Vorlesungsmitschrift Numerik 1

Vorlesung von Prof. P. Kunkel

Universität Leipzig, Sommersemester 2008

Inhaltsverzeichnis

0 Vorbemerkungen

Ziel der Numerischen Mathematik ist die Entwicklung und Bewertung von Verfahren zur näherungsweisen zahlenmäßigen Lösung von (mathematischen) Problemen. Die Problematik ist dabei die folgende:

- Endlichkeit der Menge der Symbole (π als Symbol, π^2 nicht, etc.)

- Endlichkeit der Anzahl der Rechenoperationen

Beispiel 0.1. Das Polynom p mit $p(x) = x^2 - 2$ besitzt die positive Nullstelle $\sqrt{2}$. Diese Aussage ist insofern leer, als dass $\sqrt{2}$ gerade so 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 Entwicklung weder abbrechend noch periodisch. Man muss sich also mit einer rationalen Approximation von $\sqrt{2}$ zufriedengeben. Ein Taschenrechner liefert z.B.

$$\sqrt{2} \approx 1.414213562 \tag{0.1}$$

Die Frage ist nun, wie man (oder der Taschenrechner) überhaupt eine solche Approximation bestimmen kann. Anforderungen an eine entsprechende Methode wären

- Effizienz (das Resultat soll „schnell“ verfügbar sein)
- Genauigkeit (das Resultat soll eine „gute“ Näherung an den gesuchten Wert darstellen)

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

$$f(x) = \begin{cases} \frac{1-\sqrt{1-x^2}}{x^2} & \text{für } x \neq 0 \\ \frac{1}{2} & x = 0. \end{cases} \tag{0.2}$$

Wegen

$$\frac{1-\sqrt{1-x^2}}{x^2} = \frac{(1-\sqrt{1-x^2})(1+\sqrt{1-x^2})}{x^2(1+\sqrt{1-x^2})} = \frac{1}{1+\sqrt{1-x^2}}, \quad x \neq 0$$

kann man f auch durch

$$f(x) = \frac{1}{1+\sqrt{1-x^2}} \tag{0.3}$$

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

i	$f(10^{-i})$ nach (0.2)	$f(10^{-i})$ nach (0.3)
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 (0.2) weniger geeignet als auf der Basis von (0.3).

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 begnügen. 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 $d_i \in \{0, \ldots, n-1\}$, $r \in \mathbb{Z}$.

Beweis. Sei $x_0 \in \mathbb{R}$ und o.B.d.A. $x_0 \geq 0$. Dann gibt es ein $r \in \mathbb{Z}$ mit $0 \leq x_0 < b^r$. Setzt man $d_1 = \left\lfloor \frac{x_0}{b^{r-1}} \right\rfloor$, so ist $d_1 \in \{0, \ldots, b-1\}$. Wegen $d_1 \leq \frac{x_0}{b^{r-1}} < d_1 + 1 \Leftrightarrow d_1 b^{r-1} \leq x_0 < d_1 b^{r-1} + b^{r-1}$ folgt für $x_1 = x_0 - d_1 b^{r-1}$ die Ungleichung $0 \leq x_1 < b^{r-1}$. 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 $0 \leq x_s < b^{r-s}$. 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

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

v ... Vorzeichen
b ... Basis
e ... Exponent
m ... Mantisse

dargestellt werden. Fordert man zusätzlich, dass $e = r$ ist, so gilt $m \in [0, 1]$. Fordert man außerdem für $x \neq 0$, dass $d_1 \neq 0$ ist, 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 wieweit normalisierte Darstellungen 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 eine abbrechend, d.h. von der Form

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

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.B.d.A. $x > 0$ mit den Darstellungen

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

wovon o.B.d.A. die erste normalisiert ist. Weiter sei $s = \min\{i \in \mathbb{N}^+ \mid d_i \neq e_i\}$. Ist die zweite Darstellung nicht normalisiert, so gilt $d_1 > 0 = e_1$. Ist die zweite Darstellung normalisiert, so können wir o.B.d.A. annehmen, dass $d_s > e_s$. Damit folgt:

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

und es muss überall das Gleichheitszeichen gelten, d.h. $d_s - e_s = 1$ und $e_i - d_i = b-1$, $i = s+1,\ s+2, \ldots$ beziehungsweise $e_i = b-1$, $d_i = 0$, $i = s+1,\ s+2, \ldots$. □

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 (sogenannte Maschinenzahlen) seien gegeben durch

$$\mathbb{M}_{b,l} = \left\{ \pm \sum_{i=1}^{l} d_i b^{r-i} \mid d_i \in \{0, \ldots, b-1\}, d_1 \neq 0, r \in \mathbb{Z} \right\} \cup \{0\}. \tag{1.5}$$

Es gilt $\mathbb{M}_{b,l} \subseteq \mathbb{R}$, 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 Handrechnung ist $b = 10$. Außerdem ist bei einem Rechner der Exponent r zusätzlich beschränkt gemäß

$$r_{min} \leq r \leq r_{max}. \tag{1.6}$$

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

Um mit einem $x \in \mathbb{R}$ 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 $fl : \mathbb{R} \to \mathbb{M}_{b,l}$ mit

$$\begin{array}{lcll} fl(x) & = & x \ \forall x \in \mathbb{M}_{b,l} & \text{(Projektionseigenschaft)} \\ fl(x) & \leq & fl(y) \ \forall x, y \in \mathbb{R} \text{ mit } x \leq y & \text{(Monotonie)} \end{array} \tag{1.7}$$

heißt Rundung bezüglich $M_{b,l}$.

Für $b = 10$ kennt man die sogenannte 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 und im mehrdeutigen Fall abbrechender Darstellung

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

Durch

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

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

$$fl(-x) = -fl(x) \ \forall x \in \mathbb{R} \tag{1.10}$$

besitzt.

Beweis. Zunächst muss gezeigt werden, dass $fl(x) \in \mathbb{M}_{b,l}$ für alle $x \in \mathbb{R}$. Dies ist offensichtlich bis auf den zweiten Fall in (1.9). Ist dort $d_l \leq b - 2$, so gilt

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

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

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

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

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

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

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

keine Maschinenzahlen liegen und alle x in

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

auf die linke Intervallgrenze beziehungsweise alle x in

$$\left[\sum_{i=1}^{l} d_i b^{r-i} + \left[\frac{b+1}{2}\right] b^{r-l-1}, \sum_{i=1}^{l} d_i b^{r-i} + b^{r-l}\right]$$

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

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

$$|fl(x) - x| \leq \frac{1}{2} b^{-l+1} |x|. \tag{1.11}$$

Beweis. Sei o.B.d.A. $x > 0$ und damit $fl(x) \geq 0$. Unter Verwendung von (1.8) und (1.9) erhält man die folgenden Abschätzungen:

- 1. Fall: $d_{l+1} \leq \left[\frac{b-1}{2}\right] = \frac{b}{2} - 1$
  Es gilt

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

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

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

In beiden Fällen erhält man also $|fl(x) - x| \leq \frac{1}{2} b^{r-l}$. Wegen $|x| \geq b^{r-1}$ folgt dann

$$|fl(x) - x| \leq \frac{1}{2} b^{r-1-l+1} \leq \frac{1}{2} b^{-l+1} |x|.$$

□

Definition 1.10 (Maschinengenauigkeit)**.** Sei b gerade. Die Größe

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

heißt (zu der obigen Rundung gehörige) Maschinengenauigkeit.

Bemerkung 1.11. Die Abschätzung

$$|fl(x) - x| \leq \text{eps}|x| \ \forall x \in \mathbb{R} \tag{1.13}$$

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

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

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 Grundrechenarten nicht erwarten kann, das die Verknüpfung zweier Maschinenzahlen wieder eine Maschinenzahl ist.
Als Modell für die Grundrechenarten verwenden wir die Fließkommarealisierungen

$$\begin{array}{ll} x \oplus y = fl(x + y) & x \ominus y = fl(x - y) \\ x \odot y = fl(x \cdot y) & x \oslash y = fl(x/y) \end{array} \tag{1.15}$$

(◯-Operationen im Rechner), entsprechend für die sogenannten Standardfunktionen

$$\textcircled{f}(x) = fl(f(x)), \tag{1.16}$$

etwa mit $f \in \{\text{sqrt, exp, log, sin, cos, arctan}, \dots\}$. Zusammen mit (1.5) und (1.9) haben wir damit ein Modell eines Rechners (idealer Rechner) aufgestellt, das wir hier ausschließlich verwenden wollen.

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

Beispiel 1.13. Sei $b = 10$, $l = 2$. Gegeben seien $x = 104$, $y = 4.22$, $z = 3.86$. Diese werden gerundet zu $\tilde{x} = 0.10 \cdot 10^3$, $\tilde{y} = 0.42 \cdot 10^1$, $\tilde{z} = 0.39 \cdot 10^1$. Damit erhält man statt $x + y + z = 112.08$

$$(\tilde{x} \oplus \tilde{y}) \oplus \tilde{z} = fl(0.1042 \cdot 10^3) \oplus 0.39 \cdot 10^1 = \dots = 0.10 \cdot 10^3,$$

beziehungsweise

$$\tilde{x} \oplus (\tilde{y} \oplus \tilde{z}) = 0.10 \cdot 10^3 \oplus fl(0.31 \cdot 10^1) = \dots = 0.11 \cdot 10^3,$$

wobei letzteres zufällig gleich dem exakten gerundeten Ergebnis ist.

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

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

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

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

bestimmen.

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

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

oder implizit durch

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

beschrieben wird. Dabei sind $\mathbb{X}$, $\mathbb{Y}$ und $\mathbb{Z}$ üblicherweise Teilmengen von normierten Vektorräumen, meist Teilmengen irgendwelcher $\mathbb{R}^n$. Um ein Ergebnis y zahlenmäßig angeben zu können, muss dieses eindeutig durch die Problemstellung festgelegt sein.
Da wir außerdem (eventuell schon bei der Eingabe der Daten x) auf dem Rechner durch Rundung Fehler machen, muss sich das gestellte Problem unter Störungen gutartig verhalten.

Definition 1.15 (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 dadurch gegebene 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.

Definition 1.16. Sei $\mathbb{X}$ eine Teilmenge eines normierten Vektorraumes mit Norm $\|.\|_{\mathbb{X}}$ und sei $\tilde{x} \in \mathbb{X}$ eine Näherung an $x \in \mathbb{X}$. Dann heißt $\|x - \tilde{x}\|_{\mathbb{X}}$ der zugehörige absolute Fehler von $\tilde{x}$. Ist $x \neq 0$, so heißt $\|x - \tilde{x}\|_{\mathbb{X}}/\|x\|_{\mathbb{X}}$ der zugehörige relative Fehler von $\tilde{x}$.

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

Definition 1.17 (Kondition eines Problems)**.** Sei durch $f : \mathbb{X} \to \mathbb{Y}$ ein wohlgestelltes Problem beschrieben. Dann heißt

$$\kappa = \sup_{\substack{x_1, x_2 \in \mathbb{X} \\ x_1 \neq x_2}} \frac{\|f(x_2) - f(x_1)\|_{\mathbb{X}}}{\|x_2 - x_1\|_{\mathbb{X}}} \in [0, \infty] \tag{1.20}$$

die Kondition des Problems.

Die Bedingung (1.20) impliziert für $\kappa < \infty$, dass f Lipschitz-stetig ist mit Lipschitzkonstante κ. Man beachte, dass κ von der Wahl der Normen auf $\mathbb{X}$ und $\mathbb{Y}$ abhängt, insbesondere davon, ob man absolute oder relative Fehler betrachtet.

Bemerkung 1.18. Sind $\mathbb{X}, \mathbb{Y} \subseteq \mathbb{R}$ mit $\|.\|_{\mathbb{X}} = \|.\|_{\mathbb{Y}} = |.|$ und ist $f : \mathbb{X} \to \mathbb{Y}$ stetig differenzierbar, so gilt nach dem Mittelwertsatz:

$$\kappa = \sup_{x \in \mathbb{X}} |f'(x)|. \tag{1.21}$$

Bemerkung 1.19 (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 Genauigkeit $r \geq 0$, so kann man wegen (1.20) nur im Fall

$$r \geq \kappa\varepsilon \tag{1.22}$$

mit der Einhaltung der Genauigkeit gemäß $\|\tilde{y} - y\|_{\mathbb{Y}} \leq r$ rechnen. Man sagt in diesem Fall, das Problem sei gut konditioniert. Anderenfalls sagt man, das Problem sei schlecht konditioniert. Will man im Extremfall bei Eingabefehlern von $\varepsilon = \text{eps}$ noch die Größenordnung des Resultats entsprechend $r = 1$ erfassen, muss für die Kondition bezüglich des relativen Fehlers

$$\kappa \leq \frac{1}{\text{eps}} \tag{1.23}$$

gelten.

Beispiel 1.20. Das durch (0.2) und $\mathbb{X} = [-1, 1]$ und $\mathbb{Y} = \mathbb{R}$ gegebene Problem ist wohlgestellt. Für $\varepsilon > 0$ hinreichend klein gilt

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

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

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

mit $x!! = x(x-2)(x-4)\ldots$ die Potenzreihenentwicklung

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

mit Konvergenzreadius $R = 1$, d.h. f ist stetig differenzierbar auf $\mathbb{X}$ mit Ableitung $f'(x)$

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

Also ist f' monoton und punktsymmetrisch und es gilt mit (1.21)

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

das heißt 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 0 nicht sensitiv gegenüber Fehlern in x sind, was die Auswertung mit Hilfe von (0.3) bestätigt. Die Probleme bei der Verwendung von (0.2) kommen also nicht ovn einer schlechten Kondition der Probleme, sondern müssen vom gewählten Rechenweg herrühren.

Definition 1.21 (Algorithmus)**.** Ein Algorithmus ist eine endliche Folge von Rechenvorschriften, bestehend aus Elementaroperationen (Grundrechenarten, Auswertung von Standardfunktionen), die angibt, wie y aus x zu bestimmen ist, das heißt eine Zerlegung von $f : \mathbb{X} \to \mathbb{Y}$ entsprechend

$$f = \varphi_l \circ \varphi_{l-1} \circ \cdots \circ \varphi_1 \tag{1.24}$$

mit

$$\varphi_i : \mathbb{X}_i \to \mathbb{X}_{i+1},\ i = 1, \ldots, l \text{ mit } \mathbb{X}_1 = \mathbb{X}, \mathbb{X}_{l+1} = \mathbb{Y}, \tag{1.25}$$

wobei 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 in Beispiel 0.2 basierend auf den Darstellungen (0.2) und (0.3).
Da man mit jedem φ_i Zwischenresultate und damit Rundungsfehler erzeugt, muss man auch die Auswirkung dieser Fehler auf das Resultat berücksichtigen. Diese sollten wenn möglich nicht wesentlich größer sein als die der unvermeidlichen Fehler (Eingabefehler, Rundung des Resultats).

Definition 1.22 (Stabilität eines Algorithmus)**.** Zu einem Algorithmus (1.24) seien die sogenannten Restabbildungen ψ_i definiert durch

$$\psi_i = \varphi_l \circ \varphi_{l-1} \circ \cdots \circ \varphi_i,\ i = 1, \ldots, l,\ \psi_i : \mathbb{X}_i \to \mathbb{Y} \tag{1.26}$$

mit Konditionen κ_i. Gilt

$$\kappa_i \leq (c+1)\max\{\kappa_1, 1\}\,,\ i = 2, \ldots, l, \tag{1.27}$$

wobei c die Anzahl der im Algorithmus durchzuführenden Elementaroperationen und κ_1 die Kondition des Problems ist, so spricht man von einem stabilen Algorithmus.
Davon unabhängig kann die Zahl $\max\limits_{i=2,\ldots,l}\{\kappa_i\}/\max\{\kappa_1, 1\}$ als Stabilitätsmaß angesehen werden.

Bemerkung 1.23 (Vorwärts/Rückwärtsanalyse)**.** Ein Algorithmus liefert eine Realisierung von $x \mapsto y$ auf dem Rechner durch $x \mapsto \tilde{y}$. Stabilität entspricht dann der Forderung, dass $\|\tilde{y} - y\|_{\mathbb{Y}}$ hinreichend klein ist. Man spricht genauer von Stabilität im Sinne der Vorwärtsanalyse. Kann man $\tilde{y}$ als exaktes Ergebnis zu gestörten Daten $\tilde{x}$ auffassen, d.h. $\tilde{y} = f(\tilde{x})$, und ist $\|\tilde{x} - x\|_{\mathbb{X}}$ hinreichend klein, so spricht man von Stabilität im Sinne der Rückwärtsanalyse.

1.3 Differenzielle Fehleranalyse

Wie man an Beispiel 1.20 schon gesehen hat, sind Berechnungen basierend auf (1.20) nicht leicht durchführbar. Geht man davon aus, dass Fehler klein sind, so bieten sich Linearisierungen an, d.h. man vernachlässigt quadratische Terme in den Fehlern. Man spricht von differenzieller Fehleranalyse.
Ist $f : \mathbb{X} \to \mathbb{Y}$ stetig differenzierbar, so gilt für $x \in \mathbb{X}$ und gestörtes $\tilde{x} \in \mathbb{X}$ mit dazu gehörigen $y = f(x)$, $\tilde{y} = f(\tilde{x})$:

$$f(\tilde{x}) - f(x) = f(x + (\tilde{x} - x)) - f(x) \doteq f(x) + f'(x)(\tilde{x} - x) - f(x) = f'(x)(\tilde{x} - x). \tag{1.28}$$

Damit beschreibt

$$x := \|f'(x)\|_{\mathbb{Y}\leftarrow\mathbb{X}}\,,\ \|A\|_{\mathbb{Y}\leftarrow\mathbb{X}} = \sup_{\substack{x\in\mathbb{X}\\ x\neq 0}} \frac{\|A\|_{\mathbb{X}}}{\|x\|_{\mathbb{X}}} \tag{1.29}$$

in erster Näherung die Verstärkung des Ausgangsfehlers, die sogenannte differentielle Kondition des Problems.

$$\kappa := |f'(x)| \text{ falls } \mathbb{Y}, \mathbb{X} \subseteq \mathbb{R} \quad \text{mit} \quad \|.\|_{\mathbb{X}} = \|.\|_{\mathbb{Y}} = |.| \tag{1.30}$$

bezeichnet man als differentielle Kondition bezüglich des absoluten Fehlers.
Wählt man stattdessen $\|.\|_{\mathbb{X}} = |.|/|x|$ und $\|.\|_{\mathbb{Y}} = |.|/|y|$, falls $x, y \neq 0$, so erhält man aus (1.28) in der Form

$$\frac{\tilde{y} - y}{y} = \frac{xf'(x)}{y}\frac{\tilde{x} - x}{x} \tag{1.31}$$

stattdessen

$$\kappa = \left|\frac{xf'(x)}{f(x)}\right|, \tag{1.32}$$

die sogenannte differentielle Kondition bezüglich des relativen Fehlers. Ist $\mathbb{X} \subseteq \mathbb{R}^n$, so hat die Jacobimatrix von $f'(x)$ die Form

$$f'(x) = \left[\frac{\partial f}{\partial x_1}(x)\ \ldots\ \frac{\partial f}{\partial x_n}(x)\right] \tag{1.33}$$

und (1.28) wird mit $x = (x_1, \ldots, x_n)^{\mathrm{T}}$ und $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_n)^{\mathrm{T}}$ zu

$$\tilde{y} - y \doteq \sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(x)(\tilde{x}_j - x_j). \tag{1.34}$$

Damit gilt in erster Näherung

$$|\tilde{y} - y| \dot{\leq} \sum_{j=1}^{n} \left|\frac{\partial f}{\partial x_j}(x)\right| \cdot |\tilde{x}_j - x_j| \tag{1.35}$$

beziehungsweise

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

Für $\mathbb{Y} \subseteq \mathbb{R}^n$ braucht man lediglich die einzelnen Komponenten von y zu betrachten. Da Rundungsfehler nach (1.11) als relative Fehler beschränkt sind, betrachtet man oft nur die differentielle Kondition bezüglich des relativen Fehlers. Durch entsprechendes Vorgehen kann man die differentielle Fehleranalyse auch zur Untersuchung der Stabilität eines Algorithmus' verwenden. Man muss nur nach Definition 1.22 statt f die Restabbildung ψ_i betrachten. Sei dazu $\mathbb{X}_i \subseteq \mathbb{R}^{n_i}$, $x_1 = x$, sowie $x_{n_1} = \psi_1(x_1)$, $i = 1, \ldots, l$, außerdem $x_i = (x_{i_1}, \ldots, x_{i_{n_i}})^{\mathrm{T}}$. Damit sind die maßgeblichen Konditionen bezüglich des relativen Fehlers entsprechend (1.36) gegeben durch:

$$\kappa_{i,j} = \left| \frac{x_{i,j}}{y} \frac{\partial \psi_i}{\partial x_{i,j}}(x_i) \right|, \ j = 1, .., n_i, \ i = 1, \ldots, l. \tag{1.37}$$

Dabei beschreibt

$$\kappa = \max\{\kappa_{1,1}, \ldots, \kappa_{1,n_1}\} \tag{1.38}$$

die differentielle Kondition des Problems und (1.27) wird ersetzt durch die Bedingung

$$\kappa_{i,j} \leq (c+1) \max\{\kappa_{1,1}, \ldots, \kappa_{1,n_1}, 1\}, \ j = 1, \ldots, n_i, \ i = 2, \ldots, l. \tag{1.39}$$

Beispiel 1.24 (Kondition von Addition/Subtraktion)**.** Sei $y = x_1 + x_2 = f(x_1, x_2)$ mit $f : \mathbb{R}^2 \to \mathbb{R}$. 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

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

beziehungsweise für den relativen Fehler $(x_1, x_2, y \neq 0)$

$$\frac{|\tilde{y} - y|}{|y|} \leq \frac{|x_1|}{|x_1 + x_2|} \frac{|\tilde{x}_1 - x_1|}{|x_1|} + \frac{|x_2|}{|x_1 + x_2|} \frac{|\tilde{x}_2 - x_2|}{|x_2|} \leq \max\left\{ \frac{|x_1|}{|x_1 + x_2|}, \frac{|x_2|}{|x_1 + x_2|} \right\} \left(\frac{|\tilde{x}_1 - x_1|}{|x_1|} + \frac{|\tilde{x}_2 - x_2|}{|x_2|} \right).$$

Für $\|.\|_{\mathbb{X}} = \|.\|_1$ und $\|.\|_{\mathbb{Y}} = |.|$ (absoluter Fehler) hat man wegen $\kappa = 1$ ein gut konditioniertes Problem. Wählt man $\|.\|_{\mathbb{X}}$ stattdessen gemäß

$$\|(z_1, z_2)\|_{\mathbb{X}} = \left| \frac{z_1}{x_1} \right| + \left| \frac{z_2}{x_2} \right|$$

und $\|.\|_{\mathbb{Y}} = |.|/|y|$ (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. Anderenfalls hat man ein schlecht konditioniertes Problem (sogenannte subtraktive Auslöschung). Das gleiche folgt auch bei differentieller Fehleranalyse.

Beispiel 1.25 (Kondition der Multiplikation)**.** Sei $y = f(x_1, x_2) = x_1 \cdot x_2$, $f : \mathbb{R}^2 \to \mathbb{R}$. Für die differentielle Kondition bezüglich des relativen Fehlers $(x_1, x_2 \neq 0)$ erhält man nach (1.36)

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

Die Multiplikation ist also bezüglich des relativen Fehlers im differentiellen Sinn gut konditioniert. Unter Verwendung von Bemerkung 1.11 kann man auch folgendermaßen schließen:
Sei $|\tilde{x}_i - x_i| \leq \mathrm{eps}|x_i|, \ i = 1, 2$, d.h. es gebe ein ε_i mit $|\varepsilon_i| \leq \mathrm{eps}$, so dass

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

Dann gilt

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

und damit

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

Dabei entsprechen die obigen Konditionszahlen gerade den Vorfaktoren von $|\varepsilon_1|$ und $|\varepsilon_2|$. Ein entsprechendes Resultat kann man auch für die Division zeigen.

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

$$a^x = \exp(x \ln a)$$

den folgenden Algorithmus anwenden:

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

Für die differentielle Kondition des relativen Fehlers gemäß (1.37) erhält man:

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

Es gilt hiermit

$$\kappa_{i,j} \leq \max\{1, |x|, |x \ln a|\},$$

also handelt es sich um einen stabilen Algorithmus.

Beispiel 1.27. Der auf (0.2) aufbauende Algorithmus zur Berechnung von $f(x)$ kann folgendermaßen beschrieben werden:

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

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

$$\begin{aligned}
x_1 &= x \\
x_2 &= \varphi_1(x_1) = x_1^2 \\
x_3 &= \varphi_2(x_2) = \begin{pmatrix} 1-x^2 \\ x^2 \end{pmatrix} \\
x_4 &= \varphi_3(x_3) = \begin{pmatrix} \sqrt{1-x^2} \\ x^2 \end{pmatrix} \doteq \begin{pmatrix} 1-\frac{1}{2}x^2 \\ x^2 \end{pmatrix} \\
x_5 &= \varphi_4(x_4) \doteq \begin{pmatrix} \frac{1}{2}x^2 \\ x^2 \end{pmatrix} \\
y &= \varphi_5(x_5) \doteq \frac{1}{2}.
\end{aligned}$$

Wegen $\psi_1 = f$ liefert die Reihenentwicklung aus Bespiel 1.20:

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

beziehungsweise

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

und damit für die differentielle Kondition bezüglich des relativen Fehlers:

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

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

2 Lineare Gleichungssysteme

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

$$\mathbf{A}x = b, \tag{2.1}$$

wobei $\mathbf{A} \in \mathbb{R}^{n \times n}$ regulär und $b \in \mathbb{R}^n$ seien.

2.1 Kondition des Problems

Wegen $x = \mathbf{A}^{-1}b$ und Stetigkeit der Abbildungen

$$^{-1} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n} \quad \text{und} \quad \cdot : \mathbb{R}^{n\times n} \times \mathbb{R}^n \to \mathbb{R}^n$$

ist das Problem wohlgestellt. In erster Näherung gilt für gestörtes Problem:

$$\begin{aligned} x + \Delta x &= (\mathbf{A} + \Delta\mathbf{A})^{-1}(b + \Delta b) \\ &= \left[\mathbf{A}(id + \mathbf{A}^{-1}\Delta\mathbf{A})\right]^{-1}(b + \Delta b) \\ &= (id + \mathbf{A}^{-1}\Delta\mathbf{A})^{-1}\mathbf{A}^{-1}(b + \Delta b) \\ &\doteq (id - \mathbf{A}^{-1}\Delta\mathbf{A})\mathbf{A}^{-1}(b + \Delta b) \\ &\doteq \mathbf{A}^{-1}b + \mathbf{A}^{-1}\Delta b - \mathbf{A}^{-1}(\Delta\mathbf{A})\mathbf{A}^{-1}b, \end{aligned}$$

also

$$\Delta x \doteq \mathbf{A}^{-1}\Delta b - \mathbf{A}^{-1}(\Delta\mathbf{A})x. \tag{2.2}$$

Mit einer Vektornorm $\|.\|$ und der zugehörigen Operatornorm, definiert durch

$$\|\mathbf{A}\| = \sup_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} \frac{\|\mathbf{A}x\|}{\|x\|} = \sup_{\|x\|=1} \|\mathbf{A}x\|, \tag{2.3}$$

gelangt man zu der Abschätzung

$$\|\Delta x\| \dot{\leq} \|\mathbf{A}^{-1}\| \cdot \|\Delta b\| + \|\mathbf{A}^{-1}\| \cdot \|x\| \cdot \|\Delta\mathbf{A}\| \tag{2.4}$$

für den absoluten Fehler und

$$\frac{\|\Delta x\|}{\|x\|} \dot{\leq} \frac{\|\mathbf{A}^{-1}\| \cdot \|\mathbf{A}x\|}{\|x\|} \cdot \frac{\|\Delta b\|}{\|b\|} + \frac{\|\mathbf{A}^{-1}\| \cdot \|x\| \cdot \|\mathbf{A}\|}{\|x\|} \cdot \frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|} \dot{\leq} \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}\right) \tag{2.5}$$

für den relativen Fehler. Die Größe

$$\mathrm{cond}_{\|.\|}\mathbf{A} = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| \tag{2.6}$$

heißt Kondition der Matrix $\mathbf{A}$ und beschreibt die differentielle Verstärkung des relativen Fehlers der Daten $\mathbf{A}$ und b.

2.2 Gauß-Eliminierung

Schreibt man (2.1) in der Form

$$\begin{array}{ccccccccc} a_{11}x_1 & + & \dots & + & a_{1n}x_n & = & b_1 \\ \vdots & & & & \vdots & & \vdots \\ a_{n1}x_1 & + & \dots & + & a_{nn}x_n & = & b_n, \end{array} \tag{2.7}$$

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

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ & & a'_{22}x_2 & + & \dots & + & a'_{2n}x_n & = & b'_2 \\ & & \vdots & & & & & & \vdots \\ & & a'_{n2}x_2 & + & \dots & + & a'_{nn}x_n & = & b'_n. \end{array} \tag{2.8}$$

Dabei ändert sich die Lösung nicht. Sukzessives Anwenden liefert nach $n-1$ Schritten:

$$\begin{array}{ccccccccc} r_{11}x_1 & + & r_{12}x_2 & + & \dots & + & r_{1n}x_n & = & y_1 \\ & & r_{22}x_2 & + & \dots & + & r_{2n}x_n & = & y_2 \\ & & & & \ddots & & \vdots & & \vdots \\ & & & & & & r_{nn}x_n & = & y_n \end{array} \tag{2.9}$$

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

Algorithmus 2.1 (Gauß-Eliminierung)**.** *Der folgende Algorithmus erzeugt* (2.9) *ausgehend von* (2.7)*.*

Schleife $i = 1, \dots, n$
Suche ein Element $a_{pi} \neq 0$ *mit* $p = i, \dots, n$*, sogenannte Pivotsuche.* *Sind alle* $a_{pi} = 0$*, so ist* $\mathbf{A}$ *singulär* $\Rightarrow$ *Stop.* *Anderenfalls vertausche* i*-te und* p*-te Zeile*
Schleife $k = i+1, \dots, n$
Setze $l_{ki} = a_{ki}/a_{ii}$
Setze $a_{kj} = a_{kj} - l_{ki}a_{ij}, \quad j = i+1, \dots, n$
Setze $b_k = b_k - l_{ki}b_i$

(2.10)

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*.*

Der Aufwand von (2.10) 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 Näherung

$$(n-1)^2 + (n-2)^2 + \ldots + 4 + 1 = \sum_{i=1}^{n-1} i^2 \doteq \frac{1}{3}n^3$$

Operationen. Die gebräuchlichste Pivotsuche in (2.10) ist gegeben durch

$$|a_{pi}| = \max_{k=i,\dots,n} |a_{ki}| \qquad \text{(Spaltenpivotsuche)}. \tag{2.11}$$

Eine andere Möglichkeit ist

$$|a_{pq}| = \max_{\substack{k=i,\dots,n \\ j=i,\dots,n}} |a_{kj}| \qquad \text{(Totalpivotsuche)}. \tag{2.12}$$

In diesem Fall ist in (2.10) auch noch die i-te und q-te Spalte zu tauschen, was einer Umbenennung der Unbekannten entspricht. In beiden Fällen gilt

$$|l_{ki} \leq 1|, \quad 1 \leq i \leq k \leq n. \tag{2.13}$$

Im folgenden betrachten wir nur Spaltenpivotsuche. Wegen eventueller Rundungsfehler muss die Abfrage

$$|a_{pi} = 0|$$

noch ersetzt werden, etwa durch $|a_{pi}| \leq \|\mathbf{A}\| \cdot \text{eps}$. Hat man Algorithmus 2.1 erfolgreich durchgeführt, so erhält man die gesuchte Lösung x folgendermaßen.

Algorithmus 2.2 (Rückwärtssubstitution)**.** *Im Anschluss an* (2.10) *ist folgendes durchzuführen:*

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

(2.14)

Der Aufwand in (2.14) beträgt in erster Näherung:

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

Operationen. Algorithmus 2.1 zeigt, dass es zu einer nichtsingulären Matrix $\mathbf{A}$ stets eine Permutation $\mathbf{\Pi} \in \mathbb{R}^{n \times n}$ gibt, so dass (2.10) für die permutierte Matrix $\mathbf{\Pi A}$ ohne Pivoting durchführbar ist. Dabei kann der Übergang von (2.7) zu (2.8) beschrieben werden als Multiplikation von $\mathbf{\Pi A}$ mit $\mathbf{E_1}$ von links, wobei

$$\mathbf{E_1} = \begin{pmatrix} 1 & & & \\ -l_{21} & 1 & & \\ \vdots & & \ddots & \\ -l_{n1} & & & 1 \end{pmatrix} \qquad \text{(Frobeniusmatrix).}$$

Bezeichnet

$$\begin{aligned} e_i &= (0, \ldots, 0, 1, 0, \ldots, 0)^{\mathrm{T}}, \quad i = 1, \ldots, n \\ l_i &= (0, \ldots, 0, l_{i+1}, \ldots, l_{ni})^{\mathrm{T}}, \quad i = 1, \ldots, n-1 \\ \mathbf{E_i} &= \mathbf{I} - l_i e_i^{\mathrm{T}}, \quad i = 1, \ldots, n-1 \\ \mathbf{R} &= \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix}, \end{aligned} \tag{2.15}$$

so gilt $\mathbf{E_{n-1}} \cdots \mathbf{E_2 E_1 \Pi A} = \mathbf{R}$, beziehungsweise

$$\mathbf{\Pi A} = \mathbf{E_1^{-1} E_2^{-1}} \cdots \mathbf{E_{n-1}^{-1} R}. \tag{2.16}$$

Wegen

$$\begin{aligned} e_j^{\mathrm{T}} l_i = l_i^{\mathrm{T}} e_j = 0 \quad &\text{für } j \le i \\ \mathbf{E_i^{-1}} = \mathbf{I} + l_i e_i^{\mathrm{T}} \quad & \end{aligned} \tag{2.17}$$

folgt induktiv

$$\mathbf{E_1^{-1} E_2^{-1}} \cdots \mathbf{E_{n-1}^{-1}} = (\mathbf{I} + l_1 e_1^{\mathrm{T}})(\mathbf{I} + l_2 e_2^{\mathrm{T}}) \cdots (\mathbf{I} + l_{n-1} e_{n-1}^{\mathrm{T}}) = \mathbf{I} + l_1 e_1^{\mathrm{T}} + l_2 e_2^{\mathrm{T}} + \cdots + l_{n-1} e_{n-1}^{\mathrm{T}}, \tag{2.18}$$

das heißt

$$\mathbf{L} = \mathbf{I} + \sum_{i=1}^{n-1} l_i e_i^{\mathrm{T}} = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{pmatrix} \tag{2.19}$$

ist eine normierte untere Dreiecksmatrix. Damit haben wir bewiesen:

Satz 2.3. *Sei $\mathbf{A} \in \mathbb{R}^{n \times n}$ regulär. Dann gibt es eine Permutation $\mathbf{\Pi} \in \mathbb{R}^{n \times n}$, eine reguläre obere Dreiecksmatrix $\mathbf{R} \in \mathbb{R}^{n \times n}$ und eine normierte untere Dreiecksmatrix $\mathbf{L} \in \mathbb{R}^{n \times n}$, so dass*

$$\mathbf{\Pi A} = \mathbf{LR} \qquad \textit{(Dreieckszerlegung).} \tag{2.20}$$

□

Algorithmus 2.4 (Dreieckszerlegung, L-R-Zerlegung)**.** *Man erhält in Algorithmus 2.1 den Faktor $\mathbf{L}$ ohne zusätzlichen Speicher, indem man l_{ki} nach a_{ki} schreibt, dessen Wert nicht mehr benötigt wird, und diese Werte beim Pivoting mittauscht.*

Unter Verwendung der Dreieckszerlegung zerfällt die Lösung von 2.1 in die drei Schritte

$$\begin{array}{ll} \mathbf{\Pi A} = \mathbf{LR} & \text{(Dreieckszerlegung)} \\ \mathbf{L}y = \mathbf{\Pi}b & \text{(Vorwärtssubstitution)} \\ \mathbf{R}x = y & \text{(Rückwärtssubstitution).} \end{array} \tag{2.21}$$

Zur Berechnung der Güte einer Näherungslösung $\tilde{x}$ verwendet man wegen der leichten Berechenbarkeit gerne das Residuum

$$r = b - \mathbf{A}\tilde{x}. \tag{2.22}$$

Wie klein dieses sein sollte, beantwortet der nächste Satz, der gleichzeitig einen ersten Schritt zur Rückwärtsanalyse der Gauß-Elimination darstellt. Dabei sind im folgenden Beträge beziehungsweise Ungleichungen von Vektoren und Matrizen stets komponentenweise zu verstehen.

Satz 2.5 (Satz von Prager/Oettli)**.** *Sei $\tilde{x}$ eine Näherungslösung von $\mathbf{A}x = b$ mit Residuum r. Weiter sei $\Delta\mathbf{A} \in \mathbb{R}^{n\times n}$ und $\Delta b \in \mathbb{R}^n$ mit $\Delta\mathbf{A} \geq 0$ und $\Delta b \geq 0$ gegeben. Dann ist $\tilde{x}$ genau dann exakte Lösung von*

$$\tilde{\mathbf{A}}\tilde{x} = \tilde{b} \tag{2.23}$$

mit

$$\Delta\mathbf{A} \geq |\mathbf{A} - \tilde{\mathbf{A}}|, \qquad |b - \tilde{b}| \leq \Delta b, \tag{2.24}$$

wenn gilt:

$$|r| \leq \Delta\mathbf{A} \cdot |\tilde{x}| + \Delta b. \tag{2.25}$$

Beweis. Sei $\tilde{\mathbf{A}}\tilde{x} = \tilde{b}$ mit (2.24), d.h. sei $\tilde{\mathbf{A}} = \mathbf{A} + \delta\mathbf{A}$, $\tilde{b} = b + \delta b$ mit $|\delta\mathbf{A}| \leq \Delta\mathbf{A}$, $|\delta b| \leq \Delta b$. Dann folgt

$$|r| = |b - \mathbf{A}\tilde{x}| = |\tilde{b} - \delta b - \tilde{\mathbf{A}}\tilde{x} + \delta\mathbf{A}\tilde{x}| = |\delta\mathbf{A}\tilde{x} - \delta b| \leq |\delta\mathbf{A} \cdot \tilde{x}| + |\delta b| \leq \Delta\mathbf{A}|\tilde{x}| + \Delta b.$$

Sei umgekehrt $|r| \leq \Delta\mathbf{A}\cdot|\tilde{x}|+\Delta b$. Mit $r = (r_i)$, $s = (s_i) = \Delta\mathbf{A}\cdot|\tilde{x}|+\Delta b \geq 0$, $\tilde{x} = (\tilde{x}_i)$ sowie $\Delta\mathbf{A} = (\Delta A_{ij})$, $\Delta b = (\Delta b_i)$ definiert man

$$\delta A_{ij} = \Delta A_{ij}\text{sign}(\tilde{x}_j)\frac{r_i}{s_i}, \qquad \delta b_i = -\Delta b_i\frac{r_i}{s_i}$$

mit der Konvention, dass $r_i/s_i = 0$, falls $s_i = 0$. Also folgt

$$|\delta\mathbf{A}| \leq \Delta A, \qquad |\delta b| \leq \Delta b$$

sowie

$$(\delta\mathbf{A} \cdot \tilde{x})_i = \sum_{j=1}^{n} \delta A_{ij}\tilde{x}_j = \sum_{j=1}^{n} \Delta A_{ij}|\tilde{x}_j|\frac{r_i}{s_i} = (\Delta\mathbf{A} \cdot |\tilde{x}|)_i\frac{r_i}{s_i} = (s_i - \Delta b_i)\frac{r_i}{s_i} = r_i - \Delta b_i\frac{r_i}{s_i} = (r + \delta b)_i.$$

Setzt man nun $\tilde{\mathbf{A}} = \mathbf{A} + \delta\mathbf{A}$, $\tilde{b} = b + \delta b$, so ergibt sich

$$\tilde{\mathbf{A}}\tilde{x} = (\mathbf{A} + \delta\mathbf{A})\tilde{x} = \mathbf{A}\tilde{x} + \delta\mathbf{A}\tilde{x} = b - r + r + \delta b = \tilde{b}. \qquad \square$$

Für die weitere Untersuchung nehmen wir an, dass $\mathbf{A}$ bereits so vorliegt, dass kein Zeilentausch durchgeführt werden muss, d.h. sei

$$\mathbf{A} = \mathbf{LR}. \tag{2.26}$$

Mit $\mathbf{L} = (l_{ij})$, $\mathbf{R} = (r_{ij})$ gilt

$$\begin{array}{llll} l_{ii} = 1, & l_{ij} = 0 & \text{für } i < j, & j, i = 1, \ldots, n \\ r_{ii} \neq 0, & r_{ij} = 0 & \text{für } i > j. & \end{array} \tag{2.27}$$

In

$$a_{ij} = \sum_{k=1}^{n} l_{ik}r_{kj} \tag{2.28}$$

genügt es also, nur $k \leq \min\{i,j\}$ zu betrachten. Man erhält dann

$$\begin{array}{ll} a_{ij} = \sum_{k=1}^{i} l_{ik} r_{kj} = r_{ij} + \sum_{k=1}^{i-1} l_{ik} r_{kj} & \text{für } i \leq j \\ a_{ij} = \sum_{k=1}^{j} l_{ik} r_{kj} = l_{ij} r_{jj} + \sum_{k=1}^{j-1} l_{ik} r_{kj} & \text{für } i > j \end{array} \tag{2.29}$$

beziehungsweise

$$\begin{array}{ll} r_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} r_{kj} & \text{für } i \leq j \\ l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} r_{kj})/r_{jj} & \text{für } i > j. \end{array} \tag{2.30}$$

Mit (2.30) kann man die Dreieckszerlegung (2.26) direkt berechnen. Algorithmus 2.4 entspricht daher dem sukzessiven Aufaddieren der Summanden. Damit besteht dieser Algorithmus aus der Lösung einer Reihe von Problemen der Form

$$c - a_1 b_1 - \cdots - a_{n-1} b_{n-1} - a_n b_n = 0, \quad a_n \neq 0 \tag{2.31}$$

bezüglich

$$b_n = (c - a_1 b_1 - \cdots - a_{n-1} b_{n-1})/a_n \tag{2.32}$$

mit dem Algorithmus

$$\begin{aligned} &s_0 = c \\ &s_j = s_{j-1} - a_j b_j, \quad j = 1, \ldots, n-1 \\ &b_n = s_{n-1}/a_n. \end{aligned} \tag{2.33}$$

Unter der Annahme, dass alle Daten Maschinenzahlen sind, wird dieser auf dem Rechner realisiert durch

$$\begin{aligned} &\tilde{s}_0 = c \\ &\tilde{s}_j = [\tilde{s}_{j-1} - a_j b_j (1+\mu_j)](1+\sigma_j), \quad j = 1, \ldots, n-1 \\ &\tilde{b}_n = (\tilde{s}_{n-1}/a_n)(1+\delta), \end{aligned} \tag{2.34}$$

wobei $|\mu_j|, |\sigma_j|, |\delta| \leq \text{eps}$.

Lemma 2.6. Unter den obigen Annahmen gilt für

$$r = c - \sum_{j=1}^{n-1} a_j b_j - a_n \tilde{b}_n \tag{2.35}$$

die Abschätzung

$$|r| \leq \frac{\text{eps}}{1 - n \cdot \text{eps}} \left[\sum_{j=1}^{n-1} j |a_j b_j| + n |a_n \tilde{b}_n| \right], \tag{2.36}$$

solange $n \cdot \text{eps} < 1$.

Beweis. Aus (2.34) folgt

$$\tilde{b}_n = \left[c \prod_{j=1}^{n-1} (1+\sigma_j) - \sum_{j=1}^{n-1} a_j b_j (1+\mu_j) \prod_{k=j}^{n-1} (1+\sigma_k) \right] \frac{1+\delta}{a_n}.$$

Es gibt dann ε_j, $j = 1, \ldots, n$ mit $|\varepsilon_j| \leq \text{eps}/(1 - n \cdot \text{eps})$, so dass

$$c = \sum_{j=1}^{n-1} a_j b_j (1+\mu_j) \prod_{k=1}^{j-1} (1+\sigma_k)^{-1} + a_n \tilde{b}_n (1+\delta)^{-1} \prod_{j=1}^{n-1} (1+\sigma_j)^{-1} = \sum_{j=1}^{n-1} a_j b_j (1+j\varepsilon_j) + a_n \tilde{b}_n (1+n\varepsilon_n),$$

siehe Übungen. Die Behauptung folgt nun sofort aus

$$r = \sum_{j=1}^{n-1} j a_j b_j \varepsilon_j + n a_n \tilde{b}_n \varepsilon_n.$$

□

Bemerkung 2.7. Ist speziell $a_n = 1$ in (2.31), so ist $\delta = 0$ und der Faktor n vor $|a_n \tilde{b}_n|$ kann durch $n-1$ ersetzt werden.

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:

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

vergleiche Abschnitt **??**. 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

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

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

$$\begin{aligned} &\mathbf{A} = (a_{ij}), \quad && a_{ij} = \varphi_j(t_i), \\ &b = (b_i), && b_i = f_i, \end{aligned} \tag{3.3}$$

so ist (3.1) äquivalent zum linearen Gleichungssystem $\mathbf{A}x = b$. 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 (sogenannte Lagrange-Polynome) Für diese gilt

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

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

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

das heißt das Polynom $p_2 - p_1 \in \Pi_n$ besitzt die $n + 1$ Nullstellen t_i, $i = 0, \ldots, n$, muss deshalb also das Nullpolynom sein, das heißt $p_2 - p_1 = 0$ beziehungsweise $p_1 = p_2$. □

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 $\mathbf{A}$ zu berechen und das zugehörige Gleichungssystem zu lösen, abhängen. Darüber hinaus ist es wichtig, die zugehörige Darstellung (3.2) des Interpolationspolynoms effizient auswerten zu können.

Bemerkung 3.2. Mögliche Basen sind zum Beispiel:

- Lagrange-Basis:
$$\varphi_j(t) = l_j(t), \quad j = 0, \dots, n,$$
$$\mathbf{A} = (l_j(t_i)) = (\delta_{ij}) = \mathbf{I};$$

- Monom-Basis:
$$\varphi_j(t) = t^j, \quad j = 0, \dots, n,$$
$$\mathbf{A} = (t_i^j); \quad \textit{(Vandermonde-Matrix)}$$

- Newton-Basis:
$$\varphi_j(t) = \omega_j(t) = \prod_{k=0}^{j-1}(t - t_k), \quad j = 0, \dots, n$$
$$\mathbf{A} = (\omega_j(t_i)). \quad \textit{(untere Dreiecksmatrix wegen } \omega_j(t_i) = 0 \textit{ für } i < j\textit{)}$$

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. Wegen $P(\cdot; t_{k+1}, \dots, t_{k+l}), P(\cdot; t_k, \dots, t_{k+l-1}) \in \Pi_{l-1}$ gilt $Q \in \Pi_l$, wenn man die rechte Seite von (3.5) mit $Q(t)$ bezeichnet. 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

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

□

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

$$P(t; t_k, \dots, t_{k+l}) = P(t; t_k, \dots, t_{k+l-1}) - \frac{t - t_k}{t_{k+l} - t_k}\left[P(t; t_k, \dots, t_{k+l-1}) - P(t; t_{k+1}, \dots, t_{k+l})\right], \tag{3.6}$$

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

Setze $P_{k,0} = f_k$, $k = 0, \dots, n$
Schleife $l = 1, \dots, n$
Setze $P_{k,l} = P_{k,l-1} - \dfrac{t - t_k}{t_{k+l} - t_k}(P_{k,l-1} - P_{k+1,l-1})$, $k = 0, \dots, n-l$

(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

$$2\sum_{l=1}^{n}(n-l) = 2\sum_{l=0}^{n-1} l \doteq n^2$$

Multiplikationen/Divisionen.

Definition 3.5 (Dividierte Differenzen)**.** Die sogenannten dividierten Differenzen sind rekursiv definiert durch

$$\begin{aligned} &f[t_k] = f_k, \quad k = 0, \ldots, n, \\ &f[t_k, \ldots, t_{k+l}] = \frac{f[t_k, \ldots, t_{k+l-1}] - f[t_{k+1}, \ldots, t_{k+l}]}{t_k - t_{k+l}}, \quad k = 0, \ldots, n-l, \quad l = 1, \ldots, n. \end{aligned} \tag{3.8}$$

Lemma 3.6. Für $k = 0, \ldots, n-l$, $l = 0, \ldots, n$ gilt

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

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. Dabei ist die Behauptung trivial für $l = 0$. Sei also $l \geq 1$. Nach Induktionsvoraussetzung ist $f[t_{k+1}, \ldots, t_{k+l}]$ der Koeffizient der höchsten Potenz von t in $P(t; t_{k+1}, \ldots, t_{k+l})$, entsprechend $f[t_k, \ldots, t_{k+l-1}]$ der in $P(t; t_k, \ldots, t_{k+l-1})$. Wegen (3.5) ist dann der Koeffizient der höchsten Potenz von t in $P(t; t_k, \ldots, t_{k+l})$ gegeben durch

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

letzteres wegen (3.8). □

Algorithmus 3.7. *Der folgende Algorithmus berechnet die dividierten Differenzen für das Polynom* $P(t; t_0, \ldots, t_n)$*.*

Setze $D_{k,0} = f_k$*,* $k = 0, \ldots, n$
Schleife $l = 1, \ldots, n$
$\quad$ *Setze* $D_{k,l} = \dfrac{D_{k,l-1} - D_{k+1,l-1}}{t_k - t_{k+l}}$*,* $k = 0, \ldots, n-l$

(3.10)

Benötigt werden die Werte $D_{0,l} = f[t_0, \ldots, t_l]$*,* $l = 0, \ldots, n$*. 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

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

Divisionen.

Algorithmus 3.8 (Horner-Schema)**.** *Naives Vorgehen bei der Auswertung eines Polynoms*

$$p(t) = a_0 + a_1 t + \cdots + a_n t^n \tag{3.11}$$

in der Darstellung bezüglich der Monom-Basis führt auf den Algorithmus

Setze $s = a_0$, $r = 1$
Schleife $i = 1, \dots, n$
$\quad$ *Setze* $r = r \cdot t$
$\quad$ *Setze* $s = s + a_i r$
Setze $p(t) = s$

(3.12)

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

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

so erhält man das sogenannte Horner-Schema

Setze $s = a_n$
Schleife $i = 1, \dots, n$
$\quad$ *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 $i = 1, \dots, n$
$\quad$ *Setze* $s = s(t - t_{n-i}) + v_{n-i}$
Setze $p(t) = s$

(3.16)

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

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

eine gegebene Funktion f approximiert.

Satz 3.9. *Sei* $\bar{t} \in \mathbb{R}$ *und* $f \in \mathcal{C}^{n+1}(\mathbb{I}, \mathbb{R})$*, wobei* $\mathbb{I}$ *das kleinste Intervall mit* $\bar{t}, t_0, \dots, t_n \in \mathbb{I}$ *ist. Dann gibt es ein* $\tau \in \mathbb{I}$ *mit*

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

Beweis. Die Behauptung ist richtig für $\bar{t} = t_i$, $i = 0, \dots, n$. Sei also im folgenden $\bar{t} \neq t_i$, $i = 0, \dots, n$. 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 hat die Funktion $g(t) = f(t) - p(t) - C\omega(t)$ in $\mathbb{I}$ $n + 2$ Nullstellen, nämlich $\bar{t}, t_0, \dots, t_n$. 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, sagen wir $\tau \in \mathbb{I}$. Wegen $p^{(n+1)} = 0$ gilt dafür

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

das heißt

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

□

Bemerkung 3.10. Wählt man $\bar{p} \in \Pi_{n+1}$ derart, dass

$$\bar{p}(t_i) = f(t_i), \quad i = 0, \ldots, n, \quad \bar{p}(\bar{t}) = f(\bar{t}) \tag{3.19}$$

mit $\bar{t} \neq t_i$, $i = 0, \ldots, n$, so gilt

$$\bar{p}(t) = p(t) + f[t_0, \ldots, t_n, \bar{t}]\omega(t). \tag{3.20}$$

Setzt man $t = \bar{t}$ und vergleicht mit (3.18), so gibt es ein $\tau \in \mathbb{I}$ mit

$$f[t_0, \ldots, t_n, \bar{t}\,] = \frac{f^{(n+1)}(\tau)}{(n+1)!}. \tag{3.21}$$

Bemerkung 3.11. Zwar kann man jede auf einem Intervall $[a, b\,]$ stetige Funktion gleichmäßig, das heißt bezüglich der Norm $\|f\|_{\mathcal{C}^0} = \sup_{t\in[a,b]} |f(t)|$ beliebig genau durch Polynome approximieren (Approximationssatz von Weierstraß), aber typischerweise findet man diese Polynome nicht durch Interpolation. Ist o.B.d.A. $t_0 < t_1 < \cdots < t_n$ und setzt man $h := \max_{i=0,\ldots,n-1} h_i$, $h_i := t_{i+1} - t_i$, so gilt für $n \to \infty$ bei gleichzeitigem $h \to 0$ für die Interpolationspolynome p_n einer stetigen Funktion f meist $\|f - p_n\|_{\mathcal{C}^0} \to \infty$, vergleiche den Satz von Faber.

3.2 Trigonometrische Interpolation

Ist f periodisch, so sollte auch die Interpolierende periodisch sein. Sei o.B.d.A. die Periode 2π. Man macht dann (für eine ungerade Anzahl von Interpolationsbedingungen) den Ansatz

$$q(t) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos kt + b_k \sin kt). \tag{3.22}$$

Bezeichnet i im folgenden die imaginäre Einheit, so folgt aus

$$e^{i\varphi} = \cos\varphi + i \sin\varphi \tag{3.23}$$

beziehungsweise

$$\cos\varphi = \frac{1}{2}(e^{i\varphi} + e^{-i\varphi}), \qquad \sin\varphi = \frac{1}{2i}(e^{i\varphi} - e^{-i\varphi}) \tag{3.24}$$

die Darstellung

$$q(t) = \frac{a_0}{2} + \sum_{k=1}^{n} \left[\left(\frac{a_k}{2} + \frac{b_k}{2i} \right) e^{ikt} + \left(\frac{a_k}{2} - \frac{b_k}{2i} \right) e^{-ikt} \right]. \tag{3.25}$$

Setzt man

$$c_{n+k} = \frac{a_k}{2} + \frac{b_k}{2i}, \quad c_{n-k} = \frac{a_k}{2} - \frac{b_k}{2i}, \quad k = 0, \ldots, n \tag{3.26}$$

mit $b_0 := 0$, so erhält man mit

$$p(z) = e^{int} q(t) = \sum_{k=0}^{2n} c_k e^{ikt} = \sum_{k=0}^{2n} c_k z^k, \quad z = e^{it} \tag{3.27}$$

ein Polynom p in z vom Grad höchstens $2n$. Satz 3.1, der bei gleichem Beweis auch im Komplexen gilt, besagt, dass es genau eine Funktion $p \in \Pi_{2n}$ gibt, die

$$p(z_j) = f_j, \quad j = 0, \ldots, 2n \qquad \text{mit } z_j, f_j \in \mathbb{C}, \quad z_j \neq z_l \text{ für } j \neq l \tag{3.28}$$

erfüllt. Hat man für q die Interpolationsbedingungen

$$q(t_j) = g_j, \quad j = 0, \ldots, 2n \tag{3.29}$$

mit $t_j \in [0, 2\pi]$, $g_j \in \mathbb{R}$, $t_j \neq t_l$ für $j \neq l$, so setzt man

$$f_j = e^{int_j} g_j, \quad z_j = e^{it_j}, \quad j = 0, \ldots, 2n \tag{3.30}$$

und erhält eindeutig bestimmte Koeffizienten c_k, $k = 0, \ldots, 2n$. Wegen $t_j, g_j \in \mathbb{R}$ gilt neben

$$e^{int_j} g_j = \sum_{k=0}^{2n} c_k e^{ikt_j}, \quad j = 0, \ldots, 2n \tag{3.31}$$

auch

$$e^{-int_j} g_j = \sum_{k=0}^{2n} \overline{c_k} e^{-ikt_j}, \quad j = 0, \ldots, 2n. \tag{3.32}$$

Multiplikation von (3.32) mit e^{2int_j} liefert

$$e^{int_j} g_j = \sum_{k=0}^{2n} \overline{c_{2n-k}} e^{ikt_j}, \quad j = 0, \ldots, 2n. \tag{3.33}$$

Wegen der Eindeutigkeit des Interpolationspolynoms folgt

$$\overline{c_{2n-k}} = c_k, \quad k = 0, \ldots, 2n \tag{3.34}$$

beziehungsweise

$$\overline{c_{n+k}} = c_{n-k}, \quad k = 0, \ldots, n. \tag{3.35}$$

Aus (3.26) erhält man

$$\begin{aligned} a_k &= c_{n+k} + c_{n-k} = \overline{c_{n+k}} + c_{n+k} = 2\Re(c_{n+k}) \\ b_k &= i(c_{n+k} - c_{n-k}) = i(c_{n+k} - \overline{c_{n+k}}) = 2\Im(c_{n+k}) \end{aligned} \tag{3.36}$$

und es ist $a_k, b_k \in \mathbb{R}$, $k = 0, \ldots, n$.

Satz 3.12. *Gegeben seien $t_j \in [0, 2\pi)$, $g_j \in \mathbb{R}$, $j = 0, \ldots, 2n$, mit $t_j \neq t_l$ für $j \neq l$. Dann gibt es genau ein trigonometrisches Polynom* (3.22) *mit* (3.29).

□

Im folgenden sei angenommen, dass die Knoten t_j äquidistant sind, d.h.

$$t_j = \frac{2\pi j}{2n+1}, \quad j = 0, \ldots, 2n. \tag{3.37}$$

Lemma 3.13. Im Spezialfall (3.37) sind die Koeffizienten von (3.22) gegeben durch

$$\begin{aligned} a_k &= \frac{2}{2n+1} \sum_{l=0}^{2n} g_l \cos lt_k, \\ b_k &= \frac{2}{2n+1} \sum_{l=0}^{2n} g_l \sin lt_k, \quad k = 0, \ldots, n. \end{aligned} \tag{3.38}$$

Beweis. Setzt man mit (3.30)

$$c_k = \frac{1}{2n+1} \sum_{l=0}^{2n} f_l e^{-ilt_k} = \frac{1}{2n+1} \sum_{l=0}^{2n} f_l e^{-\frac{2\pi ikl}{2n+1}},$$

so gilt

$$p(z_j) = \sum_{k=0}^{2n} c_j z_j^k = \frac{1}{2n+1} \sum_{k=0}^{2n} \sum_{l=0}^{2n} f_l e^{-\frac{2\pi ikl}{2n+1}} e^{\frac{2\pi ijk}{2n+1}} = \sum_{l=0}^{2n} \left[\frac{1}{2n+1} \sum_{k=0}^{2n} e^{\frac{2\pi i(j-l)k}{2n+1}} \right] f_l = \sum_{l=0}^{2n} \delta_{jl} f_l = f_j.$$

Damit ist (unter Verwendung von $t_{n\pm k} = t_n \pm t_k$ und $lt_n = nt_l$)

$$a_k = \frac{1}{2n+1} \sum_{l=0}^{2n} f_l \left[e^{-ilt_{n+k}} + e^{-ilt_{n-k}} \right] = \frac{1}{2n+1} \sum_{l=0}^{2n} g_l \left[e^{-ilt_k} + e^{ilt_k} \right] = \frac{2}{2n+1} \sum_{l=0}^{2n} g_l \cos lt_k.$$

Entsprechende Rechnung für b_k zeigt die Behauptung. □

Zur Berechnung der a_k und b_k sowie zur Auswertung von q müssen Summen der Form

$$A = \sum_{l=0}^{2n} g_l \cos lt, \qquad B = \sum_{l=0}^{2n} g_l \sin lt \tag{3.39}$$

berechnet werden. Dabei würde man gerne möglichst wenige Auswertungen von Standardfunktionen verwenden.

Satz 3.14. *Sei $t \in \mathbb{R}$ mit $\sin t \neq 0$ und*

$$\begin{aligned} U_j &= \frac{1}{\sin t} \sum_{l=j}^{2n} g_l \sin\left[(l-j+1)t\right], \quad j = 0, \ldots, 2n \\ U_{2n+1} &= U_{2n+2} = 0. \end{aligned} \tag{3.40}$$

Dann gilt

$$U_j = g_j + 2U_{j+1} \cos t - U_{j+2}, \quad j = 2n, \ldots, 0 \tag{3.41}$$

und

$$A = g_0 + U_1 \cos t - U_2, \qquad B = U_1 \sin t. \tag{3.42}$$

Beweis. Die Darstellung von B folgt sofort aus der Definition von U_1, für die andere Darstellung gilt

$$\begin{aligned} g_0 + U_1 \cos t - U_2 &= g_0 + \frac{\cos t}{\sin t} \sum_{l=1}^{2n} g_l \sin lt - \frac{1}{sint} \sum_{l=2}^{2n} g_l \sin\left[(l-1)t\right] \\ &= g_0 + \frac{1}{\sin t} \sum_{l=1}^{2n} g_l \left[\sin lt \cos t - sin\left[(l-1)t\right]\right] \\ &= g_0 + \frac{1}{\sin t} \sum_{l=1}^{2n} g_l \cos lt \sin t = A. \end{aligned}$$

Weiter folgt aus (3.40):

$$U_{2n} = \frac{1}{\sin t} g_{2n} \sin t = g_{2n}$$

sowie

$$U_{2n-1} = \frac{1}{\sin t} \left(g_{2n-1} \sin t - g_{2n} \sin 2t\right) = g_{2n-1} + g_{2n} \cdot 2 \cos t.$$

Für die Induktion fehlt noch

$$\begin{aligned} g_j + 2U_{j+1} \cos t - U_{j+2} &= g_j + \frac{2}{\sin t} \sum_{l=j+1}^{2n} g_l \sin(l-j)t \cos t - \frac{1}{\sin t} \sum_{l=j+2}^{2n} g_l \sin(l-j-1)t \\ &= g_j + \frac{1}{\sin t} \sum_{l=j+1}^{2n} g_l \left[2 \sin(l-j)t \cos t - \sin(l-j-1)t\right] \\ &= g_j + \frac{1}{\sin t} \sum_{l=j+1}^{2n} g_l \sin(l-j+1)t = U_j. \end{aligned}$$

□

Algorithmus 3.15 (Algorithmus von Goertzel)**.** *Satz 3.14 liefert folgenden Algorithmus zur Berechnung von A und B zu gegebenem $t \in \mathbb{R}$ sowie $g_0, \ldots, g_{2n} \in \mathbb{R}$.*

Setze $c = \cos t$, $d = 2c$
Setze $U_{2n+2} = U_{2n+1} = 0$
Schleife $j = 2n, \ldots, 1$
$\quad$ *Setze* $U_j = g_j + dU_{j+1} - U_{j+2}$
Setze $A = g_0 + cU_1 - U_2$, $B = U_1 \sin t$

(3.43)

Dieser Algorithmus ist aber für $|\sin t|$ hinreichend klein instabil. Gleich zu Beginn der Berechnung wird die Information über t durch c ersetzt. Schreibt man $A = f(t, g_0, \ldots, g_{2n})$, so hat man als differentielle Auswirkung eines Fehlers in t:

$$\Delta A \doteq \frac{\partial f}{\partial t}(t, g_0, \ldots, g_{2n})\Delta t = -\sum_{l=0}^{2n} g_l \sin lt \cdot lt \frac{\Delta t}{t}. \tag{3.44}$$

Wegen $c = \cos t$ wirkt sich ein Fehler in c stattdessen entsprechend

$$\Delta A \doteq \left(\frac{\frac{\partial f}{\partial t}}{\frac{\partial c}{\partial t}}\right)(t, g_0, \ldots, g_{2n})\Delta c = \sum_{l=0}^{2n} g_l \sin lt \cdot l \cdot \cot t \frac{\Delta c}{c} \tag{3.45}$$

aus. Die Instabilität ergibt sich sofort aus $|\cot t| \to \infty$ für $t \to 0$ ($|\sin t| \to 0$). Für $\cos t \geq 0$ kann man Algorithmus 3.15 wie folgt stabilisieren: Mit

$$\delta U_j = U_j - U_{j+1} \tag{3.46}$$

ergibt sich anstatt (3.41) die Rekursion

$$\delta U_j = g_j + 2U_{j+1}\cos t - U_{j+2} - U_{j+1} = g_j + \lambda U_{j+1} + \delta U_{j+1} \tag{3.47}$$

mit

$$\lambda = 2(\cos t - 1) = -4\sin^2 \frac{t}{2}. \tag{3.48}$$

Dafür gilt

$$\Delta A \doteq \left(\frac{\frac{\partial f}{\partial t}}{\frac{\partial \lambda}{\partial t}}\right)(t, g_0, \ldots, g_{2n})\Delta\lambda = -\sum_{l=0}^{2n} g_l \sin lt \cdot l \tan\frac{t}{2} \cdot \frac{\Delta\lambda}{\lambda} \tag{3.49}$$

mit $|\tan \frac{t}{2}| \leq 1$ für $\cos t \geq 0$. Entsprechend kann man auch für $\cos t \leq 0$ vorgehen. Insgesamt erhält man den folgenden Algorithmus.

Algorithmus 3.16 (Algorithmus von Goertzel/Reinsch)**.** *Die stabilisierte Form von Algorithmus 3.15 lautet*

Setze $\lambda = \begin{cases} -4\sin^2\frac{t}{2} & \text{für } \cos t \geq 0 \\ 4\cos^2\frac{t}{2} & \text{für } \cos t < 0 \end{cases}$
Setze $U_{2n+2} = \delta U_{2n+1} = 0$
Schleife $j = 2n, \ldots, 0$
$\quad$ *Setze* $U_{j+1} = \begin{cases} \delta U_{j+1} + U_{j+2} & \text{für } \cos t \geq 0 \\ \delta U_{j+1} - U_{j+2} & \text{für } \cos t < 0 \end{cases}$
$\quad$ *Setze* $\delta U_j = \begin{cases} g_j + \lambda U_{j+1} + \delta U_{j+1} & \text{für } \cos t \geq 0 \\ g_j + \lambda U_{j+1} - \delta U_{j+1} & \text{für } \cos t < 0 \end{cases}$
Setze $A = \delta U_0 - \frac{\lambda}{2}U_1$, $B = U_1 \sin t$

(3.50)

Heutzutage benutzt man hauptsächlich FFT (Fast Fourier Transform), die beiden besprochenen Algorithmen haben also eher historischen Wert.

3.3 Spline-Interpolation

Um Effekte der Polynominterpolation bei größerem Grad n wie in Bemerkung 3.11 beschrieben zu vermeiden, kann man versuchen, Polynome geringeren Grades stückweise zu hinreichend glatten Funktionen zusammenzusetzen. Man nennt solche Funktionen Splines nach biegsamen Schablonen, die im Schiffsbau eingesetzt wurdenn, um die Krümmung des Schiffsrumpfes abzunehmen.

Definition 3.17 (kubische Splines)**.** Gegeben sei eine Unterteilung

$$a = t_0 < t_1 < \cdots < t_n = b \tag{3.51}$$

des Intervalls $[a, b]$. Eine Funktion $s : [a, b] \to \mathbb{R}$ heißt (zu (3.51) gehöriger) kubischer Spline, wenn $s \in \mathcal{C}^2([a, b], \mathbb{R})$ und $s|_{[t_i, t_{i+1}]}$ für jedes $i = 0, \ldots, n-1$ mit einem kubischen Polynom auf $[t_i, t_{i+1}]$ übereinstimmt.

Satz 3.18. *Sei $f \in \mathcal{C}^2([a, b], \mathbb{R})$. Dann gilt für jeden zu (3.51) gehörigen kubischen Spline s mit*

$$s(t_i) = f(t_i), \quad i = 0, \ldots, n \tag{3.52}$$

die Identität

$$\|f - s\|^2 = \|f\|^2 - \|s\|^2 \tag{3.53}$$

und damit die Minimaleigenschaft

$$\|s\| \le \|f\|, \tag{3.54}$$

wobei

$$\|u\|^2 = \int_a^b \ddot{u}(t)^2 \, dt \qquad \text{(Halbnorm auf } \mathcal{C}^2([a, b], \mathbb{R}) \text{ bzw. Norm auf } \mathcal{C}_0^2([a, b], \mathbb{R})\text{)}, \tag{3.55}$$

wenn s zusätzlich eine der drei Bedingungen

$$\begin{array}{ll} \ddot{s}(a) = 0 \text{ und } \ddot{s}(b) = 0 & \text{(natürlicher Spline)} \\ \dot{s}(a) = \dot{f}(a) \text{ und } \dot{s}(b) = \dot{f}(b) & \text{(eingespannter Spline)} \\ \dot{s}(a) = \dot{s}(b), \quad \ddot{s}(a) = \ddot{s}(b) & \text{(periodischer Spline)} \end{array} \tag{3.56}$$

erfüllt. Dabei wird im letzten Fall zusätzlich vorausgesetzt, dass f periodisch auf $[a, b]$ ist.

Beweis. Die Identität (3.53) ergibt sich aus

$$\|f-s\|^2 = \int_a^b \Big(\ddot{f}(t) - \ddot{s}(t)\Big)^2 \, dt = \|f\|^2 + \|s\|^2 - 2\int_a^b \ddot{f}(t)\ddot{s}(t)\, dt = \|f\|^2 - \|s\|^2 - 2\int_a^b \Big(\ddot{f}(t) - \ddot{s}(t)\Big)\ddot{s}(t)\, dt$$

und

$$\begin{aligned} \int_a^b \Big(\ddot{f}(t) - \ddot{s}(t)\Big)\ddot{s}(t)\, dt &= \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} \Big(\ddot{f}(t) - \ddot{s}(t)\Big)\ddot{s}(t)\, dt \\ &= \sum_{i=1}^{n-1} \Big(\dot{f}(t) - \dot{s}(t)\Big)\ddot{s}(t)\Big|_{t_i}^{t_{i+1}} - \int_{t_i}^{t_{i+1}} \Big(\dot{f}(t) - \dot{s}(t)\Big) \underbrace{\dddot{s}(t)}_{=d_i \in \mathbb{R}} \, dt \\ &= \Big(\dot{f}(t) - \dot{s}(t)\Big)\ddot{s}(t)\Big|_a^b - \sum_{i=0}^{n-1} d_i (f(t) - s(t))\Big|_{t_i}^{t_{i+1}} = 0 \end{aligned}$$

in allen drei Fällen. Dabei wurde ausgenutzt, dass $\dddot{s}(t) = d_i$ auf $[t_i, t_{i+1})$. □

Korollar 3.19. In allen drei Fällen in (3.56) ist der Spline s durch (3.52) eindeutig festgelegt.

Beweis. Seien s_1 und s_2 zwei solche Splines. Mit (3.53) gilt dann

$$\|s_2 - s_1\|^2 = \|s_2\|^2 - \|s_1\|^2 = 0,$$

das heißt

$$\int_a^b \left(\ddot{s}_2(t) - \ddot{s}_1(t)\right)^2 \, dt = 0.$$

Wegen der Stetigkeit des Integranden folgt daraus

$$\ddot{s}_1(t) = \ddot{s}_2(t), \quad t \in [a, b],$$

also

$$s_2(t) - s_1(t) = ct + d.$$

Mit $s_1(a) = s_2(a)$ und $s_1(b) = s_2(b)$ nach (3.52) folgt $c = d = 0$. □

Im folgenden betrachten wir nur natürliche Splines, d.h. (3.52) zusammen mit der ersten Bedingung aus (3.56). Zur Konstruktion von s setzen wir

$$h_i = t_{i+1} - t_i, \quad i = 0, \ldots, n-1 \tag{3.57}$$

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

$$s_i(t) = a_i + b_i(t - t_i) + \frac{1}{2}c_i(t - t_i)^2 + \frac{1}{6}d_i(t - t_i)^3 \tag{3.58a}$$

$$\dot{s}_i(t) = b_i + c_i(t - t_i) + \frac{1}{2}d_i(t - t_i)^2 \tag{3.58b}$$

$$\ddot{s}_i(t) = c_i + d_i(t - t_i). \tag{3.58c}$$

Zu erfüllen sind die Bedingungen

$$s_i(t_i) = f_i, \quad s_i(t_{i+1}) = f_{i+1}, \quad i = 0, \ldots, n-1 \tag{3.59a}$$

$$\dot{s}_i(t_{i+1}) = \dot{s}_{i+1}(t_{i+1}), \quad i = 0, \ldots, n-2 \tag{3.59b}$$

$$\ddot{s}_i(t_{i+1}) = \ddot{s}_{i+1}(t_{i+1}), \quad i = 0, \ldots, n-2 \tag{3.59c}$$

$$\ddot{s}_0(t_0) = 0, \quad \ddot{s}_{n-1}(t_n) = 0. \tag{3.59d}$$

Aus (3.59a) ergibt sich sofort

$$a_i = f_i, \quad i = 0, \ldots, n-1 \tag{3.60}$$

und aus (3.59c)

$$c_{i+1} = c_i + d_i h_i \tag{3.61}$$

oder

$$d_i = \frac{c_{i+1} - c_i}{h_i}, \quad i = 0, \ldots, n-1, \tag{3.62}$$

wobei $c_0 = c_n = 0$ entsprechend (3.59d). Damit folgt aus (3.59a)

$$b_i = \frac{f_{i+1} - f_i}{h_i} - \frac{1}{2}c_i h_i - \frac{1}{6}\frac{c_{i+1} - c_i}{h_i}h_i^2 = \frac{f_{i+1} - f_i}{h_i} - \frac{2c_i + c_{i+1}}{6}h_i, \quad i = 0, \ldots, n-1 \tag{3.63}$$

und alle Größen sind in den Unbekannten c_i ausgedrückt. Aus (3.59b) ergibt sich nun

$$b_i = b_{i-1} + c_{i-1}h_{i-1} + \frac{1}{2}d_{i-1}h_{i-1}^2, \quad i = 1, \ldots, n-1 \tag{3.64}$$

beziehungsweise

$$\frac{f_{i+1} - f_i}{h_i} - \frac{2c_i + c_{i+1}}{6}h_i = \frac{f_i - f_{i-1}}{h_{i-1}} - \frac{2c_{i-1} + c_i}{6}h_{i-1} + c_{i-1}h_{i-1} + \frac{1}{2}\frac{c_i - c_{i-1}}{h_{i-1}}h_{i-1}^2 \tag{3.65}$$

oder

$$\frac{h_{i-1}}{6}c_{i-1} + \frac{h_{i-1}+h_i}{3}c_i + \frac{h_i}{6}c_{i+1} = \frac{f_{i+1}-f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}}, \tag{3.66}$$

jeweils für $i = 1, \ldots, n-1$. 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 tridiagonaler, streng diagonaldominanter, symmetrischer, positiv definiter und damit regulärer Koeffizientenmatrix.
Mit der zugehörigen eindeutigen Lösung und den Beziehungen (3.60), (3.62) und (3.63) ist dann s bestimmt.

Bemerkung 3.20. Im Gegensatz zu Bemerkung 3.11 kann man hier für genügend glattes f zeigen, dass bei immer feiner werdendem Gitter $h = \max_{i=0,\ldots,n-1} h_i \to 0$ für die zugehörigen kubischen Splines s tatsächlich $s \to f$ gilt (punktweise).

4 Differentiation

Sei $a < b$ und $f \in \mathbb{X} = \mathcal{C}^1([a,b],\mathbb{R})$. Zu $\bar{t} \in [a,b]$ ist der Wert $\dot{f}(\bar{t})$ zu bestimmen. Wir betrachten also für festes $\bar{t}$ die Abbildung

$$D : \mathbb{X} \to \mathbb{R}, \quad f \mapsto \dot{f}(\bar{t}) \tag{4.1}$$

4.1 Kondition des Problems

Durch die Wahl

$$\|f\|_{\mathbb{X}} = \|f\|_{\mathcal{C}^0} + \|\dot{f}\|_{\mathcal{C}^0} \qquad (\|u\|_{\mathcal{C}^0} = \max_{t\in[a,b]} |u(t)|) \tag{4.2}$$

wird $\mathbb{X}$ zu einem Banachraum und es gilt

$$|D(f_2) - D(f_1)| = \left|\dot{f}_2(\bar{t}) - \dot{f}_1(\bar{t})\right| = \left|\frac{d}{dt}(f_2 - f_1)(\bar{t})\right| \le \left\|\frac{d}{dt}(f_2 - f_1)\right\|_{\mathcal{C}^0} \le \|f_2 - f_1\|_{\mathbb{X}}. \tag{4.3}$$

In diesem Sinn wäre die Differentiation ein gut konditioniertes Problem mit Kondition $\kappa = 1$. Dies würde aber voraussetzen, dass man als Daten auch $\dot{f}$ zur Verfügung hätte. Will man allerdings nur Auswertungen von f verwenden, so muss man

$$\|f\|_{\mathbb{X}} = \|f\|_{\mathcal{C}^0} \tag{4.4}$$

betrachten. In diesem Fall ist $\mathbb{X}$ kein Banachraum. Es stellt sich heraus, dass das entsprechende D jetzt unbeschränkt ist. Dies folgt etwa aus

$$f(t) = \tanh[c(t-\bar{t})] \tag{4.5a}$$

$$\dot{f}(t) = \frac{c}{\cosh^2[c(t-\bar{t})]} \tag{4.5b}$$

da hier

$$\|f\|_{\mathbb{X}} \le 1, \quad \left|\dot{f}(\bar{t})\right| = |c\,| \tag{4.6}$$

ist und $c \in \mathbb{R}$ beliebig gewählt werden kann. In diesem Sinne ist die Differentiation kein wohlgestelltes Problem.

4.2 Differenzenverfahren

Der Raum $\mathbb{X}$ ist nicht endlichdimensional, das heißt er kann (bei Auszeichnung einer Basis) nicht durch endlich viele reelle Zahlen beschrieben werden. Für eine numerische Behandlung muss $\mathbb{X}$ also durch einen endlichdimensionalen Raum $\mathbb{V} \subseteq \mathbb{X}$ ersetzt werden. Man spricht von Diskretisierung. Die damit verbundenen Fehler in der Lösung heißen Diskretisierungsfehler.

Im folgenden wählen wir $\mathbb{V} = \Pi_n$ und ersetzen f durch das an gegebenen, paarweise verschiedenen Stellen t_i, $i = 0, \ldots, n$, interpolierende Polynom p. Ausgehend von

$$p(t) = \sum_{j=0}^{n} f_j l_j(t), \quad l_j = \prod_{\substack{m=0 \\ m \neq j}}^{n} \frac{t - t_m}{t_j - t_m}, \quad f_j = f(t_j) \tag{4.7}$$

erhält man

$$\begin{aligned} \dot{p}(\bar{t}) &= \sum_{j=0}^{n} f_j \dot{l}_j(\bar{t}) = \sum_{j=0}^{n} b_j f_j \\ b_j &= \dot{l}_j(\bar{t}) = \sum_{\substack{l=0 \\ l \neq j}}^{n} \frac{1}{t_j - t_l} \prod_{\substack{m=0 \\ m \neq j,l}}^{n} \frac{\bar{t} - t_m}{t_j - t_m}. \end{aligned} \tag{4.8}$$

Bemerkung 4.1. Per Konstruktion sind die so festgelegten Differenzenverfahren exakt für alle Polynome $p \in \Pi_n$, d.h.

$$\dot{p}(\bar{t}) = \sum_{j=0}^{n} b_j p(t_j). \tag{4.9}$$

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

$$\sum_{j=0}^{n} b_j t_j^i = \frac{d}{dt}(t^i)\Big|_{t=\bar{t}} = i\bar{t}^{\,i-1}, \quad i = 0, \ldots, n. \tag{4.10}$$

Die b_j genügen also einem linearen Gleichungssystem mit einer regulären Vandermonde-Matrix.

Wählt man das Gitter äquidistant, d.h. $t_i = t_0 + ih$, $i = 0, \ldots, n$, $h > 0$, derart, dass $\bar{t} = t_k$ ein Gitterpunkt ist, so ist das Verfahren festgelegt durch die Angabe von $n \in \mathbb{N}$ und $k \in \{0, \ldots, n\}$. Insbesondere gilt:

$$\begin{aligned} b_j &= \sum_{\substack{l=0 \\ l \neq j}}^{n} \frac{1}{h(j-l)} \prod_{\substack{m=0 \\ m \neq j,l}}^{n} \frac{h(k-m)}{h(j-m)} = \frac{1}{h}\sigma_j \\ \sigma_j &= \sum_{\substack{l=0 \\ l \neq j}}^{n} \frac{1}{(j-l)} \prod_{\substack{m=0 \\ m \neq j,l}}^{n} \frac{(k-m)}{(j-m)} \end{aligned} \tag{4.11}$$

mit σ_j, $j = 0, \ldots, n$, unabhängig von t_0 und h.

Beispiel 4.2. Im Fall $n = 1$ und $k = 0$, d.h. $t_0 = \bar{t}$, $t_1 = \bar{t} + h$ erhält man aus

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

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

$$\dot{f}(t) \approx \frac{f(t+h) - f(t)}{h}, \tag{4.12}$$

d.h. man hat gerade die Grenzwertbildung bei der Definition der Ableitung rückgängig gemacht.
Im Fall $n = 2$, $k = 1$, d.h. $t_0 = \bar{t} - h$, $t_1 = \bar{t}$, $t_2 = \bar{t} + h$, erhält man aus

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

die Approximation

$$\dot{f}(t) \approx \frac{f(t+h) - f(t-h)}{2h}. \tag{4.13}$$

Der Diskretisierungsfehler kann mit Hilfe des Interpolationsfehlers (3.18) abgeschätzt werden.

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

$$\left|\dot{f}(\bar{t}) - \dot{p}(\bar{t})\right| \leq Ch^n \tag{4.14}$$

mit $C \geq 0$ unabhängig von h.

Beweis. Die Funktion $f - p$ besitzt die $n+1$ paarweise verschiedenen Nullstellen $t_0, \ldots, t_n \in [t_0, t_n]$. Damit besitzt $g = \dot{f} - \dot{p}$ in $[t_0, t_n]$ mindestens n paarweise verschiedene Nullstellen $\bar{t}_1, \ldots, \bar{t}_n$.
Interpolation von g in diesen Nullstellen liefert das Nullpolynom in Π_{n+1} mit der Fehlerdarstellung

$$g(\bar{t}) = \frac{g^{(n)}(\tau)}{n!} \prod_{i=1}^{n} (\bar{t} - \bar{t}_i)$$

nach Satz 3.9 für $\bar{t} \in [t_0, t_n]$. Wegen $g^{(n)} = f^{(n+1)}$ gilt

$$\left|\dot{f}(t) - \dot{p}(\bar{t})\right| = \left|\frac{f^{(n+1)}(\tau)}{n!} \prod_{i=1}^{n} (\bar{t} - t_i)\right|.$$

Die Äquidistanz des Gitters liefert

$$\left|\prod_{i=1}^{n} (\bar{t} - t_i)\right| \leq \tilde{C}h^n.$$

□

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. Es gibt damit irgendwo ein optimales $h > 0$, für das der Gesamtfehler minimal wird. Dieses h ist jedoch schwierig zu bestimmen. Üblicherweise setzt man für (4.12)

$$h = \sqrt{\text{eps}}\,|f(\bar{t})|\,. \tag{4.15}$$

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 dann durch Auslöschung zugunsten des Diskretisierungsfehlers geopfert.

4.3 Extrapolationsverfahren

Ist $f \in \mathcal{C}^{2N+3}([a,b],\mathbb{R})$, $N \in \mathbb{N}$, so besitzt f die Taylorentwicklung

$$f(t+h) = \sum_{k=0}^{2N+2} \frac{f^{(k)}(t)}{k!} h^k + \frac{f^{(2N+3)}(r)}{(2N+3)!} h^{2N+3} \quad \text{mit } r \in [t, t+h\,] \tag{4.16}$$

für $h > 0$, analog für $h < 0$. Für (4.13) ergibt sich damit

$$\frac{f(t+h) - f(t-h)}{2h} = \sum_{k=0}^{n} \frac{f^{(2k+1)}(t)}{(2k+1)!} h^{2k} + \frac{f^{(2N+3)}(r_1) + f^{(2N+3)}(r_2)}{2(2N+3)!} h^{2N+2}$$

mit $r_1 \in [t, t+h\,]$, $r_2 \in [t-h, t\,]$, beziehungsweise

$$\frac{f(t+h) - f(t-h)}{2h} = p(h^2) + R_{2N+2} \cdot h^{2N+2}, \tag{4.17}$$

wobei $p \in \Pi_n$.
Man beachte, dass $p(0) = \dot{f}(t)$ ist. Die Idee der Extrapolation ist nun, den Rest R_{2N+2} zu vernachlässigen, für Schrittweiten

$$h_0 > h_1 > \cdots > h_N > 0 \tag{4.18}$$

die Werte

$$p_i = \frac{f(t+h_i) - f(t-h_i)}{2h_i}, \quad i = 0, \ldots, N \tag{4.19}$$

zu berechnen und den Wert des Interpolationspolynoms zu den Daten (x_i, p_i), $i = 0, \ldots, N$ mit $x_i = h_i^2$ für $x = 0$ zu bestimmen. Da $0 \notin [x_N, x_0]$, spricht man von Extrapolation nach $h = 0$. Benutzt man für die Auswertung das Verfahren von Aitken-Neville, so erhält (3.7) die Form

$$\begin{aligned} P_{k,l} &= P_{k,l-1} + \frac{x_k}{x_{k+l} - x_k} \left(P_{k,l-1} - P_{k+1,l-1}\right) \\ &= P_{k,l-1} + \frac{h_k^2}{h_{k+l}^2 - h_k^2} \left(P_{k,l-1} - P_{k+1,l-1}\right) \\ &= P_{k,l-1} - \frac{1}{1 - \left(\frac{h_{k+l}}{h_k}\right)^2} \left(P_{k,l-1} - P_{k+1,l-1}\right). \end{aligned} \tag{4.20}$$

Definiert man $g : [0, x_0] \to \mathbb{R}$ durch

$$g(h^2) = \begin{cases} \frac{1}{2}(f(t+h) - f(t-h)) & \text{für } h \neq 0 \\ \dot{f}(t) & \text{für } h = 0, \end{cases} \tag{4.21}$$

so gilt entsprechend (3.18)

$$g(0) - P_{0,N} = \frac{g^{(N+1)}(\xi)}{(N+1)!} (-1)^{N+1} \prod_{i=0}^{N} h_i^2. \tag{4.22}$$

Wird die Schrittweise sukzessive halbiert, d.h. ist

$$h_i = \frac{h}{2^i}, \quad i = 0, \ldots, N, \tag{4.23}$$

so wird (4.20) zu

$$P_{k,l} = P_{k,l-1} - \frac{4^l}{4^l - 1} \left(P_{k,l-1} - P_{k+1,l-1}\right) \tag{4.24}$$

und (4.22) impliziert (für $\bar{t} = t$)

$$\left| \dot{f}(\bar{t}) - P_{0,N} \right| \leq C h^{2N+2}. \tag{4.25}$$

4.4 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 eine 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, sogenanntes symbolisches Differenzieren, vergleiche das Arbeitsblatt.

In einer Programmiersprache, in der man wie 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, sogenanntes automatisches Differenzieren.

5 Integration

5.1 Newton-Cotes-Formeln

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

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

d.h. wir betrachten die Abbildung $I : \mathbb{X} \to \mathbb{R}$, $f \mapsto \int_a^b f(t)\,dt$. Für diese gilt

$$|I(f_2) - I(f_1)| \le \int_a^b |f_2(t) - f_1(t)|\,dt \le (b-a)\|f_2 - f_1\|_{\mathbb{X}}, \tag{5.2}$$

d.h. die Kondition des absoluten Fehlers wird durch $\kappa = b - a$ beschrieben. Ausgehend von

$$a \le t_0 < t_1 < \cdots < t_n \le b \quad \text{und } (t_i, f_i), \quad f_i = f(t_i), \quad i = 0,\dots,n \tag{5.3}$$

erhält man ein Interpolationspolynom $p \in \Pi_n$. Integration liefert

$$\int_a^b p(t)\,dt = \int_a^b \sum_{j=0}^n f_j l_j(t)\,dt = \sum_{j=0}^n f_i \int_a^b l_j(t)\,dt = \sum_{j=0}^n b_j f_j \tag{5.4}$$

mit sogenannten Gewichten

$$b_j = \int_a^b l_j(t)\,dt. \tag{5.5}$$

Damit hat man eine Näherungsformel der Form

$$I(f) \approx \sum_{j=0}^n b_j f(t_j) \tag{5.6}$$

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

Bemerkung 5.1. Nach Konstruktion sind die Quadraturformeln (5.6) mit (5.5) exakt für alle $p \in \Pi_n$, d.h.

$$I(p) = \sum_{j=0}^n b_j p(t_j) \quad \forall p \in \Pi_n. \tag{5.7}$$

Wählt man die Monom-Basis, so gilt

$$\sum_{j=0}^n b_j t_j^i = \frac{1}{1+i}(b^{i+1} - a^{i+1}), \quad i = 0,\dots,n, \tag{5.8}$$

das heißt die b_j genügen einem linearen Gleichungssystem mit regulärer Vandermonde-Matrix.

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

Nach Bemerkung 5.1 haben die Quadraturformeln (5.6) mit (5.5) die Ordnung $n+1$.

Lemma 5.3. Sei $f \in \mathcal{C}^{n+1}([a,b],\mathbb{R})$. Dann gilt für (5.6) mit (5.5)

$$|I(f) - I(p)| \le CH^{n+2}, \tag{5.9}$$

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

Beweis. Mit (3.18) erhält man

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

□

Sei das Gitter im folgenden äquidistant gemäß

$$t_i = a + ih, \quad i = 0, \ldots, n, \quad h = \frac{b-a}{n}. \tag{5.10}$$

Dann gilt mit $t = a + sh$

$$b_j = \int_a^b \prod_{\substack{k=0 \\ k\neq j}}^{n} \frac{t - t_k}{t_j - t_k}\, dt = h \int_0^n \prod_{\substack{k=0 \\ k\neq j}}^{n} \frac{s-k}{j-k}\, ds = h\sigma_j. \tag{5.11}$$

Die so erhaltenen Quadraturformeln heißen Newton/Cotes-Formeln. Diese sind symmetrisch im folgenden Sinn, siehe Übung.

Definition 5.4. Eine Quadraturformel (5.6) heißt symmetrisch, wenn gilt

$$t_{n-j} = (a+b) - t_j, \quad b_{n-j} = b_j, \quad j = 0, \ldots, n. \tag{5.12}$$

Lemma 5.5. Gegeben sei eine symmetrische Quadraturformel (5.6) mit mindestens ungerader Ordnung $m+1$. Dann hat sie 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 bezüglich 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. □

Damit ist gezeigt, dass symmetrische Quadraturformen immer eine gerade Ordnung besitzen.

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

$$\sigma_0 = \int_0^1 \frac{s-1}{0-1}\, ds = \frac{1}{2}$$
$$\sigma_1 = \int_0^1 \frac{s-0}{1-0}\, ds = \frac{1}{2},$$

und man erhält die sogenannte Trapezregel

$$I(f) \approx \frac{b-a}{2}(f(a) + f(b)) \tag{5.13}$$

mit der Ordnung 2. Für $n = 2$ gilt

$$\sigma_0 = \int_0^2 \frac{s-1}{0-1}\frac{s-2}{0-2}\,ds = \frac{1}{3}$$
$$\sigma_1 = \int_0^2 \frac{s-0}{1-0}\frac{s-2}{1-2}\,ds = \frac{4}{3}$$
$$\sigma_2 = \int_0^2 \frac{s-0}{2-0}\frac{s-1}{2-1}\,ds = \frac{1}{3},$$

und man erhält die sogenannte Simpson-Regel

$$I(f) \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) \tag{5.14}$$

mit der Ordnung 4, vergleiche Lemma 5.5.

Leider treten ab $n = 7$ negative Gewichte b_j auf, was zu Auslöschungseffekten bei der Berechnung von $I(f)$ führt.

Bemerkung 5.7. Sei $f \in \mathcal{C}^2([a,b],\mathbb{R})$. Die Trapezregel

$$T(f) = \frac{b-a}{2}(f(a) + f(b)) \tag{5.15}$$

besitzt dann einen Diskretisierungsfehler der Form

$$I(f) - T(f) = -\frac{H^3}{12}\ddot{f}(\tau), \quad H = b-a, \quad \tau \in [a,b]. \tag{5.16}$$

Beweis. Sei p die zugehörige lineare Interpolierende. Nach Satz 3.9 zusammen mit (3.21) gilt für jedes $t \in [a,b]$

$$f(t) - p(t) = f[a,b,t](t-a)(t-b)$$

und damit

$$I(f) - I(p) = \int_a^b f[a,b,t](t-a)(t-b)\,dt.$$

Da $(t-a)(t-b) \le 0$ auf $[a,b]$ ist, folgt (Mittelwertsatz Integralrechnung)

$$I(f) - I(p) = f[a,b,\bar{\tau}]\int_a^b (t-a)(t-b)\,dt = \frac{\ddot{f}(\tau)}{2}\frac{(a-b)^3}{6} = -\frac{\ddot{f}(\tau)}{12}(b-a)^3$$

mit $\tau, \bar{\tau} \in [a,b]$. □

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

$$t_k = a + kH, \quad k = 0,\ldots,m, \quad H = \frac{b-a}{m}, \tag{5.17}$$

so erhält man die zugehörigen summierten Newton/Cotes-Formeln, indem man auf jedem Teilintervall $[t_k, t_{k+1}]$ den Integranden f durch $p_k \in \Pi_n$ an den Stellen

$$t_{ik} = t_k + ih, \quad i = 0,\ldots,n, \quad h = \frac{H}{n} = \frac{b-a}{m\cdot n} \tag{5.18}$$

interpoliert. Bezeichnet I_k die Integration über $[t_k, t_{k+1}]$, so haben die summierten Newton/Cotes-Formeln die Form

$$I(f) = \sum_{k=0}^{m-1} I_k(p_k). \tag{5.19}$$

Satz 5.8. *Sei $f \in \mathcal{C}^{n+1}([a,b],\mathbb{R})$. Dann gilt für die summierten Newton/Cotes-Formeln:*

$$\left| I(f) - \sum_{k=0}^{m-1} I_k(p_k) \right| \leq Ch^{n+1} \tag{5.20}$$

mit $C \geq 0$ unabhängig von h.

Beweis. Nach Satz 3.9 gilt

$$\mathbb{I}(f) - \sum_{k=0}^{m-1} I_k(p_k) = \sum_{k=0}^{m-1} \int_{t_k}^{t_{k+1}} (f(t) - p_k(t))\, dt = \sum_{k=0}^{m-1} \int_{t_k}^{t_{k+1}} \frac{f^{(n+1)}(\tau(t))}{(n+1)!} \omega_k(t)\, dt$$

mit $\tau(t) \in [t_k, t_{k+1}]$ für $t \in [t_k, t_{k+1}]$ und $\omega_k(t) = (t - t_{0k}) \cdot \cdots \cdot (t - t_{nk})$. Damit folgt

$$\left| I(f) - \sum_{k=0}^{m-1} I_k(p_k) \right| \leq \sum_{k=0}^{m-1} (t_{k+1} - t_k) \frac{\|f^{(n+1)}\|_{\mathcal{C}^0}}{(n+1)!} (n+1)! h^{n+1} = Ch^{n+1}.$$

□

5.2 Extrapolation

Im folgenden bezeichne $T(h)$ das Ergebnis der summierten Trapezregel, angewendet auf eine vorgegebene Funktion f. Man beachte, dass hier $h = H$ wegen $n = 1$ gilt.

Satz 5.9 (Euler/Maclaurinsche Summenformel)**.** *Sei $f \in \mathcal{C}^{2N+1}([a,b],\mathbb{R})$ und $h = \frac{b-a}{m}$ mit $m \in \mathbb{N}$. Dann gilt für die summierte Trapezregel*

$$T(h) = p(h^2) + R_{2N+2} \cdot h^{2N+2} \tag{5.21}$$

mit $p \in \Pi - N$, $p(0) = I(f)$ und R_{2N+2} unabhängig von h.

Beweis. Siehe Literatur. □

Wir haben also dieselbe Situation wie in (4.17) bei der Differentiation. Um wie dort ein Extrapolationsverfahren zu konstruieren, wählen wir

$$n_0 < n_1 < \cdots < n_N, \quad n_i \in \mathbb{N}, \quad i = 0, \ldots, N \tag{5.22}$$

und setzen entsprechend der Notation in Abschnitt 4.3

$$h_i = \frac{b-a}{n_i}, \quad i = 0, \ldots, N \tag{5.23}$$

sowie $x_i = h_i^2$ und

$$p_i = T(h_i). \tag{5.24}$$

Die für (5.22) klassisch verwendete Folge ist gegeben durch

$$n_i = 2^i. \tag{5.25}$$

Man nennt das Extrapolationsverfahren in diesem Fall Romberg-Quadratur. Wegen

$$\begin{aligned} T(\frac{h}{2}) &= \frac{h}{4} \left(f(a) + 2 \sum_{i=1}^{2m-1} f\left(a + \frac{ih}{2}\right) + f(b) \right) \\ &= \frac{h}{4} \left(f(a) + 2 \sum_{k=1}^{m-1} f(a+kh) + f(b) \right) + \frac{h}{2} \sum_{k=1}^{m} f\left(a + (2k-1)\frac{h}{2}\right) \\ &= \frac{1}{2} T(h) + \frac{h}{2} \sum_{k=1}^{m} f\left(a + (2k-1)\frac{h}{2}\right) \end{aligned} \tag{5.26}$$

kann man die p_i hier rekursiv berechnen. Für $f \in \Pi_{2N}$ gilt $R_{2N+2} = 0$ und damit $T(h) = p(h^2)$ beziehungsweise $I(f) = p(0) = P_{0,N}$, das heißt die Ordnung ist mindestens $2N + 2$ (Symmetrie). Man beachte, dass man wiederum $[a, b]$ in kleinere (nicht unbedingt gleich große) Intervalle unterteilen kann und jedes Teilintervall mit Extrapolation (bei nicht unbedingt gleichem N) behandeln kann.

5.3 Gauß-Quadratur

Zu berechnen sei etwas allgemeiner

$$I(f) = \int_a^b w(t) f(t)\, dt \tag{5.27}$$

mit fester Gewichtsfunktion $w \in \mathcal{C}([a, b], \mathbb{R}_0^+)$, wobei

$$\int_a^b w(t)\, dt > 0 \tag{5.28}$$

mit $w(t) = 0$ nur für abzählbar viele $t \in [a, b)$. Entsprechend (5.6) approximieren wir $I(f)$ durch

$$\tilde{I}(f) = \sum_{j=0}^{n} b_j f(t_j) \tag{5.29}$$

mit paarweise verschiedenen $t_j \in [a, b]$. Bei fest vorgegebenen t_j, $j = 0, \ldots, n$ gibt es nach Bemerkung 5.1 eindeutig bestimmte b_j, $j = 0, \ldots, n$, so dass

$$I(p) = \tilde{I}(p) \quad \forall p \in \Pi_n. \tag{5.30}$$

Es ändert sich nur die rechte Seite. Die Frage ist nun, ob man nicht die Ordnung in (5.29) durch geschickte Wahl von t_j erhöhen kann.

Lemma 5.10. Die Ordnung einer Quadraturformel (5.29) ist höchstens $2n + 2$.

Beweis. Sei $p \in \Pi_{2n+2}$ mit

$$p(t) = \prod_{j=0}^{n} (t - t_j)^2.$$

Dann gilt $\tilde{I}(p) = 0$, aber $I(p) > 0$. □

Im folgenden soll eine Quadraturformel hergeleitet werden, deren Ordnung gerade die eben bewiesene obere Schranke ist. Dazu definiert man

$$(f, g) = \int_a^b w(t) f(t) g(t)\, dt. \tag{5.31}$$

Weiter sei

$$\tilde{\Pi}_n = \{p \in \Pi_n | p(t) = t^n + a_{n-1} t^{n-1} + \cdots + a_0\} \tag{5.32}$$

die Menge der normierten Polynome vom Grad n.

Satz 5.11 (Orthogonalpolynome)**.** *Es gibt eindeutig bestimmte Polynome $p_i \in \tilde{\Pi}_i$, $i \in \mathbb{N}_0$ mit*

$$(p_j, p_k) = 0 \quad \textit{für } j \neq k. \tag{5.33}$$

Diese genügen der Rekursionsformel

$$p_{i+1}(t) = (t - d_{i+1}) p_i(t) - c_{i+1}^2 p_{i-1}(t), \quad i \in \mathbb{N}^+ \tag{5.34a}$$

$$p_0(t) = 1, \quad p_1(t) = t - d_1 \tag{5.34b}$$

mit

$$d_{i+1} = (t p_i, p_i)/(p_i, p_i) \tag{5.35a}$$

$$c_{i+1}^2 = (p_i, p_i)/(p_{i-1}, p_{i-1}). \tag{5.35b}$$

Beweis. Offensichtlich ist $p_0(t) = 1$. Für $p_1 \in \tilde{\Pi}_1$ muss gelten

$$(p_1, p_0) = 0, \quad p_1(t) = t - d_1.$$

Einsetzen ergibt

$$0 = (tp_0 - d_1 p_0, p_0) = (tp_0, p_0) - d_1(p_0, p_0)$$

und damit

$$d_1 = (tp_0, p_0)/(p_0, p_0).$$

Sei nun $i \in \mathbb{N}^+$. Wir nehmen an, dass die $p_j \in \tilde{\Pi}_j$, $j = 0, \ldots, i$, die obigen Bedingungen erfüllen. Jedes $p_{i+1} \in \tilde{\Pi}_{i+1}$ kann in der Form

$$p_{i+1}(t) = (t - d_{i+1})p_i(t) - \tilde{c}_{i-1}p_{i-1}(t) - \cdots - \tilde{c}_0 p_0(t)$$

mit eindeutig bestimmten Koeffizienten $d_{i+1}, \tilde{c}_{i-1}, \ldots, \tilde{c}_0$ geschrieben werden. Wegen (5.33) für $j, k \leq i$ ist

$$0 = (p_{i+1}, p_i) = (tp_i - d_{i+1}p_i, p_i) = (tp_i, p_i) - d_{i+1}(p_i, p_i)$$

und für $j \leq i - 1$

$$0 = (p_{i+1}, p_j) = (tp_i - \tilde{c}_j p_j, p_j) = (p_i, tp_j) - \tilde{c}_j(p_j, p_j)$$

zu erfüllen. Die erste Bedingung liefert die Festlegung von d_{i+1} nach (5.35a). In die zweite Bedingung setzen wir die Rekursion (5.34a) in der Form

$$tp_j(t) = p_{j+1}(t) + d_{j+1}p_j(t) + c_{j+1}^2 p_{j-1}(t)$$

ein. Es ergibt sich

$$0 = (p_i, p_{j+1}) - \tilde{c}_j(p_j, p_j)$$

beziehungsweise

$$\tilde{c}_j = \frac{(p_i, p_{j+1})}{(p_j, p_j)} = \begin{cases} c_{i+1}^2 & \text{für } j = i - 1 \\ 0 & \text{sonst} \end{cases}$$

und es gilt (5.34a). □

Die p_i, $i = 0, \ldots, n$ bilden eine Orthogonalbasis von Π_n. Insbesondere gilt wegen (5.33)

$$(p, p_{n+1}) = 0 \quad \text{für alle } p \in \Pi_n. \tag{5.36}$$

Satz 5.12. *Die Nullstellen t_j, $j = 0, \ldots, n$ von p_{n+1} sind reell und einfach und liegen in (a, b).*

Beweis. Seien $t_0, \ldots, t_l$ die Nullstellen von p_{n+1} in (a, b), an denen p_{n+1} das Vorzeichen wechselt und sei $l < n$ angenommen, das heißt es gebe nicht in (a, b) liegende oder mehrfache Nullstellen. Setzt man

$$q(t) = \prod_{j=0}^{l} (t - t_j),$$

so gilt $q \in \tilde{\Pi}_{l+1}$, so dass $(q, p_{n+1}) = 0$. Da aber $qp_{n+1} \neq 0$ stetig ist und in (a, b) keinen Vorzeichenwechsel hat, ist per Konstruktion

$$\int_a^b w(t)p_{n+1}(t)q(t)\,dt \neq 0.$$

□

Satz 5.13. *Für beliebige, paarweise verschiedene t_j, $j = 0, \ldots, n$ ist die Matrix*

$$\mathbf{A} = \begin{pmatrix} p_0(t_0) & \ldots & p_0(t_n) \\ \vdots & & \vdots \\ p_n(t_0) & \ldots & p_n(t_n) \end{pmatrix} \tag{5.37}$$

regulär.

Beweis. Wäre $\mathbf{A}$ singulär, so gäbe es ein $v = (v_0, \ldots, v_n)^{\mathrm{T}} \in \mathbb{R}^{n+1}$ mit $v \neq 0$ und $v^{\mathrm{T}}\mathbf{A} = 0$. Definiert man $q \in \Pi_n$ durch

$$q(t) = \sum_{k=0}^{n} v_k p_k(t),$$

so besagt $v^{\mathrm{T}}\mathbf{A} = 0$, dass $q(t_j) = 0$ ist für $j = 0, \ldots, n$, das heißt dass $q = 0$ ist. Da aber p_k, $k = 0, \ldots, n$ eine Basis von Π_n bilden, folgt $v = 0$, Widerspruch. □

Satz 5.14. *Seien die t_j, $j = 0, \ldots, n$, die Nullstellen von p_{n+1} und b_j, $j = 0, \ldots, n$ die Lösung des linearen Gleichungssystems*

$$\tilde{I}(p_k) = \sum_{j=0}^{n} p_k(t_j) b_j = \begin{cases} (p_0, p_0) & \text{für } k = 0 \\ 0 & \text{sonst} \end{cases}, \tag{5.38}$$

so gilt

$$I(p) = \tilde{I}(p) \quad \text{für alle } p \in \Pi_{2n+1}. \tag{5.39}$$

Beweis. Nach Satz 5.13 sind die b_j durch (5.38) eindeutig bestimmt. Sei $p \in \Pi_{2n+1}$. Dann lässt sich p schreiben als

$$p = q p_{n+1} + r$$

mit $q, r \in \Pi_n$. Mit $r = a_n p_n + \cdots + a_0 p_0$ gilt dann

$$I(p) = \int_a^b w(t) p(t)\, dt = \int_a^b w(t) q(t) p_{n+1}(t)\, dt + \int_a^b w(t) r(t)\, dt = (q, p_{n+1}) + (r, p_0) = a_0 (p_0, p_0)$$

und

$$\tilde{I}(p) = \sum_{j=0}^{n} b_j p(t_j) = \sum_{j=0}^{n} b_j r(t_j) = \sum_{j=0}^{n} b_j \sum_{k=0}^{n} a_k p_k(t_j) = \sum_{k=0}^{n} a_k \sum_{j=0}^{n} p_k(t_j) b_j = a_0 (p_0, p_0).$$

□

Satz 5.15. *Gegeben seien b_j, t_j, $j = 0, \ldots, n$, derart, dass* (5.38) *und* (5.39) *gilt. Außerdem seien alle t_j paarweise verschieden. Dann ist*

$$b_j > 0, \quad j = 0, \ldots, n. \tag{5.40}$$

Beweis. Definiert man $\tilde{p}_i \in \Pi_{2n}$, $i = 0, \ldots, n$, durch

$$\tilde{p}_i(t) = \prod_{\substack{k=0 \\ k \neq i}}^{n} (t - t_k)^2,$$

so folgt aus $I(\tilde{p}_i) = \tilde{I}(\tilde{p}_i)$ sofort

$$0 < \int_a^b w(t) \tilde{p}_i(t)\, dt = \sum_{j=0}^{n} b_j \tilde{p}_i(t_j) = b_i \tilde{p}_i(t_i).$$

□

Satz 5.16. *Gegeben seien b_j, t_j, $j = 0, \ldots, n$, derart, dass* (5.39) *gilt. Dann sind die t_j die Nullstellen von p_{n+1} und die b_j erfüllen* (5.38).

Beweis. Sind die t_j nicht alle paarweise verschieden, so kann man auf eine maximale Zahl von paarweise verschiedenen Knoten reduzieren. Seien dies o.E. $t_0, \ldots, t_l$ mit $l < n$. Dann wird aber $p \in \Pi_{2l+2} \subseteq \Pi_{2n+1}$ mit

$$p(t) = \prod_{j=0}^{l} (t - t_j)^2$$

nicht exakt integriert. Also sind alle t_j, $j = 0, \ldots, n$ paarweise verschieden. Wegen $I(p_k) = \tilde{I}(p_k)$, $k = 0, \ldots, n$, gilt

$$\sum_{j=0}^{n} b_j p_k(t_j) = \int_a^b p_k(t) w(t)\, dt = (p_k, p_0) = \begin{cases} (p_0, p_0) & \text{für } k = 0 \\ 0 & \text{sonst} \end{cases},$$

das heißt die b_j erfüllen (5.38). Nach Satz 5.15 gilt $b_j > 0$. wählt man $p =_k p_{n+1}$, $k \in \{0, \ldots, n\}$, so gilt $p \in \Pi_{2n+1}$ und damit

$$0 = (p_k, p_{n+1}) = I(p) = \sum_{j=0}^{n} b_j p_k(t_j) p_{n+1}(t_j),$$

das heißt der Vektor $u = (b_j p_{n+1}(t_j))$ erfüllt $\mathbf{A}u = 0$ mit $\mathbf{A}$ nach (5.37). Da aber die t_j paarweise verschieden sind, ist $\mathbf{A}$ regulär und damit $u = 0$. □

Die auf den so hergeleiteten Verfahren beruhende numerische Integration heißt Gauß-Quadratur. Im Fall $w(t) = 1$ hat man im Vergleich zu den Newton/Cotes-Formeln eine wesentlich höhere Ordnung bei gleicher Anzahl von Funktionsauswertungen. Außerdem sind die Gewichte b_j immer positiv. Die zu $w(t) = 1$ und $a = -1, b = 1$ gehörenden Orthogonalpolynome heißen Legendre-Polynome. Schließlich kann man auch Quadraturformeln für Integranden mit sogenannten schwachen Singularitäten herleiten, z.B. für die Wahl

$$\begin{array}{lll} w(t) = \sqrt{1-t^2}, & a = -1, b = 1 & \\ w(t) = (1-t^2)^{-\frac{1}{2}}, & a = -1, b = 1 & \text{(Tschebyscheff-Polynome)} \\ w(t) = e^{-t}, & a = 0, b = \infty & \text{(Laguerre-Polynome)} \\ w(t) = e^{-t^2}, & a = -\infty, b = \infty & \text{(Hermite-Polynome)} \end{array} \tag{5.41}$$

5.4 Gitteranpassung

Um die Genauigkeit einer Quadraturformel zu erhöhen, kann man die Intervalle verkleinern, auf die man sie anwendet. Für die Effizienz (Anzahl der Funktionsauswertungen) sollte die Länge der Intervalle auf dem jeweiligen Funktionsverlauf angepasst werden. Man spricht von Gitteranpassung oder auch von Schrittweitensteuerung. Zwei Methoden sollen hier vorgestellt werden.

Die erste Methode geht davon aus, dass man die Integration von a bis zu einem Punkt $t_k \in (a, b]$ bereits durchgeführt hat. Man möchte jetzt t_{k+1} in irgendeiner Weise passend wählen. Ausgehend von

$$a = t_0 < t_1 < \cdots < t_k < t_{k+1} \leq b \tag{5.42}$$

erhält man

$$t_{k+1} = t_k + h_k \tag{5.43}$$

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

$$I_k(f) = \int_{t_k}^{t_{k+1}} f(t)\, dt. \tag{5.44}$$

Es muss dabei geklärt werden, ob h_k eine sinnvolle Schrittweite ist. Dazu bezeichne $\tilde{I}_k$ das Ergebnis einer Quadraturformel der Ordnung q und $\hat{I}_k$ das Ergebnis einer Quadraturformel mit einer Ordnung, die höher als q ist, jeweils angewendet auf das Intervall $[t_k, t_{k+1}]$. In Anlehnung an (5.9) macht man die Annahme

$$\left| I_k(f) - \tilde{I}_k(f) \right| \approx C h_k^{q+1} \tag{5.45}$$

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

$$\hat{I}_k(f) \approx I_k(f) \tag{5.46}$$

gerechtfertigt ist. Wegen

$$\texttt{ERR} = \left| \hat{I}_k(f) - \tilde{I}_k(f) \right| \approx \left| I_k(f) - \tilde{I}_k(f) \right| \tag{5.47}$$

macht man dann die Modellannahme

$$\texttt{ERR} = Ch_k^{q+1}. \tag{5.48}$$

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

$$\texttt{ERR} < \texttt{TOL} \tag{5.49}$$

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

$$\texttt{TOL} = C\bar{h}^{\,q+1}. \tag{5.50}$$

Elimination von C liefert

$$\frac{\texttt{ERR}}{\texttt{TOL}} = \left(\frac{h_k}{\bar{h}}\right)^{q+1} \tag{5.51}$$

beziehungsweise

$$\bar{h} = h_k \sqrt[q+1]{\frac{\texttt{TOL}}{\texttt{ERR}}}. \tag{5.52}$$

Dabei ist (5.49) äquivalent zu

$$\bar{h} \geq h_k. \tag{5.53}$$

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, $\tilde{I}$ oder $\hat{I}$ 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 (5.52) typischerweise in etwas konservativerer Form

$$\bar{h} = \texttt{RED} \cdot h_k \sqrt[q+1]{\frac{\texttt{TOL}}{\texttt{ERR} + \texttt{SAV}}} \tag{5.54}$$

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

$$\texttt{RMIN} \leq \frac{\bar{h}}{h_k} \leq \texttt{RMAX}. \tag{5.55}$$

Die Wahl der Konstanten RED, SAV, sowie RMIN, RMAX ist von philosophischer Natur. Eine mögliche Wahl wäre

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

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

Algorithmus 5.17. *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 h entsprechend (5.54) und (5.55)
Ist $\texttt{ERR} > \texttt{TOL}$, so setze $h_k = \bar{h}$ und beginne Schleife
Setze $h_{k+1} = \min\{\bar{h}, b - t_{k+1}\}$, $I = I + \tilde{I}_k(f)$
Ist $t_{k+1} = b$, so akzeptiere I als Approximation an $I(f) \Rightarrow$ Stop
$k = k + 1$

(5.56)

Man beachte, dass dieser Algorithmus eine Richtung (die von a nach b) auszeichnet.

Die zweite Methode geht davon aus, dass man eine Unterteilung des gesamten Intervalls $[a, b]$ etwa gemäß

$$a = t_0 < t_1 < \cdots < t_N = b \tag{5.57}$$

vorliegen hat. Bezeichnen $T_k(f)$ und $S_k(f)$ die zu $[t_k, t_{k+1}]$ gehörigen mit Trapezregel beziehungsweise Simpson-Regel erzielten Approximationen an $I_k(f)$, sowie

$$T(f) = \sum_{k=0}^{N-1} T_k(f), \qquad S(f) = \sum_{k=0}^{N-1} S_k(f), \tag{5.58}$$

so trifft man hier die Annahme

$$S_k(f) \approx I_k(f) \tag{5.59}$$

und

$$\varepsilon_k(f) = |S_k(f) - T_k(f)| \approx |I_k(f) - T_k(f)|\,. \tag{5.60}$$

Wegen (5.16) macht man dann die Modellannahme

$$\varepsilon_k(f) = Ch_k^3, \quad h_k = t_{k+1} - t_k. \tag{5.61}$$

Ziel ist, durch Verfeinerung von (5.57) zu ereichen, dass im neuen Gitter $\varepsilon_k(f)$ weitgehend unabhängig von k ist. Als Maßnahme zur Verfeinerung verwenden wir Halbierung von Teilintervallen.
Durch Halbieren von $[t_k, t_{k+1}]$ hat man statt $\varepsilon_k(f)$ für die Teilintervalle $[t_k, t_k + \frac{1}{2}h_k]$ bzw. $[t_k + \frac{1}{2}h_k, t_{k+1}]$ Fehlerabschätzungen $\varepsilon_{k,1}$ und $\varepsilon_{k,2}$ entsprechend (5.60). Für diese gilt

$$\varepsilon_{k,i} \approx \frac{1}{8}\varepsilon_k(f), \quad i = 1, 2. \tag{5.62}$$

Man unterteilt jetzt alle Intervalle, für die

$$\varepsilon_k(f) \geq \frac{1}{8} \max_{l=0,\ldots,N-1} \varepsilon_l(f) \tag{5.63}$$

gilt, und verfährt so weiter, bis der geschätzte Gesamtfehler

$$\varepsilon(f) = \sum_{k=0}^{N-1} \varepsilon_k(f) \tag{5.64}$$

eine vorgegebene Toleranz unterschreitet.

Algorithmus 5.18. *Gegeben sei eine Unterteilung* (5.57)*, etwa* $N = 2$ *und* $t_1 = \frac{a+b}{2}$*, sowie eine Toleranz* `TOL` > 0*.*

Schleife über erlaubte Anzahl von Versuchen
Berechne $T_k(f)$*,* $S_k(f)$ *und* $\varepsilon_k(f)$*,* $k = 0, \ldots, N-1$
Berechne $S(f)$ *und* $\varepsilon(f)$
Ist $\varepsilon(f) \leq$ `TOL`*, so akzeptiere* $S(f)$ *als Approximation an* $I(f) \Rightarrow$ *Stop*
Verfeinere Unterteilung durch Halbieren aller Intervalle, für die (5.63) *gilt und berechne neues Gitter wieder mit* (5.57)

(5.65)

6 Nichtlineare Gleichungssysteme

Gesucht ist eine Lösung $x^\star$ der Gleichung

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

wobei $f : \mathbb{D} \to \mathbb{R}^n$, $\mathbb{D} \subseteq \mathbb{R}^n$ offen, stetig sei. Ist f stetig differenzierbar und $f'(x^\star)$ reguläre Matrix, so besagt der Satz über die Umkehrfunktion, dass f in einer Umgebung $\mathbb{V}$ von $y^\star = 0$ die Umkehrfunktion f^{-1} besitzt. Inbesondere ist $x^\star = f^{-1}(y^\star)$ 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:

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

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.

$$f(x^\star) = 0 \Leftrightarrow x^\star = \varphi(x^\star). \tag{6.4}$$

Man nennt $x^\star$ mit $x^\star = \varphi(x^\star)$ 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, in der Hoffnung, dass dadurch eine Folge $(x_\nu)_{\nu\in\mathbb{N}_0}$ definiert wird mit $x_\nu \to x^\star$. 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 $x^\star$, die bei zehnstelliger Rechnung gegeben ist durch

$$x^\star = -0.56714\ 32904.$$

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.

Man beachte, dass selbst im Fall $x_\nu \to x^\star$ auf dem Rechner die Iteration irgendwann abgebrochen werden muss und das letzte berechnete x_ν als Ersatz für $x^\star$ akzeptiert werden muss. Der zugehörige Fehler $x^\star - x_\nu$ 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*

$$\|\varphi(x_2) - \varphi(x_1)\| \le L\|x_2 - x_1\| \tag{6.6}$$

für alle $x_1, x_2 \in \mathbb{D}$ 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 $x^\star$ von φ konvergiert. Insbesondere gilt

$$\|x^\star - x_{\nu+1}\| \le L\|x^\star - x_\nu\| \tag{6.7a}$$

$$\|x^\star - x_\nu\| \le \frac{L^\nu}{1-L}\|x_1 - x_0\| \tag{6.7b}$$

$$\|x^\star - x_{\nu+1}\| \le \frac{L}{1-L}\|x_{\nu+1} - x_\nu\| \tag{6.7c}$$

und man spricht von linearer Konvergenz.

Beweis. Siehe gängige Lehrbücher zur Analysis. □

Die Voraussetzungen des Banachschen Fixpunktsatzes sind oft schwierig nachzuweisen. Der folgende Satz gibt hinreichende Bedingungen dafür an, dass der Banachsche Fixpunktsatz zumindest in einer hinreichend kleinen Umgebung eines gegebenen Fixpunktes gilt.

Satz 6.3. *Sei $x^\star \in \mathbb{D}$ ein Fixpunkt von $\varphi \in \mathcal{C}^1(\mathbb{D}, \mathbb{R}^n)$ mit $\mathbb{D} \subseteq \mathbb{R}^n$ offen. Außerdem sei*

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

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

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

$$\|\varphi'(x)\| \leq 1 - \frac{\delta}{2} \quad \text{für alle } x \in \mathbb{U} = \overline{K_\varepsilon(x^\star)}.$$

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

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

das heißt $\varphi|_{\mathbb{U}}$ ist kontraktiv mit $L = 1 - \frac{\delta}{2}$. Wählt man speziell $x_1 = x^\star$ und $x_2 = x \in \mathbb{U}$, so liefert die obige Ungleichung

$$\|\varphi(x) - x^\star\| = \|\varphi(x) - \varphi(x^\star)\| \leq \left(1 - \frac{\delta}{2}\right) \|x - x^\star\| \leq \|x - x^\star\| \leq \varepsilon,$$

d.h. $\varphi(x) \in \mathbb{U}\ \forall x \in \mathbb{U}$ bzw. $\varphi\big|_{\mathbb{U}} : \mathbb{U} \to \mathbb{U}$. □

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

6.2 Intervallmethoden

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

Algorithmus 6.4 (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 $f \Rightarrow$ Stop
Ist $f(a_k)f(c_k) \leq 0$, so setze $a_{k+1} = a_k$, $b_{k+1} = c_k$, ansonsten setze $a_{k+1} = c_k$, $b_{k+1} = b_k$
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 von f und man könnte sofort abbrechen.

Satz 6.5. *Ist $f : [a, b] \to \mathbb{R}$ stetig und $f(a) \cdot f(b) \leq 0$, so definiert Algorithmus 6.4 bei Weglassen der Abbrechbedingung eine Intervallschachtelung $[a_k, b_k]$, $k \in \mathbb{N}_0$, mit*

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

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

$$|x^\star - c_k| \leq \left(\frac{1}{2}\right)^{k+1} (b-a) \tag{6.11}$$

in linearer Konvergenz.

□

Bemerkung 6.6 (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 6.4 ergebende Methode heißt *regula falsi.* Außerdem muss das Abbruchkriterium geändert werden, etwa in

$$c_k - a_k \leq \texttt{TOL} \quad \text{oder} \quad b_k - c_k \leq \texttt{TOL},$$

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

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 Approximationen, so erhält man die iterative Methode

$$x_{\nu+1} = \frac{x_{\nu-1}f(x_\nu) - x_\nu f(x_{\nu-1})}{f(x_\nu) - f(x_{\nu-1})}, \tag{6.14}$$

das sogenannte Sekantenverfahren. Während man im letzten Abschnitt immer mit Intervallen arbeitete, die eine Nullstelle $x^\star$ enthalten, 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 $\nu \in \mathbb{N}_0$ gilt.

Satz 6.7. *Sei $x^\star$ eine Nullstelle von $f \in \mathcal{C}^2(\mathbb{D}, \mathbb{R})$, $\mathbb{D} \subseteq \mathbb{R}$ offen mit $f'(x^\star) \neq 0$. 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 $x_\nu \to x^\star$. Inbesondere gilt*

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

mit $C \geq 0$, $K \in (0,1)$ und $q = \frac{1}{2}(1+\sqrt{5}) \approx 1,618\ldots$.

Beweis. Aus (6.14) bzw. (6.13) ergibt sich

$$\begin{aligned} x^\star - x_{\nu+1} &= x^\star - x_\nu + f(x_\nu)\frac{x_\nu - x_{\nu-1}}{f(x_\nu) - f(x_{\nu-1})} = (x^\star - x_\nu) + \frac{f(x_\nu) - f(x^\star)}{f[x_{\nu-1}, x_\nu]} \\ &= (x^\star - x_\nu)\left(1 - \frac{f[x_\nu, x^\star]}{f[x_{\nu-1}, x_\nu]}\right) = -(x^\star - x_\nu)(x^\star - x_{\nu-1})\frac{f[x_{\nu-1}, x_\nu, x^\star]}{f[x_{\nu-1}, x_\nu]}. \end{aligned}$$

Sei $x_{\nu-1}, x_\nu \in \mathbb{U} = \overline{K_\varepsilon(x^\star)}$. Wegen (3.21) ist

$$f[x_{\nu-1}, x_\nu] = f'(\xi_1), \quad f[x_{\nu-1}, x_\nu, x^\star] = \frac{1}{2}f''(\xi_2)$$

mit $\xi_1, \xi_2 \in \mathbb{U}$. Ist ε hinreichend klein, so existiert wegen $f'(x^\star) \neq 0$

$$M = \max_{x,y \in \mathbb{U}} \left| \frac{\frac{1}{2} f''(y)}{f'(x)} \right|.$$

Setzt man $\varepsilon_\nu = M|x^\star - x_\nu|$, so folgt aus der Ungleichung

$$|x^\star - x_{\nu+1}| \leq |x^\star - x_\nu| \cdot |x^\star - x_{\nu-1}| \cdot M$$

die Ungleichung $\varepsilon_{\nu+1} \leq \varepsilon_\nu \varepsilon_{\nu-1}$. Seien $\varepsilon, \varepsilon_0, \varepsilon_1$ so klein, dass $\varepsilon_0 \varepsilon_1 \leq \varepsilon M < 1$. Mit

$$K = \max\left\{\varepsilon_0, \sqrt[q]{\varepsilon_1}\right\} < 1,$$

wobei q die positive Wurzel von $\lambda^2 - \lambda - 1 = 0$ ist, d.h. $q = \frac{1}{2}(q + \sqrt{5})$, folgt dann $\varepsilon_\nu \leq K^{q^\nu}$. Dies ist nämlich für $\nu = 0$ mit $\varepsilon_0 \leq K$ und $\varepsilon_1 \leq K^q$ per Ansatz richtig und es gilt

$$\varepsilon_{\nu+1} \leq \varepsilon_\nu \cdot \varepsilon_{\nu-1} \leq K^{q^\nu} \cdot K^{q^{\nu-1}} = K^{q^\nu + q^{\nu+1}} = K^{q^{\nu-1}(q+1)} = K^{q^{\nu-1} q^2} = K^{q^{\nu+1}}.$$

□

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 0 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.

6.4 Newton-Verfahren

Statt die zu $x_{\nu-1}$ und x_ν gehörige Sekante zu verwenden, kann man bei differenzierbarem f auch die zu x_ν gehörige Tangente verwenden. Diese ist gegeben durch (auch für $f : \mathbb{D} \to \mathbb{R}^n$, $\mathbb{D} \subseteq \mathbb{R}^n$ durchführbar):

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

mit der Nullstelle

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

Die Vorschrift (6.17) entspricht dem Iterationsverfahren (6.5) mit

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

Wegen

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

gilt hier

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

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 6.8. *Sei $f \in \mathcal{C}^1(\mathbb{D}, \mathbb{R}^n)$, $\mathbb{D} \subseteq \mathbb{R}^n$ offen, $f'(x)$ regulär für alle $x \in \mathbb{D}$ und*

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

in einer Vektornorm mit zugehöriger Matrixnorm. Weiter sei $x^\star \in \mathbb{D}$ eine Lösung von (6.1) und x_0 genüge

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

Dan ist durch (6.17) eine Folge $(x_\nu)_{\nu \in \mathbb{N}_0}$ definiert mit $x_\nu \to x^\star$. Insbesondere gilt

$$= \|x_{\nu+1} - x^\star\| \leq \frac{\omega}{2} \|x^\star - x_\nu\|^2 \tag{6.22a}$$

$$= \|x^\star - x_\nu\| \leq \frac{2}{\omega} K^{2^\nu}, \quad K \in (0, 1) \tag{6.22b}$$

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

Beweis. Seien $x_0, \dots, x_\nu \in \overline{K_\varrho(x^\star)}$. Dann gilt

$$\begin{aligned}
x^\star - x_{\nu+1} &= x^\star - x_\nu + f'(x_\nu)^{-1} f(x_\nu) = x^\star - x_\nu + f'(x_0)^{-1}(f(x_\nu) - f(x^\star)) \\
&= f'(x_\nu)^{-1}[f(x_\nu) - f(x^\star) - f'(x_\nu)(x_\nu - x^\star)] = f'(x_\nu)^{-1}\left[f(x^\star + s(x_\nu - x^\star))\Big|_0^1 - f'(x_\nu)(x_\nu - x^\star)\right] \\
&= \int_0^1 f'^{-1}\left[f'(x^\star + s(x_\nu - x^\star)) \cdot x_\nu - x^\star) - f'(x_\nu)(x_\nu - x^\star)\right] ds \\
&= \int_0^1 f'^{-1}\left[f'(x^\star + s(x_\nu - x^\star)) - f'(x_\nu)\right](x_\nu - x^\star)\, ds
\end{aligned}$$

beziehungsweise

$$\|x^\star - x_{\nu+1}\| \le \int_0^1 \omega \|x^\star + s(x_\nu - x^\star) - x_\nu\| \cdot \|x_\nu - x^\star\|\, ds = \omega \|x_\nu - x^\star\|^2 \int_0^1 (1-s)\, ds = \frac{\omega}{2}\|x^\star - x_\nu\|^2.$$

Damit folgt

$$\|x^\star - x_{\nu+1}\| \le \frac{\omega}{2}\varrho\|x^\star - x_\nu\| \le \|x^\star - x_\nu\| \le \varrho.$$

Das heißt $x_{\nu+1} \in \overline{K_\varrho(x^\star)}$. Mit

$$h_\nu = \frac{\omega}{2}\|x^\star - x_\nu\|$$

kann man (6.22a) schreiben in der Form

$$0 \le h_{\nu+1} \le h_\nu^2, \quad h_0 = \frac{\omega}{2}\|x^\star - x_0\| \le 1$$

das heißt

$$h_\nu \le h_0^{2^\nu} \xrightarrow{\nu\to\infty} 0,$$

also $x_\nu \to x^\star$. Insbesondere folgt (6.22b) mit $K = h_0$. Ist $x^{\star\star} \in K_{\frac{2}{\omega}}(x^\star) \cap \mathbb{D}$ eine zweite Lösung von (6.1), so gilt

$$\begin{aligned}
\|x^{\star\star} - x^\star\| &= \|f'(x^\star)^{-1}[f(x^{\star\star}) - f(x^\star) - f'(x^\star)(x^{\star\star} - x^\star)]\| \\
&\le \frac{\omega}{2}\|x^{\star\star} - x^\star\|^2 = \sigma\|x^{\star\star} - x^\star\|,
\end{aligned}$$

wobei $\sigma = \frac{\omega}{2}\|x^{\star\star} - x^\star\| < 1$ und damit $\|x^{\star\star} - x^\star\| = 0$. □

Die Bedingung (6.21) verlangt wiederum, dass x_0 hinreichend nahe bei $x^\star$ liegen muss. Man sagt deshalb kurz, das Newton-Verfahren sei lokal und quadratisch konvergent.

Index