefert dann die Abschätzung

$$\|g'\|_2 \leq \frac{1}{\sqrt{2}} h_{max} \|g''\|_2.$$

Zusammen mit (5.19) und (5.20) folgen

$$\begin{aligned} \|g\|_\infty &\leq \frac{1}{2} h_{max}^{3/2} \|g''\|_2, \\ \|g\|_2 &\leq \frac{1}{4} h_{max}^2 \|g''\|_2. \end{aligned}$$

Wegen (5.12) und einer der Zusatzeigenschaften (i), (ii) oder (iii) aus Satz 5.2.3 folgt schließlich

$$\|g''\|_2 = \|f''\|_2 - \|s''\|_2 \leq \|f''\|_2,$$

woraus die Behauptungen folgen. □

Die Differenzierbarkeitseigenschaften der Funktion f können weiter abgeschwächt werden, da im Beweis nur verwendet wird, daß f'' eine L_2 -integrierbare Funktion ist.

Setzt man umgekehrt eine höhere Glattheit von f voraus, so können unter zusätzlichen Annahmen bessere Approximationsaussagen bis zur Ordnung h_{max}^4 erzielt werden.

Kapitel 6

Numerische Integration

Viele praktisch relevante Anwendungen, wie z.B. die Lösung einer elliptischen partiellen Differentialgleichung mit der Finite-Element-Methode oder die Fourieranalyse, erfordern die Auswertung von bestimmten Integralen der Form

$$I[f] := \int_a^b f(x) dx, \quad a, b \in \mathbb{R}, a < b. \quad (6.1)$$

Da diese Integrale oftmals nicht exakt berechnet werden können, betrachte z.B.

$$\int_0^{\pi/2} \left[1 - \left(\frac{x}{r} \right)^2 \sin^2 \theta \right]^{1/2} d\theta,$$

werden numerische Methoden zur approximativen Berechnung von (6.1) benötigt.

Zur numerischen Berechnung des Integrals (6.1) verwendet man in der Regel den Ansatz

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

mit $N \in \mathbb{N}_0$, Gewichten w_i und Stützstellen

$$a \leq x_0 \leq x_1 \leq \dots \leq x_N \leq b.$$

Die Aufgabe besteht nun darin, geeignete Werte für w_i und x_i zu finden, so daß die sogenannte **Quadraturformel**

$$Q[f] := \sum_{i=0}^N w_i f(x_i) \quad (6.2)$$

den Wert des Integrals möglichst gut approximiert.

Beispiel 6.0.1 (Trapezregel)

Die Trapezregel basiert auf der linearen Interpolation der Stützstellen $(a, f(a))$ und $(b, f(b))$, vgl. Abbildung 6.1.

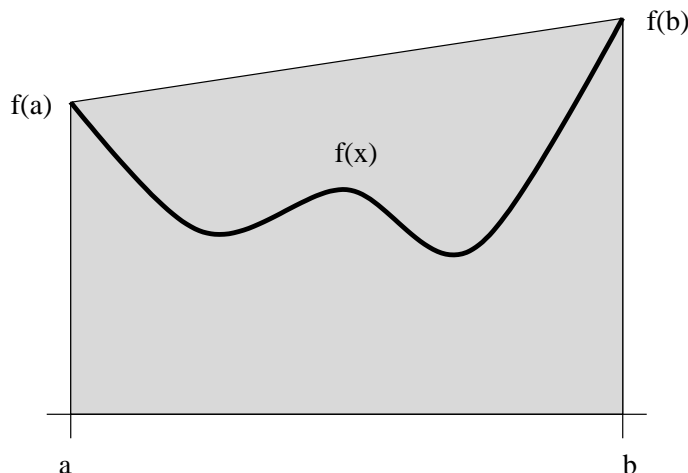


Abbildung 6.1: Trapezregel

Der Integrand f wird dann durch das lineare Interpolationspolynom

$$p_1(x) = f(a) + \frac{x-a}{b-a}(f(b) - f(a)) \quad (6.3)$$

ersetzt. Das resultierende Integral über p_1 kann berechnet werden und lautet

$$\int_a^b f(x)dx \approx \int_a^b p_1(x)dx = \frac{b-a}{2}(f(a) + f(b)) =: T_1[f].$$

Geometrisch ist dies gerade der Flächeninhalt des Trapezes, welches durch die Punkte $(a, 0)$, $(a, f(a))$, $(b, f(b))$, $(b, 0)$ gegeben ist, vgl. Abbildung 6.1. Die Trapezregel $T_1[f]$ ist also eine spezielle Quadraturformel (6.2) mit

$$N = 1, \quad w_0 = w_1 = \frac{b-a}{2}, \quad x_0 = a, \quad x_1 = b.$$

Ein Maß für die Güte der Approximation, welches später noch häufiger auftreten wird, ist u.a. der Exaktheitsgrad einer Quadraturformel.

Definition 6.0.2 (Exaktheitsgrad einer Quadraturformel)

Eine Quadraturformel $Q[f]$ hat **Exaktheitsgrad** p , falls alle Polynome bis zum Grad p exakt integriert werden.

Durch einfaches Nachrechnen zeigt sich, daß die Trapezregel den Exaktheitsgrad eins hat. Polynome ersten Grades werden also durch die Trapezregel exakt integriert.

Wie Abbildung 6.1 bereits erahnen läßt, ist die Approximation von f durch p_1 im Falle der Trapezregel i.a. nicht sehr genau. Häufig verwendet man daher sogenannte **zusammengesetzte Quadraturformeln**. Dabei wird das Intervall $[a, b]$ (in der Regel) äquidistant

mit Schrittweite $h = (b - a)/n$ für $n \in \mathbb{N}$ unterteilt und in jedem Teilintervall $[x_i, x_{i+1}]$ mit $x_i = a + ih$, $i = 0, \dots, n - 1$, wird dann eine Quadraturformel der Form (6.2) auf

$$I_i[f] = \int_{x_i}^{x_{i+1}} f(x)dx, \quad i = 0, \dots, n - 1,$$

angewendet. Eine Approximation von $I[f]$ ergibt sich dann durch Summation über alle Teilintervalle:

$$I[f] = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x)dx \approx \sum_{i=0}^{n-1} Q[f|_{[x_i, x_{i+1}]}] = Q_n[f].$$

Beispiel 6.0.3 (zusammengesetzte Trapezregel)

Das Intervall $[a, b]$ wird durch Einführung von äquidistanten Stützstellen

$$x_i = a + ih, \quad i = 0, \dots, n, \quad h = \frac{b - a}{n} \quad (6.4)$$

in Teilintervalle $[x_i, x_{i+1}]$ zerlegt. Die Trapezregel wird dann auf jedes Teilintervall angewendet, vgl. Abbildung 6.2.

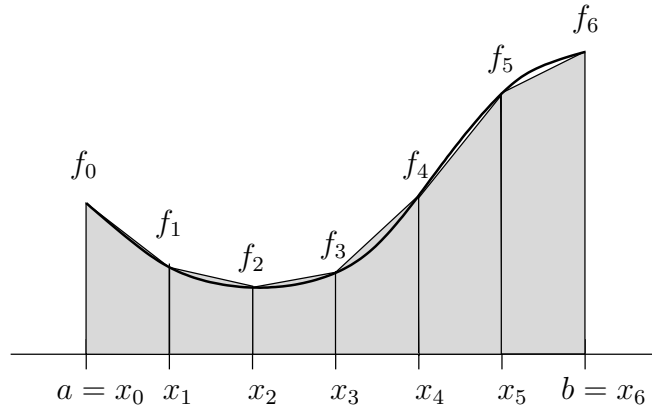


Abbildung 6.2: Zusammengesetzte Trapezregel für $n = 6$.

Dies liefert die **zusammengesetzte Trapezregel** oder **iterierte Trapezregel**

$$T_n[f] := \frac{h}{2}f(a) + h \sum_{i=1}^{n-1} f(x_i) + \frac{h}{2}f(b).$$

Die zusammengesetzte Trapezregel entspricht der Approximation von f durch eine stetige, stückweise lineare Funktion. Sie ist ebenfalls eine Quadraturformel der Form (6.2) mit

$$N = n, \quad w_0 = w_n = \frac{h}{2}, \quad w_i = h, \quad i = 1, \dots, n - 1, \quad x_i = a + ih, \quad i = 0, \dots, n.$$

Bei zusammengesetzten Quadraturformeln spielt die Konvergenzordnung bzgl. der Schrittweite h eine große Rolle.

Definition 6.0.4 (Konvergenzordnung zusammengesetzte Quadraturformel)

Eine zusammengesetzte Quadraturformel $Q_n[f]$ hat die **Konvergenzordnung** p (ist von Ordnung p), falls

$$|Q_n[f] - I[f]| = \mathcal{O}(h^p)$$

mit $h = (b - a)/n$ gilt.

6.1 Interpolatorische Quadraturformeln

Die Idee der interpolatorischen Quadraturformeln besteht darin, die Funktion f im Integranden von $I[f]$ durch ein interpolierendes Polynom mit Höchstgrad N zu ersetzen, welches f an den Stützstellen $a \leq x_0 < x_1 < \dots < x_{N-1} < x_N \leq b$ interpoliert. Für dieses interpolierende Polynom kann das Integral dann leicht ausgewertet werden. Als Spezialfall haben wir die Trapezregel bereits kennen gelernt.

Allgemeiner wird nun ein Polynom p_N höchstens N -ten Grades zur Interpolation der $N+1$ Stützpunkte $(x_i, f(x_i))$, $i = 0, \dots, N$, verwendet. Dieses Interpolationspolynom ist nach Satz 5.1.2 eindeutig bestimmt und besitzt nach (5.4) die Darstellung

$$p_N(x) = \sum_{k=0}^N f(x_k) L_k(x),$$

wobei $L_k(x)$ das k -te Lagrange'sche Polynom bezeichnet.

Ersetzen von f durch p_N in $I[f]$ in (5.1) liefert die Approximation

$$I[f] = \int_a^b f(x) dx \approx \int_a^b p_N(x) dx,$$

wobei das rechts stehende Integral explizit berechnet werden kann. Es gilt

$$\int_a^b p_N(x) dx = \sum_{k=0}^N f(x_k) \underbrace{\int_a^b L_k(x) dx}_{=: w_k} = \sum_{k=0}^N w_k f(x_k)$$

mit den Koeffizienten

$$w_k = \int_a^b L_k(x) dx.$$

Insgesamt erhalten wir daraus die interpolatorischen Quadraturformeln.

Definition 6.1.1 (interpolatorische Quadraturformeln)

Seien $a < b$, $N \in \mathbb{N}_0$ und Stützstellen x_i , $i = 0, \dots, N$, mit

$$a \leq x_0 < x_1 < \dots < x_{N-1} < x_N \leq b$$

gegeben. Die Quadraturformel

$$Q[f] = \sum_{k=0}^N w_k f(x_k), \quad w_k = \int_a^b L_k(x) dx \quad (6.5)$$

heißt **interpolatorische Quadraturformel**.

Im Fall äquidistanter Stützstellen

$$x_i = a + ih, \quad i = 0, \dots, N, \quad h = \frac{b-a}{N}$$

heißt eine interpolatorische Quadraturformel **geschlossenes Newton-Cotes-Verfahren**, wobei hierbei $(a, f(a))$ und $(b, f(b))$ interpoliert werden.

Im Fall äquidistanter Stützstellen

$$x_i = a + (i+1)h, \quad i = 0, \dots, N, \quad h = \frac{b-a}{N+2}$$

heißt eine interpolatorische Quadraturformel **offenes Newton-Cotes-Verfahren**, wobei hierbei $(a, f(a))$ und $(b, f(b))$ nicht interpoliert werden.

Gemäß Definition 6.1.1 ist die Trapezregel also ein geschlossenes Newton-Cotes-Verfahren. Ein Beispiel für ein offenes Newton-Cotes/-Verfahren ist die folgende Mittelpunktsregel.

Beispiel 6.1.2 (Mittelpunktsregel)

Sei $[a, b]$, $a < b$, gegeben. Wähle $N = 0$ und $x_0 = (a+b)/2$ (Mittelpunkt des Intervalls). Das interpolierende Polynom 0-ten Grades zum Stützwert $(x_0, f(x_0))$ ist gegeben durch $p_0(x) = f(x_0)$. Damit wird das Integral $I[f]$ in (6.1) approximiert durch die sogenannte **Mittelpunktsregel**

$$I[f] \approx M_0[f] = \int_a^b p_0(x) dx = (b-a) f\left(\frac{a+b}{2}\right),$$

vgl. Abbildung 6.3.

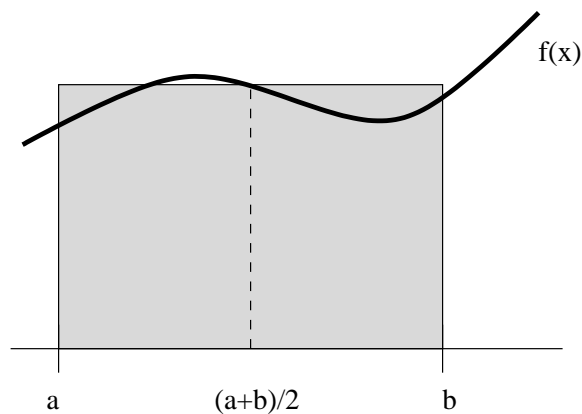


Abbildung 6.3: Mittelpunkstregel

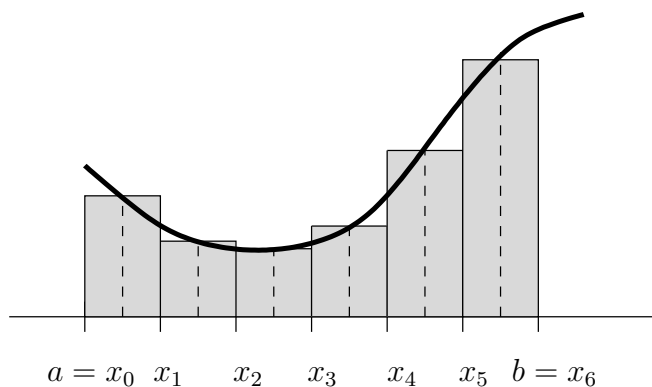
Dies ist ein offenes Newton-Cotes-Verfahren der Form (6.2) mit

$$N = 0, w_0 = b - a, x_0 = (a + b)/2.$$

Die daraus resultierende zusammengesetzte Quadraturformel lautet

$$M_n[f] = \sum_{i=0}^{n-1} (x_{i+1} - x_i) f\left(\frac{x_i + x_{i+1}}{2}\right) = h \sum_{i=0}^{n-1} f\left(\frac{x_i + x_{i+1}}{2}\right)$$

mit $h = (b - a)/n$, vgl. Abbildung 6.4.

Abbildung 6.4: Iterierte Mittelpunktsregel für $n = 6$.

Wir interessieren uns nun für den Approximationsfehler interpolatorischer Quadraturformeln, wofür wir auf Satz 5.1.11 zurück greifen.

Satz 6.1.3

Seien $f : [a, b] \rightarrow \mathbb{R}$ $(N + 1)$ -mal stetig differenzierbar und $a \leq x_0 < x_1 < \dots < x_N \leq b$ gegeben. Dann gilt

$$|I[f] - Q[f]| \leq \frac{1}{(N + 1)!} \max_{x \in [a, b]} |f^{(N+1)}(x)| \cdot \int_a^b \prod_{i=0}^N |x - x_i| dx$$

für die interpolatorische Quadraturformel $Q[f]$ in (6.5).

Insbesondere besitzt die interpolatorische Quadraturformel $Q[f]$ (mindestens) den Exaktheitsgrad N .

Beweis: Es gilt

$$|I[f] - Q[f]| = \left| \int_a^b f(x) - p_N(x) dx \right| \leq \int_a^b |f(x) - p_N(x)| dx.$$

Die Fehlerdarstellung des interpolierenden Polynoms p_N aus Satz 5.1.11 liefert

$$|f(x) - p_N(x)| \leq \frac{1}{(N + 1)!} \max_{x \in [a, b]} |f^{(N+1)}(x)| \cdot \prod_{i=0}^N |x - x_i|$$

für alle $x \in [a, b]$. Integration liefert die erste Behauptung.

Da $f^{(N+1)} \equiv 0$ für alle Polynome mit Höchstgrad N gilt, folgt aus der Fehlerdarstellung sofort $|I[f] - Q[f]| = 0$, so daß Polynome bis zum Höchstgrad N exakt integriert werden.

□

Der maximale Exaktheitsgrad von $Q[f]$ kann größer als N ausfallen. So besitzt die Simpsonregel (geschlossenes Newton-Cotes-Verfahren mit $N = 2$) sogar den Exaktheitsgrad $N + 1 = 3$.

Tabelle 6.1 faßt einige gebräuchliche geschlossene Newton-Cotes-Verfahren und deren Exaktheitsgrad zusammen. Hierbei gilt $x_i = a + ih$, $i = 0, \dots, N$, $h = (b - a)/N$.

Tabelle 6.1: Einige gebräuchliche geschlossene Newton-Cotes-Verfahren

N	Gewichte	Name	Fehler	E.-Grad
1	$w_0 = w_1 = h/2$	Trapezregel	$-\frac{h^3}{12}f^{(2)}(\xi)$	1
2	$w_0 = w_2 = h/3,$ $w_1 = 4h/3$	Simpsonregel	$-\frac{h^5}{90}f^{(4)}(\xi)$	3
3	$w_0 = w_3 = 3h/8,$ $w_1 = w_2 = 9h/8$	3/8-Regel (pulcherrima)	$-\frac{3h^5}{80}f^{(4)}(\xi)$	3
4	$w_0 = w_4 = 28h/90,$ $w_1 = w_3 = 128h/90,$ $w_2 = 48h/90$	Milneregel	$-\frac{8h^7}{945}f^{(6)}(\xi)$	5

Tabelle 6.2 faßt einige gebräuchliche offene Newton-Cotes-Verfahren und deren Exaktheitsgrad zusammen. Hierbei gilt $x_i = a + (i + 1)h$, $i = 0, \dots, N$, $h = (b - a)/(N + 2)$.

Tabelle 6.2: Einige gebräuchliche offene Newton-Cotes-Verfahren

N	Gewichte	Name	Fehler	E.-Grad
0	$w_0 = 2h$	Mittelpunktsregel	$\frac{h^3}{3}f^{(2)}(\xi)$	1
1	$w_0 = w_1 = 3h/2,$		$\frac{3h^3}{4}f^{(2)}(\xi)$	1
2	$w_0 = w_2 = 8h/3,$ $w_1 = -4h/3$		$\frac{28h^5}{90}f^{(4)}(\xi)$	3
3	$w_0 = w_3 = 55h/24,$ $w_1 = w_2 = 5h/24,$		$\frac{95h^5}{144}f^{(4)}(\xi)$	3

Für jede Regel gilt jeweils $\xi \in [a, b]$ mit eigenem ξ .

6.1.1 Zusammengesetzte Trapezregel und Euler-MacLaurin'sche Summenformel

Für jede interpolatorische Quadraturformel läßt sich eine zugehörige zusammengesetzte interpolatorische Quadraturformel entwickeln. Auf Grund der besonderen Wichtigkeit der zusammengesetzten Trapezregel für die Romberg-Extrapolation, die im folgenden Abschnitt besprochen wird, betrachten wir hier stellvertretend für andere zusammengesetzte Quadraturformeln die zusammengesetzte Trapezregel

$$T_n[f] := \frac{h}{2}f(a) + h \sum_{i=1}^{n-1} f(x_i) + \frac{h}{2}f(b) \quad (6.6)$$

mit $x_i = a + ih$, $i = 0, \dots, n$, $h = (b - a)/n$, $n \in \mathbb{N}$, zur Approximation von $I[f]$. Insbesondere interessieren wir uns für die Konvergenzordnung und die Darstellung des Approximationsfehlers.

Hauptresultat dieses Abschnitts ist folgender Satz.

Satz 6.1.4 (asymptotische Fehlerdarstellung der zusammengesetzten Trapezregel)

Sei $f : [a, b] \rightarrow \mathbb{R}$ $(2m + 2)$ -mal stetig differenzierbar. Dann gilt für die zusammengesetzte Trapezregel $T_n[f]$ die **asymptotische Fehlerdarstellung**

$$T_n[f] - I[f] = \gamma_1 h^2 + \gamma_2 h^4 + \dots + \gamma_m h^{2m} + r(h) h^{2m+2}$$

mit Konstanten γ_i , $i = 1, \dots, m$, und beschränktem Restgliedkoeffizienten $|r(h)| \leq K$.

Beispiel 6.1.5

Betrachte

$$f(x) = \tan(x), \quad I[f] = \int_0^1 f(x) dx = [-\ln(\cos(x))]_0^1 = -\ln(\cos(1)).$$

Wir wenden die iterierte Trapezregel an und schätzen numerisch die Ordnung der Approximation. Hierzu wird angenommen, daß $|T_n[f] - I[f]| \approx Ch^p$ mit $h = 1/n$ und einer Konstanten C gilt. Dann gilt $|T_{2n}[f] - I[f]| \approx C(h/2)^p$ und somit

$$\frac{|T_n[f] - I[f]|}{|T_{2n}[f] - I[f]|} = \frac{Ch^p}{C(h/2)^p} = 2^p$$

und daraus folgt

$$p = \ln \left(\frac{|T_n[f] - I[f]|}{|T_{2n}[f] - I[f]|} \right) / \ln(2).$$

Ergebnisse der iterierten Trapezregel:

N	Wert	Fehler	rel. Fehler	Ordnung
4	0.6279861835	0.0123597131	0.02007664338	
8	0.6187665645	0.0031400941	0.00510064828	1.97
16	0.6164148747	0.0007884043	0.00128065367	1.99

Die numerischen Werte deuten auf die Ordnung 2 hin.

Zum Beweis des Satzes benötigen wir die berühmte Euler-MacLaurin'sche Summenformel, welche mit Hilfe der Bernoulli-Polynome gezeigt werden kann.

Definition 6.1.6 (Bernoulli-Polynome)

Die **Bernoulli-Polynome** $B_k : [0, 1] \rightarrow \mathbb{R}$ sind rekursiv definiert durch

$$\begin{aligned} B_0(x) &= 1, \\ B'_k(x) &= kB_{k-1}(x) \quad \text{mit} \quad \int_0^1 B_k(x) dx = 0, \quad k = 1, 2, \dots \end{aligned}$$

Die Zahlen $B_k := B_k(0)$ heißen **Bernoulli-Zahlen**.

Die Bernoulli-Polynome bis zur Ordnung 4 lauten

$$\begin{aligned} B_0(x) &= 1, \\ B_1(x) &= x - \frac{1}{2}, \\ B_2(x) &= x^2 - x + \frac{1}{6}, \\ B_3(x) &= x^3 - \frac{3}{2}x^2 + \frac{1}{2}x, \\ B_4(x) &= x^4 - 2x^3 + x^2 - \frac{1}{30}. \end{aligned}$$

Wir fassen einige Eigenschaften der Bernoulli-Polynome zusammen:

Hilfsatz 6.1.7

Die Bernoulli-Polynome besitzen folgende Eigenschaften:

- (i) $B_k(0) = B_k(1)$ für alle $k \geq 2$
- (ii) $B_k(x) = (-1)^k B_k(1-x)$ für alle $k \geq 0$, $x \in [0, 1]$
- (iii) $B_{2m-1}(1/2) = 0$ für alle $m \geq 1$ und $B_{2m-1}(0) = B_{2m-1}(1) = 0$ für alle $m \geq 2$.
- (iv) $(-1)^m B_{2m-1}(x) > 0$ für alle $m \geq 1$ und alle $x \in (0, 1/2)$
- (v) $(-1)^m (B_{2m}(x) - B_{2m}) > 0$ für alle $m \geq 1$ und alle $x \in (0, 1)$

(vi) $(-1)^{m+1}B_{2m}(x) > 0$ für alle $m \geq 1$.

Beweis: Übung

Durch Ausnutzen der Eigenschaften in Hilfsatz 6.1.7, kann die folgende Euler-MacLaurin'sche Summenformel bewiesen werden.

Satz 6.1.8 (Euler-MacLaurin'sche Summenformel)

Sei $g : [0, 1] \rightarrow \mathbb{R}$ eine $(2m + 2)$ -mal stetig differenzierbare Funktion. Dann existiert $\xi \in (0, 1)$ mit

$$\int_0^1 g(x)dx = \frac{1}{2}(g(0) + g(1)) + \sum_{k=1}^m \frac{B_{2k}}{(2k)!} (g^{(2k-1)}(0) - g^{(2k-1)}(1)) - \frac{B_{2m+2}}{(2m+2)!} g^{(2m+2)}(\xi).$$

Beweis: Aus der Definition der Bernoulli-Polynome folgt

$$B_k(x) = \frac{1}{k+1} B'_{k+1}(x).$$

Mit dieser Formel und partieller Integration folgt

$$\int_0^1 g(x)dx = \int_0^1 g(x)B_0(x)dx = B_1(x)g(x)\Big|_0^1 - \int_0^1 B_1(x)g'(x)dx.$$

Partielle Integration angewendet auf das letzte Integral liefert dann in analoger Weise

$$\int_0^1 B_1(x)g'(x)dx = \frac{1}{2}B_2(x)g'(x)\Big|_0^1 - \frac{1}{2}\int_0^1 B_2(x)g''(x)dx.$$

Dieser Vorgang kann fortgesetzt werden und es folgt

$$\int_0^1 B_{k-1}(x)g^{(k-1)}(x)dx = \frac{1}{k}B_k(x)g^{(k-1)}(x)\Big|_0^1 - \frac{1}{k}\int_0^1 B_k(x)g^{(k)}(x)dx$$

für $k = 1, 2, \dots, 2m + 2$. Insgesamt ergibt sich daraus durch wiederholtes Einsetzen

$$\begin{aligned} \int_0^1 g(x)dx &= -\sum_{k=1}^m \frac{1}{(2k)!} B_{2k}(x)g^{(2k-1)}(x)\Big|_0^1 + \sum_{k=1}^m \frac{1}{(2k-1)!} B_{2k-1}(x)g^{(2k-2)}(x)\Big|_0^1 \\ &\quad - \frac{1}{(2m+1)!} \int_0^1 B_{2m+1}(x)g^{(2m+1)}(x)dx. \end{aligned}$$

Aus $B_k(0) = B_k(1) = B_k$ für alle $k \geq 2$, vgl. (i) in Hilfsatz 6.1.7, folgt

$$B_k(x)g^{(k-1)}(x)\Big|_0^1 = B_k(g^{(k-1)}(1) - g^{(k-1)}(0)).$$

Für ungerade Indizes gilt nach (iii) in Hilfsatz 6.1.7 jedoch $B_{2k-1} = 0$ für alle $k \geq 2$ und zusammen mit $B_1(1) = 1/2$ und $B_1(0) = -1/2$ folgt insgesamt

$$\begin{aligned} \int_0^1 g(x) dx &= \frac{1}{2} (g(0) + g(1)) - \sum_{k=1}^m \frac{B_{2k}}{(2k)!} (g^{(2k-1)}(1) - g^{(2k-1)}(0)) \\ &\quad - \frac{1}{(2m+1)!} \int_0^1 B_{2m+1}(x) g^{(2m+1)}(x) dx. \end{aligned}$$

Partielle Integration des letzten Integrals liefert

$$\begin{aligned} \int_0^1 B_{2m+1}(x) g^{(2m+1)}(x) dx &= \frac{1}{2m+2} (B_{2m+2}(x) - B_{2m+2}) g^{(2m+1)}(x) \Big|_0^1 \\ &\quad - \frac{1}{2m+2} \int_0^1 (B_{2m+2}(x) - B_{2m+2}) g^{(2m+2)}(x) dx \\ &= -\frac{1}{2m+2} \int_0^1 (B_{2m+2}(x) - B_{2m+2}) g^{(2m+2)}(x) dx. \end{aligned}$$

Da $B_{2m+2}(x) - B_{2m+2}$ auf $[0, 1]$ gemäß (v) in Hilfsatz 6.1.7 dasselbe Vorzeichen besitzt, gibt es nach dem Mittelwertsatz der Integralrechnung ein $\xi \in (0, 1)$ mit

$$\begin{aligned} \int_0^1 (B_{2m+2}(x) - B_{2m+2}) g^{(2m+2)}(x) dx &= g^{(2m+2)}(\xi) \int_0^1 (B_{2m+2}(x) - B_{2m+2}) dx \\ &= -B_{2m+2} g^{(2m+2)}(\xi). \end{aligned}$$

Zusammen mit (6.7) ergibt sich also die Behauptung. \square

Mit Hilfe der Euler-MacLaurin'schen Summenformel kann schließlich auch die asymptotische Fehlerdarstellung der zusammengesetzten Trapezregel bewiesen werden.

Beweis: von Satz 6.1.4. Wir wenden die Euler-MacLaurin'sche Summenformel auf die durch $g(t) := f(x_i + ht)$ mit $h = x_{i+1} - x_i$ und $t \in [0, 1]$ definierte Funktion an und erhalten für ein $\xi_i \in (0, 1)$ die Gleichung

$$\begin{aligned} \int_{x_i}^{x_{i+1}} f(x) dx &= h \int_0^1 g(t) dt \\ &= \frac{h}{2} (g(0) + g(1)) + \sum_{k=1}^m \frac{B_{2k}}{(2k)!} h (g^{(2k-1)}(0) - g^{(2k-1)}(1)) \\ &\quad - \frac{B_{2m+2}}{(2m+2)!} h g^{(2m+2)}(\xi_i) \\ &= \frac{h}{2} (f(x_i) + f(x_{i+1})) + \sum_{k=1}^m \frac{B_{2k}}{(2k)!} h^{2k} (f^{(2k-1)}(x_i) - f^{(2k-1)}(x_{i+1})) \\ &\quad - \frac{B_{2m+2}}{(2m+2)!} h^{2m+3} f^{(2m+2)}(x_i + \xi_i h). \end{aligned}$$

Summation liefert

$$\begin{aligned}
 \int_a^b f(x)dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x)dx \\
 &= T_n[f] + \sum_{k=1}^m \frac{B_{2k}}{(2k)!} h^{2k} \sum_{i=0}^{n-1} (f^{(2k-1)}(x_i) - f^{(2k-1)}(x_{i+1})) \\
 &\quad - \frac{B_{2m+2}}{(2m+2)!} h^{2m+3} \sum_{i=0}^{n-1} f^{(2m+2)}(x_i + \xi_i h) \\
 &= T_n[f] + \sum_{k=1}^m \frac{B_{2k}}{(2k)!} h^{2k} (f^{(2k-1)}(x_0) - f^{(2k-1)}(x_n)) \\
 &\quad - \frac{B_{2m+2}}{(2m+2)!} h^{2m+3} \sum_{i=0}^{n-1} f^{(2m+2)}(x_i + \xi_i h)
 \end{aligned}$$

Wegen

$$\min_{i=0, \dots, n-1} f^{(2m+2)}(x_i + h\xi_i) \leq \frac{1}{n} \sum_{i=0}^{n-1} f^{(2m+2)}(x_i + \xi_i h) \leq \max_{i=0, \dots, n-1} f^{(2m+2)}(x_i + h\xi_i)$$

gibt es nach dem Zwischenwertsatz ein $\xi_h \in (a, b)$ mit

$$f^{(2m+2)}(\xi_h) = \frac{1}{n} \sum_{i=0}^{n-1} f^{(2m+2)}(x_i + \xi_i h).$$

Zusammen mit $x_0 = a$ und $x_n = b$ folgt

$$\begin{aligned}
 \int_a^b f(x)dx - T_n[f] &= \sum_{k=1}^m \frac{B_{2k}}{(2k)!} h^{2k} (f^{(2k-1)}(a) - f^{(2k-1)}(b)) \\
 &\quad - \frac{B_{2m+2}}{(2m+2)!} h^{2m+3} (b-a) f^{(2m+2)}(\xi_h).
 \end{aligned}$$

Mit $\gamma_k := -\frac{B_{2k}}{(2k)!} (f^{(2k-1)}(a) - f^{(2k-1)}(b))$, $k = 1, \dots, m$, und $r(h) = \frac{B_{2m+2}}{(2m+2)!} (b-a) f^{(2m+2)}(\xi_h)$ folgt die Behauptung. Beachte dabei, daß $r(h)$ für alle h beschränkt ist durch

$$|r(h)| \leq \frac{|B_{2m+2}|}{(2m+2)!} (b-a) \max_{\xi \in [a, b]} |f^{(2m+2)}(\xi)|,$$

wobei das Maximum von $f^{(2m+2)}$ auf $[a, b]$ auf Grund der Stetigkeit von $f^{(2m+2)}$ angenommen wird. \square

Interessant ist folgende Beobachtung. Für periodische Funktionen mit Periodenlänge $b-a$ gilt unter den Voraussetzungen des Satzes $f^{(2k-1)}(a) = f^{(2k-1)}(b)$ für $k = 1, \dots, m$ und somit

$$T_n[f] - \int_a^b f(x)dx = \frac{B_{2m+2}}{(2m+2)!} h^{2m+3} (b-a) f^{(2m+2)}(\xi_h) = \mathcal{O}(h^{2m+3}).$$

Die zusammengesetzte Trapezregel liefert also für periodische Funktionen eine extrem genaue Approximation.

Das Besondere an der Fehlerdarstellung der zusammengesetzten Trapezregel ist die Tatsache, daß ausschließlich gerade Potenzen von h auftreten. Dieser Umstand wird von sogenannten Extrapolationsverfahren ausgenutzt.

Allerdings benötigt die Fehlerdarstellung eine hinreichend hohe Glätte der Funktion f . Das folgende Resultat besagt jedoch, daß die zusammengesetzte Trapezregel sogar für Funktionen f , die nur Riemann-integrierbar sind, konvergiert.

Satz 6.1.9

Sei $f : [a, b] \rightarrow \mathbb{R}$ Riemann-integrierbar. Dann gilt

$$\lim_{n \rightarrow \infty} |T_n[f] - I[f]| = 0.$$

Beweis: Es gilt

$$T_n[f] = h \left(\frac{f(a)}{2} + \sum_{k=1}^{n-1} f(x_k) + \frac{f(b)}{2} \right) = \frac{1}{2}h \sum_{k=0}^{n-1} f(x_k) + \frac{1}{2}h \sum_{k=1}^n f(x_k).$$

In dieser Darstellung wird $T_n[f]$ dargestellt als Durchschnitt von zwei Riemann-Summen für $I[f]$, die jeweils auf der Schrittweite h basieren. Da f als Riemann-integrierbar vorausgesetzt ist, gilt

$$\lim_{h \downarrow 0} h \sum_{k=1}^{n-1} f(x_k) = I[f] = \lim_{h \downarrow 0} h \sum_{k=1}^n f(x_k),$$

woraus die Behauptung folgt. □

6.2 Romberg-Verfahren und Extrapolation

Die Idee des Romberg-Verfahrens besteht darin, für verschiedene Schrittweiten h_i Näherungen für $I[f]$ mit der zusammengesetzten Trapezregel zu berechnen und diese Näherungen geeignet zu kombinieren, um bessere Näherungen zu erhalten. Konkret ergibt sich folgender

Algorithmus 6.2.1 (Romberg-Verfahren)

(0) Wähle eine Folge von Zahlen $n_i \in \mathbb{N}$, $i = 0, 1, \dots, m$ mit

$$n_0 < n_1 < n_2 < \dots < n_m$$

und berechne die zugehörigen Schrittweiten $h_i = (b - a)/n_i$, $i = 0, 1, 2, \dots, m$.

(1) Berechne für $i = 0, 1, 2, \dots, m$ die zusammengesetzten Trapezsummen

$$T_{i,0} := T_{n_i}[f].$$

(2) Berechne für $k = 1, \dots, i$, $k \leq m$ rekursiv die Werte

$$T_{i,k} := T_{i,k-1} + \frac{T_{i,k-1} - T_{i-1,k-1}}{\left(\frac{h_{i-k}}{h_i}\right)^2 - 1}.$$

Die Berechnungsvorschrift kann in Tableauform („Neville-Schema“) zusammengefaßt werden

$$\begin{array}{ccccccc} T_{n_0}[f] & = & T_{0,0} & & & & \\ T_{n_1}[f] & = & T_{1,0} & T_{1,1} & & & \\ T_{n_2}[f] & = & T_{2,0} & T_{2,1} & T_{2,2} & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\ T_{n_m}[f] & = & T_{m,0} & T_{m,1} & T_{m,2} & \cdots & T_{m,m} \end{array}$$

In der Praxis werden die **Romberg-Folge** (Schrittweitenhalbierung)

$$\{n_0, n_1, n_2, n_3, \dots, n_i, \dots\} = \{1, 2, 4, 8, \dots, 2n_{i-1}, \dots\}$$

oder die **Bulirsch-Folge**

$$\{n_0, n_1, n_2, n_3, n_4, n_5, \dots, n_i, \dots\} = \{1, 2, 3, 4, 6, 8, \dots, 2n_{i-2}, \dots\} \quad (i \geq 3)$$

verwendet.

Beispiel 6.2.2

Betrachte

$$f(x) = \tan(x), \quad \int_0^1 f(x) dx = [-\ln(\cos(x))]_0^1 = -\ln(\cos(1)).$$

Das Romberg-Verfahren liefert folgende Ausgabe:

```
EXAKTER WERT:    0.6156264704E+00
ERGEBNIS DER ROMBERGEXTRAPOLATION:
0.7787038623E+00
0.6625031761E+00  0.6237696140E+00
0.6279861833E+00  0.6164805191E+00  0.6159945794E+00
0.6187665645E+00  0.6156933582E+00  0.6156408808E+00  0.6156352666E+00
0.6164148747E+00  0.6156309781E+00  0.6156268194E+00  0.6156265962E+00  0.6156265622E+00
FEHLER DER ROMBERGEXTRAPOLATION:
0.1630773919E+00
0.4687670570E-01  0.8143143619E-02
0.1235971295E-01  0.8540486976E-03  0.3681090361E-03
0.3140094122E-02  0.6688784635E-04  0.1441045627E-04  0.8796193098E-05
0.7884043243E-03  0.4507725118E-05  0.3490503687E-06  0.1258534497E-06  0.9185211769E-07
```

Eine genauere Betrachtung der Fehler in Spalte 1 zeigt, daß sich der Fehler bei jeder Schritthalbierung viertelt, was auf die Approximationsordnung 2 der iterierten Trapezregel schließen läßt. In der zweiten Spalte erkennt man (mit etwas gutem Willen), daß sich der Fehler bei Schrittweithalbierung etwa um den Faktor $1/16$ verringert, was auf die Ordnung 4 hindeutet.

Es stellt sich die Frage, was es mit den in Schritt (2) des Algorithmus berechneten Werten $T_{i,k}$ auf sich hat. Für eine $(2m+2)$ -mal stetig differenzierbare Funktion f gilt nach Satz 6.1.4 die asymptotische Fehlerdarstellung

$$T_n[f] - I[f] = \gamma_1 h^2 + \gamma_2 h^4 + \dots + \gamma_m h^{2m} + r(h) h^{2m+2}$$

für die zusammengesetzte Trapezregel, wobei die γ_i 's Konstanten und $|r(h)| \leq K$ sind. Zur Vereinfachung betrachten wir die Romberg-Folge $h_i = h_{i-1}/2$. Für zwei konkrete Schrittweiten $h_i = h_{i-1}/2$ gelten die Fehlerdarstellungen

$$T_{i-1,0} - I[f] = \sum_{j=1}^m \gamma_j h_{i-1}^{2j} + r(h_{i-1}) h_{i-1}^{2m+2}, \quad (6.7)$$

$$T_{i,0} - I[f] = \sum_{j=1}^m \gamma_j h_i^{2j} + r(h_i) h_i^{2m+2} \quad (6.8)$$

$$= \sum_{j=1}^m \gamma_j \left(\frac{h_{i-1}}{2} \right)^{2j} + r \left(\frac{h_{i-1}}{2} \right) \left(\frac{h_{i-1}}{2} \right)^{2m+2}. \quad (6.9)$$

Betrachte das Interpolationspolynom p der Form

$$p(h) = \alpha_0 + \alpha_1 h^2$$

in h , welches die Werte $(h_{i-1}, T_{i-1,0})$ und $(h_i, T_{i,0})$ interpoliert. Es folgt

$$\alpha_0 = \frac{h_i^2 T_{i-1,0} - h_{i-1}^2 T_{i,0}}{h_i^2 - h_{i-1}^2}, \quad \alpha_1 = \frac{T_{i,0} - T_{i-1,0}}{h_i^2 - h_{i-1}^2}.$$

Auswertung des Polynoms an der Stelle $h = 0$ (**Extrapolation**) liefert mit einigen Umformungen

$$p(0) = \frac{h_i^2 T_{i-1,0} - h_{i-1}^2 T_{i,0}}{h_i^2 - h_{i-1}^2} = T_{i,0} + \frac{T_{i,0} - T_{i-1,0}}{\left(\frac{h_{i-1}}{h_i} \right)^2 - 1} = T_{i,1}. \quad (6.10)$$

Für $T_{i,1}$ folgt durch Einsetzen von (6.7) und (6.8) die Fehlerdarstellung

$$\begin{aligned} T_{i,1} - I[f] &= \sum_{j=2}^m \tilde{\gamma}_j h_{i-1}^{2j} + \tilde{r}(h_{i-1}) h_{i-1}^{2m+2} \\ &= \tilde{\gamma}_2 h_{i-1}^4 + \tilde{\gamma}_3 h_{i-1}^6 + \dots + \tilde{\gamma}_m h_{i-1}^{2m} + \tilde{r}(h_{i-1}) h_{i-1}^{2m+2} \end{aligned}$$

mit

$$\tilde{\gamma}_j = \frac{\gamma_j}{3} \left(\frac{1}{2^{2j-2}} - 1 \right), \quad \tilde{r}(h_{i-1}) = \frac{4}{3} \left(\frac{r(h_{i-1}/2)}{2^{2m+2}} - 4r(h_{i-1}) \right).$$

Also hat $T_{i,1}$ genau wie $T_{i-1,0}$ und $T_{i,0}$ eine asymptotische Darstellung mit beschränktem Restglied \tilde{r} . Während der führende Term in den Fehlerdarstellungen (6.7) und (6.8) noch h_{i-1}^2 enthielt, lautet dieser für $T_{i,1}$ aber $\tilde{\gamma}_2 h_{i-1}^4$. Wir haben also durch eine geeignete Linearkombination von Approximationen $T_{i-1,0}$ und $T_{i,0}$ der Ordnung 2 gemäß (6.10) eine bessere Approximation $T_{i,1}$ der Ordnung 4 berechnet, indem wir ein geeignetes interpolierendes Polynom an der Stelle $h = 0$ ausgewertet haben (**Extrapolationsvorgang**). Betrachte das Neville-Schema. Die Werte $T_{i-1,0}$ und $T_{i,0}$ stehen in der ersten Spalte, die Werte $T_{i,1}$ stehen in der zweiten Spalte. Beim Übergang von der ersten Spalte zur zweiten Spalte gewinnen wir also zwei Potenzen in h an Genauigkeit. Die Vermutung liegt nahe, daß beim Übergang von der zweiten zur dritten Spalte wieder zwei Potenzen in h gewonnen werden. Allgemein sei

$$p_{ik}(h) = \alpha_0 + \alpha_1 h^2 + \dots + \alpha_k h^{2k}$$

dasjenige Interpolationspolynom, welches die Werte $(h_{i-k}, T_{i-k,0}), (h_{i-k+1}, T_{i-k+1,0}), \dots, (h_i, T_{i,0})$ interpoliert, vgl. Abbildung 6.5, d.h.

$$p_{ik}(h_j) = T_{j,0} = T_{n_j}[f], \quad j = i-k, i-k+1, \dots, i.$$

Dann gilt

$$T_{i,k} = p_{ik}(0).$$

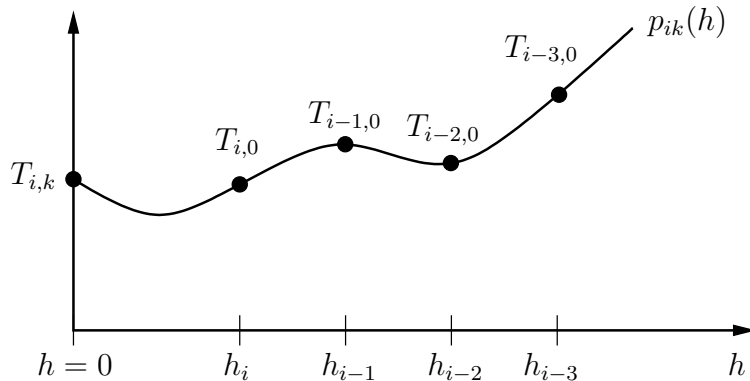


Abbildung 6.5: Prinzip der Extrapolation für $k = 3$.

Für die Romberg-Folge ist die Fehlerordnung

$$T_{i,k} - I[f] = \mathcal{O}(h_{i-k}^{2k+2}).$$

Mit Hilfe des Extrapolationsverfahrens können sehr schnell extrem genaue Approximationen berechnet werden. Der Zusammenhang des Tableaus mit dem Neville-Schema zur Berechnung des Wertes eines Interpolationspolynoms wird deutlich, wenn p_{ik} als Polynom in h^2 aufgefaßt wird:

$$p_{ik}(h) = \alpha_0 + \alpha_1(h^2)^1 + \alpha_2(h^2)^2 + \dots + \alpha_k(h^2)^k.$$

Grundvoraussetzung zur Durchführbarkeit der Extrapolation ist die Existenz einer asymptotischen Entwicklung. Diese muß nicht notwendig nur gerade Potenzen von h enthalten, sondern kann für eine Funktion T , die das exakte Ergebnis γ_0 approximiert, allgemein wie folgt aussehen:

$$T(h) = \gamma_0 + \gamma_1 h^{\tau_1} + \dots + \gamma_m h^{\tau_m} + r(h) h^{\tau_{m+1}}$$

mit Konstanten $\gamma_0, \dots, \gamma_m$, $0 < \tau_1 < \tau_2 < \dots < \tau_{m+1}$ und beschränkter Funktion r . Die Beschränktheit von r ist wichtig, da ansonsten nicht gewährleistet ist, daß der Fehler für $h \rightarrow 0$ beschränkt bleibt. Der Nachweis der Existenz einer asymptotischen Entwicklung für ein konkretes Approximationsverfahren T ist keinesfalls trivial. Desweiteren besitzt nicht jedes Approximationsverfahren eine asymptotische Entwicklung.

Die Existenz einer asymptotischen Entwicklung ermöglicht auch eine Schrittweitensteuerung. Die Idee ist, daß aus zwei gegebenen Näherungen eine neue Schrittweite berechnet wird, so daß der Fehler der zugehörigen Approximation eine gegebene Toleranz nicht übersteigt.

6.3 Gauss'sche Quadraturformeln

Ziel ist es nun, Quadraturformeln der Form (6.2) mit **maximalem Exaktheitsgrad** herzuleiten, wobei neben den Gewichten w_i auch die Stützstellen x_i **frei wählbar** sind. Wir betrachten hier etwas allgemeinere Integrale der Form

$$I_\alpha[f] := \int_a^b \alpha(x) f(x) dx$$

mit einer gegebenen stetigen und positiven Gewichtsfunktion $\alpha : [a, b] \rightarrow \mathbb{R}_+$, die durch die Quadraturformel

$$Q_N[f] = \sum_{k=0}^N w_k f(x_k)$$

approximiert werden sollen. Als Spezialfall kann (und wird) stets $\alpha \equiv 1$ betrachtet werden. Wir haben im Abschnitt über interpolatorische Quadraturformeln gesehen, daß bei fest gewählten Knoten x_k , $k = 0, \dots, N$, Exaktheitsgrad N erreicht werden kann. Es ist daher zu vermuten, daß durch optimale Wahl der $N + 1$ Knoten, der Exaktheitsgrad $2N + 1$

erreicht werden kann. Ein noch höherer Exaktheitsgrad kann nicht erreicht werden, wie das Gegenbeispiel

$$p(x) = \prod_{k=0}^N (x - x_k)^2$$

vom Grad $2N + 2$ zeigt. Es ist nichtnegativ und verschwindet nicht identisch, daher gilt wegen $\alpha(x) > 0$ für alle $x \in [a, b]$ auch $I_\alpha[p] > 0$. Andererseits gilt aber $Q_N[p] = 0$ für beliebige Wahl der Stützstellen x_k , da diese gerade die Nullstellen von p sind. Also beträgt der maximale Exaktheitsgrad von $Q_N[f]$ höchstens $2N + 1$.

Der folgende Satz liefert ein hinreichendes Kriterium für den Exaktheitsgrad $2N + 1$.

Satz 6.3.1 (Gauss Quadratur)

Seien α eine stetige nichtnegative Gewichtungsfunktion und q ein nichttriviales Polynom vom Grad $N + 1$ mit

$$\int_a^b x^k \alpha(x) q(x) dx = 0 \quad (0 \leq k \leq N), \quad (6.11)$$

d.h. q ist α -orthogonal zu allen Polynomen vom Höchstgrad N . Seien x_0, x_1, \dots, x_N die Nullstellen von q . Dann ist die Quadraturformel

$$Q_N[f] = \sum_{k=0}^N w_k f(x_k) \quad \text{mit} \quad w_k = \int_a^b \alpha(x) L_k(x) dx \quad (6.12)$$

exakt für alle Polynome vom Höchstgrad $2N + 1$.

Beweis: Sei f ein Polynom vom Grad $2N + 1$. Division mit Rest von f durch q liefert $f = qp + r$ mit Polynomen p und r vom Höchstgrad N . Nach Definition von q gilt $f(x_k) = r(x_k)$, $k = 0, \dots, N$. Weiter gilt wegen (6.11)

$$I_\alpha[f] = \int_a^b \alpha(x) q(x) p(x) + \alpha(x) r(x) dx = \int_a^b \alpha(x) r(x) dx.$$

Analog zu Satz 6.1.3 kann gezeigt werden, daß die Quadraturformel (6.12) exakt ist für alle Polynome bis zum Höchstgrad N , so daß

$$\int_a^b \alpha(x) r(x) dx = \sum_{k=0}^N w_k r(x_k) = \sum_{k=0}^N w_k f(x_k)$$

gilt, woraus die Behauptung folgt. □

Gelingt es, α -orthogonale Polynome zu konstruieren, so besitzen die Nullstellen dieser Polynome und die Gewichte weitere schöne Eigenschaften.

Satz 6.3.2

Sei α eine positive und stetige Gewichtsfunktion und q ein Polynom $N+1$ -ten Grades mit (6.11). Für die Nullstellen x_0, \dots, x_N von q und die Gewichte w_0, \dots, w_N gemäß (6.12) gelten folgende Aussagen:

(i) $x_k \in (a, b)$ für alle $k = 0, \dots, N$ und die Nullstellen sind reell und paarweise verschieden.

(ii) $w_k > 0$ für alle $k = 0, \dots, N$ und $\sum_{k=0}^N w_k = \int_a^b \alpha(x) dx$.

Beweis:

(i) Wir zeigen, daß q mindestens $N+1$ -mal das Vorzeichen in (a, b) wechselt. Da q stetig ist, muß es nach dem Zwischenwertsatz mindestens (und nach dem Fundamentalsatz der Algebra genau) $N+1$ reelle Nullstellen von q in (a, b) geben.

Aus (6.11) folgt für $k = 1$, daß

$$\int_a^b \alpha(x) q(x) dx = 0.$$

Da $\alpha(x) > 0$ in $[a, b]$ gilt, muß q mindestens einmal das Vorzeichen in (a, b) wechseln.

Angenommen, q wechselt das Vorzeichen lediglich $r \leq N$ mal in (a, b) . Dann gibt es Punkte $a = t_0 < t_1 < \dots < t_r < t_{r+1} = b$, so daß q auf jedem Intervall (t_i, t_{i+1}) , $i = 0, \dots, r$, ein einheitliches Vorzeichen besitzt. Das Polynom

$$p(x) = \prod_{i=1}^r (x - t_i)$$

besitzt dieselbe Eigenschaft und daher gilt

$$\int_a^b p(x) \alpha(x) q(x) dx \neq 0.$$

Dies ist ein Widerspruch zu (6.11), da p ein Polynom vom Grad $r \leq N$ ist.

(ii) Definiere $p_k(x) = q(x)/(x - x_k)$ für $k \in \{0, \dots, N\}$. Nach (i) ist x_k einfache Nullstelle von q . Dann ist p_k^2 ein nichtnegatives Polynom mit Höchstgrad $2N$, welches nicht identisch verschwindet. Da die Gauss'sche Quadraturformel exakt ist für p_k folgt

$$0 < \int_a^b \alpha(x) p_k(x)^2 dx = \sum_{j=0}^N w_j p_k(x_j)^2 = w_k p_k(x_k)^2$$

und somit $w_k > 0$, wobei hier ausgenutzt wurde, daß $p(x_j) = q(x_j)/(x_j - x_k) = 0$ für $j \neq k$ gilt.

Für das Polynom 0-ten Grades $f(x) \equiv 1$ liefert die Gauss Quadratur

$$\int_a^b \alpha(x) dx = \sum_{k=0}^N w_k,$$

womit alles gezeigt ist. □

Nach Satz 6.3.1 gilt es, α -orthogonale Polynome q mit (6.11) zu konstruieren. Dies kann wie im Abschnitt über das konjugierte Gradientenverfahren mit Hilfe des **Gram-Schmidtschen Orthogonalisierungsverfahrens** bzgl. des Skalarprodukts

$$\langle f, g \rangle_\alpha := \int_a^b \alpha(x) f(x) g(x) dx$$

geschehen. Die Polynome $\{1, x, x^2, \dots, x^k, \dots\}$ sind linear unabhängig, so daß α -orthogonale Polynome durch die Vorschrift

$$q_k = x^k - \sum_{j=0}^{k-1} \frac{\langle q_j, x^k \rangle_\alpha}{\langle q_j, q_j \rangle_\alpha} q_j, \quad k = 1, 2, \dots,$$

berechnet werden können.

Wir beschränken uns zur Illustration auf das Intervall $[a, b] = [-1, 1]$. Dies ist keine wirkliche Einschränkung, da das Integral (6.1) durch die lineare Transformation $x(t) = (a(1-t) + b(1+t))/2$ auf das Intervall $[-1, 1]$ transformiert werden kann:

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f((a(1-t) + b(1+t))/2) dt$$

Ausserdem betrachten wir nur den Fall $\alpha \equiv 1$. Für andere Gewichtsfunktionen ergeben sich andere α -orthogonale Polynome und damit auch andere Quadraturformeln.

Beispiel 6.3.3

Startend mit $q_0(x) \equiv 1$ liefert das Gram-Schmidtsche Orthogonalisierungsverfahren:

$$\begin{aligned} q_1(x) &= x - \frac{\int_{-1}^1 1 \cdot x dx}{\int_{-1}^1 1^2 dx} \cdot 1 = x, \\ q_2(x) &= x^2 - \frac{\int_{-1}^1 1 \cdot x^2 dx}{\int_{-1}^1 1^2 dx} \cdot 1 - \frac{\int_{-1}^1 x \cdot x^2 dx}{\int_{-1}^1 x^2 dx} \cdot x = x^2 - \frac{1}{3}. \end{aligned}$$

Jedes Vielfache ungleich Null dieser Polynome ist ebenfalls zulässig, so daß noch eine Normierung vorgenommen werden könnte.

Wir konstruieren die dazu passende Gauss'sche Quadraturformel für $N = 1$. Dazu benötigen wir die Nullstellen des Polynoms $q_{N+1} = q_2$, welche durch

$$x_0 = -\sqrt{\frac{1}{3}}, \quad x_1 = \sqrt{\frac{1}{3}}$$

gegeben sind. Daraus ergeben sich folgende Gewichte:

$$\begin{aligned} w_0 &= \int_{-1}^1 L_0(x) dx \\ &= \int_{-1}^1 \frac{x - x_1}{x_0 - x_1} dx \\ &= 1, \\ w_1 &= \int_{-1}^1 L_1(x) dx \\ &= \int_{-1}^1 \frac{x - x_0}{x_1 - x_0} dx \\ &= 1. \end{aligned}$$

Die Gauss'sche Quadraturformel für $N = 1$ und $[a, b] = [-1, 1]$ lautet also

$$\int_{-1}^1 f(x) dx \approx w_0 f(x_0) + w_1 f(x_1) = f(-\sqrt{1/3}) + f(\sqrt{1/3}).$$

Allgemein gilt folgender Satz zur Konstruktion α -orthogonaler Polynome.

Satz 6.3.4 (α -orthogonale Polynome)

Die durch die Rekursion

$$\begin{aligned} q_0(x) &= 1, \\ q_1(x) &= x - a_1, \\ q_k(x) &= (x - a_k)q_{k-1}(x) - b_k q_{k-2}(x), \quad k = 2, 3, \dots, \end{aligned}$$

mit

$$\begin{aligned} a_k &= \frac{\langle x q_{k-1}, q_{k-1} \rangle_\alpha}{\langle q_{k-1}, q_{k-1} \rangle_\alpha}, \\ b_k &= \frac{\langle x q_{k-1}, q_{k-2} \rangle_\alpha}{\langle q_{k-2}, q_{k-2} \rangle_\alpha} \end{aligned}$$

definierten Polynome sind α -orthogonal.

Beweis: Aus der Rekursion ergibt sich induktiv, daß q_k ein Polynom vom Grad k ist und somit nicht das Nullpolynom ist. Daher sind die Nenner in a_k und b_k ungleich Null.

Wir zeigen induktiv, daß $\langle q_k, q_i \rangle_\alpha = 0$ für alle $i < k$ gilt.

Für $k = 0$ ist nichts zu zeigen. Für $k = 1$ gilt

$$\langle q_1, q_0 \rangle_\alpha = \langle x - a_1, q_0 \rangle_\alpha = \langle xq_0, q_0 \rangle_\alpha - a_1 \langle q_0, q_0 \rangle_\alpha = 0.$$

Sei die Behauptung gezeigt für $k \geq 2$. Dann gilt

$$\begin{aligned} \langle q_{k+1}, q_k \rangle_\alpha &= \langle xq_k, q_k \rangle_\alpha - a_{k+1} \langle q_k, q_k \rangle_\alpha - b_{k+1} \underbrace{\langle q_{k-1}, q_k \rangle_\alpha}_{=0} = 0, \\ \langle q_{k+1}, q_{k-1} \rangle_\alpha &= \langle xq_k, q_{k-1} \rangle_\alpha - a_{k+1} \underbrace{\langle q_k, q_{k-1} \rangle_\alpha}_{=0} - b_{k+1} \langle q_{k-1}, q_{k-1} \rangle_\alpha = 0. \end{aligned}$$

Für $i = 1, \dots, k-2$ folgt

$$\begin{aligned} \langle q_{k+1}, q_i \rangle_\alpha &= \langle xq_k, q_i \rangle_\alpha - a_{k+1} \langle q_k, q_i \rangle_\alpha - b_{k+1} \langle q_{k-1}, q_i \rangle_\alpha \\ &= \langle q_k, xq_i \rangle_\alpha \\ &= \langle q_k, q_{i+1} + a_{i+1}q_i + b_{i+1}q_{i-1} \rangle_\alpha \\ &= 0, \end{aligned}$$

wobei die Rekursionsgleichung ausgenutzt wurde, um xq_i darzustellen. Schließlich gilt für $i = 0$

$$\begin{aligned} \langle q_{k+1}, q_0 \rangle_\alpha &= \langle xq_k, q_0 \rangle_\alpha - a_{k+1} \langle q_k, q_0 \rangle_\alpha - b_{k+1} \langle q_{k-1}, q_0 \rangle_\alpha \\ &= \langle q_k, xq_0 \rangle_\alpha \\ &= \langle q_k, q_1 + a_1q_0 \rangle_\alpha \\ &= 0, \end{aligned}$$

womit die Behauptung gezeigt ist. □

Für $[a, b] = [-1, 1]$ und $\alpha \equiv 1$ ergeben sich aus der Rekursion die sogenannten **Legendre-Polynome**, deren erste Vertreter wie folgt lauten:

$$\begin{aligned} q_0(x) &= 1, \\ q_1(x) &= x, \\ q_2(x) &= x^2 - \frac{1}{3}, \\ q_3(x) &= x^3 - \frac{3}{5}x, \\ q_4(x) &= x^4 - \frac{6}{7}x^2 + \frac{3}{35}. \end{aligned}$$

Tabelle 6.3 faßt Knoten und Gewichte von einigen Gauss-Quadraturformeln für $\alpha \equiv 1$ und $[a, b] = [-1, 1]$ zusammen. Zu beachten ist, daß die angegebenen Stützstellen und

Koeffizienten ausschließlich für das Intervall $[-1, 1]$ und die Gewichtsfunktion $\alpha \equiv 1$ gültig sind.

Tabelle 6.3: Einige Gauss Quadraturformeln für $\alpha \equiv 1$ und $[a, b] = [-1, 1]$.

N	Knoten x_i	Gewichte A_i
0	$x_0 = 0$	$w_0 = 2$
1	$x_0 = -\sqrt{1/3}$ $x_1 = +\sqrt{1/3}$	$w_0 = 1$ $w_1 = 1$
2	$x_0 = -\sqrt{3/5}$ $x_1 = 0$ $x_2 = +\sqrt{3/5}$	$w_0 = 5/9$ $w_1 = 8/9$ $w_2 = 5/9$
3	$x_0 = -\sqrt{\frac{3}{7} + \frac{1}{7}\sqrt{\frac{24}{5}}}$ $x_1 = -\sqrt{\frac{3}{7} - \frac{1}{7}\sqrt{\frac{24}{5}}}$ $x_2 = +\sqrt{\frac{3}{7} - \frac{1}{7}\sqrt{\frac{24}{5}}}$ $x_3 = +\sqrt{\frac{3}{7} + \frac{1}{7}\sqrt{\frac{24}{5}}}$	$w_0 = \frac{18-\sqrt{30}}{36}$ $w_1 = \frac{18+\sqrt{30}}{36}$ $w_2 = \frac{18+\sqrt{30}}{36}$ $w_3 = \frac{18-\sqrt{30}}{36}$

Wir betrachten noch ein Beispiel.

Beispiel 6.3.5

Approximiere das Integral

$$I = \int_{-2}^2 \exp(-t^2) dt.$$

Da das Intervall auf $[a, b] := [-2, 2]$ und nicht auf $[-1, 1]$ definiert ist, muss es zunächst auf $[-1, 1]$ transformiert werden. Dies wird erreicht durch die lineare Transformation

$$t(x) = \frac{a(1-x) + b(1+x)}{2} = \frac{-2(1-x) + 2(1+x)}{2} = 2x \quad \Rightarrow \quad dt = 2dx.$$

Transformation des Integrals:

$$\int_{-2}^2 \exp(-t^2) dt = \int_{-1}^1 \exp(-(2x)^2) \cdot 2dx = \int_{-1}^1 2 \exp(-4x^2) dx.$$

Anwendung der Formeln:

$$\int_{-2}^2 \exp(-t^2) dt = \int_{-1}^1 \underbrace{2 \exp(-4x^2)}_{=f(x)} dx \approx \sum_{k=0}^N w_k f(x_k).$$

Für $N = 1$ folgt

$$\begin{aligned}
 \int_{-2}^2 \exp(-t^2) dt &= \int_{-1}^1 \underbrace{2 \exp(-4x^2)}_{=f(x)} dx \\
 &\approx w_0 f(x_0) + w_1 f(x_1) \\
 &= 1 \cdot f(-\sqrt{1/3}) + 1 \cdot f(\sqrt{1/3}) \\
 &= 2 \exp(-4/3) + 2 \exp(-4/3) \\
 &= 4 \exp(-4/3) \approx 1.054388553.
 \end{aligned}$$

Für $N = 2$ erhalten wir den Wert 1.979373230 und für $N = 3$ 1.714546068. Der exakte Wert des Integrals beträgt ungefähr $I \approx 1.764162782$.

Bemerkung 6.3.6

Wie bereits erwähnt, ergeben sich für verschiedene Intervalle $[a, b]$ und Gewichtsfunktionen α verschiedene Gauss-Quadraturformeln.

- Für die Gewichtsfunktion $\alpha(x) = \frac{1}{\sqrt{1-x^2}}$ und $[a, b] = [-1, 1]$ ergeben sich die α -orthogonalen Tschebyscheff-Polynome.
- Für die Gewichtsfunktion $\alpha(x) = \exp(-x)$ und $[a, b] = [0, \infty)$ ergeben sich die sogenannten **Laguerre-Polynome**, deren erste Glieder wie folgt lauten:

$$1, \quad 1 - x, \quad 2 - 4x + x^2, \quad 6 - 18x + 9x^2 - x^3$$

- Für die Gewichtsfunktion $\alpha(x) = \exp(-x^2)$ und $[a, b] = (-\infty, \infty)$ ergeben sich die sogenannten **Hermite-Polynome**, deren erste Glieder wie folgt lauten:

$$1, \quad x, \quad x^2 - \frac{1}{2}, \quad x^3 - \frac{3}{2}x.$$

Abschließend zeigen wir noch ein Konvergenzresultat für die Gauss Quadratur für den Fall $N \rightarrow \infty$.

Satz 6.3.7

Sei f stetig in $[a, b]$. Dann gilt

$$\lim_{N \rightarrow \infty} Q_N[f] = I[f]$$

für die Gauss Quadratur.

Beweis: Sei $\varepsilon > 0$. Der Approximationssatz von Weierstrass besagt, daß die Menge der Polynome dicht in der Menge der stetigen Funktionen liegt, d.h. zu jedem $\varepsilon > 0$ gibt es ein Polynom p mit

$$\|f - p\|_\infty < \varepsilon.$$

Sei r der Grad des Polynoms p . Für jedes N mit $2N+1 \geq r$ integriert die Quadraturformel $Q_N[p]$ p exakt. Es folgt

$$\begin{aligned}
 & \left| \int_a^b \alpha(x) f(x) dx - Q_N[f] \right| \\
 & \leq \left| \int_a^b \alpha(x) f(x) dx - \int_a^b \alpha(x) p(x) dx \right| + |Q_N[p] - Q_N[f]| \\
 & \leq \int_a^b \alpha(x) |f(x) - p(x)| dx + \sum_{k=0}^N w_k |p(x_k) - f(x_k)| \\
 & \leq \varepsilon \int_a^b \alpha(x) dx + \varepsilon \sum_{k=0}^N w_k \\
 & = 2\varepsilon \int_a^b \alpha(x) dx,
 \end{aligned}$$

wobei im letzten Schritt Satz 6.3.2, (ii), ausgenutzt wurde. Da $\varepsilon > 0$ beliebig war, folgt die Behauptung. \square

6.4 Adaptive Verfahren am Beispiel der Simpson-Regel

In den vorangehenden Abschnitten wurden stets eine feste Unterteilung des Intervalls $[a, b]$ gewählt. In diesem Abschnitt wird diese Unterteilung nicht a priori festgelegt, sondern **adaptiv** in Abhängigkeit vom Approximationsfehler gewählt. Der Grund hierfür ist, daß in einigen Abschnitten von $[a, b]$ mitunter (z.B. wenn f dort stark variiert) eine sehr feine Unterteilung notwendig ist, um eine gute Approximation zu erreichen, während in anderen Bereichen (z.B. weil f sich dort kaum ändert) eine grobe Unterteilung ausreicht, vgl. Abbildung 6.6.

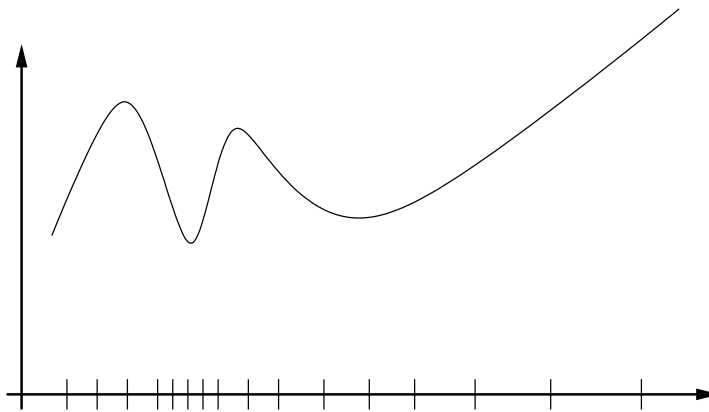


Abbildung 6.6: Idee eines adaptiven Verfahrens: Die Schrittweite wird lokal adaptiv an die Eigenschaften von f angepasst.

Zur Demonstration wählen wir die

elementare Simpson-Regel:

$$\int_a^b f(x)dx \approx S(a, b) := \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right].$$

Falls f in $[a, b]$ 4-mal stetig differenzierbar ist, dann besitzt die Simpson-Regel folgende Fehlerdarstellung:

Fehler:

$$I[f] - S(a, b) = -\frac{1}{90} \left(\frac{b-a}{2} \right)^5 f^{(4)}(\xi) \quad (\xi \in (a, b)).$$

Die adaptive Simpson-Regel basiert auf einer Methode zur numerischen Schätzung des Interpolationsfehlers:

Adaptive Simpson-Regel::

- (i) Seien das Intervall $[a, b]$ und eine Toleranz $\varepsilon > 0$ gegeben.
- (ii) Wende die Simpson-Regel auf $[a, b]$ an und berechne $S^{(1)} := S(a, b)$.
- (iii) Wende die elementare Simpsonregel auf $[a, c]$ und $[c, b]$ mit $c = (a+b)/2$ an und berechne $S^{(2)} := S(a, c) + S(c, b)$ (**zusammengesetzte Simpson-Regel!**).
- (iv) Falls die Ungleichung

$$|S^{(2)} - S^{(1)}| < 15\varepsilon \tag{6.13}$$

erfüllt ist, STOP (der Approximationsfehler ist hinreichend klein) und verwende

$$S^{(2)} + \frac{1}{15} (S^{(2)} - S^{(1)}) \tag{6.14}$$

als Approximation an $\int_a^b f(x)dx$.

Falls die Ungleichung **nicht** erfüllt ist:

- Teile $[a, b]$ in zwei Teilintervalle $[a, c]$ und $[c, b]$ auf,
- Wende diesen Algorithmus rekursiv auf $[a, c]$ **und** $[c, b]$ mit ε ersetzt durch $\varepsilon/2$ an.

Hintergrund der Fehlerschätzung (6.13):

Seien $h = b - a$ und $C := f^{(4)}(c)$ ¹. Mit Hilfe der Fehlerformel der elementaren Simpson-Regel folgt

$$E^{(1)} := I[f] - S^{(1)} = -\frac{1}{90} \left(\frac{h}{2}\right)^5 C + \mathcal{O}(h^6) \quad (6.15)$$

$$\begin{aligned} E^{(2)} := I[f] - S^{(2)} &= -\frac{1}{90} \left(\frac{h/2}{2}\right)^5 C - \frac{1}{90} \left(\frac{h/2}{2}\right)^5 C + \mathcal{O}(h^6) \\ &= \frac{1}{16} \left(-\frac{1}{90} \left(\frac{h}{2}\right)^5\right) C + \mathcal{O}(h^6). \end{aligned} \quad (6.16)$$

Ein Vergleich der rechten Seiten liefert

$$\begin{aligned} I[f] - S^{(1)} &\approx 16E^{(2)}, \\ I[f] - S^{(2)} &= E^{(2)}. \end{aligned}$$

Subtraktion der zweiten von der ersten Gleichung liefert

$$S^{(2)} - S^{(1)} \approx 15E^{(2)},$$

und daher

$$I[f] - S^{(2)} = E^{(2)} \approx \frac{1}{15} (S^{(2)} - S^{(1)}).$$

Deshalb kann der Wert

$$\frac{1}{15} (S^{(2)} - S^{(1)}) \quad (6.17)$$

als Approximation des Fehlers $I[f] - S^{(2)}$ verwendet werden. Beachte, daß $E^{(2)}$ nicht bekannt ist, aber wir können (6.17) berechnen. Gleichung (6.13) sichert (approximativ), daß $E^{(2)}$ kleiner als ε ist. Es ist wichtig zu bemerken, daß die Approximation in (6.14) den Fehler $\mathcal{O}(h^6)$ hat. Dies folgt aus den Fehlerdarstellungen (6.15) und (6.16) (multipliziere die erste Gleichung mit $-1/15$, die zweite mit $16/15$ und addiere beide Gleichungen).

Die Division von ε durch zwei in Schritt (iv) soll garantieren, daß der Fehler in jedem der zwei Teilintervalle kleiner als $\varepsilon/2$ verbleibt, so daß der Fehler des gesamten Intervalls kleiner als $\varepsilon/2 + \varepsilon/2 = \varepsilon$ ist.

6.5 Mehrdimensionale Integrale

Ziel dieses Abschnitts ist es, n-dimensionale Integrale der Form

$$I[f] = \int_{a_1}^{b_1} \cdots \int_{a_n}^{b_n} f(x_1, \dots, x_n) dx_n \cdots dx_1$$

¹Falls f 5-mal stetig differenzierbar ist, liefert Taylorentwicklung $f^{(4)}(\xi) = f^{(4)}(c) + \frac{1}{2}f^{(5)}(\zeta)(\xi - c) = f^{(4)}(c) + \mathcal{O}(h)$ für jedes $\xi \in (a, b)$ und $\zeta \in (c, \xi)$.

zu approximieren. Zur Vereinfachung sei im folgenden $n = 2$ gewählt, d.h.

$$I[f] = \int_a^b \int_c^d f(x, y) dx dy. \quad (6.18)$$

Man kann leicht Quadraturformeln entwickeln, indem die Quadraturformeln für eindimensionale Integrale verwendet werden. Dies soll am Beispiel der Trapezregel verdeutlicht werden. Zunächst wird das innere Integral

$$\int_c^d f(x, y) dx$$

für festes y durch die zusammengesetzte Trapezregel approximiert:

$$\int_c^d f(x, y) dx \approx h_1 \left(\frac{f(x_0)}{2} + \sum_{i=1}^{n_1-1} f(x_i, y) + \frac{f(x_{n_1})}{2} \right) =: \sum_{i=0}^{n_1} A_i f(x_i, y)$$

wobei $h_1 = (d - c)/n_1$ und $x_i = c + ih_1$, $i = 0, 1, \dots, n_1$. Dann wird diese Approximation in das ursprüngliche Integral eingesetzt und die iterierte Trapezregel wird angewendet auf das äußere Integral mit $h_2 = (b - a)/n_2$, $y_j = a + jh_2$, $j = 0, 1, \dots, n_2$:

$$\begin{aligned} \int_a^b \int_c^d f(x, y) dx dy &\approx \int_a^b \sum_{i=0}^{n_1} A_i f(x_i, y) dy \\ &= \sum_{i=0}^{n_1} A_i \left(\int_a^b f(x_i, y) dy \right) \\ &\approx \sum_{i=0}^{n_1} A_i \left(\sum_{j=0}^{n_2} B_j f(x_i, y_j) \right) \\ &= \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} A_i B_j f(x_i, y_j). \end{aligned}$$

Entsprechend kann für $n > 2$ verfahren werden. Ebenso können auch Newton-Cotes-Formeln oder zusammengesetzte Verfahren verwendet werden. Es gibt auch Verfahren, die auf der Gauss-Quadratur basieren.

Kapitel 7

Trigonometrische Approximation und Schnelle Fouriertransformation (FFT)

Ist – etwa auf Grund physikalischer Gesetze – bekannt, daß der einer Messung zu Grunde liegende Vorgang periodisch ist, werden konsequenterweise periodische Ansatzfunktionen – etwa Sinus und Cosinus – an Stelle von Polynomen zur Interpolation der Meßwerte verwendet. Im folgenden beschränken wir uns auf 2π -periodische Funktionen $f : \mathbb{R} \rightarrow \mathbb{C}$ mit

$$f(x + 2\pi) = f(x) \quad \text{für alle } x \in \mathbb{R}.$$

Da f dann bereits durch die Werte in $[0, 2\pi]$ vollständig definiert ist, genügt es, das Intervall $[0, 2\pi]$ zu betrachten. Die Einschränkung auf Periodenlänge 2π und das Intervall $[0, 2\pi]$ ist hierbei nicht wesentlich, da jede periodische Funktion durch eine lineare Transformation auf dieses Intervall transformiert werden kann.

7.1 Trigonometrische Approximation

Zur Approximation periodischer Funktionen werden sogenannte trigonometrische Polynome verwendet.

Definition 7.1.1 (komplexes trigonometrisches Polynom)

Die Funktion

$$t(x) = \sum_{k=-n}^n \alpha_k \exp(ikx), \quad \alpha_k \in \mathbb{C}, \quad (7.1)$$

heißt (komplexes) trigonometrisches Polynom vom Grad n . Die Menge

$$\mathcal{T}_n := \text{span}\{\exp(ikx) \mid k = 0, \pm 1, \dots, \pm n\}$$

bezeichnet den Vektorraum aller trigonometrischen Polynome vom Grad n .

Beachte, daß hier und im folgenden i für die komplexe Zahl $i \in \mathbb{C}$ mit $i^2 = -1$ steht. Wir werden gelegentlich die **Euler-Formel**

$$\exp(ix) = \cos(x) + i \sin(x) \quad \forall x \in \mathbb{R} \quad (7.2)$$

und die daraus für $x \in \mathbb{R}$ resultierenden Formeln

$$\cos(x) = \frac{1}{2} (\exp(ix) + \exp(-ix)), \quad \sin(x) = \frac{1}{2i} (\exp(ix) - \exp(-ix)) \quad (7.3)$$

verwenden. Mit Hilfe der Euler-Formel (7.2) lässt sich das komplexe trigonometrische Polynom auch schreiben als

$$\begin{aligned} t(x) &= \sum_{k=-n}^n \alpha_k \exp(ikx) \\ &= \sum_{k=-n}^n \alpha_k \cos(kx) + i\alpha_k \sin(kx) \\ &= \alpha_0 + \sum_{k=1}^n (\alpha_k + \alpha_{-k}) \cos(kx) + i(\alpha_k - \alpha_{-k}) \sin(kx). \end{aligned}$$

Mit den Setzungen $\alpha_0 = a_0/2$, $a_k = \alpha_k + \alpha_{-k}$ und $b_k = i(\alpha_k - \alpha_{-k})$ folgt

$$t(x) = \frac{a_0}{2} + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)). \quad (7.4)$$

Umgekehrt kann aus der Darstellung (7.4) mit Hilfe von (7.3) ein trigonometrisches Polynom der Form (7.1) hergeleitet werden:

$$\begin{aligned} t(x) &= \frac{a_0}{2} + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)) \\ &= \frac{a_0}{2} + \sum_{k=1}^n \left(\frac{a_k}{2} (\exp(ikx) + \exp(-ikx)) + \frac{b_k}{2i} (\exp(ikx) - \exp(-ikx)) \right) \\ &= \frac{a_0}{2} + \sum_{k=1}^n \left(\frac{1}{2} (a_k - ib_k) \exp(ikx) + \frac{1}{2} (a_k + ib_k) \exp(-ikx) \right) \\ &= \underbrace{\frac{a_0}{2}}_{=: \alpha_0} + \underbrace{\sum_{k=1}^n \frac{1}{2} (a_k - ib_k) \exp(ikx)}_{=: \alpha_k, k=1, \dots, n} + \underbrace{\sum_{k=-n}^{-1} \frac{1}{2} (a_{-k} + ib_{-k}) \exp(ikx)}_{=: \alpha_k, k=-n, \dots, -1} \\ &= \sum_{k=-n}^n \alpha_k \exp(ikx). \end{aligned}$$

Falls $\alpha_k = \overline{\alpha_{-k}}$ für $k = 0, \dots, n$ gilt, so gilt $a_k = 2\operatorname{Re}(\alpha_k)$ und $b_k = -2\operatorname{Im}(\alpha_k)$ in (7.4) und sämtliche Koeffizienten in (7.4) sind reell. In diesem Fall spricht man vom reellen trigonometrischen Polynom.

Definition 7.1.2 (reelles trigonometrisches Polynom)

Die Funktion

$$t(x) = \frac{a_0}{2} + \sum_{k=1}^n (a_k \cos(kx) + b_k \sin(kx)), \quad a_k, b_k \in \mathbb{R},$$

heißt (reelles) **trigonometrisches Polynom vom Grad n** .

Wegen der 2π -Periodizität von Sinus und Cosinus sind trigonometrische Polynome 2π -periodisch, d.h. es gilt $t(x + 2\pi) = t(x)$, und eignen sich daher zur Approximation bzw. Interpolation von 2π -periodischen Funktionen.

Der folgende Satz besagt, daß die elementaren trigonometrischen Polynome $\exp(ikx)$, $k \in \mathbb{Z}$, ein Orthonormalsystem bzgl. des auf dem Raum der 2π -periodischen L_2 -integrierbaren Funktionen

$$L_2([0, 2\pi]) := \{f : [0, 2\pi] \rightarrow \mathbb{C} \mid \int_0^{2\pi} |f(x)|^2 dx < \infty\}$$

definierten Skalarprodukts

$$\langle f, g \rangle := \frac{1}{2\pi} \int_0^{2\pi} f(x) \overline{g(x)} dx \quad (f, g \in L_2([0, 2\pi])). \quad (7.5)$$

Das Skalarprodukt induziert eine Norm auf $L_2([0, 2\pi])$ gemäß $\|f\|_{L_2} := \sqrt{\langle f, f \rangle}$.

Satz 7.1.3

Die Funktionen $e_k(x) := \exp(ikx)$, $k \in \mathbb{Z}$, bilden ein Orthonormalsystem auf $L_2([0, 2\pi])$ bezüglich des Skalarprodukts (7.5), d.h. für $k, j \in \mathbb{Z}$ gilt

$$\langle e_k, e_j \rangle = \begin{cases} 1, & \text{falls } k = j, \\ 0, & \text{sonst} \end{cases}$$

Beweis: $k = j$ impliziert

$$\langle e_k, e_k \rangle = \frac{1}{2\pi} \int_0^{2\pi} \exp(ikx) \overline{\exp(ikx)} dx = \frac{1}{2\pi} \int_0^{2\pi} \exp(i0x) dx = \frac{1}{2\pi} \int_0^{2\pi} 1 dx = 1.$$

Für $k \neq j$ gilt

$$\begin{aligned} \langle e_k, e_j \rangle &= \frac{1}{2\pi} \int_0^{2\pi} \exp(ikx) \overline{\exp(ijx)} dx \\ &= \frac{1}{2\pi} \int_0^{2\pi} \exp(i(k-j)x) dx \\ &= \frac{1}{2\pi} \frac{1}{i(k-j)} \exp(i(k-j)x) \Big|_0^{2\pi} = 0. \end{aligned}$$

□

Wir untersuchen nun das folgende Approximationsproblem.

Problem 7.1.4

Gegeben sei $f \in L_2([0, 2\pi])$. Bestimme ein trigonometrisches Polynom $\hat{t} \in \mathcal{T}_n$ mit

$$\|\hat{t} - f\|_{L_2} \leq \|t - f\|_{L_2} \quad \forall t \in \mathcal{T}_n.$$

Wir erinnern uns an den Projektionssatz 3.1.2. Demnach existiert höchstens eine Bestapproximierende

$$\hat{t}(x) = \sum_{k=-n}^n \hat{\alpha}_k \exp(ikx) \in \mathcal{T}_n, \quad (7.6)$$

und $\hat{t} \in \mathcal{T}_n$ ist Bestapproximierende an f in \mathcal{T}_n genau dann, wenn

$$\langle f - \hat{t}, v \rangle = 0 \quad \forall v \in \mathcal{T}_n.$$

Da die Funktionen $e_k(x) = \exp(ikx)$, $k = 0, \pm 1, \dots, \pm n$, ein Orthonormalsystem und somit eine Basis von \mathcal{T}_n darstellen, folgt, daß $\hat{t} \in \mathcal{T}_n$ genau dann Bestapproximierende an f in \mathcal{T}_n ist, wenn für alle $j = 0, \pm 1, \dots, \pm n$ gilt

$$\begin{aligned} 0 &= \langle f - \hat{t}, e_j \rangle \\ &= \langle f - \sum_{k=-n}^n \hat{\alpha}_k e_k, e_j \rangle \\ &= \langle f, e_j \rangle - \sum_{k=-n}^n \hat{\alpha}_k \langle e_k, e_j \rangle \\ &= \langle f, e_j \rangle - \hat{\alpha}_j. \end{aligned}$$

Dieses Gleichungssystem besitzt die eindeutige Lösung

$$\hat{\alpha}_j = \langle f, e_j \rangle = \frac{1}{2\pi} \int_0^{2\pi} f(x) \overline{\exp(ijx)} dx, \quad j = 0, \pm 1, \dots, \pm n, \quad (7.7)$$

und folglich haben wir bewiesen:

Satz 7.1.5 (Bestapproximierende)

Seien $f \in L_2([0, 2\pi])$ und $n \in \mathbb{N}$ gegeben. Das trigonometrische Polynom \hat{t} aus (7.6) mit Koeffizienten (7.7) ist eindeutig bestimmt und löst Problem 7.1.4, d.h. es gilt

$$\|\hat{t} - f\|_{L_2} \leq \|t - f\|_{L_2} \quad \forall t \in \mathcal{T}_n.$$

Aus (7.7) erhält man für die Darstellung (7.4) die Koeffizienten

$$\begin{aligned} a_0 &= \frac{1}{\pi} \int_0^{2\pi} f(x) dx, \\ a_k &= \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(kx) dx, \quad k = 1, \dots, n, \\ b_k &= \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(kx) dx, \quad k = 1, \dots, n. \end{aligned}$$

Ist f reellwertig, so lassen sich demnach sämtliche Rechnungen im reellen durchführen.

Beispiel 7.1.6

Die 2π -periodische **Sägezahnfunktion**

$$f(x) = x - \pi, \quad x \in [0, 2\pi],$$

soll im Intervall $[0, 2\pi]$ durch trigonometrische Polynome approximiert werden. Als Lösung erhält man

$$t(x) = -2 \sum_{k=1}^n \frac{\sin(kx)}{k}.$$

Abbildung 7.1 zeigt die Lösungen für $n = 1, 2, 3, 4, 20$.

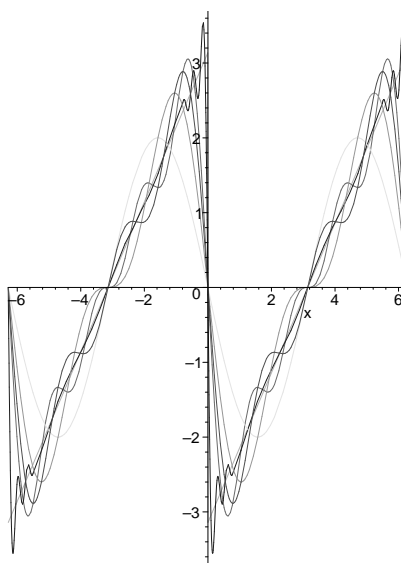


Abbildung 7.1: Approximation der Sägezahnfunktion durch trigonometrische Polynome vom Grad 1, 2, 3, 4 und 20.

Typisch bei der Approximation mit trigonometrischen Funktionen sind die Über- und Unterschwingen an Unstetigkeitsstellen von f , die sich auch durch Erhöhung des Polynomgrades nicht vermeiden lassen. Dieses Phänomen ist als **Gibbs-Phänomen** bekannt und man beobachtet, daß die Höhe der Über- und Unterschwingung in etwa 18 % Prozent der Sprunghöhe beträgt.

Nachdem wir nun wissen, wie die Bestapproximierende aussieht, stellt sich die Frage, was beim Grenzübergang $n \rightarrow \infty$ passiert. Formal kann einer Funktion $f \in L_2([0, 2\pi])$ die sogenannte **Fourierreihe**

$$F_f(x) := \sum_{k=-\infty}^{\infty} \hat{\alpha}_k \exp(ikx)$$

mit **Fourierkoeffizienten**

$$\hat{\alpha}_j = \langle f, e_j \rangle = \frac{1}{2\pi} \int_0^{2\pi} f(x) \overline{\exp(ijx)} dx, \quad j \in \mathbb{Z},$$

zugeordnet werden. Die Reihe konvergiert genau dann, wenn die Folge $\{S_f^n\}$ der Partialsummen

$$S_f^n(x) := \sum_{k=-n}^n \hat{\alpha}_k \exp(ikx)$$

konvergiert. Interessant sind folgende Fragen:

- Unter welchen Voraussetzungen an f konvergiert die Fourierreihe F_f punktweise gegen f ?
- Unter welchen Voraussetzungen an f konvergiert die Fourierreihe F_f gleichmäßig gegen f ?
- Welche weiteren Arten der Konvergenz gibt es?

Diese Fragen sind Gegenstand der Fourieranalyse, auf die wir hier nicht weiter eingehen können. Dem interessierten Leser sei z.B. das Buch „Approximationstheorie“ von Schönhage [Sch71, Kap. 4] empfohlen.

7.2 Diskrete trigonometrische Approximation und Interpolation

Für eine gegebene Funktion f sind die Fourierkoeffizienten $\hat{\alpha}_k$, $k = 0, \pm 1, \dots, \pm n$, in (7.7) i.a. schwer zu berechnen, da die Integrale i.a. nicht analytisch berechenbar sind. Daher bietet es sich an, diese durch ein geeignetes Verfahren zur Approximation von Integralen zu approximieren. Hierzu eignet sich auf Grund ihrer periodischen Struktur insbesondere die zusammengesetzte Trapezregel, die wir für äquidistante Stützstellen

$$x_k = \frac{2\pi k}{N}, \quad k = 0, \dots, N,$$

mit $N \in \mathbb{N}$ anwenden werden. Mit der Abkürzung $g(x) = f(x)\overline{\exp(ikx)}$ erhalten wir die Approximationen

$$\begin{aligned}\hat{\alpha}_k &= \frac{1}{2\pi} \int_0^{2\pi} g(x) dx \\ &\approx \frac{1}{2\pi} \frac{2\pi}{N} \left(\frac{1}{2}g(x_0) + \sum_{j=1}^{N-1} g(x_j) + \frac{1}{2}g(x_N) \right) \\ &= \frac{1}{N} \sum_{j=0}^{N-1} g(x_j) \\ &= \frac{1}{N} \sum_{j=0}^{N-1} f(x_j) \overline{\exp(ikx_j)}\end{aligned}$$

für $k = 0, \pm 1, \dots, \pm n$. Es wird sich herausstellen, daß diese diskreten Fourierkoeffizienten eine analoge Rolle wie die kontinuierlichen Fourierkoeffizienten bei der Approximation von Funktionen auf dem Gitter $\{x_0, \dots, x_N\}$ spielen. Um dies zu sehen, definieren wir

$$\begin{aligned}\omega(x) &= \exp(ix), \\ \omega_j^k &= \exp(ikx_j) = \omega(x_j)^k, \quad k = 0, \dots, n, \quad j = 0, \dots, N-1.\end{aligned}$$

Analog zu Satz 7.1.3 gilt

Satz 7.2.1

Die Vektoren

$$\omega^k = \begin{pmatrix} \omega_0^k \\ \vdots \\ \omega_{N-1}^k \end{pmatrix}, \quad k = 0, \dots, n-1, \quad (7.8)$$

bilden ein Orthonormalsystem in \mathbb{C}^N bzgl. des Skalarproduktes

$$\langle y, z \rangle := \frac{1}{N} z^* y.$$

Beweis: Für $k \in \{0, \dots, n-1\}$ gilt

$$\langle \omega^k, \omega^k \rangle = \frac{1}{N} \sum_{j=0}^{N-1} \overline{\exp(ikx_j)} \exp(ikx_j) = \frac{1}{N} \sum_{j=0}^{N-1} 1 = 1.$$

Für $k \neq \ell$ gilt

$$\begin{aligned}
 \langle \omega^\ell, \omega^k \rangle &= \frac{1}{N} \sum_{j=0}^{N-1} \overline{\exp(ikx_j)} \exp(i\ell x_j) \\
 &= \frac{1}{N} \sum_{j=0}^{N-1} \exp(i(\ell - k)x_j) \\
 &= \frac{1}{N} \sum_{j=0}^{N-1} \exp(i(\ell - k)2\pi/N)^j \\
 &= \frac{1}{N} \cdot \frac{1 - \exp(i(\ell - k)2\pi/N)^N}{1 - \exp(i(\ell - k)2\pi/N)} \\
 &= \frac{1}{N} \cdot \frac{1 - 1}{1 - \exp(i(\ell - k)2\pi/N)} \\
 &= 0.
 \end{aligned}$$

□

Wir möchten nun gegebene Funktionswerte f_j an den Stützstellen x_j , $j = 0, \dots, N-1$, bestmöglich approximieren. Analog zu Satz 7.1.5 erhalten wir die folgende diskrete Variante:

Satz 7.2.2

Seien f_j , $j = 0, \dots, N-1$, gegebene komplexe Zahlen, $n \leq N$ und

$$\hat{\alpha}_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j \overline{\exp(ikx_j)} = \frac{1}{N} \sum_{j=0}^{N-1} f_j \overline{\omega_j^k}, \quad k = 0, \dots, n-1.$$

Dann ist

$$\hat{t} = \sum_{k=0}^{n-1} \hat{\alpha}_k \omega^k$$

die eindeutig bestimmte Bestapproximierende an $f = (f_0, \dots, f_{N-1})^\top$ in dem durch $\{\omega^k \mid k = 0, \dots, n-1\}$ aufgespannten Vektorraum bzgl. der euklidischen Norm, d.h. es gilt

$$\|f - \hat{t}\|_2 \leq \|f - t\|_2 \quad \forall t = \sum_{k=0}^{n-1} \alpha_k \omega^k.$$

Im Fall $n = N$ gilt $\|f - \hat{t}\|_2 = 0$, d.h.

$$f_j = \sum_{k=0}^{n-1} \hat{\alpha}_k \omega_j^k, \quad j = 0, \dots, N-1.$$

Beweis: Für $\ell = 0, \dots, n-1$ gilt

$$\langle f - \hat{t}, \omega^\ell \rangle = \langle f - \sum_{k=0}^{n-1} \hat{\alpha}_k \omega^k, \omega^\ell \rangle = \langle f, \omega^\ell \rangle - \sum_{k=0}^{n-1} \hat{\alpha}_k \langle \omega^k, \omega^\ell \rangle = \langle f, \omega^\ell \rangle - \hat{\alpha}_\ell = 0.$$

Nach dem Projektionssatz 3.1.2 ist \hat{t} also die eindeutig bestimmte Bestapproximierende an f . Im Fall $n = N$ ist $f \in \text{span}\{\omega^0, \dots, \omega^{n-1}\}$, woraus die Behauptung folgt. \square

Wir betrachten nun den Fall $N = n$ etwas genauer. Wegen

$$\sum_{k=0}^{n-1} \hat{\alpha}_k \exp(ikx_j) = \sum_{k=0}^{n-1} \hat{\alpha}_k \omega_j^k = f_j, \quad j = 0, \dots, N-1,$$

definiert

$$\hat{t}(x) = \sum_{k=0}^{n-1} \hat{\alpha}_k \exp(ikx) \tag{7.9}$$

ein interpolierendes trigonometrisches Polynom mit $\hat{t}(x_j) = f_j$, $j = 0, \dots, N-1$. Definiere für $k = 0, 1, \dots, n-1$

$$A_k = \frac{1}{n} \sum_{j=0}^{n-1} f_j \cos(kx_j), \quad B_k = \frac{1}{n} \sum_{j=0}^{n-1} f_j \sin(kx_j).$$

Damit gilt

$$\begin{aligned} A_0 &= \frac{1}{n} \sum_{j=0}^{n-1} f_j, & B_0 &= 0, \\ A_{n/2} &= \frac{1}{n} \sum_{j=0}^{n-1} (-1)^j f_j, & B_{n/2} &= 0 \quad (\text{falls } n \text{ gerade}), \\ \hat{\alpha}_k &= A_k - iB_k, & k &= 1, \dots, n-1, \\ \hat{\alpha}_{n-k} &= A_k + iB_k, & k &= 1, \dots, n-1. \end{aligned}$$

Weiter folgt für $k = 1, \dots, n-1$

$$\begin{aligned} &\hat{\alpha}_k \exp(ikx) + \hat{\alpha}_{n-k} \exp(i(n-k)x) \\ &= (A_k - iB_k)(\cos(kx) + i\sin(kx)) + (A_k + iB_k)(\cos(kx) - i\sin(kx)) \\ &= 2(A_k \cos(kx) + B_k \sin(kx)). \end{aligned}$$

Zusammen mit dem interpolierenden trigonometrischen Polynom (7.9) erhalten wir daraus folgendes Resultat, welches den Begriff trigonometrische Interpolation rechtfertigt.

Satz 7.2.3 (trigonometrische Interpolation)

Seien f_j , $j = 0, \dots, n-1$, gegebene komplexe Zahlen und

$$\begin{aligned} a_0 &= \frac{1}{n} \sum_{j=0}^{n-1} f_j, \\ a_{n/2} &= \frac{1}{n} \sum_{j=0}^{n-1} (-1)^j f_j \quad \text{für } n \text{ gerade}, \\ a_k &= \frac{2}{n} \sum_{j=0}^{n-1} f_j \cos(kx_j), \quad k \in \mathbb{N}, k < \frac{n}{2}, \\ b_k &= \frac{2}{n} \sum_{j=0}^{n-1} f_j \sin(kx_j), \quad k \in \mathbb{N}, k < \frac{n}{2}. \end{aligned}$$

Setze für $n = 2\ell + 1$ (n ungerade)

$$\hat{t}(x) = a_0 + \sum_{k=1}^{\ell} (a_k \cos(kx) + b_k \sin(kx))$$

bzw. für $n = 2\ell$ (n gerade)

$$\hat{t}(x) = a_0 + \sum_{k=1}^{\ell-1} (a_k \cos(kx) + b_k \sin(kx)) + a_{\ell} \cos(\ell x).$$

Dann gilt $t(x_j) = f_j$ für $j = 0, \dots, n-1$.

7.3 Schnelle Fourier-Transformation

In diesem Abschnitt beschäftigen wir uns mit der schnellen Berechnung der diskreten Fourierkoeffizienten

$$\hat{\alpha}_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j \overline{\exp(ikx_j)} = \frac{1}{N} \sum_{j=0}^{N-1} f_j \overline{\omega_j^k}, \quad k = 0, \dots, N-1,$$

für gegebenes $f = (f_0, \dots, f_{N-1})^\top \in \mathbb{C}^N$ im Fall $N = n$. Die Abbildung $f \mapsto \hat{\alpha} = (\hat{\alpha}_0, \dots, \hat{\alpha}_{N-1})^\top$ wird als **diskrete Fourier-Transformation (DFT)** bezeichnet. In Vektorschreibweise gilt

$$\frac{1}{N} \underbrace{\begin{pmatrix} \overline{\omega_0^0} & \overline{\omega_1^0} & \cdots & \overline{\omega_{N-1}^0} \\ \overline{\omega_0^1} & \overline{\omega_1^1} & \cdots & \overline{\omega_{N-1}^1} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{\omega_0^{N-1}} & \overline{\omega_1^{N-1}} & \cdots & \overline{\omega_{N-1}^{N-1}} \end{pmatrix}}_{=: V^*} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{pmatrix} = \begin{pmatrix} \hat{\alpha}_0 \\ \hat{\alpha}_1 \\ \vdots \\ \hat{\alpha}_{N-1} \end{pmatrix}$$

bzw.

$$\frac{1}{N} V^* f = \hat{\alpha}. \quad (7.10)$$

Die Zeilen der Matrix V^* sind gerade die Vektoren $\overline{\omega^k}$ mit ω^k aus (7.8), welche nach Satz 7.2.1 ein Orthonormalsystem bzgl. des Skalarproduktes $\langle y, z \rangle := \frac{1}{N} z^* y$ bilden. Daher ist die Matrix $\bar{V} := \frac{1}{\sqrt{N}} V$ unitär und es gilt

$$\bar{V}^* \bar{V} = I \quad \Leftrightarrow \quad V^* V = N \cdot I \quad \Leftrightarrow \quad (V^*)^{-1} = \frac{1}{N} V.$$

Es ist daher besonders einfach, die Umkehrabbildung auszurechnen:

$$f = V \hat{\alpha}.$$

Dieser Vorgang wird als **inverse diskrete Fourier-Transformation (IDFT)** bezeichnet.

Der Aufwand zur Berechnung von $\hat{\alpha}$ in (7.10) beträgt $\mathcal{O}(N^2)$ (eine Matrix-Vektor-Multiplikation).

In vielen Anwendungen ist N extrem groß, so daß $\mathcal{O}(N^2)$ Rechenoperationen zu teuer sind.

Es zeigt sich, daß dieser Aufwand mit Hilfe der **schnellen Fourier-Transformation (FFT)** auf $\mathcal{O}(N \log_2 N)$ reduziert werden kann, wenn $N = 2^p$ eine Zweierpotenz ist. In diesem Fall gilt

$$\begin{aligned} \hat{\alpha}_k &= \frac{1}{N} \sum_{j=0}^{N-1} f_j \overline{\exp(ikx_j)} \\ &= \frac{1}{N} \sum_{j=0}^{N/2-1} f_{2j} \overline{\exp(ikx_{2j})} + \frac{1}{N} \sum_{j=0}^{N/2-1} f_{2j+1} \overline{\exp(ikx_{2j+1})} \\ &= \frac{1}{N} \sum_{j=0}^{N/2-1} f_{2j} \exp\left(-i \frac{2\pi j k}{N/2}\right) + \frac{1}{N} \sum_{j=0}^{N/2-1} f_{2j+1} \exp\left(-i \frac{2\pi k 2j}{N}\right) \exp\left(-i \frac{2\pi k}{N}\right) \\ &= \frac{1}{2} \frac{1}{N/2} \sum_{j=0}^{N/2-1} f_{2j} \exp\left(-i \frac{2\pi j k}{N/2}\right) + \frac{1}{2} \frac{1}{N/2} \exp\left(-i \frac{2\pi k}{N}\right) \sum_{j=0}^{N/2-1} f_{2j+1} \exp\left(-i \frac{2\pi k j}{N/2}\right). \end{aligned}$$

Mit $m = N/2$ und $\overline{\omega_1^k} = \exp\left(-i \frac{2\pi k}{N}\right)$ steht dort

$$\hat{\alpha}_k = \frac{1}{2} \left(\underbrace{\frac{1}{m} \sum_{j=0}^{m-1} f_{2j} \exp\left(-i \frac{2\pi j k}{m}\right)}_{=: \beta_k^g} + \overline{\omega_1^k} \underbrace{\frac{1}{m} \sum_{j=0}^{m-1} f_{2j+1} \exp\left(-i \frac{2\pi k j}{m}\right)}_{=: \beta_k^u} \right).$$

Diese Formel läßt sich folgendermaßen interpretieren: Der Term β_k^g ist gerade der diskrete Fourierkoeffizient zu den „geraden“ Stützstellen (x_{2j}, f_{2j}) , $j = 0, \dots, m-1$, während β_k^u der diskrete Fourierkoeffizient zu den „ungeraden“ Stützstellen (x_{2j+1}, f_{2j+1}) , $j =$

$0, \dots, m-1$, ist. Da β_k^g und β_k^u formal dieselbe Struktur wie $\hat{\alpha}_k$ besitzen, kann diese Aufteilung **rekursiv** fortgesetzt werden und somit kann der diskrete Fourierkoeffizient $\hat{\alpha}_k$ rekursiv aus β_k^g und β_k^u berechnet werden.

Berücksichtigt man noch die Identitäten

$$\exp\left(-i\frac{2\pi jk}{m}\right) = \exp\left(-i\frac{2\pi j(k-m)}{m}\right) \quad \text{und} \quad \overline{\omega_1^k} = -\overline{\omega_1^{k-m}},$$

so kann lassen sich die diskreten Fourierkoeffizienten $\hat{\alpha}_k$ für $k = 0, \dots, m-1$ berechnen gemäß

$$\hat{\alpha}_k = \frac{1}{2} \left(\frac{1}{m} \sum_{j=0}^{m-1} f_{2j} \exp\left(-i\frac{2\pi jk}{m}\right) + \overline{\omega_1^k} \frac{1}{m} \sum_{j=0}^{m-1} f_{2j+1} \exp\left(-i\frac{2\pi jk}{m}\right) \right)$$

und für $k = m, \dots, N-1$ gemäß

$$\hat{\alpha}_k = \frac{1}{2} \left(\frac{1}{m} \sum_{j=0}^{m-1} f_{2j} \exp\left(-i\frac{2\pi j(k-m)}{m}\right) - \overline{\omega_1^{k-m}} \frac{1}{m} \sum_{j=0}^{m-1} f_{2j+1} \exp\left(-i\frac{2\pi j(k-m)}{m}\right) \right).$$

Hierbei ist zu beachten, daß die Summanden für $k = m, \dots, N-1$ mit denen für $k = 0, \dots, m-1$ jeweils übereinstimmen und daher nur einmalig berechnet werden müssen.

Ist $N = 2^p$ eine Zweierpotenz, so läßt sich die Berechnung der diskreten Fourierkoeffizienten mit Hilfe des folgenden rekursiven Algorithmus berechnen.

Algorithmus 7.3.1 (Schnelle Fourier-Transformation (FFT))

Gegeben: $N = 2^p$, $f = (f_0, \dots, f_{N-1})^\top$

Output: diskrete Fourierkoeffizienten $\alpha = (\alpha_0, \dots, \alpha_{N-1})^\top$

function $\alpha = FFT(f, N)$

If $N = 1$ **then**

$$\alpha = \frac{1}{N} f$$

else

$$f_g = (f_0, f_2, \dots, f_{N-2})^\top$$

$$f_u = (f_1, f_3, \dots, f_{N-1})^\top$$

$$m = N/2$$

$$\alpha_g = FFT(f_g, m)$$

$$\alpha_u = FFT(f_u, m)$$

$$\alpha_k = \frac{1}{2} \left(\alpha_{g,k} + \overline{\omega_1^k} \alpha_{u,k} \right), \quad k = 0, \dots, m-1$$

$$\alpha_{k+m} = \frac{1}{2} \left(\alpha_{g,k} - \overline{\omega_1^k} \alpha_{u,k} \right), \quad k = 0, \dots, m-1$$

end

Abschließend betrachten wir noch die inverse diskrete Fouriertransformation $f = V\hat{\alpha}$, wobei wir die Symmetrie von V^* ausnutzen. Diese kann man schreiben als

$$f = V\alpha = N \overline{\frac{1}{N} V^* \bar{\alpha}}.$$

Sie kann mit Hilfe der FFT berechnet werden:

Algorithmus 7.3.2 (Schnelle inverse Fourier-Transformation (IFFT))

Gegeben: $N = 2^p$, $\alpha = (\alpha_0, \dots, \alpha_{N-1})^\top$

Output: $f = (f_0, \dots, f_{N-1})^\top$

function $f = \text{IFFT}(\alpha, N)$

$f = N \cdot \overline{\text{FFT}(\bar{\alpha}, N)}.$

Der Aufwand der FFT berechnet sich wie folgt. Sei $N = 2^p$, $p \in \mathbb{N}$, also $p = \log_2(N)$. Desweiteren wird angenommen, daß die Werte $\overline{\omega_1^k}$ vorab berechnet und gespeichert werden. Sei $c(N)$ der Aufwand der FFT, wobei hier nur komplexe Additionen und Multiplikationen gezählt werden. Offenbar gilt dann die Rekursion

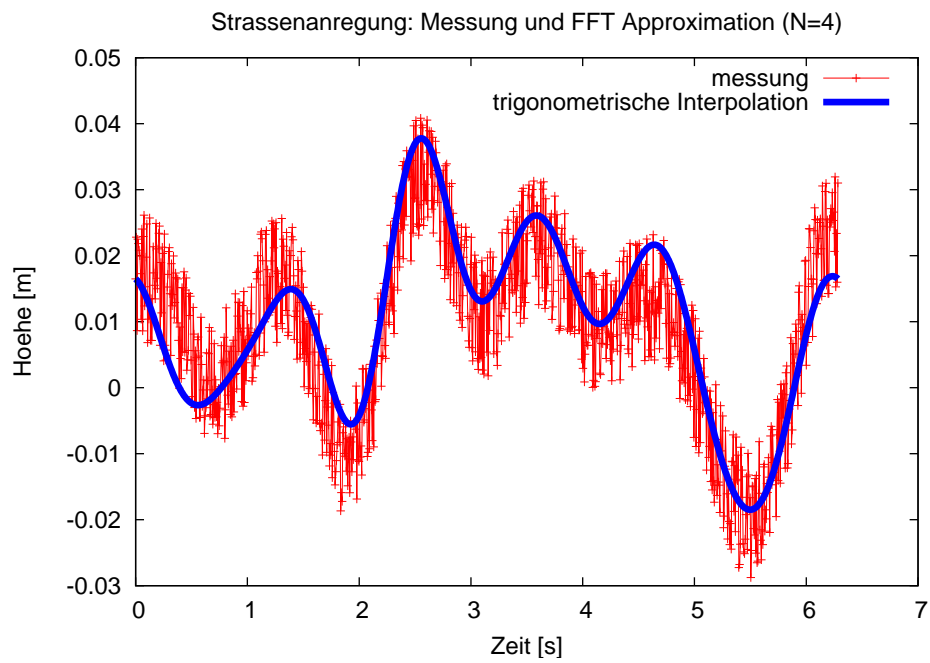
$$c(N) = 2c(N/2) + 3N/2,$$

da in jedem Rekursionsschritt die Koeffizienten α_g und α_u , sowie die α_k berechnet werden müssen. Rekursive Auswertung (p-mal) liefert

$$\begin{aligned} c(N) &= 2c(N/2) + 3N/2 \\ &= 2(2c(N/4) + 3N/4) + 3N/2 = 4c(N/4) + 2 \frac{3N}{2} \\ &= 8c(N/8) + 3 \frac{3N}{2} \\ &\vdots \\ &= 2^p c(N/2^p) + p \frac{3N}{2} = \mathcal{O}(N \log_2 N). \end{aligned}$$

Beispiel 7.3.3

Die folgende Abbildung zeigt das interpolierende trigonometrische Polynom t (glatte Kurve) für $n = 8$, welches für verrauschte Messdaten berechnet wurde:



Kapitel 8

Numerische Differentiation

In diesem Kapitel beschäftigen wir uns mit der Approximation von Ableitungen einer hinreichend glatten Funktion f an einem Punkt x .

8.1 Approximation der ersten Ableitung mittels Taylorentwicklung

Ziel ist es, die erste Ableitung $f'(x)$ zu approximieren.

8.1.1 Finite Differenzen Verfahren erster Ordnung

Sei f zweimal stetig differenzierbar in x . Der Satz von Taylor liefert dann

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(\xi).$$

Umsortierung führt auf

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{1}{2}hf''(\xi).$$

Vernachlässigung des Fehlerterms $-\frac{1}{2}hf''(\xi)$ liefert

Finite Differenzen Approximation erster Ordnung an $f'(x)$:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}. \quad (8.1)$$

Fehlerabschätzung:

$$\left| f'(x) - \frac{f(x+h) - f(x)}{h} \right| \leq C \cdot h = \mathcal{O}(h), \quad C := \frac{1}{2} \max_{\zeta \in [x, x+h]} |f''(\zeta)|.$$

Bemerkung 8.1.1

In der Praxis kann h auf Grund von Rundungsfehlereinfluß nicht beliebig klein gewählt werden. Andererseits sollte h auch nicht zu groß gewählt werden, da dann der Approximationsfehler dominiert. In der Praxis hat sich $h = \sqrt{\text{eps}} \max\{1, |x|\}$ bewährt, wobei eps die Maschinengenauigkeit bezeichnet.

8.1.2 Finite Differenzen zweiter Ordnung

Sei f dreimal stetig differenzierbar in x . Taylorentwicklung liefert

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \frac{1}{6}h^3 f'''(\xi_1), \\ f(x-h) &= f(x) - hf'(x) + \frac{1}{2}h^2 f''(x) - \frac{1}{6}h^3 f'''(\xi_2). \end{aligned}$$

Subtraktion liefert

$$f(x+h) - f(x-h) = 2hf'(x) + \underbrace{\frac{1}{6}h^3(f'''(\xi_1) + f'''(\xi_2))}_{\text{error term}}.$$

Umsortieren und Vernachlässigen des Fehlerterms führt auf

Finite Differenzen Approximation 2. Ordnung an $f'(x)$:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}. \quad (8.2)$$

Fehlerabschätzung:

$$\left| f'(x) - \frac{f(x+h) - f(x-h)}{2h} \right| \leq C \cdot h^2 = \mathcal{O}(h^2), \quad C := \frac{1}{6} \max_{\zeta \in [x-h, x+h]} |f'''(\zeta)|.$$

8.2 Approximation zweiter Ableitungen mittels Taylorentwicklung

Wir möchten $f''(x)$ approximieren.

Sei f viermal stetig differenzierbar in x . Dann gilt

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \frac{1}{6}h^3 f'''(x) + \frac{1}{24}h^4 f^{(4)}(\xi_1), \\ f(x-h) &= f(x) - hf'(x) + \frac{1}{2}h^2 f''(x) - \frac{1}{6}h^3 f'''(x) + \frac{1}{24}h^4 f^{(4)}(\xi_2). \end{aligned}$$

Addition liefert

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + \underbrace{\frac{1}{24}h^4(f^{(4)}(\xi_1) + f^{(4)}(\xi_2))}_{\text{Fehler}}.$$

Umsortierung und Vernachlässigung des Fehlers führt auf

Finite Differenzen Approximation zweiter Ordnung an $f''(x)$:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \quad (8.3)$$

Fehlerabschätzung:

$$\left| f''(x) - \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \right| \leq C \cdot h^2 = \mathcal{O}(h^2), \quad C := \frac{1}{12} \max_{\zeta \in [x-h, x+h]} |f^{(4)}(\zeta)|.$$

8.3 Richardson Extrapolation

Für festes f und x definiere das Verfahren $\varphi(h)$ durch

$$\varphi(h) := \frac{f(x+h) - f(x-h)}{2h}.$$

Unter der Annahme, daß f beliebig hohe Ableitungen besitzt, liefert Taylorentwicklung

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{1}{2!}h^2f''(x) + \frac{1}{3!}h^3f'''(x) + \frac{1}{4!}h^4f^{(4)}(x) + \dots, \\ f(x-h) &= f(x) - hf'(x) + \frac{1}{2!}h^2f''(x) - \frac{1}{3!}h^3f'''(x) + \frac{1}{4!}h^4f^{(4)}(x) - \dots \end{aligned}$$

Subtraktion und Umsortierung liefert

$$\begin{aligned} \varphi(h) - f'(x) &= \frac{1}{3!}h^2f'''(x) + \frac{1}{5!}h^4f^{(5)}(x) + \dots \\ &=: -a_2h^2 - a_4h^4 - a_6h^6 - \dots \\ &= -\sum_{i=1}^{\infty} a_{2i}h^{2i}. \end{aligned} \tag{8.4}$$

Die Koeffizienten a_{2i} hängen von x aber nicht von h ab!

Bemerkung 8.3.1 (Asymptotische Fehlerdarstellung)

Gleichung 8.4 ist eine **asymptotische Fehlerdarstellung**. Die Existenz dieser Darstellung ist die Basis für die folgende Richardson Extrapolation (vgl. auch die zusammengesetzte Trapezregel zur Integration von Funktionen).

Betrachte

$$\begin{aligned} \varphi(h) - f'(x) &= -a_2h^2 - a_4h^4 - a_6h^6 - \dots, \\ \varphi\left(\frac{h}{2}\right) - f'(x) &= -a_2\left(\frac{h}{2}\right)^2 - a_4\left(\frac{h}{2}\right)^4 - a_6\left(\frac{h}{2}\right)^6 - \dots \end{aligned}$$

Beachte, daß

$$\varphi\left(\frac{h}{2}\right) + \frac{1}{3} \left[\varphi\left(\frac{h}{2}\right) - \varphi(h) \right] - f'(x) = \frac{1}{4}a_4h^4 + \frac{5}{16}a_6h^6 + \dots$$

Durch eine geeignete Kombination der beiden Approximationen $\varphi(h)$ und $\varphi(h/2)$, wobei jede den Fehler $\mathcal{O}(h^2)$ besitzt, erhalten wir eine Approximation der Ordnung $\mathcal{O}(h^4)$. Wir haben also zwei Ordnungen gewonnen.

Dieses Verfahren heißt **Richardson Extrapolation**.

Algorithmus 8.3.2 (Richardson Extrapolation)

- (i) Wähle N und h .
- (ii) Für $i = 0, 1, \dots, N$, berechne $D(i, 0) := \varphi\left(\frac{h}{2^i}\right)$.
- (iii) Für $j = 1, \dots, i$, $i = 1, \dots, N$, berechne

$$D(i, j) = D(i, j-1) + \frac{D(i, j-1) - D(i-1, j-1)}{4^j - 1}.$$

Der Algorithmus führt auf das folgende Schema („Neville Schema“):

	$\mathcal{O}(h^2)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^6)$	\dots	$\mathcal{O}(h^{2(N+1)})$
$\varphi(h)$	$= D(0, 0)$				
$\varphi(h/2)$	$= D(1, 0)$	$D(1, 1)$			
$\varphi(h/4)$	$= D(2, 0)$	$D(2, 1)$	$D(2, 2)$		
\vdots	\vdots	\vdots	\vdots	\ddots	
$\varphi(h/2^N)$	$= D(N, 0)$	$D(N, 1)$	$D(N, 2)$	\dots	$D(N, N)$

Es gilt

Satz 8.3.3 (Richardson Extrapolation)

Sei f beliebig oft differenzierbar. Für $D(n, m)$ gilt

$$D(n, m) - f'(x) = \sum_{k=m+1}^{\infty} A(k, m) \left(\frac{h}{2^n}\right)^{2k} \quad (0 \leq m \leq n)$$

mit Konstanten $A(k, m)$, die nicht von h abhängen.

Kapitel A

Software

Available software in the world wide web:

- SCILAB: A Free Scientific Software Package; <http://www.scilab.org>
- GNU OCTAVE: A high-level language, primarily intended for numerical computations; <http://www.octave.org/octave.html>
- GAMS: Guide to Available Mathematical Software; <http://gams.nist.gov/>
- NETLIB: collection of mathematical software, papers, and databases; <http://www.netlib.org/>
- NEOS GUIDE: www-fp.mcs.anl.gov/otc/Guide

Empfohlene Literatur für Numerische Mathematik I:

- Deuffhard, P. und Hohmann, A. *Numerische Mathematik*. de Gruyter, Berlin, 1991.
- Hämmerlin, G. und Hoffmann, K.-H. *Numerische Mathematik*. Springer, Berlin-Heidelberg-New York, 1994, 4. Auflage.
- Schaback, R. und Werner, H. *Numerische Mathematik*. Springer, Berlin-Heidelberg-New York, 1992, 4. Auflage.
- Stoer, J. *Numerische Mathematik 1*. Springer, Berlin-Heidelberg-New York, 1993, 6. Auflage.
- Opfer, G. *Numerische Mathematik für Anfänger*. Vieweg, 2008, 5. Auflage.
- Kincaid, D. und Cheney, W. *Numerical Analysis: Mathematics of Scientific Computing*. Brooks/Cole Series in Advanced Mathematics, Thomson Learning, 3rd Edition, 2002.
- Lempio, F. *Numerische Mathematik I und II*. Bayreuther Mathematische Schriften, 51 und 55, 1997 und 1998.

Literaturverzeichnis

- [DB94] Deuffhard, P. and Bornemann, F. *Numerische Mathematik II*. de Gruyter, Berlin, 1994.
- [Dem91] Demailly, J.-P. *Gewöhnliche Differentialgleichungen*. Vieweg, Braunschweig, 1991.
- [DH91] Deuffhard, P. and Hohmann, A. *Numerische Mathematik*. de Gruyter, Berlin, 1991.
- [GK99] Geiger, C. and Kanzow, C. *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben*. Springer, Berlin-Heidelberg-New York, 1999.
- [HH94] Hämmerlin, G. and Hoffmann, K.-H. *Numerische Mathematik*. Springer, Berlin-Heidelberg-New York, 4th edition, 1994.
- [JS04] Jarre, F. and Stoer, J. *Optimierung*. Springer, Berlin-Heidelberg-New York, 2004.
- [Kan07] Kanzow, C. *Numerische Mathematik I*. Vorlesungsskript, Institut für Mathematik, Universität Würzburg, 2007.
- [KC02] Kincaid, D. and Cheney, W. *Numerical Analysis: Mathematics of Scientific Computing*. Brooks/Cole–Thomson Learning, Pacific Grove, CA, 3rd edition, 2002.
- [Kel95] Kelley, C. T. *Iterative Methods for Solving Linear and Nonlinear Equations*, volume 16 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, 1995.
- [KL94] Kortüm, W. and Lugner, P. *Systemdynamik und Regelung von Fahrzeugen*. Springer, Berlin-Heidelberg-New York, 1994.
- [Kos93] Kosmol, P. *Methoden zur numerischen Behandlung nichtlinearer Gleichungen und Optimierungsaufgaben*. Teubner, Stuttgart, 2nd edition, 1993.
- [Lem97] Lempio, F. *Numerische Mathematik I – Methoden der linearen Algebra*. volume 51 of *Bayreuther Mathematische Schriften*. Bayreuth, 1997.
- [Lem98] Lempio, F. *Numerische Mathematik II – Methoden der Analysis*. volume 55 of *Bayreuther Mathematische Schriften*. Bayreuth, 1998.

- [Opf08] Opfer, G. *Numerische Mathematik für Anfänger*. Vieweg, 5th edition, 2008.
- [Sch71] Schönhage, A. *Approximationstheorie*. DeGruyter, Berlin, 1971.
- [Sto93] Stoer, J. *Numerische Mathematik I*. Springer, Berlin-Heidelberg-New York, 6th edition, 1993.
- [SW92] Schaback, R. and Werner, H. *Numerische Mathematik*. Springer, Berlin-Heidelberg-New York, 4th edition, 1992.