Numerische Mathematik
für Informatiker

Peter Kunkel

Literatur

[1] Deuflhard/Hohmann: Numerische Mathematik I; Walter de Gruyter.

[2] Huckle/Schneider: Numerik für Informatiker; Springer-Verlag.

Inhalt

Kapitel 0

Vorbemerkungen

Ziel der Numerischen Mathematik ist die Entwicklung und Bewertung von Verfahren zur (näherungsweisen) zahlenmäßigen Lösung von (mathematischen) Problemen.
Problematik:
— Endlichkeit der Menge der Symbole;
— Endlichkeit der Anzahl der Rechenoperationen.

Beispiel 0.1
Das Polynom p mit

$$p(x) = x^2 - 2 \tag{0.1}$$

besitzt die positive Nullstelle $\sqrt{2}$. Diese Aussage ist insofern leer, da das Symbol $\sqrt{2}$ gerade dadurch definiert ist. Die Frage nach der Größe von $\sqrt{2}$ könnte so präzisiert werden, daß man eine Dezimalbruchentwicklung von $\sqrt{2}$ angeben soll. Da aber $\sqrt{2}$ nicht rational ist, ist diese weder abbrechend noch periodisch. Man muß sich also mit einer rationalen Approximation von $\sqrt{2}$ zufrieden geben. Ein Taschenrechner liefert z. B.

$$\sqrt{2} \approx 1.414213562\,. \tag{0.2}$$

Die Frage ist nun, wie man (oder der Taschenrechner) überhaupt eine solche Approximation bestimmen kann. Anforderungen an eine entsprechende Methode wären Effizienz (d.h., das Resultat soll "schnell" verfügbar sein) und Genauigkeit (d.h., das Resultat soll eine "gute" Approximation an den gesuchten Wert darstellen).

□

Beispiel 0.2
Ggeben sei $f : [-1,1] \to \mathbb{R}$ mit

$$f(x) = \begin{cases} \frac{1-\sqrt{1-x^2}}{x^2} & \text{für } x \neq 0, \\ \frac{1}{2} & \text{für } x = 0. \end{cases} \tag{0.3}$$

Wegen

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

für $x \neq 0$, kann man f auch durch

$$f(x) = \frac{1}{1+\sqrt{1-x^2}} \tag{0.4}$$

darstellen. Berechnet man $f(10^{-i})$, $i = 0, 1, 2, \ldots$, mit den beiden Darstellungen z. B. auf einem Taschenrechner, so erhält man folgende Resultate:

i	$f(10^{-i})$ nach (0.3)	$f(10^{-i})$ nach (0.4)
0	1.00000 00000	1.00000 00000
1	0.50125 62894	0.50125 62893
2	0.50001 25100	0.50001 25006
3	0.50000 10000	0.50000 01250
4	0.50001 00000	0.50000 00013
5	0.51000 00000	0.50000 00000
6	0.00000 00000	0.50000 00000

Offensichtlich ist die Berechnung von $f(x)$ für $x \approx 0$ auf der Basis von (0.3) weniger geeignet als auf der Basis von (0.4).

□

Die Frage ist nun, wie man obiges Phänomen erklären und vermeiden kann.

Kapitel 1

Rechnerarithmetik und Fehleranalyse

1.1 Arithmetik der Maschinenzahlen

Wegen der Endlichkeit eines Rechners kann man auf ihm nicht mit ganz $\mathbb{R}$ arbeiten, sondern muß sich mit einer endlichen Teilmenge davon zufrieden geben. Im folgenden soll als Basis für weitere Überlegungen ein Modell für eine Rechnerarithmetik entwickelt werden. Tatsächliche Rechner können zum Teil davon abweichen.

Satz 1.1 (Zahldarstellung zu einer Basis)
Sei $b \in \mathbb{N} \setminus \{1\}$. Jedes $x \in \mathbb{R}$ besitzt eine Darstellung

$$x = \pm \sum_{i=1}^{\infty} d_i b^{r-i} \tag{1.1}$$

mit $d_i \in \{0, \dots, b-1\}$, $r \in \mathbb{Z}$.
Beweis:
Sei $x_0 \in \mathbb{R}$ und o. E. $x_0 \geq 0$. Dann gibt es ein $r \in \mathbb{Z}$ mit

$$0 \leq x_0 < b^r.$$

Setzt man

$$d_1 = \left[\frac{x_0}{b^{r-1}}\right], \quad [x] = \max\{l \in \mathbb{Z} \mid l \leq x\},$$

so ist $d_1 \in \{0, \ldots, b-1\}$. Wegen

$$d_1 \le \frac{x_0}{b^{r-1}} < d_1 + 1 \quad \Longleftrightarrow \quad d_1 b^{r-1} \le x_0 < d_1 b^{r-1} + b^{r-1}$$

folgt für $x_1 = x_0 - d_1 b^{r-1}$ die Ungleichung

$$0 \le x_1 < b^{r-1}.$$

Nach s Schritten erhält man so die Darstellung

$$x_0 = \sum_{i=1}^{s} d_i b^{r-i} + x_s$$

mit $d_i \in \{0, \ldots, b-1\}$ und es gilt

$$0 \le x_s < b^{r-s}.$$

Die Behauptung folgt daraus wegen $b^{r-s} \to 0$ für $s \to \infty$.

□

Bemerkung 1.2 (normalisierte Zahldarstellung)
Alternativ kann ein $x \in \mathbb{R}$ statt gemäß (1.1) auch in der Form

$$x = v \cdot m \cdot b^e, \quad v \in \{\pm 1\}, \quad m = \textstyle\sum_{i=1}^{\infty} d_i b^{r-e-i} \tag{1.2}$$

v – Vorzeichen, b – Basis, m – Mantisse, e – Exponent

dargestellt werden. Fordert man zusätzlich, daß $e = r$ ist, so gilt $m \in [0, 1]$. Fordert man außerdem für $x \neq 0$, daß $d_1 \neq 0$, so gilt sogar $m \in [\frac{1}{b}, 1]$. Im letzteren Fall spricht man von einer normalisierten Darstellung von x.

□

Auf einem Rechner muß entscheidbar sein, wann zwei Zahlen gleich sind. Eine wichtige Frage ist daher, in wie weit normalisierte Darstellungen von reellen Zahlen eindeutig sind.

Lemma 1.3 (Eindeutigkeit)
Jedes $x \in \mathbb{R} \setminus \{0\}$ besitzt höchstens zwei normalisierte Darstellungen der Form (1.1). Besitzt x zwei Darstellungen, so ist die eine abbrechend, d. h. von der Form

$$x = \pm \sum_{i=1}^{s} d_i b^{r-i}, \quad d_1 \neq 0, \tag{1.3}$$

und die andere ist gegeben durch

$$x = \pm \sum_{i=1}^{s} d_i b^{r-i} \mp b^{r-s} \pm \sum_{i=s+1}^{\infty} (b-1) b^{r-i}, \tag{1.4}$$

wobei letztere eventuell noch nicht normalisiert ist.
Beweis:
Sei o. E. $x > 0$ mit den Darstellungen

$$\sum_{i=1}^{\infty} d_i b^{r-i} = \sum_{i=1}^{\infty} e_i b^{r-i},$$

wovon die erste normalisiert sei. Weiter sei

$$s = \min\{i \in \mathbb{N} \mid d_i \neq e_i\}.$$

Ist die zweite Darstellung nicht normalisiert, so gilt

$$d_1 > 0 = e_1.$$

Sind beide Darstellungen normalisiert, so können wir o. E. annehmen, daß

$$d_s > e_s.$$

Damit folgt

$$\begin{aligned} b^{r-s} &\leq (d_s - e_s) b^{r-s} = \textstyle\sum_{i=1}^{s} (d_i - e_i) b^{r-i} = \\ &= \textstyle\sum_{i=s+1}^{\infty} (e_i - d_i) b^{r-i} \leq (b-1) \sum_{i=s+1}^{\infty} b^{r-i} = \\ &= (b-1) b^{r-s-1} \textstyle\sum_{i=0}^{\infty} b^{-i} = (b-1) b^{r-s-1} \frac{b}{b-1} = b^{r-s} \end{aligned}$$

und es muß überall das Gleichheitszeichen gelten, d. h. $d_s - e_s = 1$ sowie

$$e_i - d_i = b - 1, \quad i = s+1, s+2, \dots,$$

bzw.

$$e_i = b - 1, \quad d_i = 0, \quad i = s+1, s+2, \dots.$$

□

Die Idee ist nun, mit endlichen (normalisierten) Darstellungen der Form (1.1) zu arbeiten.

Definition 1.4 (normalisierte Fließkommazahlen)
Die auf einem (idealisierten) Rechner verfügbaren Zahlen (sog. Maschinenzahlen) seien gegeben durch

$$(1.5)\ \mathbb{M}_{b,l} = \{\pm \sum_{i=1}^{l} d_i b^{r-i} \mid d_i \in \{0, \ldots, b-1\},\ d_1 \neq 0,\ r \in \mathbb{Z}\} \cup \{0\} \subseteq \mathbb{R},$$

wobei $b \in \mathbb{N} \setminus \{1\}$ und $l \in \mathbb{N}$. Man nennt $\mathbb{M}_{b,l}$ die Menge der normalisierten Fließkommazahlen zur Basis b mit Mantissenlänge l.

□

Bemerkung 1.5
Für einen elektronischen Rechner ist $b = 2$ oder $b = 16$, bei Handrechung ist $b = 10$. Außerdem ist bei einem Rechner der Exponent r zusätzlich beschränkt gemäß

$$(1.6) \qquad r_{\min} \leq r \leq r_{\max}.$$

Die dadurch auftretenden Randeffekte wie Exponentenüberlauf oder -unterlauf spielen in unseren Untersuchungen keine Rolle und werden deshalb hier nicht berücksichtigt. Wir arbeiten also in unserem Modell mit einer unendlichen Menge $\mathbb{M}_{b,l}$ von Maschinenzahlen.

□

Um mit einem gegebenen $x \in \mathbb{R}$ auf dem Rechner arbeiten zu können, muß x durch ein $\tilde{x} \in \mathbb{M}_{b,l}$ ersetzt werden. Man spricht von Rundung.

Definition 1.6 (Rundung)
Eine Abbildung $\mathrm{fl} : \mathbb{R} \to \mathbb{M}_{b,l}$ mit

(1.7)

(a) $\mathrm{fl}(x) = x$ für alle $x \in \mathbb{M}_{b,l}$,
(Projektionseigenschaft)

(b) $\mathrm{fl}(x) \leq \mathrm{fl}(y)$ für alle $x, y \in \mathbb{R}$ mit $x \leq y$
(Monotonie)

heißt Rundung (bzgl. $\mathbb{M}_{b,l}$).

□

Für $b = 10$ kennt man die sog. kaufmännische Rundung (auch 5/4-Rundung genannt). Sie kann folgendermaßen verallgemeinert werden.

Lemma 1.7 (kaufmännische Rundung)
Sei $x \in \mathbb{R} \setminus \{0\}$ mit normalisierter (im mehrdeutigen Fall abbrechender) Darstellung

$$x = \pm \sum_{i=1}^{\infty} d_i b^{r-i}, \quad d_1 \neq 0. \tag{1.8}$$

Durch

$$\mathrm{fl}(x) = \begin{cases} \pm \sum_{i=1}^{l} d_i b^{r-i} & \text{für } d_{l+1} \in \{0, \dots, [\frac{b-1}{2}]\}, \\ \pm \sum_{i=1}^{l} d_i b^{r-i} \pm b^{r-l} & \text{für } d_{l+1} \in \{[\frac{b+1}{2}], \dots, b-1\}, \end{cases} \tag{1.9}$$

sowie $\mathrm{fl}(0) = 0$ ist eine Rundung $\mathrm{fl} : \mathbb{R} \to \mathbb{M}_{b,l}$ gemäß Definition 1.6 gegeben, die außerdem symmetrisch ist, d. h. die Eigenschaft

$$\mathrm{fl}(-x) = -\,\mathrm{fl}(x) \text{ für alle } x \in \mathbb{R} \tag{1.10}$$

besitzt.

Beweis:
Zunächst muß gezeigt werden, daß $\mathrm{fl}(x) \in \mathbb{M}_{b,l}$ für alle $x \in \mathbb{R}$ ist. Das ist offensichlich bis auf den zweiten Fall in (1.9). Ist dort $d_l \leq b-2$, so gilt

$$\mathrm{fl}(x) = \pm \sum_{i=1}^{l-1} d_i b^{r-i} \pm (d_l + 1) b^{r-l} \in \mathbb{M}_{b,l}.$$

Ist aber $d_l = b - 1$, so hat man zunächst nur

$$\mathrm{fl}(x) = \pm \sum_{i=1}^{l-1} d_i b^{r-i} \pm b^{r-l+1}.$$

Das ist aber gerade der zweite Fall in (1.9) mit um eins verkürzter Mantisse. Induktiv ergibt sich also entweder $\mathrm{fl}(x) \in \mathbb{M}_{b,l}$ durch den ersten Schritt oder schließlich

$$\mathrm{fl}(x) = \pm b^r \in \mathbb{M}_{b,l}$$

im zweiten Schritt.

Die Projektionseigenschaft und die Symmetrie ergeben sich direkt aus der Definition. Die Monotonie ergibt sich aus der Beobachtung, daß im Inneren des Intervalls

$$\Big[\sum_{i=1}^{l} d_i b^{r-i}, \sum_{i=1}^{l} d_i b^{r-i} + b^{r-l}\Big]$$

keine Maschinenzahl liegt und alle x in

$$\Big[\sum_{i=1}^{l} d_i b^{r-i}, \sum_{i=1}^{l} d_i b^{r-i} + [\tfrac{b+1}{2}] b^{r-l-1}\Big)$$

auf die linke Intervallgrenze bzw. alle x in

$$\Big[\sum_{i=1}^{l} d_i b^{r-i} + [\tfrac{b+1}{2}] b^{r-l-1}, \sum_{i=1}^{l} d_i b^{r-i} + b^{r-l}\Big]$$

auf die rechte Intervallgrenze gerundet werden, also nie über eine Maschinenzahl hinweg gerundet wird.

□

Bemerkung 1.8
Ist b gerade, so wird bei obiger Rundung zur nächstliegenden Maschinenzahl gerundet. Ist diese nicht eindeutig, so wird zur betragsgrößeren der beiden Maschinenzahl gerundet.

□

Offensichtlich wird bei jeder Rundung (falls nicht zufällig eine Maschinenzahl vorliegt) ein Fehler gemacht (sog. Rundungsfehler). Es ist deshalb wichtig zu wissen, wie dieser Fehler aussieht bzw. wie er abgeschätzt werden kann.

Lemma 1.9
Ist b gerade, so gilt für die durch (1.9) definierte Rundung die Abschätzung

$$|\operatorname{fl}(x) - x| \le \tfrac{1}{2} b^{-l+1} |x| \tag{1.11}$$

Beweis:
Sei o. E. $x > 0$ und damit $\operatorname{fl}(x) > 0$. Unter Verwendung von (1.8) und (1.9) erhält man die folgenden Abschätzungen.
1. Fall: $d_{l+1} \le [\frac{b-1}{2}] = \frac{b}{2} - 1$
Es gilt

$$\begin{aligned} 0 \le x - \operatorname{fl}(x) &= \textstyle\sum_{i=1}^{\infty} d_i b^{r-i} - \sum_{i=1}^{l} d_i b^{r-i} = \\ &= \textstyle\sum_{i=l+1}^{\infty} d_i b^{r-i} \le (\frac{b}{2} - 1) b^{r-l-1} + (b-1) \sum_{i=l+2}^{\infty} b^{r-i} = \\ &= (\tfrac{b}{2} - 1) b^{r-l-1} + (b-1) b^{r-l-2} \tfrac{b}{b-1} = \tfrac{1}{2} b^{r-l}. \end{aligned}$$

2. Fall: $d_{l+1} \geq [\frac{b+1}{2}] = \frac{b}{2}$
Es gilt

$$\begin{aligned} 0 \geq x - \mathrm{fl}(x) &= \textstyle\sum_{i=1}^{\infty} d_i b^{r-i} - \sum_{i=1}^{l} d_i b^{r-i} - b^{r-l} = \\ &= \textstyle\sum_{i=l+1}^{\infty} d_i b^{r-i} - b \cdot b^{r-l-1} = (d_{l+1} - b) b^{r-l-1} + \sum_{i=l+2}^{\infty} d_i b^{r-i} \geq \\ &\geq (\tfrac{b}{2} - b) b^{r-l-1} = -\tfrac{1}{2} b^{r-l}. \end{aligned}$$

In beiden Fällen erhält man also

$$|\,\mathrm{fl}(x) - x| \leq \tfrac{1}{2} b^{r-l}.$$

Wegen $|x| \geq b^{r-1}$ folgt dann

$$|\,\mathrm{fl}(x) - x| \leq \tfrac{1}{2} b^{r-1-l+1} = \tfrac{1}{2} b^{r-1} b^{-l+1} \leq \tfrac{1}{2} b^{-l+1} |x|.$$

□

Definition 1.10 (Maschinengenauigkeit)
Sei b gerade. Die Größe

$$\text{eps} = \tfrac{1}{2} b^{-l+1} \tag{1.12}$$

heißt (die zu obiger Rundung gehörige) Maschinengenauigkeit.

□

Bemerkung 1.11
Die Abschätzung

$$|\,\mathrm{fl}(x) - x| \leq \text{eps}\,|x| \text{ für alle } x \in \mathbb{R} \tag{1.13}$$

impliziert für gegebenes $x \in \mathbb{R}$ mit zugehörigem $\tilde{x} = \mathrm{fl}(x)$ die Existenz von $\varepsilon \in \mathbb{R}$ mit

$$\tilde{x} = x(1+\varepsilon), \quad |\varepsilon| \leq \text{eps}. \tag{1.14}$$

Beweis:
Für $x = 0$ ist der Beweis trivial, etwa mit der Wahl $\varepsilon = 0$. Für $x \neq 0$ setzt man

$$\varepsilon = \frac{\tilde{x} - x}{x}$$

Damit gilt sofort $\tilde{x} = x(1+\varepsilon)$. Außerdem folgt mit (1.13)

$$|\varepsilon| = \frac{|\tilde{x} - x|}{|x|} = \frac{|\,\mathrm{fl}(x) - x|}{|x|} \le \mathrm{eps}\,.$$

□

Nachdem wir ein Modell für die auf einem Rechner verwendbaren Zahlen zur Verfügung haben, wollen wir natürlich mit diesen Zahlen rechnen. Die Problematik dabei ist, daß man schon bei den Grundrechenarten nicht erwarten kann, daß die Verknüpfung zweier Maschinenzahlen wieder eine Maschinenzahl ist.

Als Modell für die Grundrechenarten verwenden wir die Fließkommarealisierungen

$$\text{(1.15)} \qquad \begin{array}{ll} \text{(a)} & x \oplus y = \mathrm{fl}(x+y), \\ \text{(b)} & x \ominus y = \mathrm{fl}(x-y), \\ \text{(c)} & x \odot y = \mathrm{fl}(x \cdot y), \\ \text{(d)} & x \oslash y = \mathrm{fl}(x/y), \end{array}$$

entsprechend für die sogenannten Standardfunktionen

$$\text{(1.16)} \qquad \textcircled{f}(x) = \mathrm{fl}(f(x)),$$

etwa mit $f \in \{\mathrm{sqrt}, \sin, \cos, \mathrm{atan}, \exp, \log, \ldots\}$. Zusammen mit (1.5) und (1.9) haben wir damit ein Modell eines Rechners (idealer Rechner) aufgestellt, das wir im Rahmen dieser Vorlesung ausschließlich verwenden werden.

Bemerkung 1.12
Man beachte, daß sich nicht alle Eigenschaften der obigen Verknüpfung von $\mathbb{R}$ auf $\mathbb{M}_{b,l}$ übertragen. So ist $\oplus$ zwar kommutativ, aber nicht assoziativ.

□

Beispiel 1.13
Sei $b = 10$ und $l = 2$. Gegeben seien

$$x = 104\,, \quad y = 4.22\,, \quad z = 3.86\,.$$

Diese werden gerundet zu

$$\tilde{x} = 0.10 \cdot 10^3\,, \quad \tilde{y} = 0.42 \cdot 10^1\,, \quad \tilde{z} = 0.39 \cdot 10^1\,.$$

Damit erhält man statt $x + y + z = 112.08$

$$\begin{aligned}(\tilde{x} \oplus \tilde{y}) \oplus \tilde{z} &= \mathrm{fl}(0.1042 \cdot 10^3) \oplus 0.39 \cdot 10^1 = \\ &= 0.10 \cdot 10^3 \oplus 0.39 \cdot 10^1 = \mathrm{fl}(0.1039 \cdot 10^3) = 0.10 \cdot 10^3\end{aligned}$$

bzw.

$$\begin{aligned}\tilde{x} \oplus (\tilde{y} \oplus \tilde{z}) &= 0.10 \cdot 10^3 \oplus \mathrm{fl}(0.81 \cdot 10^1) = \\ &= 0.10 \cdot 10^3 \oplus 0.81 \cdot 10^1 = \mathrm{fl}(0.1081 \cdot 10^3) = 0.11 \cdot 10^3,\end{aligned}$$

wobei letzteres gleich dem gerundeten exakten Ergebnis ist.

□

Algorithmus 1.14
Auf einem idealen Rechner (mit geradem b) gilt

$$\text{eps} = \min\{x \in \mathbb{M}_{b,l} \mid x > 0,\ 1 \oplus x > 1\}. \tag{1.17}$$

Für $b = 2$ gilt speziell eps $= 2^{-l}$. Man kann eps deshalb in diesem Fall mit dem Algorithmus

```
eps=1.0;
while (1.0+eps>1.0) eps=0.5*eps;
eps=2.0*eps;
```

bestimmen.

□

Bemerkung 1.15 (IEEE-Standard 754)
Das vorstehend entwickelte Modell eines idealen Rechners, insbesondere (1.15) und (1.16) zusammen mit (1.13), bildet die Grundlage für die Entwicklung und Beurteilung von numerischen Verfahren. Tatsächliche Rechner weichen davon nicht nur bezüglich der Endlichkeit der Menge der darstellbaren Zahlen in unterschiedlichem Maß ab. Dies betrifft vor allem Großrechner etwa von IBM oder Cray z. B. bezüglich Rundung bzw. Genauigkeit der Grundrechenarten. Im Jahr 1985 wurde vom Institute of Electronical and Electronics Engineers ein Standard herausgegeben, dem zumindest alle gängigen Workstations und (mit geringen Abweichungen) PCs genügen. Dabei werden zwei Zahlenformate fest vorgeschrieben und Mindestanforderungen für zwei zugehörige erweiterte Formate festgelegt. Diese Forderungen werden auf den letztgenannten Rechnerarchitekturen durch das

Tabelle 1: Zahlformate

Format	Bytes	Basis b	Länge l der Mantisse	hidden bit	Bits für Exponent
`single`	4	2	24	ja	8
`double`	8	2	53	ja	11
`extended` (WS)	16	2	113	ja	15
`extended` (PC)	10	2	64	nein	15

Vorhandensein von drei Zahlformaten erfüllt, vgl. Tabelle 1. Der Standard fordert für die Grundformate `single` und `double` die Eigenschaften (1.15) sowie (1.16) für die Wurzelfunktion. In Abweichung von (1.9) wird für ein x, das genau zwischen zwei Maschinenzahlen liegt, auf diejenige von beiden gerundet, deren letzte Ziffer in der Mantisse gerade ist. Daneben schreibt der Standard drei weitere wählbare Rundungsformen (Abschneiden, Runden in Richtung $+\infty$ bzw. $-\infty$) vor. Im ersten Fall gilt (1.13) wie im Modell mit $\text{eps} = 2^{-l}$, wenn man von $\mathbb{M}_{2,l}$ als Menge der verfügbaren Maschinenzahlen ausgeht. Bei den drei anderen Rundungen gilt (1.13) mit $\text{eps} = 2^{-l+1}$.

$\square$

1.2 Grundlegende Begriffe

Ein zu lösendes Problem ist dadurch gekennzeichnet, daß zu gegebenen Daten x ein Ergebnis y zu berechnen ist, wobei der Zusammenhang zwischen x und y entweder explizit durch

$$y = f(x), \quad f\colon \mathbb{X} \to \mathbb{Y} \tag{1.18}$$

oder implizit durch

$$g(x, y) = 0, \quad g\colon \mathbb{X} \times \mathbb{Y} \to \mathbb{Z} \tag{1.19}$$

beschrieben wird. Die Mengen $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ sind im folgenden typischerweise Teilmengen irgendwelcher $\mathbb{R}^n$.

Um ein Ergebnis y zahlenmäßig angeben zu können, muß dieses eindeutig durch die Problemstellung festgelegt sein. Da wir außerdem (evtl. schon bei der Eingabe der Daten x) auf dem Rechner durch Rundung Fehler machen, muß sich das gestellte Problem unter Störungen gutartig verhalten.

Definition 1.16 (Wohlgestelltheit eines Problems)
Ein Problem (1.18) oder (1.19) heißt wohlgestellt, wenn es zu jedem $x \in \mathbb{X}$ genau eine Lösung $y \in \mathbb{Y}$ gibt und die Zuordnung $x \mapsto y$ stetig ist.

□

Für ein wohlgestelltes Problem möchte man ein Maß dafür, wie stark sich Fehler in den Daten auf Fehler in der Lösung auswirken. Um Fehler messen zu können, benötigen wir Normen. Im $\mathbb{R}^n$ verwendet man als Verallgemeinerung des Betrags in $\mathbb{R}$ etwa (mit $x = (x_1, \ldots, x_n)^T$)

$$
\text{(1.20)} \quad
\begin{array}{lll}
\text{(a)} & \|x\|_1 = \sum_{i=1}^n |x_i|, & \text{(Summennorm)} \\
\text{(b)} & \|x\|_2 = (\sum_{i=1}^n |x_i|^2)^{1/2}, & \text{(euklidische Norm)} \\
\text{(c)} & \|x\|_\infty = \max_{i=1\ldots,n} |x_i|, & \text{(Maximumnorm)}
\end{array}
$$

mit den für eine Norm $\|\cdot\|$ üblichen Eigenschaften

$$
\text{(1.21)} \quad
\begin{array}{lll}
\text{(a)} & \|x\| > 0 \text{ für alle } x \neq 0, & \text{(Positivität)} \\
\text{(b)} & \|\alpha x\| = |\alpha| \, \|x\| \text{ für alle } \alpha \in \mathbb{R},\ x \in \mathbb{R}^n, & \text{(Homogenität)} \\
\text{(c)} & \|x + y\| \leq \|x\| + \|y\| \text{ für alle } x, y \in \mathbb{R}^n. & \text{(Dreiecksungl.)}
\end{array}
$$

Definition 1.17 (Absoluter/relativer Fehler)
Sei $\tilde{x} \in \mathbb{R}^n$ eine Näherung an $x \in \mathbb{R}^n$. Sei außerdem eine Norm $\|\cdot\|$ auf $\mathbb{R}^n$ gegeben. Dann heißt $\|\tilde{x} - x\|$ der zugehörige absolute Fehler von $\tilde{x}$. Ist darüberhinaus $x \neq 0$, so heißt $\|\tilde{x} - x\| / \|x\|$ der zugehörige relative Fehler von $\tilde{x}$.

□

Man beachte, daß Rundungsfehler wegen (1.11) bzw. $|\operatorname{fl}(x) - x| / |x| \leq \text{eps}$ für $x \neq 0$ als relative Fehler beschränkt sind.

Definition 1.18 (Kondition eines Problems)
Sei durch $f\colon \mathbb{X} \to \mathbb{Y}$ ein wohlgestelltes Problem beschrieben. Dann heißt

$$
\text{(1.22)} \qquad \kappa = \sup_{\substack{x_1, x_2 \in \mathbb{X} \\ x_1 \neq x_2}} \frac{\|f(x_2) - f(x_1)\|_{\mathbb{Y}}}{\|x_2 - x_1\|_{\mathbb{X}}} \in [0, \infty]
$$

Kondition des Problems.

□

Man beachte, daß κ von der Wahl der Normen auf $\mathbb{X}$ und $\mathbb{Y}$ abhängt.

Bemerkung 1.19
Sind $\mathbb{X}, \mathbb{Y} \subseteq \mathbb{R}$ mit $\|\cdot\|_{\mathbb{X}} = \|\cdot\|_{\mathbb{Y}} = |\cdot|$ und ist $f\colon \mathbb{X} \to \mathbb{Y}$ stetig differenzierbar, so gilt nach dem Mittelwertsatz

$$\kappa = \sup_{x \in \mathbb{X}} |f'(x)|. \tag{1.23}$$

□

Bemerkung 1.20 (gut/schlecht konditioniertes Problem)
Um die Kondition eines Problems zu beurteilen, seien x die wahren Daten, $\tilde{x}$ die verfügbaren Daten, jeweils mit Resultaten $y = f(x)$ und $\tilde{y} = f(\tilde{x})$. Ist $\varepsilon = \|\tilde{x} - x\|_{\mathbb{X}}$ und benötigt man das Resultat mit einer Genauigkeit $\tau \geq 0$, so kann man wegen (1.22) nur im Fall

$$\tau \geq \kappa \cdot \varepsilon \tag{1.24}$$

mit der Einhaltung der geforderten Genauigkeit entsprechend $\|\tilde{y} - y\|_{\mathbb{Y}} \leq \tau$ rechnen. Man sagt in diesem Fall, das Problem sei gut konditioniert. Andernfalls sagt man, das Problem sei schlecht konditioniert. Will man im Extremfall bei Eingabefehlern $\varepsilon = \mathrm{eps}$ noch die Größenordnung des Resultats entsprechend $\tau = 1$ erfassen, muß

$$\kappa \leq \frac{1}{\mathrm{eps}} \tag{1.25}$$

gelten.

□

Beispiel 1.21
Das durch (0.3) mit $\mathbb{X} = [-1, 1]$ und $\mathbb{Y} = \mathbb{R}$ gegebene Problem ist wohlgestellt. Für $\varepsilon > 0$ hinreichend klein gilt

$$\frac{1}{\varepsilon}(f(1) - f(1-\varepsilon)) = \frac{1}{\varepsilon}\left[1 - \frac{1 - \sqrt{1 - (1-\varepsilon)^2}}{(1-\varepsilon)^2}\right] = \frac{1 - 2\varepsilon + \varepsilon^2 - 1 + \sqrt{2\varepsilon - \varepsilon^2}}{\varepsilon(1-\varepsilon)^2} \to \infty$$

für $\varepsilon \to 0$. Damit ist $\kappa = \infty$ und das Problem ist schlecht konditioniert.

Wählt man stattdessen $\mathbb{X} = [-\frac{1}{2}, \frac{1}{2}]$, so hat f wegen

$$\sqrt{1 - x^2} = 1 - \frac{1}{2}x^2 - \sum_{k \geq 2} \frac{(2k-3)!!}{k! 2^k} x^{2k}$$

die Potenzreihenentwicklung

$$f(x) = \frac{1}{2} + \sum_{k\geq 2} \frac{(2k-3)!!}{k!2^k} x^{2k-2}$$

mit Konvergenzradius $R = 1$, d. h. f ist stetig differenzierbar auf $\mathbb{X}$ mit Ableitung

$$f'(x) = \sum_{k\geq 2} \frac{(2k-2)(2k-3)!!}{k!2^k} x^{2k-3}.$$

Also ist f' monoton und punktsymmetrisch und es gilt mit (1.23)

$$\kappa \leq f'(\frac{1}{2}) = \sum_{k\geq 2} \frac{2k-2}{k\cdot 2} \frac{2k-3}{(k-1)\cdot 2} \cdots \frac{1}{1\cdot 2} \left(\frac{1}{2}\right)^{2k-3} \leq \sum_{k\geq 2} \left(\frac{1}{2}\right)^{2k-3} \leq 1,$$

d. h. das so modifizierte Problem ist gut konditioniert.

□

Die obige Untersuchung besagt, daß die Funktionswerte $f(x)$ von f aus Beispiel 0.2 in der Nähe der Null nicht sensitiv gegenüber Fehlern in x sind, was die Auswertung mit Hilfe von (0.4) bestätigt. Die Probleme bei Verwendung der Darstellung (0.3) kommen also nicht von einer schlechten Kondition des Problems, sondern müssen vom gewählten Rechenweg herrühren.

Im Normalfall kann man nicht erwarten, daß man die Zuordnung $x \overset{f}{\mapsto} y$ auf dem Rechner in einem Schritt ohne Erzeugung von Zwischenergebnissen durchführen kann.

Definition 1.22 (Algorithmus)
Ein Algorithmus ist eine endliche Folge von Rechenvorschriften bestehend aus Elementaroperationen (Grundrechenarten, Auswerten von Standardfunktionen), die angibt, wie y aus x zu bestimmen ist, d. h. eine Zerlegung von $f\colon \mathbb{X} \to \mathbb{Y}$ entsprechend

$$f = \varphi_l \circ \varphi_{l-1} \circ \cdots \circ \varphi_1 \tag{1.26}$$

mit

$$\varphi_i : \mathbb{X}_i \to \mathbb{X}_{i+1}, \quad i = 1, \ldots, l, \tag{1.27}$$

wobei $\mathbb{X}_1 = \mathbb{X}$ und $\mathbb{X}_{l+1} = \mathbb{Y}$ und jedes durch eine Elementaroperation erzeugte Zwischenergebnis als Komponente des Bildes eines φ_i auftritt.

□

Man beachte, daß es zu einem f verschiedene Algorithmen geben kann, etwa wie im Beispiel 0.2 basierend auf den Darstellungen (0.3) und (0.4).

Da man mit jedem φ_i Zwischenresultate und damit Rundungsfehler erzeugt, muß man auch die Auswirkung dieser Fehler auf das Resultat berücksichtigen. Diese sollte wenn möglich nicht wesentlich größer sein als die der unvermeidbaren Fehler (Eingabefehler und Rundung des Resultats).

Definition 1.23 (Stabilität eines Algorithmus)
Zu einem Algorithmus (1.26) seien die sogenannten Restabblidungen ψ_i definiert durch

$$\psi_i = \varphi_l \circ \varphi_{l-1} \circ \cdots \circ \varphi_i, \quad \psi_i : \mathbb{X}_i \to \mathbb{Y}, \quad i = 1, \ldots, l, \tag{1.28}$$

mit Konditionen κ_i. Gilt

$$\kappa_i \leq (C+1) \max\{\kappa, 1\}, \tag{1.29}$$

wobei C die Anzahl der im Algorithmus durchzuführenden Elementaroperationen ist, so spricht man von einem stabilen Algorithmus. Davon unabhängig kann die Zahl $\max_{i=1,\ldots,l}\{\kappa_i\}/\max\{\kappa, 1\}$ als Stabilitätsmaß angesehen werden.

□

1.3 Differentielle Fehleranalyse

Wie man an Beispiel 1.21 schon gesehen hat, sind Berechnungen basierend auf (1.22) nicht leicht durchzuführen. Geht man davon aus, daß Fehler klein sind, so bieten sich Linearisierungstechniken an, d. h. man vernachlässigt quadratische Terme in den Fehlern. Man spricht von differentieller Fehleranalyse.

Sei $f : \mathbb{X} \to \mathbb{Y}$ mit $\mathbb{X}, \mathbb{Y} \subseteq \mathbb{R}$ stetig differenzierbar, so gilt für $x \in \mathbb{X}$ und gestörtes $\tilde{x} \in \mathbb{X}$ mit dazu gehörigen Resultaten $y = f(x)$ und $\tilde{y} = f(\tilde{x})$

$$\begin{aligned} \tilde{y} - y &= f(\tilde{x}) - f(x) = f(x + (\tilde{x} - x)) - f(x) \doteq \\ &\doteq f(x) + f'(x)(\tilde{x} - x) - f(x) = f'(x)(\tilde{x} - x). \end{aligned} \tag{1.30}$$

Mit $\|\cdot\|_{\mathbb{X}} = \|\cdot\|_{\mathbb{Y}} = |\cdot|$ erhält man speziell

$$\kappa = |f'(x)|, \tag{1.31}$$

sog. differentielle Kondition bzgl. des absoluten Fehlers. Wählt man stattdessen $\|\cdot\|_{\mathbb{X}} = |\cdot|/|x|$ und $\|\cdot\|_{\mathbb{Y}} = |\cdot|/|y|$, falls $x, y \neq 0$, so liefert (1.30) in der Form

$$\frac{\tilde{y}-y}{y} \doteq \frac{xf'(x)}{y}\frac{\tilde{x}-x}{x} \tag{1.32}$$

stattdessen

$$\kappa = \left|\frac{xf'(x)}{f(x)}\right|, \tag{1.33}$$

sog. differentielle Kondition bzgl. des relativen Fehlers.

Ist $\mathbb{X} \subseteq \mathbb{R}^n$, so hat die Jacobimatrix $f'(x)$ die Form

$$f'(x) = \left[\begin{array}{ccc} \frac{\partial f}{\partial x_1}(x) & \cdots & \frac{\partial f}{\partial x_n}(x) \end{array}\right] \tag{1.34}$$

und (1.30) wird mit $x = (x_1, \ldots, x_n)^T$ und $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_n)^T$ zu

$$\tilde{y} - y \doteq \sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(x)(\tilde{x}_j - x_j). \tag{1.35}$$

Damit gilt in erster Näherung

$$|\tilde{y} - y| \mathrel{\dot{\le}} \sum_{j=1}^{n} \left|\frac{\partial f}{\partial x_j}(x)\right| |\tilde{x}_j - x_j| \tag{1.36}$$

bzw.

$$\left|\frac{\tilde{y}-y}{y}\right| \mathrel{\dot{\le}} \sum_{j=1}^{n} \left|\frac{x_j}{y}\frac{\partial f}{\partial x_j}(x)\right| \left|\frac{\tilde{x}_j - x_j}{x_j}\right| \tag{1.37}$$

und die entsprechenden Faktoren beschreiben die differentielle Konditionen bzgl. des absoluten bzw. relativen Fehlers der einzelnen Komponenten x_j von x.

Für $\mathbb{Y} \subseteq \mathbb{R}^m$ braucht man nur entsprechend die einzelnen Komponenten von $y \in \mathbb{Y}$ zu betrachten.

Entsprechend kann man die differentielle Fehleranalyse zur Untersuchung der Stabilität eines Algorithmus verwenden. Man muß nur nach Definition 1.23 statt f die Restabbildungen ψ_i betrachten. Sei dazu $\mathbb{X}_i \subseteq \mathbb{R}^{n_i}$, $x_1 = x$ sowie $x_{i+1} = \varphi_i(x_i)$, $i = 1, \ldots, l$, außerdem $x_i = (x_{i,1}, \ldots, x_{i,n_i})^T$. Die maßgeblichen differentiellen Konditionen bzgl. des relativen Fehlers entsprechend (1.37) sind damit gegeben durch

$$\kappa_{i,j} = \left|\frac{x_{i,j}}{y}\frac{\partial \psi_i}{\partial x_{i,j}}(x_i)\right|, \quad j = 1, \ldots, n_i,\ i = 1, \ldots, l. \tag{1.38}$$

Dabei beschreibt

$$\kappa = \max\{\kappa_{1,1}, \ldots, \kappa_{1,n_1}\} \tag{1.39}$$

die differentielle Kondition des Problems und (1.29) wird ersetzt durch die Bedingung

$$\kappa_{i,j} \leq (C+1)\max\{\kappa_{1,1}, \ldots, \kappa_{1,n_1}, 1\}, \quad j = 1, \ldots, n_i,\ i = 2, \ldots, l. \tag{1.40}$$

für die Stabilität eines Algorithmus.

Beispiel 1.24 (Kondition von Addition/Subtraktion)
Sei $y = x_1 + x_2$, d. h. $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ mit $f(x_1, x_2) = x_1 + x_2$. Es gilt

$$\tilde{y} - y = (\tilde{x}_1 + \tilde{x}_2) - (x_1 + x_2) = (\tilde{x}_1 - x_1) + (\tilde{x}_2 - x_2)$$

und damit für den absoluten Fehler

$$|\tilde{y} - y| \leq |\tilde{x}_1 - x_1| + |\tilde{x}_2 - x_2| = \|\tilde{x} - x\|_1$$

bzw. für den relativen Fehler (bei $x_1, x_2, y \neq 0$)

$$\begin{aligned}\tfrac{|\tilde{y}-y|}{|y|} &\dot{\leq} \tfrac{|x_1|}{|x_1+x_2|} \cdot \tfrac{|\tilde{x}_1-x_1|}{|x_1|} + \tfrac{|x_2|}{|x_1+x_2|} \cdot \tfrac{|\tilde{x}_2-x_2|}{|x_2|} \leq \\ &\leq \max\left\{\tfrac{|x_1|}{|x_1+x_2|}, \tfrac{|x_2|}{|x_1+x_2|}\right\} \cdot \left(\tfrac{|\tilde{x}_1-x_1|}{|x_1|} + \tfrac{|\tilde{x}_2-x_2|}{|x_2|}\right).\end{aligned}$$

Für $\|\cdot\|_{\mathbb{X}} = \|\cdot\|_1$ und $\|\cdot\|_{\mathbb{Y}} = |\cdot|$ (absoluter Fehler) hat man also wegen $\kappa = 1$ ein gut konditioniertes Problem. Wählt man $\|\cdot\|_{\mathbb{X}}$ stattdessen gemäß $\|(z_1, z_2)\|_{\mathbb{X}} = |z_1/x_1| + |z_2/x_2|$ und $\|\cdot\|_{\mathbb{Y}} = |\cdot|/|y|$ (relativer Fehler), so liegt nur dann ein gut konditioniertes Problem vor, wenn $|x_1 + x_2|$ nicht wesentlich kleiner als $\max\{|x_1|, |x_2|\}$ ist. Andernfalls hat man ein schlecht konditioniertes Problem (sog. subtraktive Auslöschung). Das gleiche folgt auch mit differentieller Fehleranalyse.

□

Beispiel 1.25 (Kondition der Multiplikation)
Sei $y = x_1 \cdot x_2$, d. h. $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ mit $f(x_1, x_2) = x_1 \cdot x_2$. Für die differentielle Kondition bzgl. des relativen Fehlers (bei $x_1, x_2 \neq 0$) erhält man nach (1.37)

$$\left|\frac{x_1}{y}\frac{\partial f}{\partial x_1}(x)\right| = \left|\frac{x_1}{x_1 \cdot x_2} \cdot x_2\right| = 1, \quad \left|\frac{x_2}{y}\frac{\partial f}{\partial x_2}(x)\right| = \left|\frac{x_2}{x_1 \cdot x_2} \cdot x_1\right| = 1.$$

Die Multiplikation ist also bzgl. des relativen Fehlers im differentiellen Sinn gut konditioniert.

Unter Verwendung von Bemerkung 1.11 kann man auch folgendermaßen schließen. Sei

$$|\tilde{x}_i - x_i| \leq \text{eps}\,|x_i|, \quad i = 1, 2,$$

d. h. gebe es ε_i mit $|\varepsilon_i| \leq \text{eps}$, so daß

$$\tilde{x}_i = x_i(1 + \varepsilon_i).$$

Dann gilt

$$\tilde{y} = \tilde{x}_1 \cdot \tilde{x}_2 = x_1(1+\varepsilon_1) \cdot x_2(1+\varepsilon_2) \doteq x_1 x_2 (1 + \varepsilon_1 + \varepsilon_2)$$

und damit

$$\frac{|\tilde{y} - y|}{|y|} \doteq |\varepsilon_1 + \varepsilon_2| \leq |\varepsilon_1| + |\varepsilon_2| \leq 2\,\text{eps}.$$

Dabei entsprechen die obigen Konditionszahlen gerade den Vorfaktoren von $|\varepsilon_1|$ und $|\varepsilon_2|$.

□

Ein entsprechendes Resultat kann man auch für die Division zeigen, siehe Übungen.

Beispiel 1.26 (Stabilität der Exponentiation)
Zur Berechnung von Potenzen a^x für $a, x \in \mathbb{R}$, $a > 0$, kann man basierend auf der Darstellung

$$a^x = \exp(x \cdot \log a)$$

den folgenden Algorithmus verwenden:

$$\begin{array}{lll}
x_1 = \binom{x}{a}, & \varphi_1 : x_1 \mapsto \binom{x_{11}}{\log x_{12}}, & \psi_1 : x_1 \mapsto \exp(x_{11} \cdot \log x_{12}), \\
x_2 = \binom{x}{\log a}, & \varphi_2 : x_2 \mapsto x_{21} \cdot x_{22}, & \psi_2 : x_2 \mapsto \exp(x_{21} \cdot x_{22}), \\
x_3 = x \cdot \log a, & \varphi_3 : x_3 \mapsto \exp(x_3), & \psi_3 : x_3 \mapsto \exp(x_3).
\end{array}$$

Für die differentiellen Konditionen des relativen Fehlers gemäß (1.38) erhält man

$$\begin{aligned}
\kappa_{11} &= \left| \tfrac{x_{11}}{y} \tfrac{\partial \psi_1}{\partial x_{11}}(x_1) \right| = \left| \tfrac{x_{11}}{y} \exp(x_{11} \cdot \log x_{12}) \log x_{12} \right| = |x \cdot \log a|, \\
\kappa_{12} &= \left| \tfrac{x_{12}}{y} \tfrac{\partial \psi_1}{\partial x_{12}}(x_1) \right| = \left| \tfrac{x_{12}}{y} \exp(x_{11} \cdot \log x_{12}) \tfrac{x_{11}}{x_{12}} \right| = |x|, \\
\kappa_{21} &= \left| \tfrac{x_{21}}{y} \tfrac{\partial \psi_2}{\partial x_{21}}(x_2) \right| = \left| \tfrac{x_{21}}{y} \exp(x_{21} \cdot x_{22}) x_{22} \right| = |x \cdot \log a|, \\
\kappa_{22} &= \left| \tfrac{x_{22}}{y} \tfrac{\partial \psi_2}{\partial x_{22}}(x_2) \right| = \left| \tfrac{x_{22}}{y} \exp(x_{21} \cdot x_{22}) x_{21} \right| = |x \cdot \log a|, \\
\kappa_3 &= \left| \tfrac{x_3}{y} \tfrac{\partial \psi_3}{\partial x_3}(x_3) \right| = \left| \tfrac{x_3}{y} \exp(x_3) \right| = |x \cdot \log a|.
\end{aligned}$$

Dabei beschreiben κ_{11} und κ_{12} die differentielle Kondition des Problems. Wegen $\kappa_{21} = \kappa_{22} = \kappa_3 = \kappa_{11}$ ist der Algorithmus stabil.

□

Beispiel 1.27
Der auf (0.3) aufbauende Algorithmus zur Berechnung von $y = f(x)$ kann folgendermaßen geschrieben werden:

$$\begin{array}{lll}
x_1 = x, & \varphi_1 : x_1 \mapsto x_1^2, & \psi_1 : x_1 \mapsto \frac{1-\sqrt{1-x_1^2}}{x_1^2}, \\
x_2 = x^2, & \varphi_2 : x_2 \mapsto \binom{1-x_2}{x_2}, & \psi_2 : x_2 \mapsto \frac{1-\sqrt{1-x_2}}{x_2}, \\
x_3 = \binom{1-x^2}{x^2}, & \varphi_3 : \binom{x_{31}}{x_{32}} \mapsto \binom{\sqrt{x_{31}}}{x_{32}}, & \psi_3 : \binom{x_{31}}{x_{32}} \mapsto \frac{1-\sqrt{x_{31}}}{x_{32}}, \\
x_4 = \binom{\sqrt{1-x^2}}{x^2}, & \varphi_4 : \binom{x_{41}}{x_{42}} \mapsto \binom{1-x_{41}}{x_{42}}, & \psi_4 : \binom{x_{41}}{x_{42}} \mapsto \frac{1-x_{41}}{x_{42}}, \\
x_5 = \binom{1-\sqrt{1-x^2}}{x^2}, & \varphi_5 : \binom{x_{51}}{x_{52}} \mapsto \frac{x_{51}}{x_{52}}, & \psi_5 : \binom{x_{51}}{x_{52}} \mapsto \frac{x_{51}}{x_{52}}.
\end{array}$$

Für $x \in \mathbb{R}$ mit $|x|$ hinreichend klein erhält man in erster Näherung

$$\begin{array}{l}
x_1 = x, \quad x_2 = \varphi_1(x_1) = x^2, \quad x_3 = \varphi_2(x_2) = \binom{1-x^2}{x^2}, \\
x_4 = \varphi_3(x_3) = \binom{\sqrt{1-x^2}}{x^2} \doteq \binom{1-\frac{1}{2}x^2}{x^2}, \quad x_5 = \varphi_4(x_4) \doteq \binom{\frac{1}{2}x^2}{x^2}, \quad y = \varphi_5(x_5) \doteq \frac{1}{2}.
\end{array}$$

Wegen $\psi_1 = f$ liefert die Reihenentwicklung aus Beispiel 1.21

$$\psi_1(x_1) \doteq \frac{1}{2} + \frac{1}{8}x_1^2, \quad \psi_2(x_2) \doteq \frac{1}{2} + \frac{1}{8}x_2$$

bzw.

$$\frac{\partial \psi_1}{\partial x_1}(x_1) \doteq \frac{1}{4}x_1, \quad \frac{\partial \psi_2}{\partial x_2}(x_2) \doteq \frac{1}{8}$$

und damit für die differentiellen Konditionen bzgl. des relativen Fehlers

$$\begin{aligned}
\kappa_1 &= \left|\tfrac{x_1}{y}\tfrac{\partial \psi_1}{\partial x_1}(x_1)\right| \doteq \left|\tfrac{x_1}{y}\tfrac{1}{4}x_1\right| \doteq \tfrac{1}{2}x^2 \\
\kappa_2 &= \left|\tfrac{x_2}{y}\tfrac{\partial \psi_2}{\partial x_2}(x_2)\right| \doteq \left|\tfrac{x_2}{y}\tfrac{1}{8}\right| \doteq \tfrac{1}{4}x^2 \\
\kappa_{31} &= \left|\tfrac{x_{31}}{y}\tfrac{\partial \psi_3}{\partial x_{31}}(x_3)\right| = \left|\tfrac{x_{31}}{y}\tfrac{1}{x_{32}}\tfrac{1}{2\sqrt{x_{31}}}\right| \doteq \tfrac{1}{x^2} \\
\kappa_{32} &= \left|\tfrac{x_{32}}{y}\tfrac{\partial \psi_3}{\partial x_{32}}(x_3)\right| = \left|\tfrac{x_{32}}{y}\tfrac{1-\sqrt{x_{31}}}{x_{32}^2}\right| \doteq \left|\tfrac{x^2}{\frac{1}{2}}\tfrac{\frac{1}{2}x^2}{x^4}\right| = 1 \\
\kappa_{41} &= \left|\tfrac{x_{41}}{y}\tfrac{\partial \psi_4}{\partial x_{41}}(x_4)\right| = \left|\tfrac{x_{41}}{y}\tfrac{1}{x_{42}}\right| \doteq \tfrac{2}{x^2} \\
\kappa_{42} &= \left|\tfrac{x_{42}}{y}\tfrac{\partial \psi_4}{\partial x_{42}}(x_4)\right| = \left|\tfrac{x_{42}}{y}\tfrac{1-x_{41}}{x_{42}^2}\right| \doteq 1 \\
\kappa_{51} &= \left|\tfrac{x_{51}}{y}\tfrac{\partial \psi_5}{\partial x_{51}}(x_5)\right| = \left|\tfrac{x_{51}}{y}\tfrac{1}{x_{52}}\right| \doteq 1 \\
\kappa_{52} &= \left|\tfrac{x_{52}}{y}\tfrac{\partial \psi_5}{\partial x_{52}}(x_5)\right| = \left|\tfrac{x_{52}}{y}\tfrac{x_{51}}{x_{52}^2}\right| \doteq 1
\end{aligned}$$

Der vorliegende Algorithmus ist also für hinreichend kleines $|x|$ instabil. Insbesondere werden Fehler bei der Berechnung bzw. Rundung von x_{31} und x_{41} extrem verstärkt. Dies ist zurückzuführen auf die anschließende Subtraktion mit Auslöschung führender Stellen.

$\square$

Kapitel 2

Lineare Gleichungssysteme

Gesucht sei $x \in \mathbb{R}^n$ mit

$$Ax = b, \tag{2.1}$$

wobei $A \in \mathbb{R}^{n,n}$ nichtsingulär und $b \in \mathbb{R}^n$ ist.

Zum Messen von Fehlern bei Matrizen verwenden wir zu gegebener Vektornorm die durch

$$\|A\| = \sup_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\| \tag{2.2}$$

definierte sogenannte Operatornorm. Neben

$$\begin{array}{lll} & \text{(a)} & \|A\| > 0 \text{ für alle } A \in \mathbb{R}^{n,n},\ A \neq 0, \\ (2.3) & \text{(b)} & \|\alpha A\| = |\alpha|\,\|A\| \text{ für alle } \alpha \in \mathbb{R},\ A \in \mathbb{R}^{n,n}, \\ & \text{(c)} & \|A+B\| \leq \|A\| + \|B\| \text{ für alle } A, B \in \mathbb{R}^{n,n} \end{array}$$

haben diese noch die Eigenschaften

$$\begin{array}{lll} & \text{(a)} & \|A \cdot B\| \leq \|A\| \cdot \|B\| \text{ für alle } A, B \in \mathbb{R}^{n,n}, \\ & & \qquad \text{(Submultiplikativität)} \\ (2.4) & \text{(b)} & \|Ax\| \leq \|A\| \cdot \|x\| \text{ für alle } A \in \mathbb{R}^{n,n},\ x \in \mathbb{R}^n. \\ & & \qquad \text{(Verträglichkeit)} \end{array}$$

Zu den Vektornormen (1.20) gehören mit $A = (a_{ij})$ die Operatornormen

$$\begin{array}{lll} & \text{(a)} & \|A\|_1 = \max_{j=1,\ldots,n} \sum_{i=1}^n |a_{ij}|, \\ & & \qquad \text{(Spaltensummennorm)} \\ (2.5) & \text{(b)} & \|A\|_2 = \max\{\sqrt{\lambda} \mid \lambda \text{ Eigenwert von } A^T A\}, \\ & & \qquad \text{(Spektralnorm)} \\ & \text{(c)} & \|A\|_\infty = \max_{i=1,\ldots,n} \sum_{j=1}^n |a_{ij}|. \\ & & \qquad \text{(Zeilensummennorm)} \end{array}$$

2.1 Kondition des Problems

Für A nichtsingulär besitzt (2.1) die eindeutige Lösung

$$x = A^{-1}b. \tag{2.6}$$

Wegen der Stetigkeit der Inversenbildung und der Matrix-Vektor-Multiplikation ist das Problem (2.1) wohlgestellt. In erster Näherung gilt für ein gestörtes Problem

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

d. h.

$$\Delta x \doteq A^{-1}\Delta b - A^{-1}\Delta A x. \tag{2.7}$$

Mit einer Vektornorm und zugehöriger Operatornorm gilt

$$\begin{gathered} \|\Delta x\| \mathrel{\dot{\leq}} \|A^{-1}\| \cdot \|\Delta b\| + \|A^{-1}\| \cdot \|x\| \cdot \|\Delta A\| \\ \text{(absoluter Fehler)} \end{gathered} \tag{2.8}$$

und

$$\begin{aligned} \frac{\|\Delta x\|}{\|x\|} &\mathrel{\dot{\leq}} \frac{\|A^{-1}\|\cdot\|Ax\|}{\|x\|} \cdot \frac{\|\Delta b\|}{\|b\|} + \frac{\|A^{-1}\|\cdot\|x\|\cdot\|A\|}{\|x\|} \cdot \frac{\|\Delta A\|}{\|A\|} \leq \\ &\leq \|A\| \cdot \|A^{-1}\| \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|}\right). \\ &\text{(relativer Fehler)} \end{aligned} \tag{2.9}$$

Die Größe

$$\operatorname{cond}_{\|\cdot\|} A = \|A\| \cdot \|A^{-1}\| \tag{2.10}$$

heißt Kondition der Matrix A und beschreibt die (differentielle) Verstärkung des relativen Fehlers der Daten A und b. Im Fall (2.5) schreibt man auch

$$\operatorname{cond}_p A = \|A\|_p \cdot \|A^{-1}\|_p \tag{2.11}$$

für $p \in \{1, 2, \infty\}$.

2.2 Gauß-Elimination

Schreibt man (2.1) in der Form

$$
\begin{array}{ccccccc}
a_{11}x_1 & + & a_{12}x_2 & + \cdots + & a_{1n}x_n & = & b_1, \\
a_{21}x_1 & + & a_{22}x_2 & + \cdots + & a_{2n}x_n & = & b_2, \\
\vdots & & \vdots & & \vdots & & \vdots \\
a_{n1}x_1 & + & a_{n2}x_2 & + \cdots + & a_{nn}x_n & = & b_n,
\end{array}
\tag{2.12}
$$

so kann man durch Vertauschen von Zeilen und Umbenennen der Größen stets erreichen, daß $a_{11} \neq 0$ ist (sonst wäre A singulär). Die Idee der Gauß-Elimination ist dann, durch Addieren von Vielfachen der ersten Gleichung zu den anderen aus diesen x_1 zu eliminieren. Man erhält

$$
\begin{array}{ccccccc}
a_{11}x_1 & + & a_{12}x_2 & + \cdots + & a_{1n}x_n & = & b_1, \\
 & & a'_{22}x_2 & + \cdots + & a'_{2n}x_n & = & b'_2, \\
 & & \vdots & & \vdots & & \vdots \\
 & & a'_{n2}x_2 & + \cdots + & a'_{nn}x_n & = & b'_n.
\end{array}
\tag{2.13}
$$

Dabei ändert sich die Lösung nicht. Sukzessives Anwenden liefert nach $n-1$ Schritten ein System der Form

$$
\begin{array}{ccccccc}
r_{11}x_1 & + & r_{12}x_2 & + \cdots + & r_{1n}x_n & = & y_1, \\
 & & r_{22}x_2 & + \cdots + & r_{2n}x_n & = & y_2, \\
 & & & \ddots & \vdots & & \vdots \\
 & & & & r_{nn}x_n & = & y_n,
\end{array}
\tag{2.14}
$$

mit $r_{ii} \neq 0$, $i = 1, \ldots, n$. Daraus kann man x beginnend mit x_n leicht bestimmen (sog. Rückwärtssubstitution).

Algorithmus 2.1 (Gauß-Elimination mit Zeilentausch)
Der folgende Algorithmus erzeugt (2.14) ausgehend von (2.12).

<table>
<tr><td colspan="3">Schleife $i = 1, \ldots, n$:</td></tr>
<tr><td rowspan="5"></td><td colspan="2">Suche ein Element $a_{pi} \neq 0$ mit $p = i, \ldots, n$, sog. Pivotsuche.
Sind alle $a_{pi} = 0$, so ist A singulär $\rightarrow$ Stop!
Andernfalls vertausche i-te und p-te Zeile.</td></tr>
<tr><td colspan="2">Schleife $k = i+1, \ldots, n$:</td></tr>
<tr><td rowspan="3"></td><td>Setze $l_{ki} = a_{ki}/a_{ii}$.</td></tr>
<tr><td>Setze $a_{kj} = a_{kj} - l_{ki}a_{ij}$, $j = i+1, \ldots, n$.</td></tr>
<tr><td>Setze $b_k = b_k - l_{ki}b_i$.</td></tr>
</table>

(2.15)

Dabei sind die r_{ij}, $1 \leq i \leq j \leq n$, durch die entsprechenden letzten Werte von a_{ij} gegeben und die y_i, $1 \leq i \leq n$, durch die entsprechenden letzten Werte von b_i gegeben.

$\square$

Der Aufwand in (2.15) beträgt im wesentlichen eine Operation (Addition und Multiplikation) pro Berechnung eines Wertes a_{kj} in der innersten Schleife, d. h. der Gesamtaufwand beträgt in erster Ordnung

$$(n-1)^2 + (n-2)^2 + \cdots + 4 + 1 = \sum_{i=1}^{n-1} i^2 \doteq \frac{1}{3}n^3 \text{ Operationen.}$$

Die gebräuchlichste Pivotsuche in (2.15) ist gegeben durch

(2.16)
$$|a_{pi}| = \max_{k=i,\dots,n} |a_{ki}| \qquad \text{(Spaltenpivotsuche).}$$

Es gilt dann

(2.17)
$$|l_{ki}| \leq 1.$$

Hat man Algorithmus 2.1 erfolgreich durchgeführt, so erhält man die gesuchte Lösung x folgendermaßen.

Algorithmus 2.2 (Rückwärtssubstitution)
Im Anschluß an (2.15) ist folgendes durchzuführen.

Schleife $i = n, \dots, 1$:	
	Setze $b_i = b_i - a_{ij}x_j$, $j = n, \dots, i+1$. Setze $x_i = b_i / a_{ii}$.

(2.18)

$\square$

Der Aufwand in (2.18) beträgt in erster Ordnung

$$0 + 1 + \cdots + (n-1) = \sum_{i=1}^{n} (i-1) \doteq \frac{1}{2}n^2 \text{ Operationen.}$$

Bemerkung 2.3 (LR-Zerlegung)
Beschreibt $\Pi \in \mathbb{R}^{n,n}$ die im Algorithmus 2.1 zu gegebenem $A \in \mathbb{R}^{n,n}$ vorzunehmenden Zeilentausche und setzt man

$$L = \begin{bmatrix} 1 & & & & \\ l_{2,1} & 1 & & & \\ l_{3,1} & l_{3,2} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{n,1} & l_{n,2} & \cdots & l_{n,n-1} & 1 \end{bmatrix}, \quad R = \begin{bmatrix} r_{1,1} & \cdots & \cdots & \cdots & r_{1,n} \\ & r_{2,2} & & & \vdots \\ & & \ddots & & \vdots \\ & & & \ddots & \vdots \\ & & & & r_{n,n} \end{bmatrix},$$

so gilt

$$\Pi \cdot A = L \cdot R \qquad \text{(sog. Dreieckszerlegung).}$$

□

Bemerkung 2.4
Will man mehrere Gleichungssysteme mit gleicher Matrix A und verschiedenen Seiten b lösen, so ist es in Algorithmus 2.1 besser, die Berechnungen betreffend A und b zu entkoppeln. Zusätzlich muß man sich die vorgenommenen Zeilentausche merken, während man die Werte l_{ki} auf dem Speicherplatz von a_{ki} ablegen kann. Auf diese Weise zerfällt Algorithmus 2.1 in zwei Teile (sog. Zerlegung von A und sog. Vorwärtssubstitution zur entsprechenden Behandlung von b). In Programmbibliotheken findet man deshalb üblicherweise Routinen zur Zerlegung von A (sog. `FACTOR`) sowie zur Berechnung einer Lösung zu gegebenem b und zerlegtem A (sog. `SOLVE`), vgl. Folie.

□

Bemerkung 2.5
Der Übergang von (2.12) zu (2.13) läßt sich auch beschreiben durch

$$A' = E \cdot A, \tag{2.19}$$

wobei

$$E = \begin{bmatrix} 1 & & & \\ -l_{21} & 1 & & \\ \vdots & & \ddots & \\ -l_{n1} & & & 1 \end{bmatrix}. \tag{2.20}$$

Beachtet man

$$E^{-1} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & & \ddots & \\ l_{n1} & & & 1 \end{bmatrix}, \tag{2.21}$$

so gilt

$$\begin{aligned} \operatorname{cond}_\infty A' = \|A'\|_\infty \cdot \|A'^{-1}\|_\infty = \|E \cdot A\|_\infty \cdot \|A^{-1} \cdot E^{-1}\|_\infty \le \\ \le \|E\|_\infty \cdot \|A\|_\infty \cdot \|E^{-1}\|_\infty \cdot \|A^{-1}\|_\infty = \operatorname{cond}_\infty E \cdot \operatorname{cond}_\infty A, \end{aligned} \tag{2.22}$$

wobei im Fall (2.16) wegen (2.17) gilt

$$\operatorname{cond}_\infty E = \|E\|_\infty \cdot \|E^{-1}\|_\infty \le 2 \cdot 2 = 4. \tag{2.23}$$

Die Kondition des zu lösenden Gleichungssystems verschlechtert sich also in jedem Schritt der Gauß-Elimination evtl. um einen Faktor 4. In diesem Sinn ist Gauß-Elimination selbst mit Spaltenpivotsuche instabil. In praktischen Problemen tritt diese Instabilität allerdings normalerweise nicht auf. Man beachte, daß ohne Pivotsuche wie (2.16) der Wert $\operatorname{cond}_\infty E$ beliebig groß werden kann.

□

2.3 Cholesky-Zerlegung

Eine wichtige Klasse von Matrizen in $\mathbb{R}^{n,n}$ sind die symmetrischen, positiv-definiten Matrizen, d. h. Matrizen $A \in \mathbb{R}^{n,n}$ mit

$$\begin{array}{llll} \text{(2.24)} & \text{(a)} & A^T = A, & \text{(Symmetrie)} \\ & \text{(b)} & x^T A x > 0 \text{ für alle } x \in \mathbb{R}^n,\ x \neq 0. & \text{(positive Definitheit)} \end{array}$$

Ein Zeilentausch, wie er bei der Gauß-Elimination auftreten kann, würde die Symmetrie sofort zerstören. Als Folge könnten wichtige Eigenschaften, die auf der Symmetrie basieren, verloren gehen. Andererseits könnte man versuchen, die Symmetrie auszunutzen, um einen effizienten Algorithmus zu erhalten.

Satz 2.6 (Cholesky-Zerlegung)
Zu jeder symmetrischen, positiv-definiten Matrix $A \in \mathbb{R}^{n,n}$ gibt es eine untere Dreiecksmatrix $L \in \mathbb{R}^{n,n}$ mit positiven Diagonaleinträgen derart, daß

$$A = L \cdot L^T. \tag{2.25}$$

Beweis:
Die Behauptung ist trivial für $n = 1$ wegen

$$A = a_{11} > 0, \quad L = l_{11} = \sqrt{a_{11}}.$$

Sei die Behauptung richtig für $n \ge 1$ und sei $A \in \mathbb{R}^{n+1,n+1}$ symmetrisch und positiv definit. Partitioniert man A gemäß

$$A = \begin{bmatrix} A' & v \\ v^T & c \end{bmatrix}$$

mit $A' \in \mathbb{R}^{n,n}$, so ist A' ebenfalls symmetrisch und positiv definit. Nach Induktionsvoraussetzung gibt es also ein $L' \in \mathbb{R}^{n,n}$ mit $A' = L'L'^T$. Mit dem Ansatz

$$A = \begin{bmatrix} L'L'^T & v \\ v^T & c \end{bmatrix} = \begin{bmatrix} L' & 0 \\ l^T & d \end{bmatrix} \begin{bmatrix} L'^T & l \\ 0 & d \end{bmatrix} = \begin{bmatrix} L'L'^T & L'l \\ l^T L'^T & l^T l + d^2 \end{bmatrix}$$

erhält man die Bedingungen

$$L'l = v, \quad l^T l + d^2 = c.$$

Da positiv-definite Matrizen insbesondere nichtsingulär sind, ist mit A' auch L' nichtsingulär und damit $l = L'^{-1}v$. Um auch nach $d > 0$ auflösen zu können, muß $c - l^T l > 0$ gelten. Dies folgt aber aus

$$\begin{aligned} 0 < (-l^T L'^{-1} \;\; 1) \begin{bmatrix} L'L'^T & v \\ v^T & c \end{bmatrix} \begin{pmatrix} -L'^{-T} l \\ 1 \end{pmatrix} = \\ = (-l^T L'^{-1} \;\; 1) \begin{pmatrix} 0 \\ c - l^T l \end{pmatrix} = c - l^T l. \end{aligned}$$

□

Zur Berechnung der Cholesky-Zerlegung beachtet man

$$a_{ij} = \sum_{k=1}^{\min\{i,j\}} l_{ik} l_{jk}, \quad L = (l_{ij}), \tag{2.26}$$

wovon man wegen der Symmetrie z. B. nur $i \geq j$ betrachten muß. Es gilt also

$$\begin{aligned} &\text{(a)} \quad a_{jj} = l_{j1}^2 + l_{j2}^2 + \cdots + l_{jj}^2, \\ &\text{(b)} \quad a_{ij} = l_{i1} l_{j1} + l_{i2} l_{j2} + \cdots + l_{ij} l_{jj} \text{ für } i > j, \end{aligned} \tag{2.27}$$

bzw.

$$\begin{aligned} &\text{(a)} \quad l_{j,j}^2 = a_{j,j} - l_{j,1}^2 - \cdots - l_{j,j-1}^2, \\ &\text{(b)} \quad l_{i,j} = (a_{i,j} - l_{i,1} l_{j,1} - \cdots - l_{i,j-1} l_{j,j-1})/l_{j,j} \text{ für } i > j, \end{aligned} \tag{2.28}$$

und man kann L spaltenweise berechnen.

Algorithmus 2.7 (Cholesky-Zerlegung)
Der folgende Algorithmus berechnet für gegebenes symmetrisches, positiv-

definites $A \in \mathbb{R}^{n,n}$ die Zerlegung (2.25).

(2.29)

Schleife $j = 1, \ldots, n$:		
	Setze $s = a_{j,j} - l_{j,1}^2 - \cdots - l_{j,j-1}^2$. Ist $s \leq 0$, so ist A nicht positiv definit $\rightarrow$ Stop! Setze $l_{j,j} = \sqrt{s}$.	
	Schleife $i = j+1, \ldots, n$:	
		Setze $l_{i,j} = (a_{i,j} - l_{i,1}l_{j,1} - \cdots - l_{i,j-1}l_{j,j-1})/l_{j,j}$.

$\square$

Der Aufwand in (2.29) beträgt in erster Näherung

$$\begin{aligned} \textstyle\sum_{j=1}^n (n-j)j = \sum_{j=1}^n (nj - j^2) = n \sum_{j=1}^n j - \sum_{j=1}^n j^2 &\doteq \\ \doteq \tfrac{1}{2}n^3 - \tfrac{1}{3}n^3 = \tfrac{1}{6}n^3 &\text{ Operationen} \end{aligned}$$

sowie n Quadratwurzeln.

Zur Lösung eines zugehörigen linearen Gleichungssystems fällt dann jeweils eine Vorwärts- und Rückwärtssubstitution an.

2.4 QR-Zerlegung

Mit Sicht auf Bemerkung 2.5 wäre für (2.1) ein Lösungsverfahren wünschenswert, das durch Eliminationsmatrizen E mit $\operatorname{cond}_\infty E = 1$ beschrieben werden kann. Matrizen mit dieser Eigenschaft beschreiben aber leider nur Zeilentausche und Vorzeichenwechsel, d. h. man kann damit nicht das gegebene lineare Gleichungssystem vereinfachen. Das gleiche gilt für Matrizen E mit $\operatorname{cond}_1 E = 1$. Hingegen ist die Klasse der Matrizen E mit $\operatorname{cond}_2 E = 1$ hinreichend groß.

Definition 2.8 (orthogonale Matrizen)
Eine Matrix $Q \in \mathbb{R}^{n,n}$ heißt orthogonal, wenn

(2.30) $$Q^T Q = I.$$

$\square$

Offensichtlich ist ein orthogonales Q invertierbar mit $Q^{-1} = Q^T$. Damit gilt auch $QQ^T = I$ und außerdem $\|Q\|_2 = \|Q^T\|_2 = \|Q^{-1}\|_2 = 1$ nach (2.5b), also

$$\operatorname{cond}_2 Q = 1 \text{ für alle } Q \in \mathbb{R}^{n,n} \text{ orthogonal.} \tag{2.31}$$

Wegen $\|Qx\|_2^2 = x^T Q^T Q x = x^T x = \|x\|_2^2$ gilt

$$\|Qx\|_2 = \|x\|_2 \text{ für alle } Q \in \mathbb{R}^{n,n} \text{ orthogonal und } x \in \mathbb{R}^n. \tag{2.32}$$

Definition 2.9 (Householder-Transformationen)
Matrizen $P \in \mathbb{R}^{n,n}$ der Form

$$P = I - 2ww^T, \quad w^T w = 1, \quad w \in \mathbb{R}^n \tag{2.33}$$

heißen Householder-Transformationen.

□

Offensichtlich ist eine Householder-Transformation P symmetrisch und wegen

$$P^T P = (I - 2ww^T)(I - 2ww^T) = I - 4ww^T + 4w(w^T w)w^T = I$$

orthogonal.

Die Frage ist nun, ob man P so wählen kann, daß es wie E in Bemerkung 2.5 in der ersten Spalte von A unterhalb des ersten Eintrags Nullen erzeugt. Bezeichnet $z \in \mathbb{R}^n$, $z \neq 0$, die erste Spalte von A, so ist also ein $w \in \mathbb{R}^n$ mit $w^T w = 1$ gesucht, sodaß

$$Pz = (I - 2ww^T)z = \alpha e_1, \quad e_1 = (1, 0, \ldots, 0)^T, \quad \alpha \in \mathbb{R}. \tag{2.34}$$

Aus der Forderung, daß P orthogonal ist, folgt wegen (2.32) sofort

$$\alpha = \pm\sigma, \quad \sigma = \|z\|_2 = \sqrt{z^T z}. \tag{2.35}$$

Weiterhin folgt aus (2.34)

$$z - 2ww^T z = \alpha e_1$$

bzw.

$$2(w^T z)w = z - \alpha e_1.$$

Zusammen mit $w^T w = 1$ ergibt sich daraus

$$w = \pm \frac{z - \alpha e_1}{\|z - \alpha e_1\|_2}. \tag{2.36}$$

Um Auslöschung in $z - \alpha e_1$ zu vermeiden, wählt man das Vorzeichen von α entsprechend

$$\operatorname{sign} \alpha = -\operatorname{sign} z_1, \quad z = (z_1, \ldots, z_n)^T, \tag{2.37}$$

wenn $z_1 \neq 0$ ist. Damit gilt

$$\begin{aligned} \|z - \alpha e_1\|_2^2 &= (z_1 - \alpha)^2 + z_2^2 + \cdots + z_n^2 = \\ &= z_1^2 + 2|\alpha||z_1| + \alpha^2 + z_2^2 + \cdots + z_n^2 = \\ &= 2\sigma^2 + 2\sigma|z_1| = 2\sigma(\sigma + |z_1|) \end{aligned}$$

bzw.

$$P = I + \beta u u^T \tag{2.38}$$

mit

$$\begin{aligned} &\text{(a)} \qquad \beta = -\tfrac{1}{\sigma(\sigma + |z_1|)}, \\ &\text{(b)} \qquad u = z - \alpha e_1. \end{aligned} \tag{2.39}$$

Satz 2.10 (QR-Zerlegung)
Zu jeder nichtsingulären Matrix $A \in \mathbb{R}^{n,n}$ gibt es eine orthogonale Matrix $Q \in \mathbb{R}^{n,n}$ und eine nichtsinguläre obere Dreicksmatrix $R \in \mathbb{R}^{n,n}$, so daß

$$A = Q \cdot R. \tag{2.40}$$

Beweis:
Setzt man für $A = (a_1, \ldots, a_n)$ in der obigen Konstruktion zunächst $z = a_1 \neq 0$ ein, so erhält man mit dem zugehörigen P

$$PA = \left[\begin{array}{c|c} r_{11} & r_{12} \;\cdots\; r_{1n} \\ \hline 0 & A' \end{array}\right].$$

Per Induktion besitzt $A' \in \mathbb{R}^{n-1,n-1}$ eine QR-Zerlegung $A' = Q'R'$, d. h.

$$PA = \left[\begin{array}{c|c} r_{11} & r_{12} \;\cdots\; r_{1n} \\ \hline 0 & Q'R' \end{array}\right] = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & Q' \end{array}\right] \left[\begin{array}{c|c} r_{11} & r_{12} \;\cdots\; r_{1n} \\ \hline 0 & R' \end{array}\right].$$

Mit

$$Q = P^T \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & Q' \end{array}\right] = P \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & Q' \end{array}\right], \quad R = \left[\begin{array}{c|c} r_{11} & r_{12} \;\cdots\; r_{1n} \\ \hline & R' \end{array}\right]$$

erhält man daraus eine Zerlegung der Form (2.40).

$\square$

Beachtet man, daß man Pz nie durch Berechnen der Householder-Transformation P nach (2.38) mit anschließender Matrix-Vektor-Multiplikation mit z bestimmen sollte, sondern wegen

$$Pz = (I + \beta uu^T)z = z + \beta(u^T z)u$$

erst den Skalar $\gamma = \beta(u^T z)$ und dann die Summe $z + \gamma u$ von Vektoren bestimmen sollte, so erhält man folgenden Algorithmus zur Lösung von (2.1).

Algorithmus 2.11
Der folgende Algorithmus berechnet die Lösung von (2.1) basierend auf einer QR-Zerlegung von A.

Schleife $i = 1, \ldots, n$:		
	Setze $\tau = a_{ii}^2 + \cdots + a_{ni}^2$; $\sigma = \sqrt{\tau}$. Ist $\sigma = 0$, so ist A singulär $\to$ Stop! Setze $\alpha = \begin{cases} \sigma & \text{für } a_{ii} \le 0, \\ -\sigma & \text{für } a_{ii} > 0. \end{cases}$ Setze $d_i = \alpha$; $\beta = 1/(\alpha a_{ii} - \tau)$; $a_{ii} = a_{ii} - \alpha$.	
	Schleife $k = i+1, \ldots, n$:	
		Setze $\gamma = \beta(a_{ii}a_{ik} + \cdots + a_{ni}a_{nk})$. Setze $a_{jk} = a_{jk} + \gamma a_{ji}$, $j = i, \ldots, n$.
	Setze $\gamma = \beta(a_{ii}b_i + \cdots + a_{ni}b_n)$. Setze $b_j = b_j + \gamma a_{ji}$, $j = i, \ldots, n$.	
Schleife $i = n, \ldots, 1$:		
	Setze $b_i = b_i - a_{ij}x_j$, $j = n, \ldots, i+1$. Setze $x_i = b_i / d_i$.	

(2.41)

Die Transformationen $P_i = I + \beta_i u_i u_i^T$ werden dabei in der Form gespeichert, daß die u_i als Spalten im unteren Dreieck von A abgelegt sind und $\beta_i = 1/(\alpha_i a_{ii} - \tau_i)$ sich durch

$$\alpha_i = d_i, \quad \tau_i = \alpha_i^2, \quad a_{ii} = a_{ii} + \alpha_i$$

zurückgewinnen läßt.

□

Der obige Algorithmus benötigt in erster Näherung

$$\sum_{i=1}^n 2(n-i)^2 \doteq 2\sum_{i=1}^n i^2 \doteq \tfrac{2}{3}n^3 \text{ Operationen.}$$

Dazu kommen n Quadratwurzeln.

2.5 Lineare Ausgleichsrechnung

Gegeben seien Messungen (t_i, f_i), $t_i, f_i \in \mathbb{R}$, $i = 1, \ldots, m$. Zu gegebenen Funktionen $\varphi_j : \mathbb{R} \to \mathbb{R}$, $j = 1, \ldots, n$, $n \leq m$, sollen $x_j \in \mathbb{R}$ so bestimmt werden, daß $f : \mathbb{R} \to \mathbb{R}$, definiert durch

$$f(t) = \sum_{j=1}^n x_j \varphi_j(t), \tag{2.42}$$

die Messungen möglichst gut beschreibt. Als Maß dafür verwenden wir, daß die Abweichung $r \geq 0$ mit

$$r^2 = \sum_{i=1}^m w_i^2 (f(t_i) - f_i)^2 = \sum_{i=1}^m \Big[\sum_{j=1}^n w_i (x_j \varphi_j(t_i) - f_i)\Big]^2 \tag{2.43}$$

zu gegebenen $w_i > 0$, $i = 1, \ldots, m$, möglichst klein sein soll. Dabei erlauben die w_i noch eine unterschiedliche Gewichtung der einzelnen Messungen. Setzt man

$$\begin{array}{lll} \text{(a)} & A = (a_{ij}), & a_{ij} = w_i \varphi_j(t_i), \\ \text{(b)} & b = (b_i), & b_i = w_i f_i, \end{array} \tag{2.44}$$

so ist also zu $A \in \mathbb{R}^{m,n}$ und $b \in \mathbb{R}^m$ ein $x \in \mathbb{R}^n$ gesucht mit

$$\|Ax - b\|_2^2 = \min ! \tag{2.45}$$

Ein typisches Resultat, das auch eine hinreichende Bedingung für die Wohlgestelltheit des Problems beinhaltet, gibt der folgende Satz.

Satz 2.12
Die Spalten von $A \in \mathbb{R}^{m,n}$ seien linear unabhängig. Dann besitzt das Problem (2.45) eine eindeutige Lösung $x \in \mathbb{R}^n$. Diese ist gegeben als Lösung des linearen Gleichungssystems

$$A^T A x = A^T b, \qquad \text{(Normalgleichung)} \tag{2.46}$$

wobei $A^T A$ symmetrisch und positiv-definit ist.

□

Leider ist die Berechnung von x mittels (2.46), etwa durch Cholesky-Zerlegung von $A^T A$, nicht stabil. Als alternative Grundlage zur Entwicklung eines Algorithmus verwenden wir deshalb die folgende Eigenschaft.

Satz 2.13
Zu jeder Matrix $A \in \mathbb{R}^{m,n}$ mit linear unabhängigen Spalten gibt es eine orthogonale Matrix $Q \in \mathbb{R}^{m,m}$ und eine nichtsinguläre obere Dreiecksmatrix $R \in \mathbb{R}^{n,n}$, so daß

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}. \tag{2.47}$$

Beweis:
Analog zu Satz 2.10!

□

Damit gilt

$$\begin{aligned} \|Ax - b\|_2^2 &= \|Q \left[\begin{smallmatrix} R \\ 0 \end{smallmatrix}\right] x - b\|_2^2 = \|Q \left[\begin{smallmatrix} R \\ 0 \end{smallmatrix}\right] x - QQ^T b\|_2^2 = \\ &= \| \left[\begin{smallmatrix} R \\ 0 \end{smallmatrix}\right] x - Q^T b\|_2^2 = \| \left[\begin{smallmatrix} R \\ 0 \end{smallmatrix}\right] x - \left[\begin{smallmatrix} c \\ d \end{smallmatrix}\right] \|_2^2 = \\ &= \|Rx - c\|_2^2 + \|d\|_2^2, \quad \left[\begin{smallmatrix} c \\ d \end{smallmatrix}\right] = Q^T b, \end{aligned}$$

was gerade für

$$x = R^{-1} c \tag{2.48}$$

minimal wird.

Algorithmus 2.14
Man modifiziert einfach Algorithmus 2.11 dahingehend, daß die Schleifen bei der Berechnung der Größen τ, γ, a_{jk} und b_j bis m statt n laufen. Die Abfrage $\sigma = 0$ bedeutet dann, daß die Spalten von A nicht linear unabhängig sind.

□

Kapitel 3

Interpolation

Bei der (linearen) Interpolation werden zu gegebenen Daten (t_i, f_i), $t_i, f_i \in \mathbb{R}$, $i = 0, \ldots, n$ mit paarweise verschiedenen t_i, d. h. mit $t_i \neq t_j$ für $i \neq j$, und Funktionen $\varphi_j : \mathbb{R} \to \mathbb{R}$, $j = 0, \ldots, n$, Koeffizienten $x_j \in \mathbb{R}$ gesucht mit

$$\sum_{j=0}^{n} x_j \varphi_j(t_i) = f_i, \quad i = 0, \ldots, n, \tag{3.1}$$

vgl. Abschnitt 2.5. Gehören die Daten zu einer Funktion f, d. h. $f(t_i) = f_i$, so kann man mit der sogenannten Interpolierenden $\tilde{f}$, definiert durch

$$\tilde{f}(t) = \sum_{j=0}^{n} x_j \varphi_j(t), \tag{3.2}$$

näherungsweise Funktionswerte von f zwischen den Stützstellen t_i bestimmen. Man denke dabei an Funktionen f, deren Auswertung sehr teuer ist.

Setzt man

$$\begin{array}{lll} \text{(a)} & A = (a_{ij}), & a_{ij} = \varphi_j(t_i), \\ \text{(b)} & b = (b_i), & b_i = f_i, \end{array} \tag{3.3}$$

so ist (3.1) äquivalent zum linearen Gleichungssystem $Ax = b$. Im folgenden sollen für speziell gewählte φ_j effiziente Methoden vorgestellt werden.

3.1 Polynom-Interpolation

Gesucht ist hier ein Polynom p vom Grad höchstens n (man schreibt $p \in \Pi_n$), so daß

$$p(t_i) = f_i, \quad i = 0, \ldots, n. \tag{3.4}$$

Satz 3.1
Gegeben seien $t_i, f_i \in \mathbb{R}$, $i = 0, \ldots, n$, mit $t_i \neq t_j$ für $i \neq j$. Dann gibt es genau ein Polynom $p \in \Pi_n$ mit (3.4).

Beweis:
Existenz:
Die durch

$$l_j(t) = \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{t - t_k}{t_j - t_k}, \quad j = 0, \ldots, n$$

definierten Funktionen l_j sind offensichtlich Polynome in Π_n (sog. Lagrange-Polynome). Für diese gilt

$$l_j(t_i) = \delta_{ij} = \begin{cases} 1 & \text{für } i = j, \\ 0 & \text{für } i \neq j, \end{cases} \qquad i, j = 0, \ldots, n.$$

Damit genügt p definiert durch

$$p(t) = \sum_{j=0}^{n} f_j l_j(t)$$

den geforderten Eigenschaften.
Eindeutigkeit:
Seien $p_1, p_2 \in \Pi_n$ mit $p_1(t_i) = p_2(t_i) = f_i$, $i = 0, \ldots, n$. Dann gilt

$$(p_2 - p_1)(t_i) = 0, \quad i = 0, \ldots, n,$$

d. h. das Polynom $p_2 - p_1 \in \Pi_n$ besitzt die $n+1$ Nullstellen t_i, $i = 0, \ldots, n$. Ein Polynom vom Grad höchstens n mit mindestens $n+1$ Nullstellen muß aber das Nullpolynom sein, d. h. $p_2 - p_1 = 0$ bzw. $p_1 = p_2$.

$\square$

Um (3.4) in die Form (3.1) zu bringen, müssen wir in Π_n eine Basis auswählen. Von dieser Wahl wird der Aufwand, die Matrix A zu berechnen und das zugehörige Gleichungssystem zu lösen, abhängen. Darüberhinaus ist es wichtig, die zugehörige Darstellung (3.2) des Interpolationspolynoms effizient auswerten zu können.

Beispiel 3.2
Mögliche Basen sind z. B.

(a) Lagrange-Basis

$$\begin{aligned}\varphi_j(t) &= l_j(t), \quad j=0,\dots,n,\\ A &= (l_j(t_i)) = (\delta_{ij}) = I;\end{aligned}$$

(b) Monom-Basis

$$\begin{aligned}\varphi_j(t) &= t^j, \quad j=0,\dots,n,\\ A &= (t_i^j); \quad \text{(Vandermonde-Matrix)}\end{aligned}$$

(c) Newton-Basis

$$\begin{aligned}\varphi_j(t) &= \omega_j(t) = \textstyle\prod_{k=0}^{j-1}(t-t_k), \quad j=0,\dots,n,\\ A &= (\omega_j(t_i)). \quad \text{(untere Dreiecksmatrix wegen } \omega_j(t_i)=0 \text{ für } i<j)\end{aligned}$$

□

Benötigt man nur eine Auswertung des Interpolationspolynoms, so kann man folgendermaßen vorgehen.

Lemma 3.3
Gegeben seien $t_i, f_i \in \mathbb{R}$, $i=0,\dots,n$, mit $t_i \neq t_j$ für $i \neq j$. Bezeichnet $P(\cdot;t_k,\dots,t_{k+l}) \in \Pi_l$, $k \in \{0,\dots,n-l\}$, $l \in \{0,\dots,n\}$, das eindeutig bestimmte Interpolationspolynom zu den Knoten $t_k,\dots,t_{k+l}$, so gilt

$$P(t;t_k,\dots,t_{k+l}) = \frac{(t_k-t)P(t;t_{k+1},\dots,t_{k+l}) - (t_{k+l}-t)P(t;t_k,\dots,t_{k+l-1})}{t_k-t_{k+l}} \tag{3.5}$$

Beweis:
Bezeichnet $Q(t)$ die rechte Seite von (3.5), so gilt $Q \in \Pi_l$ wegen $P(\cdot;t_{k+1},\dots,t_{k+l}), P(\cdot;t_k,\dots,t_{k+l-1}) \in \Pi_{l-1}$. Für $i=k+1,\dots,k+l-1$ ist

$$Q(t_i) = \frac{(t_k-t_i)f_i - (t_{k+l}-t_i)f_i}{t_k-t_{k+l}} = f_i,$$

außerdem

$$Q(t_k) = \frac{-(t_{k+l}-t_k)f_k}{t_k-t_{k+l}} = f_k, \quad Q(t_{k+l}) = \frac{(t_k-t_{k+l})f_{k+l}}{t_k-t_{k+l}} = f_{k+l},$$

also

$$Q(t) = P(t; t_k, \ldots, t_{k+l}).$$

□

Algorithmus 3.4 (Verfahren von Aitken/Neville)
Schreibt man (3.5) in der Form

$$(3.6) \qquad \begin{aligned} P(t; t_k, \ldots, t_{k+l}) &= P(t; t_k, \ldots, t_{k+l-1}) - \\ &- \tfrac{t-t_k}{t_{k+l}-t_k}[P(t; t_k, \ldots, t_{k+l-1}) - P(t; t_{k+1}, \ldots, t_{k+l})], \end{aligned}$$

so ergibt sich der folgende Algorithmus zur Berechnung von $P(t; t_0, \ldots, t_n)$ für gegebenes $t \in \mathbb{R}$.

Setze $P_{k,0} = f_k$, $k = 0, \ldots, n$.	
Schleife $l = 1, \ldots, n$:	
	Setze $P_{k,l} = P_{k,l-1} - \frac{t-t_k}{t_{k+l}-t_k}(P_{k,l-1} - P_{k+1,l-1})$, $k = 0, \ldots, n-l$.

(3.7)

Das Ergebnis steht dann in $P_{0,n}$. Die Berechnung kann auf einem Vektor V ausgeführt werden, indem man $P_{k,l}$ nach V_{k+l} schreibt und die innere Schleife über k rückwärts laufen läßt.

□

Der Aufwand in (3.7) beträgt in erster Näherung

$$2\sum_{l=1}^{n}(n-l) = 2\sum_{l=0}^{n-1} l \doteq n^2 \text{ Multiplikationen/Divisionen.}$$

Beispiel 3.5
Zu $n = 3$ seien die Daten

i	0	1	2	3
t_i	-1	0	1	2
$f_i = P_{i,0}$	-2	-1	2	19

gegeben. Für $t = -2$ liefert Algorithmus 3.4

$$P_{0,1} = P_{0,0} - \frac{t-t_0}{t_1-t_0}(P_{0,0} - P_{1,0}) = (-2) - \frac{(-2)-(-1)}{0-(-1)}((-2)-(-1)) = -3,$$

entsprechend $P_{1,1} = -7$, $P_{2,1} = -49$. Weiterhin ist

$$P_{0,2} = P_{0,1} - \frac{t - t_0}{t_2 - t_0}(P_{0,1} - P_{1,1}) = (-3) - \frac{(-2) - (-1)}{1 - (-1)}((-3) - (-7)) = -1,$$

entsprechend $P_{1,2} = 35$, sowie

$$P_{0,3} = P_{0,2} - \frac{t - t_0}{t_3 - t_0}(P_{0,2} - P_{1,2}) = (-1) - \frac{(-2) - (-1)}{2 - (-1)}((-1) - 35) = -13.$$

□

Will man das Interpolationspolynom öfter auswerten, so verwendet man die Newton-Basis.

Definition 3.6 (Dividierte Differenzen)
Die sogenannten dividierten Differenzen sind rekursiv definiert durch

$$\begin{aligned} &\text{(a)} \quad f[t_k] = f_k, \ k = 0, \ldots, n, \\ (3.8) \quad &\text{(b)} \quad f[t_k, \ldots, t_{k+l}] = \tfrac{f[t_k, \ldots, t_{k+l-1}] - f[t_{k+1}, \ldots, t_{k+l}]}{t_k - t_{k+l}}, \\ &\qquad\qquad k = 0, \ldots, n-l, \ l = 1, \ldots, n. \end{aligned}$$

□

Lemma 3.7
Für $k = 0, \ldots, n-l$, $l = 0, \ldots, n$ gilt

$$\begin{aligned} P(t; t_k, \ldots, t_{k+l}) = {} & f[t_k] + f[t_k, t_{k+1}](t - t_k) + \\ (3.9) \qquad & + f[t_k, t_{k+1}, t_{k+2}](t - t_k)(t - t_{k+1}) + \cdots + \\ & + f[t_k, \ldots, t_{k+l}](t - t_k) \cdots (t - t_{k+l-1}). \end{aligned}$$

Beweis:
Wegen

$$P(t; t_k, \ldots, t_{k+l}) = P(t; t_k, \ldots, t_{k+l-1}) + f[t_k, \ldots, t_{k+l}](t - t_k) \cdots (t - t_{k+l-1})$$

genügt es zu zeigen, daß $f[t_k, \ldots, t_{k+l}]$ der Koeffizient der höchsten Potenz von t (nämlich t^l) ist.

Die Behauptung ist trivial für $l = 0$. Sei nun $l \geq 1$. Nach Induktionsvoraussetzung ist $f[t_{k+1}, \ldots, t_{k+l}]$ der Koeffizient der höchsten

Potenz von t in $P(t; t_{k+1}, \dots, t_{k+l})$, entsprechend $f[t_k, \dots, t_{k+l-1}]$ der in $P(t; t_k, \dots, t_{k+l-1})$. Wegen (3.5) ist dann der Koeffizient der höchsten Potenz von t in $P(t; t_k, \dots, t_{k+l})$ gegeben durch

$$\frac{-f[t_{k+1}, \dots, t_{k+l}] + f[t_k, \dots, t_{k+l-1}]}{t_k - t_{k+l}} = f[t_k, \dots, t_{k+l}],$$

letzteres wegen (3.8b).

□

Algorithmus 3.8
Der folgender Algorithmus berechnet die dividierten Differenzen für $P(t; t_0, \dots, t_n)$.

Setze $D_{k,0} = f_k$, $k = 0, \dots, n$.	
Schleife $l = 1, \dots, n$:	
	Setze $D_{k,l} = \frac{D_{k,l-1} - D_{k+1,l-1}}{t_k - t_{k+l}}$, $k = 0, \dots, n-l$.

(3.10)
Wegen

$$\begin{aligned} p(t) &= P(t; t_0, \dots, t_n) = \\ &= f[t_0] + f[t_0, t_1](t - t_0) + \dots + f[t_0, \dots, t_n](t - t_0) \cdots (t - t_{n-1}) \end{aligned}$$

werden nur die Werte $D_{0,l} = f[t_0, \dots, t_l]$, $l = 0, \dots, n$, benötigt. Entsprechend Algorithmus 3.4 kann man $D_{k,l}$ nach V_{k+l} speichern, um mit einem Vektor auszukommen. Dabei ist die innere Schleife über k wieder rückwärts zu durchlaufen.

□

Der Aufwand in (3.10) beträgt in erster Näherung

$$\textstyle\sum_{l=1}^{n} (n - l) \doteq \frac{1}{2} n^2 \text{ Divisionen.}$$

Algorithmus 3.9 (Horner-Schema)
Naives Vorgehen bei der Auswertung eines Polynoms

(3.11) $$p(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_n t^n$$

in der Darstellung bzgl. der Monom-Basis führt auf den Algorithmus

Setze $s = a_0$, $r = 1$.	
Schleife $i = 1, \ldots, n$:	
	Setze $r = r \cdot t$. Setze $s = s + a_i \cdot r$.
Setze $p(t) = s$.	

(3.12)

wobei $2n$ Multiplikationen benötigt werden. Schreibt man (3.11) stattdessen in der Form

$$p(t) = a_0 + t(a_1 + t(a_2 + \cdots + t(a_{n-1} + ta_n) \cdots)), \tag{3.13}$$

so erhält man das sogenannte Horner-Schema

Setze $s = a_n$.	
Schleife $i = 1, \ldots, n$:	
	Setze $s = s \cdot t + a_{n-i}$.
Setze $p(t) = s$.	

(3.14)

welches nur noch n Multiplikationen benötigt. Entsprechendes Vorgehen für

$$p(t) = V_0 + V_1(t-t_0) + V_2(t-t_0)(t-t_1) + \cdots + V_n(t-t_0)(t-t_1)\cdots(t-t_{n-1}) \tag{3.15}$$

liefert das modifizierte Horner-Schema

Setze $s = V_n$.	
Schleife $i = 1, \ldots, n$:	
	Setze $s = s(t - t_{n-i}) + V_{n-i}$.
Setze $p(t) = s$.	

(3.16)

□

Beispiel 3.10

Für die Daten aus Beispiel 3.5 erhält man mit Algorithmus 3.8 ausgehend von $D_{i,0} = f_i$

$$D_{0,1} = \frac{D_{0,0} - D_{1,0}}{t_0 - t_1} = \frac{(-2) - (-1)}{(-1) - 0} = \frac{-1}{-1} = 1,$$

entsprechend $D_{1,1} = 3$, $D_{2,1} = 17$. Weiterhin ist

$$D_{0,2} = \frac{D_{0,1} - D_{1,1}}{t_0 - t_2} = \frac{1-3}{(-1)-1} = 1,$$

entsprechend $D_{1,2} = 7$, sowie

$$D_{0,3} = \frac{D_{0,2} - D_{1,2}}{t_0 - t_3} = \frac{1-7}{(-1)-2} = 2.$$

Zur Vereinfachung der Handrechnung kann man auch das Schema

$$\begin{array}{ccccccccccc}
t_i & & f_i & & & & & & & & \\
\downarrow & & \downarrow & & & & & & & & \\
-1 & \to & -2 & & & & & & & & \\
 & & & \searrow & & & & & & & \\
 & & & & 1 & & & & & & \\
 & & & \nearrow & & \searrow & & & & & \\
0 & \to & -1 & & & & 1 & & & & \\
 & & & \searrow & & \nearrow & & \searrow & & & \\
 & & & & 3 & & & & 2 & & \\
 & & & \nearrow & & \searrow & & \nearrow & & \nwarrow & \\
1 & \to & 2 & & & & 7 & & & & V_{n-i} \\
 & & & \searrow & & \nearrow & & & & & \\
 & & & & 17 & & & & & & \\
 & & & \nearrow & & & & & & & \\
2 & \to & 19 & & & & & & & &
\end{array}$$

verwenden. Für $t = -2$ liefert das modifizierte Horner-Schema dann

$$\begin{array}{ll}
s = V_3 = 2, & \\
i = 1: & s = s(t - t_2) + V_2 = 2((-2) - 1) + 1 = -5, \\
i = 2: & s = s(t - t_1) + V_1 = (-5)((-2) - 0) + 1 = 11, \\
i = 3: & s = s(t - t_0) + V_0 = 11((-2) - (-1)) + (-2) = -13,
\end{array}$$

bzw. als Rechenschema

$$\begin{array}{rrrrl}
2 & 1 & 0 & -1 & \leftarrow t_i \\
-4 & -3 & -2 & -1 & \leftarrow t - t_i \\
\hline
2 & 1 & 1 & -2 & \leftarrow V_i \\
 & -6 & 10 & -11 & \\
\hline
2 & -5 & 11 & -13 &
\end{array}$$

$\square$

Es bleibt die Frage, wie gut ein Interpolationspolynom p, festgelegt durch

$$p(t_i) = f(t_i), \quad i = 0, \ldots, n, \quad p \in \Pi_n, \tag{3.17}$$

eine gegebene Funktion f approximiert.

Satz 3.11
Sei $\bar{t} \in \mathbb{R}$ und $f : \mathbb{I} \to \mathbb{R}$ $(n+1)$-mal stetig differenzierbar, wobei $\mathbb{I}$ das kleinste Intervall mit $\bar{t}, t_0, \ldots, t_n \in \mathbb{I}$ sei. Dann gibt es ein $\tau \in \mathbb{I}$ mit

$$f(\bar{t}) - p(\bar{t}) = \frac{f^{(n+1)}(\tau)}{(n+1)!}\omega(\bar{t}), \quad \omega(t) = \prod_{k=0}^{n}(t - t_k). \tag{3.18}$$

Beweis:
Wegen $p(t_i) = f(t_i)$ und $\omega(t_i) = 0$ für $i = 0, \ldots, n$ ist die Behauptung richtig für $\bar{t} = t_i$, $i = 0, \ldots, n$. Sei also im folgenden $\bar{t} \neq t_i$ für alle $i = 0, \ldots, n$. Dann ist $\omega(\bar{t}) \neq 0$ und es gibt ein $C \in \mathbb{R}$ mit

$$f(\bar{t}) - p(\bar{t}) - C\omega(\bar{t}) = 0,$$

Damit besitzt die Funktion $g = f - p - C\omega$ in $\mathbb{I}$ die $n + 2$ Nullstellen $\bar{t}, t_0, \ldots, t_n$. Nach dem Satz von Rolle besitzt $\dot{g}$ dann mindestens $n + 1$ Nullstellen in $\mathbb{I}$, entsprechend $\ddot{g}$ mindestens n Nullstellen usw. Also besitzt $g^{(n+1)}$ mindestens eine Nullstelle $\tau \in \mathbb{I}$. Wegen $p^{(n+1)} = 0$ gilt dafür

$$g^{(n+1)}(\tau) = f^{(n+1)}(\tau) - C(n+1)! = 0,$$

d. h.

$$C = \frac{f^{(n+1)}(\tau)}{(n+1)!}.$$

□

Bemerkung 3.12
Für größere n besitzen die Lagrange-Polynome l_j zwischen den Stützstellen t_i zum Teil Funktionswerte von sehr großem Betrag. Damit kann der Interpolationsfehler (3.18) ebenfalls sehr groß werden.

□

3.2 Spline-Interpolation

Um Effekte der Polynominterpolation bei größerem Grad n wie in Bemerkung 3.12 angedeutet zu vermeiden, kann man versuchen, Polynome geringeren Grades stückweise zu hinreichend glatten Funktionen zusammenzufügen. Man nennt solche Funktionen Splines.

Definition 3.13 (kubische Splines)
Gegeben sei eine Unterteilung

$$a = t_0 < t_1 < \cdots < t_n = b \tag{3.19}$$

des Intervalls $[a, b]$. Eine Funktion $s : [a, b] \to \mathbb{R}$ heißt (zu (3.19) gehöriger) kubischer Spline, wenn $s \in C^2([a, b], \mathbb{R})$ und $s|_{[t_i, t_{i+1}]}$ für jedes $i = 0, \dots, n-1$ auf $[t_i, t_{i+1}]$ mit einem kubischen Polynom übereinstimmt.

□

Satz 3.14
Gegeben seien $t_i, f_i \in \mathbb{R}$, $i = 0, \dots, n$, mit (3.19). Fordert man neben den Interpolationsbedingungen

$$s(t_i) = f_i, \quad i = 0, \dots, n \tag{3.20}$$

noch eine der drei Bedingungen

$$\begin{array}{ll} \text{(a)} & \ddot{s}(t_0) = 0,\ \ddot{s}(t_n) = 0, \\ & \qquad\qquad \text{(natürlicher Spline)} \\ \text{(b)} & \dot{s}(t_0) = \dot{f}_0,\ \dot{s}(t_n) = \dot{f}_n \text{ mit } \dot{f}_0, \dot{f}_n \in \mathbb{R}, \\ & \qquad\qquad \text{(eingespannter Spline)} \\ \text{(c)} & \dot{s}(t_0) = \dot{s}(t_n),\ \ddot{s}(t_0) = \ddot{s}(t_n), \\ & \qquad\qquad \text{(periodischer Spline)} \end{array} \tag{3.21}$$

so ist dadurch genau ein kubischer Spline festgelegt.

□

Im folgenden betrachten wir nur natürliche Splines, d. h. (3.20) zusammen mit (3.21a).

Zur Konstruktion von s setzen wir

$$h_i = t_{i+1} - t_i, \quad i = 0, \dots, n-1, \tag{3.22}$$

und machen auf $[t_i, t_{i+1}]$, $i = 0, \dots, n-1$, den Ansatz

$$
(3.23) \quad
\begin{array}{ll}
\text{(a)} & s_i(t) = a_i + b_i(t - t_i) + \frac{1}{2}c_i(t - t_i)^2 + \frac{1}{6}d_i(t - t_i)^3, \\
\text{(b)} & \dot{s}_i(t) = b_i + c_i(t - t_i) + \frac{1}{2}d_i(t - t_i)^2, \\
\text{(c)} & \ddot{s}_i(t) = c_i + d_i(t - t_i).
\end{array}
$$

Zu erfüllen sind die Bedingungen

$$
(3.24) \quad
\begin{array}{ll}
\text{(a)} & s_i(t_i) = f_i,\ s_i(t_{i+1}) = f_{i+1},\ i = 0, \dots, n-1, \\
\text{(b)} & \dot{s}_i(t_{i+1}) = \dot{s}_{i+1}(t_{i+1}),\ i = 0, \dots n-2, \\
\text{(c)} & \ddot{s}_i(t_{i+1}) = \ddot{s}_{i+1}(t_{i+1}),\ i = 0, \dots, n-2, \\
\text{(d)} & \ddot{s}_0(t_0) = 0,\ \ddot{s}_{n-1}(t_n) = 0.
\end{array}
$$

Aus (3.24a) ergibt sich sofort

$$
(3.25) \qquad a_i = f_i, \quad i = 0, \dots, n-1,
$$

und aus (3.24c)

$$
(3.26) \qquad c_{i+1} = c_i + d_i h_i
$$

oder (mit $c_0 = c_n = 0$ nach (3.24d))

$$
(3.27) \qquad d_i = \frac{c_{i+1} - c_i}{h_i}, \quad i = 0, \dots, n-1.
$$

Damit folgt aus (3.24a)

$$
b_i = \frac{f_{i+1} - f_i}{h_i} - \frac{1}{2}c_i h_i - \frac{1}{6}\frac{c_{i+1} - c_i}{h_i}h_i^2 = \frac{f_{i+1} - f_i}{h_i} - \frac{2c_i + c_{i+1}}{6}h_i,\ i = 0, \dots, n-1,
$$
(3.28)

und alle Größen sind in den Unbekannten c_i ausgedrückt. Aus (3.24b) ergibt sich nun

$$
(3.29) \qquad b_i = b_{i-1} + c_{i-1}h_{i-1} + \frac{1}{2}d_{i-1}h_{i-1}^2, \quad i = 1, \dots, n-1,
$$

bzw.

$$
\frac{f_{i+1} - f_i}{h_i} - \frac{2c_i + c_{i+1}}{6}h_i = \frac{f_i - f_{i-1}}{h_{i-1}} - \frac{2c_{i-1} + c_i}{6}h_{i-1} + c_{i-1}h_{i-1} + \frac{1}{2}\frac{c_i - c_{i-1}}{h_{i-1}}h_{i-1}^2,
$$
(3.30)

oder

$$
(3.31) \qquad \frac{h_{i-1}}{6}c_{i-1} + \frac{h_{i-1} + h_i}{3}c_i + \frac{h_i}{6}c_{i+1} = \frac{f_{i+1} - f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}},
$$

jeweils mit $i = 1, \ldots, n-1$. Zusammen mit $c_0 = c_n = 0$ erhält man also für die Berechnung der c_i, $i = 1, \ldots, n-1$, ein lineares Gleichungssystem mit symmetrischer, tridiagonaler Koeffizientenmatrix. Aus der Größe der Einträge kann man schließen, daß diese positiv-definit und damit nichtsingulär ist. Damit erhält man aus (3.31) eine eindeutige Lösung c_i, $i = 1, \ldots, n-1$. Die Beziehungen (3.25), (3.27) und (3.28) liefern dann eine vollständige Darstellung (3.23a) des gesuchten Splines s.

Kapitel 4

Differentiation

Zu gegebenem $f \in \mathbb{X} = C^1([a,b],\mathbb{R})$ und $\bar{t} \in [a,b]$ ist der Wert $\dot{f}(\bar{t})$ zu bestimmen. Darf man als Daten nur Funktionswerte von f verwenden, so ist dieses Problem schlecht gestellt.

Beispiel 4.1
Sei $f : \mathbb{R} \to \mathbb{R}$ gegeben durch

$$f(t) = \tanh(ct)$$

mit $c > 0$. Wegen

$$\dot{f}(t) = \frac{c}{\cosh^2(ct)}$$

gilt für $\bar{t} = 0$

$$\dot{f}(\bar{t}) = c,$$

d. h. obwohl

$$|f(t)| < 1 \text{ für alle } t \in \mathbb{R}$$

gilt, kann $|\dot{f}(\bar{t})|$ beliebig groß werden.

$\square$

Trotz der Schlechtgestelltheit des Problems werden in manchen Situationen numerische Verfahren zur Differentiation verwendet. Man muß dann besonders genau auf die Auswirkung von Rundungsfehlern achten.

4.1 Differenzenverfahren

Die Idee bei den Differenzenverfahren ist, die gegebene Funktion f durch ein Interpolationspolynom p zu ersetzen und $\dot{p}(\bar{t})$ als Approximation an $\dot{f}(\bar{t})$ zu verwenden. Da man dabei ein Element eines unendlichdimensionalen Vektorraums durch ein Element eines endlichdimensionalen Vektorraums ersetzt, spricht man von Diskretisierung.

Interpoliert man f in den Stellen t_i, $i = 0, \ldots, n$, so erhält man das Polynom

$$p(t) = \sum_{j=0}^{n} f_j l_j(t), \quad l_j(t) = \prod_{\substack{k=0\\k\neq j}}^{n} \frac{t - t_k}{t_j - t_k} \tag{4.1}$$

mit $f_j = f(t_j)$. Es gilt dann

$$\dot{p}(t) = \sum_{j=0}^{n} f_j \dot{l}_j(t), \quad \dot{l}_j(t) = \sum_{\substack{l=0\\l\neq j}}^{n} \frac{1}{t_j - t_l} \prod_{\substack{k=0\\k\neq j,l}}^{n} \frac{t - t_k}{t_j - t_k}. \tag{4.2}$$

Typischerweise wählt man das Gitter äquidistant, d. h. $t_i = t_0 + ih$, $i = 0, \ldots, n$, mit einer Schrittweite $h > 0$, und so, daß $\bar{t} = t_j$ ein Gitterpunkt ist.

Beispiel 4.2
Für $n = 1$ und $t_0 = \bar{t}$, $t_1 = \bar{t} + h$ erhält man aus

$$\dot{p}(t_0) = f_0 \frac{1}{t_0 - t_1} + f_1 \frac{1}{t_1 - t_0} = \frac{f_1 - f_0}{h}$$

die Approximation (mit t statt $\bar{t}$)

$$\dot{f}(t) \approx \frac{f(t+h) - f(t)}{h}, \tag{4.3}$$

d. h. man hat gerade die Grenzwertbildung bei der Definition der Ableitung rückgängig gemacht (der Rechner kann keinen Limes!).

Für $n = 2$ und $t_0 = \bar{t} - h$, $t_1 = \bar{t}$, $t_2 = \bar{t} + h$ erhält man aus

$$\begin{aligned}\dot{p}(t_1) = {} & f_0[\tfrac{1}{t_0-t_1}\tfrac{t_1-t_2}{t_0-t_2} + \tfrac{1}{t_0-t_2}\tfrac{t_1-t_1}{t_0-t_1}] + \\ & + f_1[\tfrac{1}{t_1-t_0}\tfrac{t_1-t_2}{t_1-t_2} + \tfrac{1}{t_1-t_2}\tfrac{t_1-t_0}{t_1-t_0}] + \\ & + f_2[\tfrac{1}{t_2-t_0}\tfrac{t_1-t_1}{t_2-t_1} + \tfrac{1}{t_2-t_1}\tfrac{t_1-t_0}{t_2-t_0}] = \\ = {} & f_0(-\tfrac{1}{2h}) + f_1(\tfrac{1}{h} - \tfrac{1}{h}) + f_2(\tfrac{1}{2h})\end{aligned}$$

die Approximation

$$\dot{f}(t) \approx \frac{f(t+h) - f(t-h)}{2h}. \tag{4.4}$$

□

Der Fehler, den man durch Ersetzen von f durch p macht, heißt Diskretisierungsfehler. Mit Hilfe der Darstellung (3.18) des Interpolationsfehlers kann man folgende Eigenschaft des Diskretisierungsfehlers zeigen.

Satz 4.3
Bei äquidistantem Gitter gilt für $f \in C^{n+1}([a,b],\mathbb{R})$

$$|\dot{f}(\bar{t}) - \dot{p}(\bar{t})| \leq Ch^n \tag{4.5}$$

mit $C \geq 0$ unabhängig von h.

□

Für $h \to 0$ gilt also für den Diskretisierungsfehler $|\dot{f}(\bar{t}) - \dot{p}(\bar{t})| \to 0$. Andererseits hat man für $h \to 0$ starke Auslöschungseffekte bei der Berechnung von $\dot{p}(\bar{t})$, da dann alle f_i ungefähr gleich sind. Für $h \to 0$ nehmen also die Rundungsfehler zu. Man darf demnach h weder zu groß noch zu klein wählen. Eine mögliche Wahl für (4.3) ist

$$h = \sqrt{\text{eps}}\,|f(\bar{t})|. \tag{4.6}$$

Ist $|\dot{f}(\bar{t})| \approx 1$, so unterscheiden sich $f(\bar{t})$ und $f(\bar{t}+h)$ in etwa auf der zweiten Hälfte der Mantisse. Die erste Hälfte wird durch Auslöschung zugunsten des Diskretisierungsfehlers geopfert.

4.2 Symbolisches/automatisches Differenzieren

Jede auf einem Rechner implementierte Funktion f besteht aus einer endlichen Komposition von Elementaroperationen. Die Ableitung einer Elementaroperation setzt sich aber wieder aus einer endlichen Zahl von Elementaroperationen zusammen. Man kann also in einem Programm zur Berechnung von $f(x)$ jede auftretende Variable als Funktion von x auffassen. Führt man für jede Variable eine neue Variable für den Ableitungswert ein, so erhält man ein Programm zur Berechnung von $f'(x)$, indem man vor jede Zuweisung eine entsprechende abgeleitete Zuweisung setzt, sog. symbolisches Differenzieren, vgl. Folie.

In einer Programmiersprache, in der man wie etwa in `C++` Operatoren überladen kann, kann man auch einen Variablentyp definieren, der aus dem ursprünglichen Wert und dem Ableitungswert besteht, und dazu alle Elementaroperationen derart überladen, daß neben den üblichen Berechnungen mit Hilfe der Ableitungsregeln der Wert der Ableitung berechnet wird, sog. automatisches Differenzieren.

Kapitel 5

Integration

Sei $a < b$ und $f \in \mathbb{X} = C([a,b],\mathbb{R})$. Zu bestimmen ist

$$I(f) = \int_a^b f(t)\,dt, \tag{5.1}$$

d. h. wir betrachten die Abbildung

$$I : \mathbb{X} \to \mathbb{R}, \quad f \mapsto \int_a^b f(t)\,dt. \tag{5.2}$$

Für diese gilt

$$\begin{aligned} |I(f_2) - I(f_1)| = |\textstyle\int_a^b (f_2(t) - f_1(t))\,dt| \le \int_a^b |f_2(t) - f_1(t)|\,dt \le \\ \le (b-a)\max_{t\in[a,b]} |f_2(t) - f_1(t)| = (b-a)\|f_2 - f_1\|_{\mathbb{X}}. \end{aligned} \tag{5.3}$$

d. h. die Kondition (bzgl. des absoluten Fehlers) wird beschrieben durch $\kappa = b - a$.

5.1 Newton/Cotes-Formeln

Wie bei der Differentiation kann das obige Problem mittels Polynominterpolation diskretisiert werden. Ausgehend von

$$a \le t_0 < t_1 < \cdots < t_n \le b \tag{5.4}$$

und Daten (t_i, f_i), $i = 0, \ldots, n$, mit $f_i = f(t_i)$ erhält man ein Interpolationspolynom $p \in \Pi_n$. Integration liefert

$$\int_a^b p(t)\,dt = \int_a^b \sum_{j=0}^n f_j l_j(t)\,dt = \sum_{j=0}^n f_j \int_a^b l_j(t)\,dt = \sum_{j=0}^n b_j f_j \tag{5.5}$$

mit sogenannten Gewichten

$$b_j = \int_a^b l_j(t)\,dt, \quad j = 0,\dots,n. \tag{5.6}$$

Damit hat man eine Nährungsformel der Form

$$I(f) \approx \sum_{j=0}^{n} b_j f(t_j) \tag{5.7}$$

für das Integral (5.1) erhalten. Man nennt Näherungsformeln zur Integration auch Quadraturformeln.

Bemerkung 5.1
Per Konstruktion sind die Quadraturformeln (5.7) mit (5.6) exakt für alle Polynome $p \in \Pi_n$, d. h.

$$I(p) = \textstyle\sum_{j=0}^{n} b_j p(t_j) \text{ für alle } p \in \Pi_n. \tag{5.8}$$

Wählt man als Basis in Π_n die Monom-Basis, so erfüllen die b_j die Beziehung

$$\sum_{j=0}^{n} b_j t_j^i = \int_a^b t^i\,dt = \frac{1}{i+1}(b^{i+1} - a^{i+1}), \quad i = 0,\dots,n. \tag{5.9}$$

Die b_j genügen also einem linearen Gleichungssystem mit einer nichtsingulären Vandermonde-Matrix als Koeffizientenmatrix.

□

Definition 5.2
Ist eine Quadraturformel exakt für alle $p \in \Pi_m$, aber nicht exakt für ein $p \in \Pi_{m+1}$, so heißt $m+1$ die Ordnung der Quadraturformel.

□

Nach Bemerkung 5.1 haben also die Quadraturformeln (5.7) mit (5.6) mindestens die Ordnung $n+1$.

Lemma 5.3
Sei $f \in C^{n+1}([a,b],\mathbb{R})$. Dann gilt für (5.7) mit (5.6)

$$|I(f) - I(p)| \le CH^{n+2}, \tag{5.10}$$

wobei $H = b - a$ und $C \geq 0$ unabhängig von H.

Beweis:
Mit (3.18) erhält man

$$|I(f) - I(p)| = |\int_a^b (f(t) - p(t))\,dt| = |\int_a^b \frac{f^{(n+1)}(\tau(t))}{(n+1)!}\omega(t)\,dt| \leq$$
$$\leq (b-a)\frac{\max_{t\in[a,b]}|f^{(n+1)}(t)|}{(n+1)!}(b-a)^{n+1} = CH^{n+2}.$$

□

Sei das Gitter im folgenden äquidistant gemäß

$$t_i = a + ih, \quad i = 0, \ldots, n, \quad h = \frac{b-a}{n} \tag{5.11}$$

Dann gilt mit $t = a + sh$

$$b_j = \int_a^b \prod_{\substack{k=0\\k\neq j}}^{n} \frac{t - t_k}{t_j - t_k}\,dt = h\int_0^n \prod_{\substack{k=0\\k\neq j}}^{n} \frac{s-k}{j-k}\,ds = h\sigma_j. \tag{5.12}$$

Die so erhaltenen Quadraturformeln heißen Newton/Cotes-Formeln. Diese sind symmetrisch in folgendem Sinn, vgl. Übungen.

Definition 5.4
Eine Quadraturformel (5.7) heißt symmetrisch, wenn gilt

$$t_{n-j} = (a+b) - t_j, \quad b_{n-j} = b_j, \qquad j = 0, \ldots, n. \tag{5.13}$$

□

Lemma 5.5
Gegeben sei eine symmetrische Quadraturformel (5.7) mit mindestens ungerader Ordnung $m+1$. Dann hat diese mindestens die Ordnung $m+2$.

Beweis:
Nach Voraussetzung ist die Quadraturformel exakt für alle $p \in \Pi_m$. Setzt man speziell

$$p(t) = \left(t - \frac{a+b}{2}\right)^{m+1},$$

so gilt

$$\int_a^b p(t)\,dt = 0,$$

da p bzgl. der Intervallmitte $\frac{a+b}{2}$ punktsymmetrisch ist. Auf der anderen Seite gilt

$$\sum_{j=0}^{n} b_j \left(t_j - \frac{a+b}{2}\right)^{m+1} = \frac{1}{2}\sum_{j=0}^{n}\left[b_j \left(t_j - \frac{a+b}{2}\right)^{m+1} + b_{n-j}\left(t_{n-j} - \frac{a+b}{2}\right)^{m+1}\right] = 0,$$

da $m+1$ ungerade ist. Also wird dieses spezielle $p \in \Pi_{m+1}$ ebenfalls von der Quadraturformel exakt integriert. Wegen der Linearität von Integral und Quadraturformel ist diese dann für alle $p \in \Pi_{m+1}$ exakt.

□

Damit ist gezeigt, daß symmetrische Quadraturformeln immer eine gerade Ordnung besitzen.

Beispiel 5.6 (Newton/Cotes-Formeln)
Für $n = 1$ gilt

$$\sigma_0 = -\textstyle\int_0^1 (s-1)\,ds = -(\frac{s^2}{2} - s)|_0^1 = \frac{1}{2}, \quad \sigma_1 = \int_0^1 s\,ds = \frac{1}{2}$$

und man erhält die sogenannte Trapez-Regel

$$I(f) \approx \frac{b-a}{2}(f(a) + f(b)) \tag{5.14}$$

mit der Ordnung 2.

Für $n = 2$ gilt

$$\begin{aligned}
\sigma_0 &= \textstyle\int_0^2 \frac{s-1}{0-1}\frac{s-2}{0-2}\,ds = \frac{1}{2}\int_0^2 (s^2 - 3s + 2)\,ds = \frac{1}{3},\\
\sigma_1 &= \textstyle\int_0^2 \frac{s-0}{1-0}\frac{s-2}{1-2}\,ds = -\int_0^2 (s^2 - 2s)\,ds = \frac{4}{3},\\
\sigma_2 &= \textstyle\int_0^2 \frac{s-2}{2-0}\frac{s-1}{2-1}\,ds = \frac{1}{2}\int_0^2 (s^2 - s)\,ds = \frac{1}{3}
\end{aligned}$$

und man erhält die sogenannte Simpson-Regel

$$I(f) \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] \tag{5.15}$$

mit Ordnung 4, vgl. Lemma 5.5.

□

Zur Verkleinerung des Diskretisierungsfehlers kann man vorab das Intervall $[a,b]$ unterteilen und die Quadraturformel auf die Teilintervalle anwenden. Man spricht von summierten Quadraturformeln.

Wählt man die Unterteilungspunkte

$$t_k = a + kH, \quad k = 0, \ldots, N, \quad H = \frac{b-a}{N}, \tag{5.16}$$

so interpoliert man im Fall der Newton/Cotes-Formeln den Integranden f auf jedem Teilintervall $[t_k, t_{k+1}]$ in den Punkten

$$t_{ik} = t_k + ih, \quad i = 0, \ldots, n, \quad h = \frac{H}{n} = \frac{b-a}{Nn}. \tag{5.17}$$

Bezeichnet p_k das Interpolationspolynom auf $[t_k, t_{k+1}]$ und I_k die Integration über $[t_k, t_{k+1}]$, so hat die summierte Formel die Form

$$I(f) \approx \sum_{k=0}^{N-1} I_k(p_k). \tag{5.18}$$

Satz 5.7
Sei $f \in C^{n+1}([a,b], \mathbb{R})$. Dann gilt für die summierten Newton/Cotes-Formeln

$$\left| I(f) - \sum_{k=0}^{N-1} I_k(p_k) \right| \leq C h^{n+1} \tag{5.19}$$

mit $C \geq 0$ unabhängig von h.

□

5.2 Gitteranpassung

Um die Genauigkeit einer Quadraturformel zu erhöhen, kann man die Intervalle verkleinern, auf die man sie anwendet. Für die Effizienz (gemessen etwa durch die Anzahl der Funktionsauswertungen) sollte die Länge der Teilintervalle an den jeweiligen Funktionsverlauf angepaßt werden. Man spricht von Gitteranpassung oder auch von Schrittweitensteuerung. Eine Methode soll hier vorgestellt werden.

Ausgehend davon, daß man die Integration von a bis zu einem Punkt $t_k \in [a, b)$ bereits durchgeführt hat, möchte man ein in irgendeiner Weise passendes t_{k+1} für das nächste Teilintervall wählen. Ausgehend von

$$a = t_0 < t_1 < \cdots < t_k < t_{k+1} < \cdots < b \tag{5.20}$$

erhält man

$$t_{k+1} = t_k + h_k \tag{5.21}$$

mit einem Schrittweitenvorschlag h_k, der aus dem letzten Schritt stammt oder für $k = 0$ vorgegeben ist.

Zu berechnen ist

$$I_k(f) = \int_{t_k}^{t_{k+1}} f(t)\, dt. \tag{5.22}$$

Dazu bezeichne $\tilde{I}_k$ das Ergebnis einer Quadraturformel der Ordnung q und $\hat{I}_k$ das Ergebnis einer Quadraturformel mit einer Ordnung höher als q, jeweils angewandt auf das Intervall $[t_k, t_{k+1}]$. Es muß jetzt geklärt werden, ob h_k eine sinnvolle Schrittweite ist.

In Anlehnung an (5.10) macht man die Annahme

$$|I_k(f) - \tilde{I}_k(f)| \approx C h_k^{q+1} \tag{5.23}$$

mit einer Konstanten C. Außerdem sei $\hat{I}_k$ wesentlich genauer als $\tilde{I}_k$, so daß die Annahme

$$\hat{I}_k(f) \approx I_k(f) \tag{5.24}$$

gerechtfertigt ist. Wegen

$$\mathtt{ERR} = |\hat{I}_k(f) - \tilde{I}_k(f)| \approx |I_k(f) - \tilde{I}_k(f)| \tag{5.25}$$

macht man dann die Modellannahme

$$\mathtt{ERR} = C h_k^{q+1}. \tag{5.26}$$

Für eine vorgegebene Toleranz $\mathtt{TOL} > 0$ soll die Bedingung

$$\mathtt{ERR} \leq \mathtt{TOL} \tag{5.27}$$

für die Genauigkeit der Integration eingehalten werden. Zu $\mathtt{TOL}$ gehört eine "optimale" Schrittweite $\bar{h}$ gemäß

$$\mathtt{TOL} = C \bar{h}^{q+1}. \tag{5.28}$$

Elimination von C liefert

$$\frac{\texttt{ERR}}{\texttt{TOL}} = \left(\frac{h_k}{\bar{h}}\right)^{q+1} \tag{5.29}$$

bzw.

$$\bar{h} = h_k \sqrt[q+1]{\frac{\texttt{TOL}}{\texttt{ERR}}} \tag{5.30}$$

Dabei ist (5.27) äquivalent zu

$$\bar{h} \geq h_k \tag{5.31}$$

Diese Überlegungen führen zu folgender Strategie: Ist $\texttt{ERR} > \texttt{TOL}$, so wird der Integrationsschritt für $[t_k, t_{k+1}]$ verworfen und mit $h_k = \bar{h}$ wiederholt. Ist $\texttt{ERR} \leq \texttt{TOL}$, so wird der Schritt akzeptiert, $\tilde{I}_k(f)$ oder $\hat{I}_k(f)$ wird zum aktuellen Wert des Integrals addiert und der nächste Schritt wird mit $h_{k+1} = \bar{h}$ durchgeführt.

Um zu viele Ablehnungen von Schritten zu vermeiden, wird (5.30) typischerweise in etwas konservativerer Form

$$\bar{h} = \texttt{RED} \cdot h_k \sqrt[q+1]{\frac{\texttt{TOL}}{\texttt{ERR} + \texttt{SAV}}} \tag{5.32}$$

verwendet. Außerdem wird die Änderung der Schrittweite eingeschränkt durch die Forderung

$$\texttt{RMIN} \leq \frac{\bar{h}}{h_k} \leq \texttt{RMAX}. \tag{5.33}$$

Die Wahl der Konstanten $\texttt{RED}$, $\texttt{SAV}$ sowie $\texttt{RMIN}$, $\texttt{RMAX}$ ist von philosophischer Natur. Eine mögliche Wahl wäre

$$\texttt{RED} = 0.9,\ \texttt{SAV} = \text{eps},\ \texttt{RMIN} = 0.2,\ \texttt{RMAX} = 2.0\,.$$

Strittig ist auch die Frage, ob man $\tilde{I}_k(f)$ oder $\hat{I}_k(f)$ zur gewünschten Approximation von $I(f)$ aufaddieren sollte (für $\tilde{I}_k(f)$ hat man eine Fehlerschätzung, wobei allerdings angenommen wurde, daß $\hat{I}_k(f)$ der genauere Wert ist).

Algorithmus 5.8

Gegeben sei $h_0 \leq b - a$, $t_0 = a$ sowie $\texttt{TOL} > 0$.

Setze $k = 0$; $I = 0$.	
Schleife über erlaubte Anzahl von Versuchen:	
	Setze $t_{k+1} = t_k + h_k$.
	Bestimme $\bar{h}$ entsprechend (5.32) und (5.33).
	Ist $\texttt{ERR} > \texttt{TOL}$, so setze $h_k = \bar{h}$ und beginne Schleife.
	Ist $\texttt{ERR} \leq \texttt{TOL}$, so setze $h_{k+1} = \min\{\bar{h}, b - t_{k+1}\}$, $I = I + \hat{I}_k(f)$ und beginne Schleife mit $k = k + 1$.

(5.34)

□

Man beachte, daß man bei dieser Methode eine Richtung (von a nach b) auszeichnet.

Kapitel 6

Nichtlineare Gleichungssysteme

Gesucht ist eine Lösung x^* der Gleichung

$$f(x) = 0, \tag{6.1}$$

wobei $f : \mathbb{D} \to \mathbb{R}^n$, $\mathbb{D} \subseteq \mathbb{R}^n$, stetig sei.

Ist f stetig differenzierbar und $f'(x^*)$ nichtsingulär, so besagt der Satz über die Umkehrfunktion, daß f in einer Umgebung $\mathbb{V}$ von $y^* = 0$ eine Inverse f^{-1} besitzt. Insbesondere ist $x^* = f^{-1}(y^*)$ in $\mathbb{U} = f^{-1}(\mathbb{V})$ die einzige Nullstelle von f (Wohlgestelltheit des Problems) und es gilt nach Abschnitt 1.3 für die differentielle Kondition

$$\kappa = \|(f^{-1})'(y^*)\| = \|f'(x^*)^{-1}\|. \tag{6.2}$$

6.1 Iterationsverfahren

Zur Lösung von (6.1) gibt es im allgemeinen keine Lösungsformel, d. h. keine geschlossene Darstellung der Umkehrfunktion. Stattdessen überführt man (6.1) in eine sogenannte Fixpunktform

$$x = \varphi(x), \tag{6.3}$$

die zu (6.1) äquivalent ist, d. h.

$$f(x^*) = 0 \quad \Longleftrightarrow \quad x^* = \varphi(x^*). \tag{6.4}$$

Man nennt ein x^* mit $x^* = \varphi(x^*)$ auch einen Fixpunkt von φ.

Die Fixpunktform (6.3) legt dann ein iteratives Vorgehen gemäß

$$x_{\nu+1} = \varphi(x_\nu), \quad \nu \in \mathbb{N}_0, \quad x_0 \text{ gegeben}, \tag{6.5}$$

nahe mit der Hoffnung, daß dadurch eine Folge $\{x_\nu\}_{\nu\in\mathbb{N}_0}$ definiert wird, für die $x_\nu \to x^*$ gilt.

Ein Problem (6.1) kann auf vielfältige Weise in eine Fixpunktform (6.3) gebracht werden, von denen nicht jede zum Ziel führt.

Beispiel 6.1

Die Gleichung

$$e^x + x = 0$$

besitzt eine eindeutige Lösung x^*, die bei zehnstelliger Rechnung gegeben ist durch

$$x^* = -0.56714\,32904\,.$$

Ausgehend von $x_0 = -0.5$ erhält man für

$$x_{\nu+1} = -e^{x_\nu}$$

z. B.

$$x_9 = -0.56755\,96343\,,$$

während für

$$x_{\nu+1} = \log(-x_\nu)$$

die Iterierte x_5 schon nicht mehr definiert ist.

$\square$

Man beachte, daß selbst im Fall $x_\nu \to x^*$ auf dem Rechner die Iteration irgendwann abgebrochen werden muß und das letzte berechnete x_ν als x^* akzeptiert werden muß. Der zugehörige Fehler $x^* - x_\nu$ heißt Abbrechfehler.

Satz 6.2 (Banachscher Fixpunktsatz)

Gegeben sei $\varphi : \mathbb{D} \to \mathbb{D}$ mit $\mathbb{D} \subseteq \mathbb{R}^n$ abgeschlossen. Ist φ kontraktiv, d. h. gilt

$$\|\varphi(x_2) - \varphi(x_1)\| \le L\,\|x_2 - x_1\| \text{ für alle } x_1,\, x_2 \in \mathbb{D} \tag{6.6}$$

mit $L < 1$, so ist durch (6.5) für jedes $x_0 \in \mathbb{D}$ eine Folge $\{x_\nu\}_{\nu\in\mathbb{N}_0}$ definiert, die gegen den einzigen Fixpunkt x^* von φ konvergiert. Insbesondere gilt

$$\left.\begin{array}{ll} \text{(a)} & \|x^* - x_{\nu+1}\| \le L\,\|x^* - x_\nu\|, \\ \text{(b)} & \|x^* - x_\nu\| \le \frac{L^\nu}{1-L}\|x_1 - x_0\|, \end{array}\right. \quad \nu \in \mathbb{N}_0, \tag{6.7}$$

und man spricht von linearer Konvergenz.

Beweis:

Wegen $\varphi(\mathbb{D}) \subseteq \mathbb{D}$ können die Iterierten $\mathbb{D}$ nicht verlassen. Weiterhin gilt

$$\|x_{\nu+1} - x_\nu\| = \|\varphi(x_\nu) - \varphi(x_{\nu-1})\| \leq L\,\|x_\nu - x_{\nu-1}\|$$

und damit induktiv

$$\|x_{\nu+1} - x_\nu\| \leq L^\nu \|x_1 - x_0\|.$$

Wegen

$$\begin{aligned}
\|x_{\nu+\mu} - x_\nu\| &= \|x_{\nu+\mu} - x_{\nu+\mu-1} + \cdots - x_{\nu+1} + x_{\nu+1} - x_\nu\| \leq \\
&\leq \|x_{\nu+\mu} - x_{\nu+\mu-1}\| + \cdots + \|x_{\nu+2} - x_{\nu+1}\| + \|x_{\nu+1} - x_\nu\| \leq \\
&\leq (L^{\nu+\mu-1} + \cdots + L^{\nu+1} + L^\nu)\|x_1 - x_0\| = \\
&= L^\nu (L^{\mu-1} + \cdots + L + 1)\|x_1 - x_0\| \leq \tfrac{L^\nu}{1-L}\|x_1 - x_0\|
\end{aligned}$$

ist $\{x_\nu\}_{\nu\in\mathbb{N}_0}$ eine Cauchy-Folge. Da $\mathbb{R}^n$ vollständig ist, gibt es ein $x^* \in \mathbb{R}^n$ mit $x_\nu \to x^*$. Da $x_\nu \in \mathbb{D}$ für $\nu \in \mathbb{N}_0$ und $\mathbb{D}$ abgeschlossen ist, folgt $x^* \in \mathbb{D}$.

Wegen

$$\begin{aligned}
\|x^* - \varphi(x^*)\| &= \|x^* - x_{\nu+1} + \varphi(x_\nu) - \varphi(x^*)\| \leq \\
&\leq \|x^* - x_{\nu+1}\| + L\,\|x_\nu - x^*\| \to 0
\end{aligned}$$

muß $\|x^* - \varphi(x^*)\| = 0$ gelten, d. h. $x^* = \varphi(x^*)$.

Sind x^* und x^{**} Fixpunkte von φ, so gilt

$$\|x^{**} - x^*\| = \|\varphi(x^{**}) - \varphi(x^*)\| \leq L\,\|x^{**} - x^*\|$$

bzw.

$$(1 - L)\|x^{**} - x^*\| \leq 0$$

oder

$$\|x^{**} - x^*\| \leq 0.$$

Also gilt $\|x^{**} - x^*\| = 0$ bzw. $x^{**} = x^*$.

Schließlich ergibt sich (6.7a) direkt aus (6.6) für $x_1 = x_\nu$, $x_2 = x^*$ und (6.7b) aus der Abschätzung für $\|x_{\nu+\mu} - x_\nu\|$ für $\mu \to \infty$.

$\square$

Die Voraussetzungen des Banachschen Fixpunktsatzes sind oft schwierig nachzuweisen. Bei dem folgenden Satz ist es oft einfacher. Dafür setzt er die Existenz eines Fixpunktes voraus.

Satz 6.3
Sei $x^* \in \mathbb{D}$ ein Fixpunkt von $\varphi : \mathbb{D} \to \mathbb{R}^n$, $\mathbb{D} \subseteq \mathbb{R}^n$ offen. Weiter sei $\varphi \in C^1(\mathbb{D}, \mathbb{R}^n)$ und

$$\|\varphi'(x^*)\| < 1 \tag{6.8}$$

in einer Operatornorm. Dann existiert eine Umgebung $\mathbb{U} \subseteq \mathbb{D}$ von x^*, sodaß $\varphi|_{\mathbb{U}}$ die Voraussetzungen des Banachschen Fixpunktsatzes erfüllt.

Beweis:
Sei $\delta = 1 - \|\varphi'(x^*)\|$. Wegen $\varphi \in C^1(\mathbb{D}, \mathbb{R}^n)$ und $\|\varphi'(x^*)\| = 1 - \delta < 1$ existiert ein $\varepsilon > 0$, sodaß

$$\|\varphi'(x)\| \leq 1 - \tfrac{\delta}{2} \text{ für alle } x \in \mathbb{U} = \overline{K_\varepsilon(x^*)} = \{x \in \mathbb{R}^n \mid \|x - x^*\| \leq \varepsilon\}.$$

Damit folgt für $x_1, x_2 \in \mathbb{U}$ in der zugehörigen Vektornorm

$$\begin{aligned} \|\varphi(x_2) - \varphi(x_1)\| &= \|\varphi(x_1 + s(x_2 - x_1))|_0^1\| = \\ &= \| \textstyle\int_0^1 \varphi'(x_1 + s(x_2 - x_1))(x_2 - x_1)\, ds\| \leq \\ &\leq \textstyle\int_0^1 \|\varphi'(x_1 + s(x_2 - x_1))\| \cdot \|x_2 - x_1\|\, ds \leq \\ &\leq (1 - \tfrac{\delta}{2})\|x_2 - x_1\|, \end{aligned}$$

d. h. $\varphi|_{\mathbb{U}}$ ist kontraktiv mit $L = 1 - \frac{\delta}{2}$.

Wählt man speziell $x_1 = x^*$ und $x_2 = x \in \mathbb{U}$, so liefert obige Ungleichung

$$\|\varphi(x) - x^*\| = \|\varphi(x) - \varphi(x^*)\| \leq (1 - \tfrac{\delta}{2})\|x - x^*\| \leq \|x - x^*\| \leq \varepsilon,$$

d. h.

$$\varphi(x) \in \mathbb{U} \text{ für alle } x \in \mathbb{U}$$

bzw. $\varphi|_{\mathbb{U}} : \mathbb{U} \to \mathbb{U}$.

$\square$

Ist also φ stetig differenzierbar und gilt (6.8), so muß man nur x_0 hinreichend nahe bei x^* wählen, um Konvergenz zu erzielen. Man spricht von lokaler Konvergenz. Dabei kann die Bestimmung eines hinreichend guten Startwertes selbst ein schwieriges Problem sein.

Beispiel 6.4
Für Beispiel 6.1 mit $\varphi(x) = -e^x$ gilt bei der Wahl $\mathbb{D} = [-1.0, -0.1]$ wegen

$$\varphi(-1.0) \approx -0.368, \quad \varphi(-0.1) \approx -0.905$$

und der Monotonie von φ, daß $\varphi(\mathbb{D}) \subseteq \mathbb{D}$. Mit dem Mittelwertsatz (vgl. Bemerkung 1.19) findet man als mögliches L in (6.6)

$$L = \sup_{x\in\mathbb{D}} |\varphi'(x)| = \sup_{x\in[-1.0,-0.1]} e^x = e^{-0.1} \approx 0.905 < 1\,.$$

Nach dem Banachschen Fixpunktsatz existiert also ein eindeutiger Fixpunkt x^* von φ in $[-1.0, -0.1]$. Satz 6.3 ist ebenfalls anwendbar mit

$$\tilde{L} = |\varphi'(x^*)| \approx 0.567\,.$$

Will man x^* auf 12 Dezimalen bestimmen, so liefert der Ansatz

$$\tilde{L}^\nu = 10^{-12},$$

wenn man in (6.7b) einfach $\frac{1}{1-L}\|x_1 - x_0\| = 1$ und $L = \tilde{L}$ setzt,

$$\nu = \log 10^{-12} / \log \tilde{L} \approx 49$$

als Schätzung für die nötige Anzahl von Iterationsschritten. Tatsächlich ist x_{44} die erste Iterierte, die auf 12 Dezimalen genau ist.

$\square$

6.2 Intervallmethoden

Sei $f : [a, b] \to \mathbb{R}$ stetig und $f(a) \cdot f(b) \leq 0$. Nach dem Zwischenwertsatz existiert dann mindestens ein $x^* \in [a, b]$ mit $f(x^*) = 0$. Wählt man ein $c \in (a, b)$, so kann man allein durch Prüfen des Vorzeichens von $f(c)$ entscheiden, ob in $[a, c]$ oder in $[c, b]$ eine Nullstelle von f liegt. Durch iteratives Vorgehen erhält man so sofort ein mögliches Verfahren zur Bestimmung einer Nullstelle von f. Wählt man jeweils die Intervallmitte, d. h. $c = \frac{1}{2}(a + b)$, so spricht man von Bisektion.

Algorithmus 6.5 (Bisektion)

Gegeben sei $a_0 = a$, $b_0 = b$ mit $f(a) \cdot f(b) \leq 0$ und eine Toleranz $\texttt{TOL} > 0$.

Setze $k = 0$.	
Schleife:	
	Setze $c_k = \frac{1}{2}(a_k + b_k)$.
	Ist $b_k - a_k \leq \texttt{TOL}$, akzeptiere c_k als Approximation an eine Nullstelle von f → Stop!
	Ist $f(a_k) \cdot f(c_k) \leq 0$, so setze $a_{k+1} = a_k$ und $b_{k+1} = c_k$, andernfalls setze $a_{k+1} = c_k$ und $b_{k+1} = b_k$.
	Setze $k = k + 1$.

(6.9)

Natürlich ist im Fall $f(a_k) \cdot f(b_k) = 0$ entweder a_k oder b_k Nullstelle und man könnte sofort abbrechen.

$\square$

Satz 6.6

Ist $f : [a, b] \to \mathbb{R}$ stetig und $f(a) \cdot f(b) \leq 0$, so definiert Algorithmus 6.5 (bei Weglassen der Abbrechbedingung) eine Intervallschachtelung $[a_k, b_k]$, $k \in \mathbb{N}_0$, mit

$$\bigcap_{k=0}^{\infty} [a_k, b_k] = \{x^*\} \tag{6.10}$$

und es gilt $f(x^*) = 0$ mit der Abschätzung

$$|x^* - c_k| \leq \left(\frac{1}{2}\right)^{k+1} (b - a). \qquad \text{(lineare Konvergenz)} \tag{6.11}$$

$\square$

Beispiel 6.7

Algorithmus 6.5 liefert angewandt auf das Problem aus Beispiel 6.1 mit $a = -1$, $b = 0$ sowie $\texttt{TOL} = 10^{-12}$ als Approximation an x^* den Wert c_{40}. Man benötigt also weniger Iterationen als in Beispiel 6.4. Aus der Forderung

$$\left(\frac{1}{2}\right)^{k+1} (b - a) = 10^{-12}$$

erhält man als Schätzung für die Anzahl der benötigten Iterationsschritte

$$k = \log 10^{-12} / \log 0.5 - 1 \approx 39\,.$$

□

Bemerkung 6.8 (Regula falsi)
Bei der Bisektion wird c unabhängig von den Beträgen von $f(a)$ und $f(b)$ gewählt. Eine mögliche Wahl von c, die diese einbezieht, ist die (eindeutige) Nullstelle der zugehörigen linearen Interpolierenden. Aus

$$p(x) = \frac{f(b) - f(a)}{b - a}(x - a) + f(a) \tag{6.12}$$

folgt mit $p(c) = 0$ die Darstellung

$$c = a - f(a)\frac{b - a}{f(b) - f(a)} = \frac{af(b) - bf(a)}{f(b) - f(a)}. \tag{6.13}$$

Die sich durch iteratives Vorgehen entsprechend Algorithmus 6.5 ergebende Methode heißt regula falsi. Außerdem muß das Abbrechkriterium geändert werden in

$$c_k - a_k \leq \mathtt{TOL} \quad \vee \quad b_k - c_k \leq \mathtt{TOL},$$

da im allgemeinen $b_k - a_k \not\to 0$ gilt.

□

Beispiel 6.9
Die regula falsi liefert mit gleicher Ausgangssituation wie in Beispiel 6.7 als Approximation an x^* den Wert c_{13}.

□

6.3 Sekantenverfahren

Sei $f : \mathbb{D} \to \mathbb{R}$ stetig mit $\mathbb{D} \subseteq \mathbb{R}$ offen. Zu gegebenen Werten $a, b \in \mathbb{D}$ kann man auch ohne die Bedingung $f(a) \cdot f(b) \leq 0$ die Nullstelle der linearen Interpolierenden berechnen, solange $f(a) \neq f(b)$ ist. Verwendet man in (6.13) stets die zwei aktuellsten Aproximationen, so erhält man die iterative Methode

$$x_{\nu+1} = \frac{x_{\nu-1}f(x_\nu) - x_\nu f(x_{\nu-1})}{f(x_\nu) - f(x_{\nu-1})}, \quad \nu \in \mathbb{N}, \quad x_0, x_1 \text{ gegeben,} \tag{6.14}$$

das sogenannte Sekantenverfahren.

Während man im letzten Abschnitt immer mit Intervallen arbeitete, die eine Nullstelle x^* enthielten, berechnet man hier wie bei den Iterationsverfahren aus Abschnitt 6.1 eine Folge $\{x_\nu\}_{\nu\in\mathbb{N}_0}$, falls man in $\mathbb{D}$ verbleibt und $f(x_\nu) \neq f(x_{\nu-1})$ für alle $\nu \in \mathbb{N}$ gilt.

Satz 6.10
Sei x^* eine Nullstelle von $f \in C^2(\mathbb{D}, \mathbb{R})$, $\mathbb{D} \subseteq \mathbb{R}$ offen, mit $f'(x^*) \neq 0$. Dann definiert (6.14) für hinreichend gute Startdaten $x_0, x_1 \in \mathbb{D}$, $x_0 \neq x_1$, eine Folge $\{x_\nu\}_{\nu\in\mathbb{N}_0}$ mit $x_\nu \to x^*$. Insbesondere gilt

$$|x^* - x_\nu| \leq CK^{q^\nu} \tag{6.15}$$

mit $C \geq 0$, $K \in (0,1)$ und $q = \frac{1}{2}(1+\sqrt{5}) \approx 1.618$.

□

Im Vergleich zu (6.15) haben (6.7b) und (6.11) eine Schranke der Form CK^ν. Da q^ν wegen $q > 1$ schneller wächst als ν, konvergiert die Schranke in (6.15) schneller gegen Null für $\nu \to \infty$. Man spricht von superlinearer Konvergenz. Da die Startwerte hier auch hinreichend nahe an der gesuchten Lösung liegen müssen, liegt wieder lokale Konvergenz vor.

Beispiel 6.11
Für das Problem aus Beispiel 6.1 erhält man mit $x_0 = -1$, $x_1 = 0$ die folgenden Werte

$$\begin{aligned}
x_2 &= -0.\underline{6}12699\,836780,\\
x_3 &= -0.\underline{57}2181\,412091,\\
x_4 &= -0.\underline{5671}02\,090172,\\
x_5 &= -0.\underline{567143\,3}27950,\\
x_6 &= -0.\underline{567143\,290410}.
\end{aligned}$$

Man erkennt, daß man aufgrund der superlinearen Konvergenz desto mehr richtige Stellen pro Iterationsschritt gewinnt, je genauer die aktuelle Iteration ist.

□

6.4 Newton-Verfahren

Statt die zu $x_{\nu-1}, x_\nu$ gehörige Sekante zu verwenden, kann man bei differenzierbarem f auch die zu x_ν gehörige Tangente verwenden. Diese ist gegeben durch

$$p(x) = f'(x_\nu)(x - x_\nu) + f(x_\nu) \tag{6.16}$$

mit der Nullstelle

$$x_{\nu+1} = x_\nu - f'(x_\nu)^{-1} f(x_\nu). \tag{6.17}$$

Dies ist auch für Abbildungen $f : \mathbb{D} \to \mathbb{R}^n$, $\mathbb{D} \subseteq \mathbb{R}^n$, durchführbar. Die Vorschrift (6.17) entspricht einem Iterationsverfahren (6.5) mit

$$\varphi(x) = x - f'(x)^{-1} f(x). \tag{6.18}$$

Wegen

$$\varphi'(x) = I - \frac{d}{dx}\left[f'(x)^{-1}\right] \cdot f(x) - f'(x)^{-1} \cdot f'(x) = -\frac{d}{dx}\left[f'(x)^{-1}\right] \cdot f(x)$$

gilt hier

$$\varphi'(x^*) = 0 \tag{6.19}$$

bei entsprechend glattem f und man kann nach Satz 6.3 gute Konvergenzeigenschaften erwarten. Man nennt das durch (6.17) definierte Verfahren Newton-Verfahren.

Satz 6.12
Sei $f \in C^1(\mathbb{D}, \mathbb{R}^n)$, $\mathbb{D} \subseteq \mathbb{R}^n$ offen, $f'(x)$ nichtsingulär für alle $x \in \mathbb{D}$ und

$$\|f'(x)^{-1}(f'(y) - f'(x))\| \leq \omega \|y - x\| \text{ für alle } x, y \in \mathbb{D}. \tag{6.20}$$

Weiter sei $x^* \in \mathbb{D}$ eine Lösung von (6.1) und x_0 genüge

$$\varrho = \|x^* - x_0\| < \tfrac{2}{\omega}, \quad \overline{K_\varrho(x^*)} \subseteq \mathbb{D}. \tag{6.21}$$

Dann ist durch (6.17) eine Folge $\{x_\nu\}_{\nu \in \mathbb{N}_0}$ definiert mit $x_\nu \in \overline{K_\varrho(x^*)}$ und $x_\nu \to x^*$. Insbesondere gilt

$$\begin{aligned} &\text{(a)} \quad \|x^* - x_{\nu+1}\| \leq \tfrac{\omega}{2}\|x^* - x^\nu\|^2, \\ &\text{(b)} \quad \|x^* - x_\nu\| \leq \tfrac{2}{\omega} K^{2^\nu}, \; K \in (0,1), \\ &\qquad\qquad \text{(quadratische Konvergenz)} \end{aligned} \tag{6.22}$$

und x^* ist eindeutig in $K_{2/\omega}(x^*) \cap \mathbb{D}$.

□

Die Bedingung (6.21) verlangt wiederum, daß x_0 hinreichend nahe bei x^* liegen muß. Man sagt deshalb kurz, das Newton-Verfahren sei lokal und quadratisch konvergent.

Beispiel 6.13
Für das Problem aus Beispiel 6.1 ist

$$f(x) = e^x + x, \quad f'(x) = e^x + 1.$$

Das zugehörige Newton-Verfahren hat also die Form

$$x_{\nu+1} = x_\nu - \frac{e^{x_\nu} + x_\nu}{e^{x_\nu} + 1}.$$

Ausgehend von $x_0 = -0.5$ erhält man die folgenden Werte

$$\begin{aligned} x_1 &= -0.\underline{566}311\,003197, \\ x_2 &= -0.\underline{567143}\,165035, \\ x_3 &= -0.\underline{567143\,290410}. \end{aligned}$$

Man erkennt die quadratische Konvergenz daran, daß sich die Anzahl der richtigen Stellen in jedem Iterationsschritt verdoppelt.

□

Bemerkung 6.14
Im folgenden soll eine Möglichkeit, wie man das Newton-Verfahren als Basis für die Auswertung der Wurzelfunktion verwenden kann, behandelt werden. Offensichtlich besitzt

$$f(x) = x^2 - a, \quad a > 0 \tag{6.23}$$

die eindeutige positive Nullstelle $x^* = \sqrt{a}$. Das zugehörige Newton-Verfahren kann auf die Form

$$x_{\nu+1} = \frac{1}{2}\left(x_\nu + \frac{a}{x_\nu}\right) \qquad \text{(Verfahren von Heron)} \tag{6.24}$$

gebracht werden. Schreibt man a in der Form

$$a = m \cdot 2^r, \tag{6.25}$$

so sind m und r eindeutig festgelegt durch die Forderung

$$m \in (\tfrac{1}{4}, 1], \ r \text{ gerade.} \tag{6.26}$$

Es ist dann

$$\sqrt{a} = \sqrt{m} \cdot 2^{r/2}, \tag{6.27}$$

d. h. man kann sich auf $a \in (\frac{1}{4}, 1]$ einschränken.

Für (6.24) gilt

$$x_{\nu+1} - \sqrt{a} = \tfrac{1}{2}(x_\nu + \tfrac{a}{x_\nu}) - \sqrt{a} = \\ = \tfrac{1}{2x_\nu}(x_\nu^2 - 2\sqrt{a}x_\nu + a) = \tfrac{1}{2x_\nu}(x_\nu - \sqrt{a})^2, \tag{6.28}$$

d. h. für $x_\nu \in [\frac{1}{2}, 1]$

$$|x_\nu - \sqrt{a}| \le |x_0 - \sqrt{a}|^{2^\nu}, \tag{6.29}$$

vgl. quadratische Konvergenz. Betrachtet man nur die ersten k Bits von a, so erhält man ein $\tilde{a} \in [\frac{1}{4}, 1]$ mit

$$|a - \tilde{a}| \le 2^{-k}. \tag{6.30}$$

Als Startwert verwendet man dann $x_0 = \sqrt{\tilde{a}}$, das man (etwa bis auf Maschinengenauigkeit) fest abgespeichert hat (sog. Table-Lookup).

Beachtet man, daß

$$|\sqrt{a} - \sqrt{\tilde{a}}| \le |a - \tilde{a}| \tag{6.31}$$

nach dem Mittelwertsatz, und wählt man k und ν so, daß

$$2^{-k} \le \sqrt[2^\nu]{\text{eps}}, \tag{6.32}$$

so gilt

$$|x_\nu - \sqrt{a}| \le |x_0 - \sqrt{a}|^{2^\nu} = |\sqrt{\tilde{a}} - \sqrt{a}|^{2^\nu} \le \\ \le |\tilde{a} - a|^{2^\nu} \le (2^{-k})^{2^\nu} \le \text{eps}. \tag{6.33}$$

Mit $\text{eps} = 2^{-l}$ ergibt sich aus (6.32)

$$k \ge \frac{l}{2^\nu}. \tag{6.34}$$

Damit hat man für $l = 53$ die Möglichkeiten

(6.35)

ν	0	1	2	3	4	5
k	53	27	14	7	4	2

.

Dabei ist ein Speicherbedarf von etwa 2^k Fließkommazahlen zu beachten. Eine g"unstige Wahl scheint deshalb $\nu = 3$, $k = 7$ zu sein. Man beachte, daß (6.24) den Nachteil besitzt, eine Division zu enthalten.

□

Kapitel 7

Eigenwertprobleme

Gegeben sei $A \in \mathbb{C}^{n,n}$. Gesucht sind einzelne oder alle Eigenwerte von A, d. h. Werte $\lambda \in \mathbb{C}$, für die

$$Ax = \lambda x \tag{7.1}$$

mit $x \in \mathbb{C}^n \setminus \{0\}$ gilt.

7.1 Kondition

Satz 7.1 (Bauer/Fike)
Seien $A, E \in \mathbb{C}^{n,n}$. Ist μ ein Eigenwert von $A + E$ und $X^{-1}AX = D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, so gilt

$$\min_{i=1,\ldots,n} |\lambda_i - \mu| \leq \operatorname{cond}_2 X \cdot \|E\|_2. \tag{7.2}$$

□

Die Kondition des Eigenwertproblems wird also f"ur diagonalisierbare Matrizen beschrieben durch

$$\kappa = \operatorname{cond}_2 X, \tag{7.3}$$

d. h. durch die Kondition der Transformationsmatrix X, die A diagonalisiert (und nicht durch die Kondition von A selbst).

Bemerkung 7.2
Ist $A \in \mathbb{C}^{n,n}$ hermitesch, d. h. $A^H = A$, wobei

$$(a_{ij})^H = (\bar{a}_{ji}), \quad A = (a_{ij}), \tag{7.4}$$

so ist

$$\kappa = 1, \tag{7.5}$$

d. h. das Eigenwertproblem ist in diesem Fall gut konditioniert.

□

Man beachte, daß (7.4) für $A \in \mathbb{R}^{n,n}$ gleichbedeutend mit $A^T = A$, d. h. mit A symmetrisch, ist.

Bemerkung 7.3
Die Eigenwerte von A sind gerade die Nullstellen des charakteristischen Polynoms

$$p(\lambda) = \det(A - \lambda I). \tag{7.6}$$

Der Algorithmus, zuerst p und dann dessen Nullstellen zu bestimmen, ist allerdings instabil.

□

7.2 Vektoriteration und inverse Iteration

Hat A einen eindeutigen betragsgrößten Eigenwert, so kann man das folgende Verfahren verwenden.

Algorithmus 7.4 (Vektoriteration bzw. Potenzmethode)
Gegeben sei $A \in \mathbb{C}^{n,n}$ und $x^{(0)} \in \mathbb{C}^n \setminus \{0\}$, sowie $\mathtt{TOL} > 0$.

Setze $k = 1$.	
Schleife über erlaubte Anzahl von Schritten:	
	Setze $z^{(k)} = Ax^{(k-1)}$.
	Setze $\lambda^{(k)} = z_i^{(k)}/x_i^{(k-1)}$, wobei $\lvert z_i^{(k)}\rvert = \lVert z^{(k)}\rVert_\infty$ mit $i \in \{1, \ldots, n\}$.
	Ist $\lvert\lambda^{(k)} - \lambda^{(k-1)}\rvert \le \mathtt{TOL}$, akzeptiere $\lambda^{(k)}$ als Eigenwert von $A \to$ Stop!
	Setze $x^{(k)} = z^{(k)}/z_i^{(k)}$.
	Setze $k = k + 1$.

(7.7)

□

Lemma 7.5
Sei $A \in \mathbb{C}^{n,n}$ diagonalisierbar mit $X^{-1}AX = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ und sei

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|. \tag{7.8}$$

Dann gilt f"ur Algorithmus 7.3 für hinreichend allgemein gew"ahltes $x^{(0)} \in \mathbb{C}^n \setminus \{0\}$, daß

$$|\lambda^{(k)} - \lambda_1| \leq C \left|\frac{\lambda_2}{\lambda_1}\right|^k \quad \text{(lineare Konvergenz)} \tag{7.9}$$

mit $C \geq 0$ unabh"angig von k.

□

Um nicht nur den betragsgr"oßten Eigenwert von A erhalten zu können, kann man folgendermaßen vorgehen. Hat A den Eigenwert λ, so hat wegen

$$Ax = \lambda x \implies (A - \mu I)x = (\lambda - \mu)x \implies (A - \mu I)^{-1}x = (\lambda - \mu)^{-1}x$$

dann $A - \mu I$ den Eigenwert $\lambda - \mu$ und $(A - \mu I)^{-1}$ den Eigenwert $(\lambda - \mu)^{-1}$. Ist λ ein einfacher Eigenwert und liegt μ näher an λ als an allen anderen Eigenwerten von A, so ist $(\lambda - \mu)^{-1}$ der betragsgr"oßte Eigenwert von $(A - \mu I)^{-1}$. Zu dessen Berechnung bietet sich damit Vektoriteration mit A ersetzt durch $(A - \mu I)^{-1}$ an, sogenannte inverse Iteration nach Wielandt.

Algorithmus 7.6 (Inverse Iteration nach Wielandt)
Gegeben sei $A \in \mathbb{C}^{n,n}$, $x^{(0)} \in \mathbb{C}^n \setminus \{0\}$ und $\mu \in \mathbb{C}$, sowie $\texttt{TOL} > 0$.

Setze $k = 1$.	
Berechne LR-Zerlegung von $A - \mu I$. Ist $A - \mu I$ singulär, so akzeptiere μ als Eigenwert von $A \to$ Stop!	
Schleife über erlaubte Anzahl von Schritten:	
	Bestimme $z^{(k)} = (A - \mu I)^{-1} x^{(k-1)}$ durch Vorwärts-/Rückwärtssubstitution.
	Setze $\tilde{\lambda}^{(k)} = z_i^{(k)} / x_i^{(k-1)}$, wobei $\lvert z_i^{(k)}\rvert = \lVert z^{(k)} \rVert_\infty$ mit $i \in \{1, \dots, n\}$.
	Ist $\lvert\tilde{\lambda}^{(k)} - \tilde{\lambda}^{(k-1)}\rvert \leq \texttt{TOL}$, akzeptiere $\lambda^{(k)} = 1/\tilde{\lambda}^{(k)} + \mu$ als Eigenwert von $A \to$ Stop!
	Setze $x^{(k)} = z^{(k)} / z_i^{(k)}$.
	Setze $k = k + 1$.

(7.10)

□