Vorlesungsskriptum

Numerische Mathematik

Teil II

Dr. Stefan Frei

basierend auf der gleichnamigen Vorlesung

an der Universität Konstanz im WS 2023/24

Inhaltsverzeichnis

4 Nichtlineare Gleichungen

In diesem Kapitel betrachten wir numerische Verfahren zur Bestimmung von Nullstellen nichtlinearer Funktionen $f : \mathbb{R}^n \to \mathbb{R}$

$$f(x) = 0, \qquad x \in \mathbb{R}^n$$

Die Fokussierung auf Nullstellensuche ist dadurch motiviert, dass man jede nichtlineare Gleichung der Form

$$g(x) = h(x)$$

durch Setzen von $f = g - h$ in obige Form bringen kann. Wir konzentrieren uns zunächst auf den Fall $n = 1$.

4.1 Intervallschachtelung (Bisektion)

Sei $f : I \to \mathbb{R}$ eine stetige Funktion auf einem Intervall $I = [a, b]$. Das Verfahren der Intervallschachtelung (auch:*Bisektion*) ist motiviert durch den Zwischenwertsatz für stetige Funktionen:

Existiert ein Teilintervall $I_0 = [a_0, b_0] \subset I$ mit $f(a_0)f(b_0) < 0$, so hat f mindestens eine Nullstelle in I_0.

Ausgehend von einem solchen Intervall $I_k = [a_k, b_k]$ definieren wir für $k \geq 0$ die Iterierte

$$x_{k+1} := \frac{1}{2}(a_k + b_k).$$

sowie die neuen Intervallgrenzen

$$[a_{k+1}, b_{k+1}] = \begin{cases} [a_k, x_{k+1}], & \text{wenn } f(a_k)f(x_{k+1}) < 0 \\ [x_{k+1}, b_k], & \text{wenn } f(a_k)f(x_{k+1}) > 0. \end{cases}$$

Gilt $f(x_{k+1}) = 0$, so ist eine Nullstelle gefunden.

Intervallschachtelung

Eingabe: Intervallgrenzen a_0, b_0 mit $a_0 < b_0$, $f(a_0) \cdot f(b_0) < 0$

Für $k = 0, 1, \ldots$

Setze $x_{k+1} := \frac{1}{2}(a_k + b_k)$

Setze $[a_{k+1}, b_{k+1}] = \begin{cases} [a_k, x_{k+1}], & \text{wenn } f(a_k)f(x_{k+1}) < 0 \\ [x_{k+1}, b_k], & \text{wenn } f(a_k)f(x_{k+1}) > 0. \end{cases}$ (4.1)

Ausgabe: Iterierte x_{k+1}, Intervallgrenzen a_{k+1}, b_{k+1}

Für eine Nullstelle $z \in I_0$ gilt nach Definition für alle $k \geq 0$

$$a_k \leq a_{k+1} \leq z \leq b_{k+1} \leq b_k$$

und außerdem

$$|x_k - z| \leq |a_k - b_k| = \frac{1}{2}|a_{k-1} - b_{k-1}| \leq \left(\frac{1}{2}\right)^k |a_0 - b_0| \to 0 \quad (k \to \infty) \tag{4.2}$$

Der Fehler wird in jeder Iteration also ca. um einen Faktor $\frac{1}{2}$ kleiner. Wir werden in diesem Fall später von *linearer* Konvergenz sprechen.

Die Intervallschachtelung ist ein stabiler numerischer Algorithmus zur Berechnung einer Nullstelle für beliebige **stetige** Funktionen f. Konvergenz ist unter der Bedingung $f(a_0)f(b_0) < 0$ sichergestellt. Ist man an einer gewissen Genauigkeit (z.B. 6 Stellen) interessiert, ist die Konvergenz in der Regel allerdings relativ langsam

$$\left(\frac{1}{2}\right)^k \leq 10^{-6} \quad \Rightarrow k \geq 20.$$

4.2 Newton-Verfahren in $\mathbb{R}^1$

Wir nehmen nun an, dass die Funktion f zweimal stetig differenzierbar ist. Das Newton-Verfahren (auch *Newton-Raphson*-Verfahren) basiert geometrisch auf der Idee eine (möglicherweise komplizierte) Funktion f lokal durch ihre Tangente anzunähern, siehe Abbildung 4.1.

Sei $x_0 \in I$ ein Startwert. Es gilt mit Taylorentwicklung für ein $\xi \in [x_0, x]$

$$f(x) = \underbrace{f(x_0) + f'(x_0)(x - x_0)}_{=:T_{x_0}(x)} + \frac{f''(\xi)}{2}(x - x_0)^2 = T_{x_0}(x) + \mathcal{O}\left(|x - x_0|^2\right).$$

Nahe bei x_0 ist $T_{x_0}(x)$ also eine gute Näherung von $f(x)$. Aufgrund des Restterms spricht man von einer *Näherung zweiter Ordnung*. Im Falle $f'(x_0) \neq 0$ kann die Nullstelle von $T_{x_0}(x)$ einfach berechnet werden

$$f(x_0) + f'(x_0)(x - x_0) = 0 \qquad \Rightarrow x = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

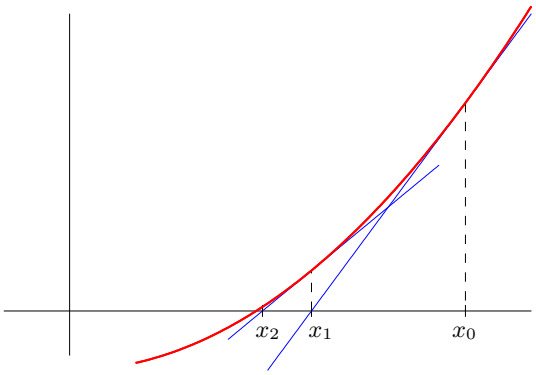

Abbildung 4.1: Graphische Darstellung des Newton-Verfahrens

Die Nullstelle x der Tangente ist im Allgemeinen noch nicht die Nullstelle von f. Da die Tangente eine Näherung zweiter Ordnung an f ist, besteht aber Hoffnung, dass diese deutlich näher an der Nullstelle von z liegt als x_0. Wir definieren die Iteration

Newton-Verfahren

Eingabe: Startwert $x_0 \in I$,

Für $k = 1, 2, \ldots$ setze $x_k = x_{k-1} - \dfrac{f(x_{k-1})}{f'(x_{k-1})}$. (4.3)

Ausgabe: Iterierte x_k

Geeignete Abbruchkriterien für die Iteration werden wir weiter unten diskutieren.

Wir wollen nun das Konvergenzverhalten der Newton-Iteration analysieren. Wir haben bereits gesehen, dass die Newton-Iteration nur für $f'(x_k) \neq 0$ wohldefiniert ist. Deswegen beschränken wir uns zunächst auf die Suche einfacher Nullstellen z, d.h. Nullstellen, für die $f'(z) \neq 0$. Da f' nach Voraussetzung stetig ist, ist dann auch $f'(x) \neq 0$ in einer Umgebung von z. Mit $C^k[a,b]$ bezeichnen wir den Raum der k-fach stetig differenzierbaren Funktionen auf (a,b), die stetig bis zum Rand fortgesetzt werden.

Satz 4.1. *Die Funktion $f \in C^2[a,b]$ habe eine Nullstelle $z \in (a,b)$ und wir nehmen an, dass*

$$m := \min_{x \in [a,b]} |f'(x)| > 0.$$

Außerdem sei $M := \max_{x\in[a,b]} |f''(x)|$ und $\rho > 0$ sei so gewählt, dass

$$\rho < \frac{2m}{M}, \quad K_\rho(z) := \{x \in \mathbb{R}, |x - z| \leq \rho\} \subset [a,b]. \tag{4.4}$$

Dann bleiben die Newton-Iterierten x_k für jeden Startwert $x_0 \in K_\rho(z)$ in $K_\rho(z)$ und konvergieren gegen die Nullstelle z. Es gilt

$$|x_k - z| \leq \frac{M}{2m}|x_{k-1} - z|^2, \quad k \in \mathbb{N}, \tag{4.5}$$

sowie die a priori Fehlerabschätzung

$$\frac{M}{2m}|x_k - z| \leq \left(\frac{M}{2m}|x_0 - z|\right)^{2^k} \leq q^{2^k}, \quad k \in \mathbb{N} \tag{4.6}$$

mit $q := \frac{M}{2m}\rho < 1$ und die a posteriori Abschätzung

$$|x_k - z| \leq \frac{1}{m}|f(x_k)| \leq \frac{M}{2m}|x_k - x_{k-1}|^2, \quad k \in \mathbb{N}. \tag{4.7}$$

Im Fall $M = 0$ (d.h. f ist linear) konvergiert das Newton-Verfahren in einem Schritt und es gilt $x_1 = z$.

Bemerkung 4.2. *Wir sprechen von einer **a posteriori** Fehlerabschätzung, wenn nur berechenbare Größen auf der rechten Seite der Abschätzung eingehen (z.B. die Iterierten x_k, x_{k-1}). Solche Fehlerabschätzungen können **nach** einem Iterationsschritt ausgerechnet werden und können z.B. bei der Definition von Abbruchkriterien verwendet werden. Bei einer **a priori** Abschätzung können dagegen auch **im Voraus** unbekannte Größen wie die unbekannte Nullstelle z eingehen. Diese sind in der Regel unabhängig von der Folge der Iterierten und können verwendet werden, um **vor** der Berechnung abzuschätzen, wie viele Iterationen maximal notwendig sein werden.*

Beweis. (i) Wir zeigen zunächst, dass die Newton-Iterierten x_k in $K_\rho(z)$ bleiben. Dazu zeigen wir, dass

$$|x_{k-1} - z| < \rho \quad \Rightarrow |x_k - z| < \rho.$$

Der Beweis basiert auf der Taylorentwicklung

$$f(z) = f(x_{k-1}) + f'(x_{k-1})(z - x_{k-1}) + \frac{f''(\xi)}{2}(z - x_{k-1})^2, \quad \text{für ein } \xi \in [x_{k-1}, z].$$

Da $f(z) = 0$ und $f'(x_{k-1}) \neq 0$ folgt daraus

$$0 = \frac{f(x_{k-1})}{f'(x_{k-1})} + z - x_{k-1} + \frac{f''(\xi)}{2f'(x_{k-1})}(z - x_{k-1})^2.$$

Damit gilt unter der Voraussetzung (4.4)

$$|x_k - z| = |x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})} - z| = \left|\frac{f''(\xi)}{2f'(x_{k-1})}(z - x_{k-1})^2\right| \leq \frac{M}{2m}|z - x_{k-1}|^2 < \frac{M}{2m}\rho^2 \leq \rho, \tag{4.8}$$

d.h. $x_k \in K_\rho(z)$.

(ii) *A priori Abschätzung*: Als Nebenprodukt erhalten wir aus (4.8) bereits (4.5). Um (4.6) zu zeigen, setzen wir

$$r_k := \frac{M}{2m}|x_k - z|.$$

(4.8) impliziert

$$r_k \leq r_{k-1}^2$$

und damit induktiv

$$\frac{M}{2m}|x_k - z| = r_k \leq r_0^{(2^k)} = \left(\frac{M}{2m}\underbrace{|x_0 - z|}_{\leq \rho}\right)^{(2^k)} \leq q^{(2^k)}.$$

(iii) Es bleibt die *a posteriori* Abschätzung (4.7) zu zeigen. Dazu verwenden wir eine Taylorentwicklung erster Ordnung

$$f(x_k) = f(z) + f'(\xi)(x_k - z), \qquad \xi \in [x_k, z].$$

Daraus folgt direkt die erste Ungleichung in (4.7)

$$|x_k - z| \leq \frac{|f(x_k)|}{|f'(\xi)|} \leq \frac{|f(x_k)|}{m}. \tag{4.9}$$

Eine weitere Taylorentwicklung zweiter Ordnung ergibt

$$f(x_k) = \underbrace{f(x_{k-1}) + f'(x_{k-1})(x_k - x_{k-1})}_{=0} + \frac{f''(\xi)}{2}(x_k - x_{k-1})^2, \quad \text{für ein } \xi \in [x_{k-1}, x_k].$$

Der erste Term auf der rechten Seite verschwindet nach Definition der Newton-Iterierten x_k und es folgt

$$\frac{|f(x_k)|}{m} \leq \frac{M}{2m}|x_k - x_{k-1}|^2.$$

□

Abbruchkriterien Die *a posteriori* Abschätzung (4.7) legt zur Erreichung einer vorgegebenen Toleranz zwei mögliche Abbruchkriterien nahe

$$\frac{1}{m}|f(x_k)| < \text{TOL} \quad \text{oder} \quad \frac{M}{2m}|x_k - x_{k-1}|^2 < \text{TOL},$$

wobei die erste Bedingung wegen der zweiten Ungleichung in (4.7) die *schärfere* ist. Oft ist man daran interessiert statt dem *absoluten* Fehler den *relativen* Fehler

$$\frac{|x_k - z|}{|z|} \approx \frac{|x_k - z|}{|x_k|}$$

zu kontrollieren (Ist die Lösung z sehr groß ist es weniger relevant die n-te Nachkommastelle richtig zu approximieren). In diesem Fall schreiben sich die obigen Kriterien als

$$\frac{1}{m}\frac{|f(x_k)|}{|x_k|} < \text{TOL} \quad \text{oder} \quad \frac{M}{2m}\frac{|x_k - x_{k-1}|^2}{|x_k|} < \text{TOL}.$$

Newton-Verfahren mit Abbruchkriterium

Eingabe: Startwert $x_0 \in I$

Solange $\frac{1}{m}\frac{|f(x_k)|}{|x_k|} < \text{TOL}$:

Setze $x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})}$.

$k \leftarrow k+1$

Ausgabe: Iterierte x_k

4.3 Konvergenzbegriffe

Definition 4.3. *Ein Iterationsverfahren zur Bestimmung einer Nullstelle z einer Funktion $f \in C[a,b]$ besitzt die* ***Konvergenzordnung*** *$p \geq 1$, wenn für beliebige Startwerte $x_0 \in [a,b]$ gilt, dass*

$$|x_k - z| \leq c|x_{k-1} - z|^p \quad \text{für } k \in \mathbb{N}. \tag{4.10}$$

Desweiteren sprechen wir auch von Konvergenz der Ordnung p, wenn es eine obere Schranke η_k für $|x_k - z|$ $(k \in \mathbb{N})$ gibt, die mit dieser Ordnung konvergiert

$$|x_k - z| \leq \eta_k, \qquad \eta_k \leq c\,\eta_{k-1}^p, \quad k \in \mathbb{N}. \tag{4.11}$$

Im Fall $p = 1$ spricht man unter der zusätzlichen Voraussetzung $c < 1$ von linearer Konvergenz, im Falle $p = 2$ von quadratischer Konvergenz. Weiter spricht man von superlinearer Konvergenz, wenn es eine Nullfolge $(c_k)_{k\in\mathbb{N}}$ gibt mit

$$|x_k - z| \leq c_k|x_{k-1} - z| \quad \text{für } k \in \mathbb{N}.$$

Im Falle $p > 1$ liegt in der Regel nur *lokale Konvergenz* vor, d.h. für Startwerte $x_0 \in K_\rho(z)$ für ein $\rho > 0$. Dies sieht man folgenderweise

$$c^{\left(\frac{1}{p-1}\right)}|x_k - z| \leq \underbrace{c^{\left(\frac{1}{p-1}\right)}c}_{=c^{\frac{p}{p-1}}}|x_{k-1} - z|^p = \left(c^{\left(\frac{1}{p-1}\right)}|x_{k-1} - z|\right)^p \leq \ldots \leq \left(\underbrace{c^{\left(\frac{1}{p-1}\right)}|x_0 - z|}_{\overset{!}{<}1}\right)^{(p^k)}. \tag{4.12}$$

Es folgt lokale Konvergenz mit $\rho < c^{-\frac{1}{p-1}}$.

In Theorem 4.1 haben wir gezeigt, dass das Newtonverfahren *lokal* quadratisch konvergiert, wobei $c = \frac{M}{2m}$ und daraus resultierend $\rho < c^{\left(\frac{1}{2-1}\right)} = \frac{2m}{M}$. Liegt der Startwert dagegen nicht genügend nahe an der Nullstelle z, ist die Konvergenz der Iteration nicht gesichert. In der Praxis werden bei solchen Problemen zum Beispiel langsamere, aber global konvergente Verfahren (z.B. die Intervallschachtelung) eingesetzt, um einen genügend guten Startwert für die Newton-Iteration zu berechnen.

Für die Intervallschachtelung ist (4.10) nicht notwendigerweise erfüllt, da $|x_k - z|$ im Allgemeinen nicht monoton fällt. Dagegen konvergiert die obere Schranke $\eta_k = |b_k - a_k|$ monoton (siehe (4.2))

$$\eta_k = |b_k - a_k| = \frac{1}{2}|b_{k-1} - a_{k-1}| = \frac{1}{2}\eta_{k-1}.$$

Es folgt also lineare Konvergenz im Sinne von (4.11).

Beispiel: Wurzelberechnung Wir wenden das Newton-Verfahren zur Berechnung der n-ten Wurzel von a an. Dazu suchen wir eine Nullstelle der Funktion

$$f(x) = x^n - a.$$

Die Newton-Iteration lautet für dieses Beispiel

$$x_{k+1} = x_k - \frac{x_k^n - 1}{n x_k^{n-1}} = \frac{1}{n}\left((n-1)x_k + \frac{a}{x_k^{n-1}}\right).$$

Die Funktion f ist konvex und streng monoton wachsend auf $\mathbb{R}_+$. In diesem Fall kann gezeigt werden, dass die Newton-Iteration für jeden beliebigen Startwert $x_0 > 0$ konvergiert (siehe Übung). Daher wird das Newton-Verfahren auf vielen Rechnern zur Berechnung von Wurzeln eingesetzt.

Wir wollen dies im Fall $n = 2$ (sprich der Berechnung der Quadratwurzel $\sqrt{a}$) veranschaulichen. Die Iterationsvorschrift lautet

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right).$$

Die rechte Seite nimmt ein Minimum im Punkt $x = \sqrt{a}$ an, d.h. für einen beliebigen Startwert x_0 gilt $x_1 \geq \sqrt{a}$ und damit $x_k \geq \sqrt{a}$ für $k \geq 1$. Weiter gilt (siehe Abb. 4.2)

$$x_k > x_{k+1} > \sqrt{a},$$

d.h. die Folge der Iterierten x_k konvergiert monoton gegen $\sqrt{a}$.

Wie in (4.8) gilt ($f'(x_k) = 2x_k$, $f''(x_k) = 2$)

$$x_{k+1} - \sqrt{a} = \frac{f''(\xi)}{2f'(x_k)}(x_k - \sqrt{a})^2 = \frac{1}{2x_k}(x_k - \sqrt{a})^2.$$

Wegen

$$\frac{1}{2x_k}(x_k - \sqrt{a}) < \frac{1}{2} \qquad \text{für } x_k \geq \sqrt{a}$$

liegt Konvergenz für beliebige Startwerte vor

$$|x_{k+1} - \sqrt{a}| \leq \frac{1}{2}|x_k - \sqrt{a}|.$$

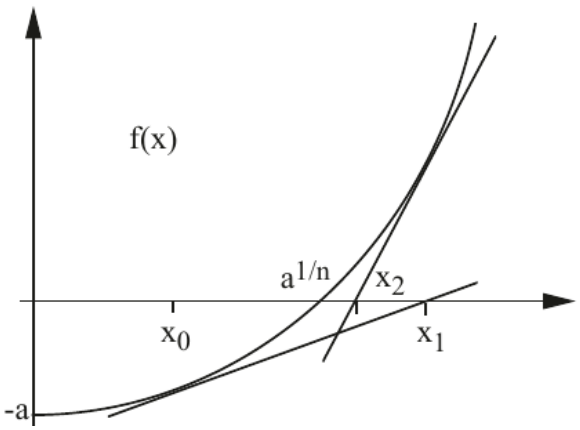

Abbildung 4.2: Monotones Konvergenzverhalten der Newton-Iteration für die Funktion $f(x) = x^n - a$ (Quelle:[4])

Iteration k	x_k	$\lvert z - x_k \rvert$
0	0.1	1.314
1	10.05	8.635
2	5.124502488...	3.710
3	2.757392138...	1.343
4	1.741357580...	$3.271 \cdot 10^{-1}$
5	1.444943382...	$3.073 \cdot 10^{-2}$
6	1.414540330...	$3.268 \cdot 10^{-3}$
7	1.414213600...	$3.774 \cdot 10^{-8}$
8	1.414213562...	$4.441 \cdot 10^{-16}$
9	1.414213562...	$2.220 \cdot 10^{-16}$
10	1.414213562...	$2.220 \cdot 10^{-16}$

Tabelle 4.1: Newton-Verfahren zur Berechnung der Nullstelle $z = \sqrt{2}$ der Funktion $f(x) = x^2 - 2$. Korrekte Dezimalstellen sind unterstrichen.

Quadratische Konvergenz liegt nach Satz 4.1 vor, wenn $\frac{M}{2m}|x_k - z| < 1$. Für $k \geq 1$ gilt $x_k > \sqrt{a}$ und wir können uns in der Definition von m auf diesen Bereich konzentrieren

$$m := \min_{x > \sqrt{a}} f'(x) = 2\sqrt{a} \qquad \Rightarrow \frac{M}{2m}|x_k - z| = \frac{1}{2\sqrt{a}}|x_k - \sqrt{a}| \overset{!}{<} 1$$

$$\Rightarrow x_k \overset{!}{<} 3\sqrt{a}.$$

Je nach Wahl des Startwerts x_0 kann das entweder schon für x_0 oder x_1 erfüllt sein, oder erst im Laufe der Iteration. Wir geben in Tabelle 4.1 als Beispiel die Berechnung von $\sqrt{2}$ ($a = 2$) mit (einem willkürlich bestimmten) Startwert $x = 0.1$.

In der ersten Iteration erhöht sich der Fehler stark. In Iterationen $2, 3$ nimmt der Fehler jeweils um etwas mehr als einen Faktor 2 ab, d.h. es liegt lineare Konvergenz vor. Ab der vierten Iteration

Iteration k	a_k	b_k	x_k	$\lvert z - x_k\rvert$
0	1	2	1	$4.14 \cdot 10^{-1}$
1	1	1.5	1.5	$8.58 \cdot 10^{-2}$
2	1.25	1.5	1.25	$1.64 \cdot 10^{-1}$
3	1.375	1.5	1.375	$3.92 \cdot 10^{-2}$
4	1.375	1.4375	1.4375	$2.33 \cdot 10^{-2}$
5	1.40625	1.4375	1.40625	$7.96 \cdot 10^{-3}$
6	1.40625	1.421875	1.421875	$7.66 \cdot 10^{-3}$
7	1.4140625	1.421875	1.414063	$1.51 \cdot 10^{-4}$
8	1.4140625	1.417969	1.417969	$3.76 \cdot 10^{-3}$
9	1.4140625	1.416016	1.416016	$1.80 \cdot 10^{-3}$
10	1.4140625	1.415039	1.415039	$8.26 \cdot 10^{-4}$

Tabelle 4.2: Intervallschachtelung zur Nullstellensuche bei $f(x) = x^2 - 2$ mit Startintervall $I_0 = [a_0, b_0]$. Korrekte Dezimalstellen sind unterstrichen.

verdoppelt sich die Anzahl der korrekten Dezimalstellen. Es liegt quadratische Konvergenz vor

$$|x_k - z| < 10^{-n} \quad \Rightarrow |x_k - z| < c|x_k - z|^2 < c \cdot 10^{-2n}.$$

Ab Iteration 9 liegt der Fehler $x_9 - z$ im Bereich der Maschinengenauigkeit $\epsilon = 2^{-53} \approx 1.1 \cdot 10^{-16}$. Aufgrund von Rundungsfehlern ist bei Verwendung von **double precision** keine Verbesserung mehr möglich.

Intervallschachtelung In Tabelle 4.2 stellen wir die Ergebnisse des Intervallschachtelungsverfahrens für das selbe Problem dar. Als Startintervall ist hier $I_0 = [1, 2]$ gewählt. Der Fehler $|x_k - z|$ reduziert sich, allerdings nicht-monoton. Während der Fehler beim Newton-Verfahren nach 8 Schritten im Bereich der Maschinengenauigkeit liegt, liegt dieser bei der Intervallschachtelung nach 8 Schritten im Bereich $\mathcal{O}(10^{-3})$

In Abb. 4.3 vergleichen wir das Konvergenzverhalten des Newton-Verfahrens und der Intervallschachtelung in einem halblogarithmischen Plot. Dabei wird in der Vertikalen der Logarithmus des Fehlers $\log(|x_k - z|)$ aufgetragen. Bei linearer Konvergenz folgt aus $|x_k - z| \leq c|x_{k-1} - z|$ mit Konstanten $d > 0$, $q \in (0, 1)$

$$|x_k - z| \leq d \cdot q^k \quad \Rightarrow \log(|x_k - z|) \leq \log(d) + k \log(q).$$

d.h. wir erhalten eine Gerade mit Steigung $\log(q) < 0$. Dies kann man bei der Reduktion der Intervalllänge $|b_k - a_k|$ beim Intervallschachtelungsverfahren beobachten. Der Fehler $|x_k - z|$ verläuft jeweils knapp unter dieser Linie. Bei Konvergenzordnung $p > 1$ folgt aus (4.12)

$$|x_k - z| \leq d \cdot q^{(p^k)} \quad \Rightarrow \log(|x_k - z|) = \log(d) + p^k \log(q),$$

d.h. die Kurve fällt exponentiell. Dies ist beim Newton-Verfahren zu beobachten, bis der Fehler die Größenordnung der Maschinengenauigkeit erreicht.

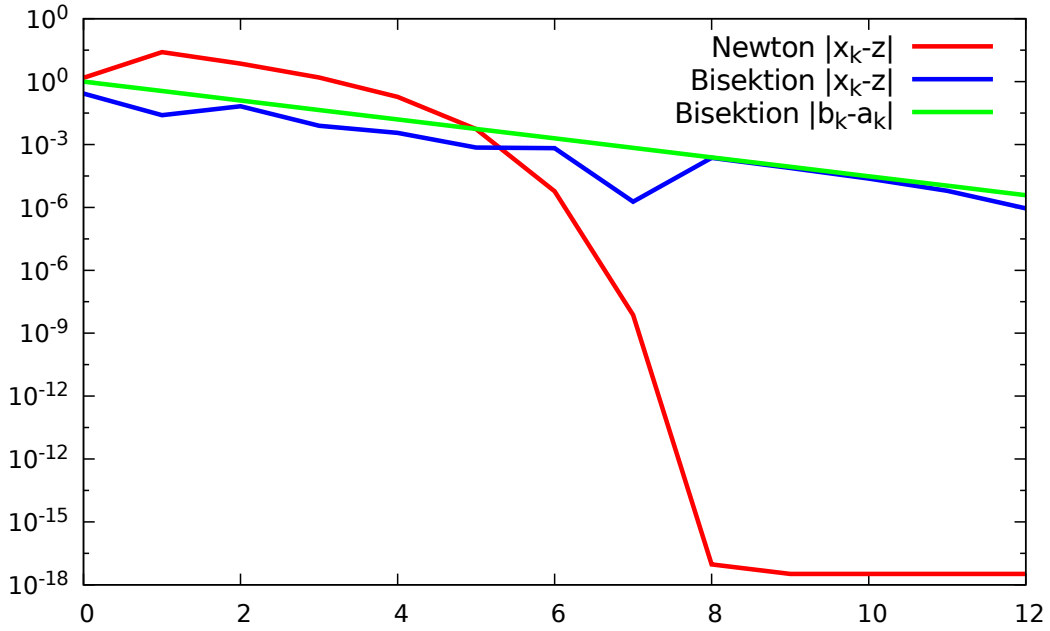

Abbildung 4.3: Vergleich des Konvergenzverhaltens des Newton-Verfahrens und der Intervallschachtelung (Bisektion) in einem halblogarithmischen Plot. Für die Intervallschachtelung liefert die linear konvergierende Intervalllänge $b_k - a_k$ eine obere Schranke für den Fehler.

4.4 Varianten des Newton-Verfahrens

4.4.1 Gedämpftes Newton-Verfahren

Das größte Problem beim Newton-Verfahren ist häufig die Wahl eines geeigneten Startwertes x_0. Ist der Einzugsbereich der quadratischen Konvergenz einmal erreicht, konvergiert das Verfahren sehr schnell. Für beliebiges x_0 ist dagegen im allgemeinen keine Konvergenz gesichert.

Eine einfache Strategie, um den Konvergenzradius zu vergrößern, ist das sogenannte *gedämpfte* Newton-Verfahren mit einer Folge von Dämpfungsparametern $(\lambda_k)_{k\geq 0}, \lambda_k \in (0, 1]$

Eingabe: Startwert $x_0 \in I$, $k = 1$

Solange $\frac{1}{m}\frac{|f(x_{k-1})|}{|x_{k-1}|} > \text{TOL}$:

Setze $x_k = x_{k-1} - \lambda_k \frac{f(x_{k-1})}{f'(x_{k-1})}$ für ein $\lambda_k \in (0, 1]$. (4.13)

$k \leftarrow k + 1$

Ausgabe: Iterierte x_k

Bei "optimaler" Wahl der Folge $(\lambda_k)_{k\geq 0}$ kann man sogar globale Konvergenz sicherstellen.

Satz 4.4. *Die Funktion $f \in C^2[a,b]$ habe eine Nullstelle $z \in (a,b)$ und wir nehmen an, dass*

$$m := \min_{x\in[a,b]} |f'(x)| > 0.$$

Für beliebige Startwerte $x_0 \in [a,b]$ kann man eine Folge $(\lambda_k)_{k>0}$ finden, so dass das gedämpfte Newton-Verfahren (4.13) gegen eine Nullstelle z konvergiert. Sobald $x_k \in K_\rho(z)$ für $\rho = \frac{m}{2M}$ und $M = \max_{x\in[a,b]} |f''(x)|$, kann $\lambda_k = 1$ gewählt werden und das gedämpfte Newton-Verfahren konvergiert nach Satz 4.1 quadratisch.

Beweis. Wir zeigen, dass für beliebiges x_k stets ein Wert $\lambda_k = \lambda \in (0,1]$ gefunden werden kann, so dass für $x(\lambda) = x_k - \lambda \frac{f(x_k)}{f'(x_k)}$ gilt

$$|f(x(\lambda))| < q|f(x_k)| \quad \text{für ein } q < 1.$$

Mithilfe von Taylor-Entwicklung gilt für ein $\xi \in [x_k, x(\lambda)]$

$$\begin{aligned} f(x(\lambda)) = f\left(x_k - \lambda\frac{f(x_k)}{f'(x_k)}\right) &= f(x_k) - \lambda\frac{f(x_k)}{f'(x_k)}f'(x_k) + \lambda^2\frac{f(x_k)^2}{f'(x_k)^2}\frac{f''(\xi)}{2} \\ &= \left(1 - \lambda + \frac{\lambda^2}{2}\frac{f(x_k)f''(\xi)}{f'(x_k)^2}\right) f(x_k). \end{aligned} \tag{4.14}$$

Für $\lambda \le 1$ folgt mit der Dreiecksungleichung

$$|f(x(\lambda))| \le \left(1 - \lambda + \frac{\lambda^2}{2}\underbrace{\frac{M|f(x_k)|}{|f'(x_k)|^2}}_{=:\alpha_k<\infty}\right)|f(x_k)|.$$

Mit der Wahl

$$\lambda_k = \min\left\{1, \frac{1}{\alpha_k}\right\} \tag{4.15}$$

gilt im Fall $\alpha_k > 1$ (d.h. $\lambda_k = \alpha_k^{-1}$)

$$1 - \lambda_k + \alpha_k\frac{\lambda_k^2}{2} = 1 - \frac{1}{2\alpha_k} \le q < 1$$

und für $\alpha_k < 1$ (d.h. $\lambda_k = 1$)

$$1 - \lambda_k + \alpha_k\frac{\lambda_k^2}{2} = \frac{\alpha_k}{2} \le \frac{1}{2}.$$

In beiden Fällen gilt bei dieser Wahl von λ_k für die nächste Iterierte x_{k+1}

$$|f(x_{k+1})| = |f(x(\lambda))| \le \underbrace{\max\{q, \frac{1}{2}\}}_{=:\tilde{q}}|f(x_k)| \le \tilde{q}^{k+1}|f(x_0)| \to 0 \quad (k\to\infty).$$

Wie in (4.9) folgert man

$$|x_k - z| < \frac{|f(x_k)|}{m} < \tilde{q}^{k+1}\frac{|f(x_0)|}{m},$$

d.h. lineare Konvergenz der Folge $(x_k)_{k\geq 0}$ gegen eine Nullstelle z im Sinne von (4.11).

Gilt $|x_k - z| < \rho = \frac{m}{2M}$, so ergibt sich wegen

$$0 = f(z) = f(x_k) + (z - x_k)f'(x_k) + (z - x_k)^2\frac{f''(\xi)}{2}, \qquad \xi \in [x_k, z]$$

die Beziehung

$$\begin{aligned}\alpha_k = \frac{M|f(x_k)|}{|f'(x_k)|^2} &\leq \frac{M}{|f'(x_k)|}|x_k - z| + \frac{M|f''(\xi)|}{2|f'(x_k)|^2}|x_k - z|^2 \\ &< \frac{M}{m}\rho + \frac{M^2}{2m^2}\rho^2 = \frac{1}{2} + \frac{1}{8} < 1 \qquad \Rightarrow \lambda_k = 1.\end{aligned}$$

Die Iteration entspricht fortan also der Standard-Newton-Iteration und konvergiert nach Satz 4.1 quadratisch. □

Der Beweis von Satz 4.4 legt die folgende Wahl der Dämpfungsparameter nahe, siehe (4.15):

$$\lambda_k = \min\left\{1, \frac{1}{\alpha_k}\right\}, \quad \text{wobei } \alpha_k\frac{M|f(x_k)|}{|f'(x_k)|^2}. \tag{4.16}$$

Allerdings ist das Maximum der zweiten Ableitungen M in der Praxis oft nicht bekannt oder es kann eine große Überschätzung der tatsächlich in (4.14) benötigten Ableitung $f''(\xi)$ darstellen.

Beispiel Wir betrachten das Beispiel

$$f(x) = x - 1 + \frac{1}{10}\sin(8x) \tag{4.17}$$

mit der einzigen reellen Nullstelle $x \approx 0.91423454$. Es gilt

$$f'(x) = 1 + \frac{4}{5}\cos(8x), \quad f''(x) = -\frac{32}{5}\sin(8x), \quad M = \frac{32}{5}.$$

Als Startwert wählen wir $x_0 = 2$. In Tabelle 4.3 zeigen wir Ergebnisse mit dem Standard-Newton-Verfahren ($\lambda_k \equiv 1$) und dem gedämpften Newton-Verfahren mit der Wahl der Dämpfungsparameter (4.16). Beim Standard-Newton-Verfahren erkennen wir, dass der erste Newton-Schritt viel zu groß ist und die Iterierte von $x_0 = 2$ auf $x_1 \approx -2.15$ springt. Ab Iteration 4 scheint sich aber quadratische Konvergenz einzustellen.

Beim gedämpften Newton-Verfahren mit der Parameterwahl (4.16) erkennt man eine monotone Konvergenz gegen die Nullstelle. Allerdings ist diese sehr langsam, da die Dämpfungsparameter bis Iteration 13 sehr klein gewählt werden. Hier zeigt sich, dass die Wahl der Dämpfungsparameter aus dem Beweis sehr pessimistisch ist, da $M \gg |f''(\xi)|$.

Iteration k	λ_k	x_k	$\lvert z - x_k\rvert$
0		2	1.09
1	1	−2.15273287	3.07
2	1	1.04566577	$1.31 \cdot 10^{-1}$
3	1	0.82730886	$8.69 \cdot 10^{-2}$
4	1	0.90693916	$7.30 \cdot 10^{-3}$
5	1	0.91413629	$9.83 \cdot 10^{-5}$
6	1	0.914234533	$1.88 \cdot 10^{-8}$
7	1	0.91423454	$< 8 \cdot 10^{-16}$

Iteration k	λ_k	x_k	$\lvert z - x_k\rvert$
0		2	1.09
1	0.009	1.96345744	1.05
2	0.006	1.93220743	1.02
3	0.008	1.89706208	$9.83 \cdot 10^{-1}$
4	0.016	1.84857017	$9.34 \cdot 10^{-1}$
5	0.045	1.76810716	$8.54 \cdot 10^{-1}$
6	0.178	1.61281843	$6.99 \cdot 10^{-1}$
7	0.745	1.33856559	$4.24 \cdot 10^{-1}$
8	0.385	1.21770606	$3.03 \cdot 10^{-1}$
9	0.048	1.18023296	$2.66 \cdot 10^{-1}$
10	0.035	1.14896471	$2.35 \cdot 10^{-1}$
11	0.045	1.11433524	$2.00 \cdot 10^{-1}$
12	0.087	1.06717244	$1.53 \cdot 10^{-1}$
13	0.264	0.98985137	$7.56 \cdot 10^{-2}$
14	1	0.89531027	$1.89 \cdot 10^{-2}$
15	1	0.91362672	$6.08 \cdot 10^{-4}$
16	1	0.91423383	$7.15 \cdot 10^{-7}$
17	1	0.91423454	$9.95 \cdot 10^{-13}$

Tabelle 4.3: Standard-Newton-Verfahren (links) und gedämpftes Newton-Verfahren mit Wahl des Dämpfungsparameters λ_k nach (4.16) angewendet auf das Beispiel (4.17). Das Newton-Verfahren "springt" in der ersten Iteration sehr weit nach links, konvergiert aber ca. ab Iteration 4 quadratisch. Beim gedämpften Newton-Verfahren beobachten wir monotone, aber langsame Konvergenz.

Iteration k	λ_k	x_k	$\lvert z - x_k\rvert$
0		2	1.08
1	0.25	0.96181678	$4.76 \cdot 10^{-2}$
2	1	0.90809245	$6.14 \cdot 10^{-3}$
3	1	0.91416432	$7.02 \cdot 10^{-5}$
4	1	0.91423454	$9.58 \cdot 10^{-9}$
5	1	0.91423454	$< 3 \cdot 10^{-16}$

Tabelle 4.4: Newton-Verfahren mit Liniensuche angewendet auf das Beispiel (4.17). Der Sprung des Standard-Newton-Verfahrens im ersten Schritt wird verhindert.

Liniensuche Stattdessen werden in der Praxis in der Regel andere Strategien zur Wahl der Dämpfungsparameter verwendet. Eine einfache Möglichkeit ist die sogenannte Liniensuche (*line search*). Hier werden die nächste Iterierte solange für Dämpfungsparameter $\lambda_k \in \{1, \frac{1}{2}, \frac{1}{4}, ...\}$ berechnet, bis $|f(x(\lambda_k))| < q|f(x)|$ für einen vorgegebenen Parameter $q < 1$ gilt:

Newton-Verfahren mit Liniensuche (*line search*)

Eingabe: Startwert $x_0 \in I$.

Solange $\frac{1}{m}\frac{|f(x_{k-1})|}{|x_{k-1}|} > \text{TOL}$:

Setze $\lambda = 1$ und berechne $x(\lambda) = x_{k-1} - \lambda\frac{f(x_{k-1})}{f'(x_{k-1})}$.

Solange $f(x(\lambda))| > q|f(x_{k-1})|$:

Setze $\lambda \leftarrow \lambda/2$ und $x(\lambda) = x_{k-1} - \lambda\frac{f(x_{k-1})}{f'(x_{k-1})}$.

Setze $x_k = x(\lambda), \quad k \leftarrow k + 1$.

Ausgabe: Iterierte x_k

Solange $\lambda_k < 1$ ist die Konvergenzgeschwindigkeit im Allgemeinen nur linear. Sobald der Einzugsbereich der quadratischen Konvergenz erreicht ist ($x_k \in K_\rho(z)$), entspricht auch das Newton-Verfahren mit Liniensuche dem Standard-Newton-Verfahren und konvergiert fortan quadratisch.

In Tabelle 4.4 sind die Ergebnisse des Newton-Verfahrens mit Liniensuche für $q = 0.5$ dargestellt, angewendet auf das obige Beispiel (4.17). Wir sehen, dass hier nur einmalig in Schritt 1 ein Parameter $\lambda_1 = 0.25 < 1$ gewählt wird und anschließend jeweils der volle Newtonschritt ($\lambda_k = 1$) ausgeführt wird. Im Vergleich zum Standard-Newton-Verfahren verhindert die Dämpfung im ersten Schritt den Sprung in die negativen Zahlen. Insgesamt konvergiert dieses Verfahren deshalb schneller als das Standard-Newton-Verfahren und erreicht bereits nach 5 Iterationen einen Fehler in der Größenordnung der Maschinengenauigkeit, im Vergleich zu 7 Iterationen beim Standard-Newton-Algorithmus.

4.4.2 Mehrfache Nullstellen

In Satz 4.1 haben wir vorausgesetzt, dass $f'(z) \neq 0$ an der Nullstelle z. Das Newton-Verfahren konvergiert jedoch auch bei mehrfachen Nullstellen. Ist die Ordnung der Nullstelle p bekannt,

bietet sich folgende Modifikation der Newton-Iteration an

$$x_{k+1} = x_k - p\frac{f(x_k)}{f'(x_k)}.$$

Für $p = 1$ ergibt sich das oben definierte Standard-Newton-Verfahren. Wir erhalten die folgenden Konvergenzresultate, welche die Ergebnisse aus Satz 4.1 verallgemeinern.

Satz 4.5. *Sei $z \in (a, b)$ eine p-fache Nullstelle der Funktion $f \in C^{p+1}([a, b])$, d.h. es gelte*

$$f(z) = f'(z) = ... = f^{(p-1)}(z) = 0, \qquad f^{(p)}(z) \neq 0.$$

Weiter seien

$$m := \min_{x\in[a,b]} |f^{(p)}(x)| > 0, \qquad M := \max_{x\in[a,b]} |f^{(p+1)}(x)|.$$

Unter der Voraussetzung, dass

$$\frac{M}{m}\rho < 1$$

konvergiert das Newton-Verfahren (4.3) für $x_0 \in K_\rho(z)$ mit linearer Konvergenzordnung gegen die Nullstelle z, d.h. mit einem $q < 1$ gilt

$$|x_k - z| < q|x_{k-1} - z|.$$

Das modifizierte Newton-Verfahren

$$x_{k+1} = x_k - p\frac{f(x_k)}{f'(x_k)}.$$

konvergiert unter der Voraussetzung $\frac{M}{m}\rho < 1$ dagegen wieder lokal quadratisch

$$|x_k - z| < \frac{M}{pm}|x_{k-1} - z|^2.$$

Beweis. Wir betrachten allgemein die Iteration

$$x_{k+1} = x_k - \omega\frac{f(x_k)}{f'(x_k)}$$

mit den Spezialfällen $\omega = 1$ und $\omega = p$. Zunächt gilt

$$|x_{k+1} - z| = \left|x_k - z - \omega\frac{f(x_k)}{f'(x_k)}\right| = \left|1 - \frac{\omega f(x_k)}{(x_k - z)f'(x_k)}\right||x_k - z|. \tag{4.18}$$

Taylorentwicklungen von f und f' ergeben für gewisse $\xi_1, \xi_2 \in [x_k, z]$

$$f(x_k) = \sum_{i=0}^{p-1} \underbrace{\frac{f^{(i)}(z)}{i!}}_{=0}(x_k - z)^i + \frac{f^{(p)}(\xi_1)}{p!}(x_k - z)^p = \frac{f^{(p)}(\xi_1)}{p!}(x_k - z)^p$$

$$f'(x_k) = \sum_{i=1}^{p-1} \underbrace{\frac{f^{(i)}(z)}{(i-1)!}}_{=0}(x_k - z)^{i-1} + \frac{f^{(p)}(\xi_2)}{(p-1)!}(x_k - z)^{p-1} = \frac{f^{(p)}(\xi_2)}{(p-1)!}(x_k - z)^{p-1}. \tag{4.19}$$

Wir entwickeln den letzten Term in der ersten Zeile weiter um ξ_2 mit einem $\xi_3 \in [x_k, z]$

$$f(x_k) = \frac{f^{(p)}(\xi_1)}{p!}(x_k - z)^p = \frac{f^{(p)}(\xi_2)}{p!}(x_k - z)^p + \frac{f^{(p+1)}(\xi_3)}{p!}(x_k - z)^p(\xi_2 - \xi_1). \tag{4.20}$$

Wir setzen (4.19) und (4.20) in (4.18) ein:

$$|x_{k+1} - z| = \Big|1 - \frac{\omega f(x_k)}{(x_k - z)f'(x_k)}\Big||x_k - z| = \Big|1 - \frac{\omega}{p} + \frac{\omega f^{(p+1)}(\xi_3)(\xi_1 - \xi_2)}{p f^{(p)}(\xi_2)}\Big||x_k - z|. \tag{4.21}$$

Im Fall $\omega = p$ folgt

$$|x_{k+1} - z| \leq \frac{M}{m}|\xi_1 - \xi_2||x_k - z|.$$

Unter der Voraussetzung, dass

$$\frac{M}{m}\rho < 1$$

folgt induktiv, dass $x_{k+1} \in K_\rho(z)$ und außerdem

$$|x_{k+1} - z| \leq \frac{M}{m}|x_k - z|^2.$$

In Falle $\omega = 1$ gilt nach (Induktions-)Voraussetzung

$$\Big|\frac{f^{(p+1)}(\xi_3)(\xi_1 - \xi_2)}{p f^{(p)}(\xi_2)}\Big| < \frac{M}{pm}|x_k - z| < \frac{1}{p}$$

und damit

$$\Big|1 - \frac{1}{p} + \frac{f^{(p+1)}(\xi_3)(\xi_1 - \xi_2)}{p f^{(p)}(\xi_2)}\Big| < 1.$$

(4.21) impliziert dann $x_{k+1} \in K_\rho(z)$ und (lokal) lineare Konvergenz. □

4.4.3 Vereinfachtes Newton-Verfahren

Der teuerste Schritt bei der Durchführung des Newton-Verfahrens ist oft die Berechnung der Ableitung $f'(x_k)$. Dies ist insbesondere in mehreren Dimensionen der Fall (siehe unten). In 1d stelle man sich Funktionen f vor, die nur implizit definiert sind.

In diesem Fall wird die Ableitung $f'(x_k)$ häufig nicht in jedem Newton-Schritt neu berechnet sondern es wird stattdessen auf in vorherigen Iterationen berechneten Ableitungen zurückgegriffen

$$f'(x_k) \approx f'(x_l) \qquad \text{für } l < k.$$

Es kann sogar gezeigt werden, dass die Ableitungen an beliebigen Stellen $y \in K_\rho(z)$ ausgewertet werden können, wobei sich die Konvergenzordnung in diesem Fall auf 1 reduziert. Wir betrachten das folgende *vereinfachte* Newtonverfahren, bei dem die Ableitung nur im ersten Schritt berechnet wird.

Eingabe: Startwert $x_0 \in I$,

Solange $\frac{1}{m}\frac{|f(x_k)|}{|x_k|} > \text{TOL}$:

Setze $x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_0)}$.

Ausgabe: Iterierte x_k

Satz 4.6. *Die Funktion $f \in C^2[a,b]$ habe eine Nullstelle $z \in (a,b)$ und wir nehmen an, dass*

$$m := \min_{x\in[a,b]} |f'(x)| > 0.$$

Außerdem sei $M := \max_{x\in[a,b]} |f''(x)|$ und $\rho > 0$ sei so gewählt, dass

$$\rho < \frac{m}{2M}, \quad K_\rho := \{x \in \mathbb{R}, |x - z| \le \rho\} \subset [a,b].$$

Dann bleiben die Newton-Iterierten x_k für jeden Startwert $x_0 \in K_\rho(z)$ in $K_\rho(z)$ und konvergieren gegen die Nullstelle z. Es gilt die a priori Fehlerabschätzung

$$|x_k - z| \le q|x_{k-1} - z| \le \frac{m}{4M}q^k, \quad k \in \mathbb{N}$$

mit $q := \frac{2M}{m}\rho < 1$.

Beweis. Übung. □

4.4.4 Sekantenverfahren

Das Sekantenverfahren kommt sogar ganz ohne Berechnung von Ableitungen aus. Statt wie beim Newton-Verfahren die Tangente $T(x)$ als lokale Approximation von f um den Punkt x zu betrachten, wählen wir hier die Sekante, d.h. die Gerade durch die Punkte $(x_k, f(x_k))$ und $(x_{k-1}, f(x_{k-1}))$ (siehe Abb. 4.4)

$$S(x) = f(x_k) + (x - x_k)\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}.$$

Die Nullstelle der Sekante zu gegebenen x_k und x_{k-1} liefert die nächste Itererierte x_{k+1} des Sekantenverfahrens.

Eingabe: Startwerte $x_0, x_1 \in I, x_0 \neq x_1$

Für $k = 1, 2, \ldots,$ setze $x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$ (4.22)

Ausgabe: Iterierte x_k

Alternativ zur geometrischen Motivation kann man das Sekantenverfahren auch als eine Variante des Newton-Verfahrens sehen, bei dem die Ableitung durch einen Differenzenquotient approximiert wird

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}.$$

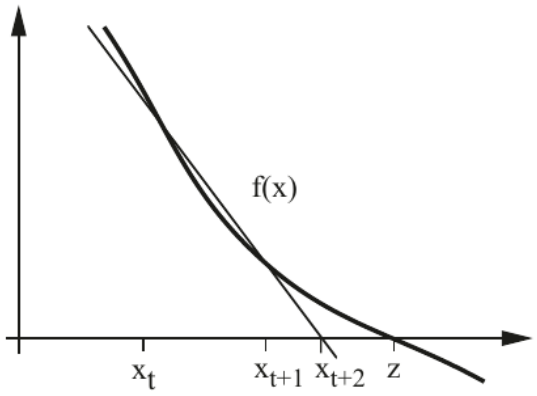

Abbildung 4.4: Graphische Darstellung des Sekantenverfahrens (Quelle: [4])

Satz 4.7. *Die Funktion $f \in C^2[a,b]$ habe eine Nullstelle $z \in (a,b)$ und wir nehmen an, dass*

$$m := \min_{x\in[a,b]} |f'(x)| > 0.$$

Außerdem sei $M := \max_{x\in[a,b]} |f''(x)|$ und $\rho > 0$ sei so gewählt, dass

$$\rho < \frac{2m}{M}, \quad K_\rho := \{x \in \mathbb{R}, |x-z| \le \rho\} \subset [a,b].$$

Dann bleiben die Iterierten x_k des Sekantenverfahrens (4.22) für beliebige Startwerte $x_0, x_1 \in K_\rho(z)$ in $K_\rho(z)$ und konvergieren gegen die Nullstelle z. Es gilt die a priori Fehlerabschätzung

$$|x_k - z| \le \frac{2m}{M} q^{\gamma_k}, \quad k \in \mathbb{N} \tag{4.23}$$

wobei $q = \frac{M}{2m}\rho < 1$ und $(\gamma_k)_{k\in\mathbb{N}}$ die Folge der Fibonacci-Zahlen. Asymptotisch ergibt sich die Konvergenzordnung $p \approx 1.618$, da

$$\lim_{k\to\infty} \gamma_k \approx 0.723 \cdot (1.618)^k.$$

Beweis. Siehe z.B. [5], Satz 6.23. □

Bei der Sekantenmethode ist pro Iterationsschritt nur eine (neue) Funktionsauswertung $f(x_k)$ und keine Berechnung von Ableitungen notwendig, weshalb ein Iterationsschritt in der Regel deutlich günstiger ist als beim Newton-Verfahren. Zusammen mit der Konvergenzordnung von 1.681 könnte man also vermuten, dass das Verfahren eine hocheffiziente Alternative zum Newton-Verfahren darstellt. Allerdings besteht die Gefahr von Auslöschung, insbesondere wenn $f(x_k)$ monoton (nicht alternierend) gegen $f(z) = 0$ konvergiert, da $f(x_k) \approx f(x_{k-1})$. Deshalb wird das Sekantenverfahren in der Praxis eher selten eingesetzt.

4.5 Fixpunktverfahren

Das Newton-Verfahren und die in Abschnitt 4.4 definierten Varianten lassen sich alle als Fixpunktiterationen in folgender Form schreiben

$$x_k = g(x_{k-1}), \quad k = 1, 2, ..., \tag{4.24}$$

wobei genau dann $z = g(z)$ gelten soll, wenn z eine Nullstelle von f ist.

Das Newton-Verfahren fällt in diese Kategorie, da unter den obigen Voraussetzungen an f' gilt

$$g(z) := z - \frac{f(z)}{f'(z)} = z \qquad \Leftrightarrow \quad f(z) = 0.$$

Für eine Nullstelle z von f gilt weiter

$$g'(z) = 1 - \frac{f'(z)}{f'(z)} + \frac{f(z)f''(z)}{(f'(z))^2} = 0. \tag{4.25}$$

Das Verhalten der Ableitungen an der Nullstelle z kann benutzt werden, um die Konvergenzordnung einer Fixpunktiteration zu bestimmen.

Satz 4.8. *Sei $g \in C^p[a, b]$ mit $p \in \mathbb{N}, p \geq 2$ und sei $z \in [a, b]$ ein Fixpunkt von g. Eine Fixpunktiteration der Form* (4.24) *hat mindestens die Ordnung p, wenn gilt*

$$g'(z) = ... = g^{(p-1)}(z) = 0. \tag{4.26}$$

Gilt für die Fixpunktiteration umgekehrt

$$|x_k - z| \leq c|x_{k-1} - z|^p \qquad \text{für alle } k \in \mathbb{N}, \tag{4.27}$$

so folgt daraus (4.26).

Beweis. "⇐" Sei zunächst $g'(z) = ... = g^{(p-1)}(z) = 0$. Taylor-Entwicklung ergibt

$$x_k - z = g(x_{k-1}) - g(z) = \sum_{k=1}^{p-1} \frac{1}{k!} \underbrace{g^{(k)}(z)}_{=0} (x_{k-1} - z)^k + \frac{g^{(p)}(\xi_k)}{p!}(x_{k-1} - z)^p, \quad \xi_k \in [x_{k-1}, z].$$

Es folgt (4.27), d.h. Konvergenz der Ordnung p im Sinne von (4.10).

"⇒" Es gelte nun umgekehrt (4.27) für $p \geq 2$ und beliebiges $x_0 \in [a, b]$. Wir zeigen induktiv für $m = 1, \ldots, p-1$, dass $g^{(m)}(z) = 0$.

Induktionsanfang: Sei zunächst $m = 1$. Wir machen die Widerspruchsannahme $g'(z) \neq 0$. Dann gilt auch $g'(\xi) \neq 0$ in einer Umgebung $K_\rho(z)$. Aus (4.27) folgt nach einer möglichen Verkleinerung von ρ außerdem die Konvergenz $x_k \to z$ für alle Startwerte $x_0 \in K_\rho(z)$. Taylor-Entwicklung (bzw. der Mittelwertsatz der Differentialrechnung) ergibt

$$x_k - z = g(x_{k-1}) - g(z) = g'(\xi_k)(x_{k-1} - z), \qquad \xi_k \in [x_{k-1}, z]. \tag{4.28}$$

Wegen $g'(\xi_k) \neq 0$ in $K_\rho(z)$ für $k \in \mathbb{N}$ wird $x_k - z$ niemals Null (wenn $x_0 - z \neq 0$). Es folgt im Widerspruch zur Annahme

$$|g'(z)| = \lim_{k\to\infty} |g'(\xi_k)| = \lim_{k\to\infty} \frac{|x_k - z|}{|x_{k-1} - z|} \leq c \lim_{k\to\infty} |x_{k-1} - z|^{p-1} = 0.$$

Induktionsschritt: Wir nehmen nun an, dass dass $g'(z) = ... = g^{(m-1)}(z) = 0$ für ein $m < p$ und machen wieder die Widerspruchsannahme, dass $g^{(m)}(z) \neq 0$. Mittels Taylor-Entwicklung erhalten wir für $x_0 \in K_\rho(z)$ für genügend kleines ρ

$$x_k - z = g(x_{k-1}) - g(z) = \frac{g^{(m)}(\xi_k)(x_{k-1} - z)^m}{m!}, \qquad \xi_k \in [x_{k-1}, z]$$

und damit im Widerspruch zur Annahme

$$|g^{(m)}(z)| = \lim_{k\to\infty} |g^{(m)}(\xi_k)| = \lim_{k\to\infty} m! \frac{|x_k - z|}{|x_{k-1} - z|^m} \leq c(m!) \lim_{k\to\infty} |x_{k-1} - z|^{p-m} = 0.$$

□

Die Taylor-Entwicklung (4.28)

$$x_k - z = g'(\xi_k)(x_{k-1} - z), \qquad \xi_k \in [x_{k-1}, z]. \tag{4.29}$$

gibt noch ein Kriterium für (lokal lineare) Konvergenz:

Lemma 4.9. *Sei $g \in C^1[a, b]$ und $z \in [a, b]$ ein Fixpunkt von g. Gilt $|g'(z)| < 1$, so folgt lokal (in einer genügend kleinen Umgebung von z) lineare Konvergenz der Fixpunktiteration gegen z*

$$|x_k - z| < q|x_{k-1} - z|, \quad \text{mit } q < 1.$$

Im Falle $|g'(z)| > 1$ konvergiert die Fixpunktiteration nicht gegen z.

Beweis. Gilt $|g'(z)| < 1$, so ist auch $|g'(\xi_k)| < 1$ in einer genügend kleinen Umgebung $K_\rho(z)$. Aus (4.29) folgt lokal lineare Konvergenz, da

$$|x_k - z| < q|x_{k-1} - z|, \quad \text{mit } q < 1.$$

Analog folgt, dass im Falle $|g'(z)| > 1$ keine Konvergenz gegen z vorliegt. □

Mithilfe der Resultate aus Theorem 4.8 können wir nun weitere Fixpunktiterationen zur Lösung des Nullstellenproblems $f(z) = 0$ konstruieren.

Beispiel 1: Für die einfache Fixpunktiteration

$$x_k = g(x_{k-1}) := x_{k-1} + f(x_{k-1})$$

gilt

$$g'(z) = 1 + f'(z).$$

Die Konvergenz der Fixpunktiteration ist abhängig von der Funktion f. Für $f'(z) \in (-2, 0)$ liegt (lokal lineare) Konvergenz vor, nur im Falle $f'(z) = -1$ ist diese lokal quadratisch.

Für die Funktion $f(x) = -\sin(\frac{1}{2}x)$ lautet die Iterationsvorschrift für gegebenen Startwert x_0 beispielsweise

$$x_k = x_{k-1} - \sin(\frac{1}{2}x_{k-1}), \qquad k \in \mathbb{N}.$$

Wegen $f'(0) = -\frac{1}{2}\cos(0) = -\frac{1}{2}$ konvergiert die Fixpunktiteration lokal linear.

Beispiel 2: Das Newton-Verfahren $g(x) := x - \frac{f(x)}{f'(x)}$ konvergiert (lokal) quadratisch, da

$$g'(z) = 0 \quad \text{(siehe (4.25))}, \text{ aber im Allgemeinen } g''(z) \neq 0.$$

Das vereinfachte Newton-Verfahren in der Form $g(x) := x - \frac{f(x)}{f'(x_0)}$ konvergiert dagegen im Allgemeinen nicht lokal quadratisch, da

$$g'(z) = 1 - \frac{f'(z)}{f'(x_0)} \neq 0.$$

In einer genügend kleinen Umgebung $K_\rho(z)$ gilt allerdings $\frac{f'(z)}{f'(x_0)} \in (0, 2)$, so dass (lokal) lineare Konvergenz sichergestellt ist.

Konstruktion von Fixpunktverfahren höherer Ordnung Zur Konstruktion eines Fixpunktverfahrens der Ordnung 3 machen wir basierend auf dem Newton-Verfahren den Ansatz

$$g(x) = x - r(x) + s(x)r^2(x), \quad \text{mit } r(x) = \frac{f(x)}{f'(x)}.$$

Aufgrund von $r(z) = 0$ und $r'(z) = 1$ gilt

$$g'(z) = 1 - r'(z) + 2s(z)r(z)r'(z) + s'(z)r(z)^2 = 0.$$

Für die Wahl

$$s(x) = \frac{r''(x)}{2r'(x)^2}$$

kann man ferner zeigen, dass $g''(z) = 0$. Nach Konstruktion hat dieses Fixpunktverfahren also die Ordnung 3

$$|x_k - z| < c|x_{k-1} - z|^3.$$

Allerdings müssen bei diesem Verfahren in jedem Iterationsschritt neben den Werten $f(x_k), f'(x_k)$ auch die höheren Ableitungen $f''(x_k)$ und $f'''(x_k)$ ausgewertet werden. Ein Schritt dieses Verfahrens ist daher in der Regel aufwändiger als 2 Newtonschritte (dort sind $f(x_k)$, $f'(x_k)$, $f(x_{k-1})$ und $f'(x_{k-1})$ auszuwerten).

Zum Vergleich mit dem Newton-Verfahren sei y_k die Folge der Newton-Iterierten und $w_k := y_{2k}$ die Folge, welche zwei Newtonschritte kombiniert. Dann erhalten wir für die Folge $(w_k)_{k\in\mathbb{N}}$

$$|w_k - z| = |y_{2k} - z| \;\leq\; q|y_{2k-1} - z|^2 \;\leq\; q^3|y_{2k-2} - z|^4 \;=\; q^3|w_{k-1} - z|^4.$$

D.h. wir erhalten ein Verfahren vierter Ordnung. Aus diesem Grund sind Iterationsverfahren höherer Ordnung im Allgemeinen kaum relevant für die Nullstellensuche.

4.6 Nullstellensuche im $\mathbb{R}^n$

Für den Rest dieses Kapitels werden wir uns der Nullstellensuche im $\mathbb{R}^n$ widmen. Wir beschränken uns dabei auf Funktionen

$$f : \mathbb{R}^n \to \mathbb{R}^n.$$

Numerisch interessant ist dabei insbesondere der Fall, dass n sehr groß ist. Besonders bei numerischen Methoden für Differentialgleichungen sind in der Praxis oft lineare oder nichtlineare Gleichungssysteme mit $n > 10^6$ zu lösen. Selbst auf den heutigen Supercomputern sind effiziente numerische Methoden essentiell, um Näherungslösungen mit einer gewissen Genauigkeit auszurechnen.

Wir beschäftigen uns wieder mit *Fixpunktverfahren* der Form

$$x_k = g(x_{k-1}), \quad k \in \mathbb{N}.$$

Zur Analyse von numerischen Methoden für die Nullstellensuche im $\mathbb{R}^n$ ist der **Banachsche Fixpunktsatz** wichtig, welcher in der Regel Gegenstand der Analysisgrundvorlesungen ist. Im folgenden bezeichne $\|\cdot\|$ eine beliebige Vektornorm im $\mathbb{R}^n$ und $|||\cdot|||$ eine mit $\|\cdot\|$ verträgliche Matrixnorm (d.h. es gilt $\|Ax\| \leq |||A||| \, \|x\|$ für $A \in \mathbb{R}^{n\times n}$, $x \in \mathbb{R}^n$).

Satz 4.10. *(Banachscher Fixpunksatz) Es sei $D \subset \mathbb{R}^n$ eine nichtleere und abgeschlossene Teilmenge und $g : D \to D$ eine Selbstabbildung und Kontraktion, d.h. eine Lipschitz-stetige Abbildung mit Lipschitzkonstante $q < 1$*

$$\|g(x) - g(y)\| \leq q\|x - y\|, \quad x, y \in D. \tag{4.30}$$

Dann besitzt g genau einen Fixpunkt $z \in D$ und die Folge $(x_k)_{k\geq 0}$ der Fixpunktiteration (4.24) konvergiert für jeden Startpunkt $x_0 \in D$ gegen z

$$x_k \to z \quad (k \to \infty).$$

Außerdem gilt die a priori Abschätzung (lineare Konvergenz)

$$\|x_k - z\| \leq \frac{q^k}{1-q}\|x_1 - x_0\|.$$

Beweis. [2], Satz 5.16. □

Zur Konstruktion und Analyse von Fixpunktverfahren werden wir wieder Ableitungen benötigen. Für Funktionen $f : \mathbb{R}^n \to \mathbb{R}^n$ sind die ersten Ableitungen in der Jacobi-Matrix

$$J_f(x) := (\partial_j f_i)_{i,j=1}^n = \begin{pmatrix} \partial_1 f_1 & \partial_2 f_1 & \dots & \partial_n f_1 \\ \partial_1 f_2 & \partial_2 f_2 & \dots & \partial_n f_2 \\ \vdots & \vdots & \ddots & \vdots \\ \partial_1 f_n & \partial_2 f_n & \dots & \partial_n f_n \end{pmatrix}$$

zusammengefasst. Zur Berechnung von J_f ist die Berechnung von n^2 partiellen Ableitungen notwendig.

Zur Anwendung des Banachschen Fixpunktsatzes müssen die Eigenschaften Selbstabbildung und Kontraktion nachgewiesen werde. Zur Überprüfung dieser Eigenschaften ist der folgende Schrankensatz nützlich.

Lemma 4.11. *(Schrankensatz) Die Funktion $g : D \to \mathbb{R}^n$ sei stetig differenzierbar auf einer konvexen Menge $D \subset \mathbb{R}^n$. Dann gilt für $x, y \in D$*

$$\|g(x) - g(y)\| \leq \sup_{\xi \in D} |||J_g(\xi)||| \, \|x - y\|. \tag{4.31}$$

Falls $q := \sup_{\xi \in D} |||J_g(\xi)||| < 1$, so ist g eine Kontraktion auf D.

Beweis. Die Beziehung (4.31) folgt direkt mithilfe des Mittelwertsatzes der Differentialrechnung. Nach diesem gibt es zu $x, y \in D$ ein $\xi = x + s(y - x)$, $s \in [0, 1]$ auf der Verbindungsstrecke zwischen x und y, so dass

$$g(x) - g(y) = J_g(\xi)(x - y)$$

Da D als konvex angenommen wurde, gilt $\xi \in D$. (4.31) folgt nach Normierung und aufgrund von $\|Ax\| \leq |||A||| \, \|x\|$. □

Gilt $|||J_g(z)||| < 1$, so kann dieses Lemma verwendet werden, um lokale Konvergenz der Fixpunktiteration zu zeigen.

Lemma 4.12. *Sei $g : D \to \mathbb{R}^n$ stetig differenzierbar auf einer offenen Menge $D \subset \mathbb{R}^n$. Zu jedem Fixpunkt $z \in D$ mit $|||J_g(z)||| < 1$ gibt es eine (abgeschlossene) Umgebung*

$$K_\rho(z) := \{x \in \mathbb{R}^n, \, \|x - z\| \leq \rho\} \subset D, \quad \rho > 0,$$

auf der g eine Selbstabbildung und Kontraktion ist. Gilt $|||J_g(x)||| < 1$ auf einer Kugelumgebung $K_r(z)$, so kann $\rho = r$ gewählt werden. Für Startwerte $x_0 \in K_\rho(z)$ konvergiert die zugehörige Fixpunktiteration gegen z.

Beweis. Aufgrund der Stetigkeit von J_g gibt es eine (abgeschlossene) Umgebung $K_\rho(z)$, so dass

$$q := \sup_{\xi \in K_\rho(z)} |||J_g(\xi)||| < 1.$$

Nach Lemma 4.11 ist g eine Kontraktion auf $K_\rho(z)$. Wir zeigen die Selbstabbildungseigenschaft. Sei dazu $x \in K_\rho(z)$. Mit der Fixpunkteigenschaft von z und der Kontraktionseigenschaft gilt

$$\|g(x) - z\| = \|g(x) - g(z)\| \leq q\|x - z\| \leq q\rho < \rho,$$

d.h. $g(x) \in K_\rho(z)$. Der Banachsche Fixpunktsatz sichert die Konvergenz der Fixpunktiteration. □

Beispiel: Fixpunktverfahren für nichtlineare Gleichungssysteme Als Fixpunktiteration zur Lösung des nichtlinearen Gleichungssystems

$$f(x) = 0, \quad f : \mathbb{R}^n \to \mathbb{R}^n$$

machen wir mit einer regulären Matrix $C \in \mathbb{R}^{n \times n}$ den Ansatz

$$x_0 \in \mathbb{R}^n, \quad x_k = x_{k-1} + Cf(x_{k-1}) \quad (k = 1, 2, ...). \tag{4.32}$$

Ist f stetig differenzierbar und ist C so gewählt, dass

$$q := \sup_{x \in K_\rho(z)} |||I + CJ_f(x)||| < 1, \tag{4.33}$$

so konvergiert die Fixpunktiteration (4.32) nach Lemma 4.12 für Startwerte $x_0 \in K_\rho(z)$. Die Beziehung (4.33) legt die Wahl $C = -J_f(x_{k-1})^{-1}$ nahe, welche weiter unten zum Newton-Verfahren im $\mathbb{R}^n$ führen wird.

4.7 Exkurs: Iterative Verfahren zur Lösung linearer Gleichungssysteme

Der Ansatz (4.32) kann auch zur Konstruktion von *iterativen Verfahren* zur Lösung von linearen Gleichungssystemen der Form

$$Ax = b, \qquad A \in \mathbb{R}^{n\times n},\, b \in \mathbb{R}^n$$

mit einer regulären Matrix A verwendet werden. Diese sind insbesondere für sehr große lineare Gleichungssysteme relevant, da das Gaußsche Eliminationsverfahren (Kapitel 2) im Fall von voll besetzten Matrizen $\mathcal{O}(n^3)$ arithmetische Operationen benötigt und sehr teuer sein kann (Im Fall $n = \mathcal{O}(10^6)$ wären $\mathcal{O}(10^{18})$ Operationen notwendig!). Die Iteration (4.32) angewendet auf die Funktion $f(x) = b - Ax$ lautet

$$x_k = x_{k-1} + C(b - Ax_{k-1}) = (I - CA)x_{k-1} + Cb \quad (k = 1, 2, ...). \tag{4.34}$$

Die Jacobi-Matrix der Verfahrensfunktion f lautet in diesem Fall $J_f = -A$ unabhängig von x. Die Matrix C sollte wieder so gewählt werden, dass (vergleiche (4.33))

$$q := |||I - CA||| < 1,$$

d.h. $C \approx A^{-1}$. Die Berechnung von A^{-1} selbst würde wie das Gaußsche Eliminationsverfahren $\mathcal{O}(n^3)$ arithmetische Operationen benötigen und kommt daher nicht in Frage.

Jacobi- und Gauß-Seidel-Verfahren Stattdessen splitten wir die Matrix A additiv in ihren Diagonalteil D, eine linke untere Dreiecksmatrix L und eine rechte obere Dreiecksmatrix R auf:

$$A = L + D + R \quad \text{mit } A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \qquad D = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & a_{nn} \end{pmatrix},$$

$$L = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{n1} & \cdots & a_{n(n-1)} & 0 \end{pmatrix}, \qquad R = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{(n-1)n} \\ 0 & \cdots & 0 & 0 \end{pmatrix}.$$

Zwei wichtige *iterative Verfahren* zur Lösung von linearen Gleichungssystemen sind das *Jacobi-* und das *Gauß-Seidel*-Verfahren. Beim Jacobi-Verfahren setzt man

$$C = D^{-1},$$

beim Gauß-Seidel-Verfahren

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

Eine erste Voraussetzung für die Durchführbarkeit beider Verfahren ist, dass D bzw. $D + L$ regulär ist, d.h. keine Nullelemente auf der Diagonalen enthält.

Algorithmen und Aufwandsanalyse Das Jacobi-Verfahren nimmt folgende Gestalt an:

Jacobi-Verfahren

Eingabe: Startwert $x_0 \in \mathbb{R}^n$

(i) Berechne Residuum $r_0 = b - Ax_0$, $k = 1$

Solange $\|r_{k-1}\| >$TOL (oder $\|x_k - x_{k-1}\| >$TOL)

(ii) Update $x_k = x_{k-1} + D^{-1} r_{k-1}$

(iii) Berechne Residuum $r_k = b - Ax_k$, $k \leftarrow k + 1$

Ausgabe: Iterierte x_{k-1}

Die Multiplikation mit der Inversen der Diagonalmatrix in (ii) kann sehr effizient mit n Divisionen und n Additionen ausgeführt werden:

$$\begin{pmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix} = \begin{pmatrix} x_1^{(k-1)} + r_1^{(k-1)}/a_{11} \\ \vdots \\ x_n^{(k-1)} + r_n^{(k-1)}/a_{nn} \end{pmatrix}.$$

Die einzig teuren Schritte sind damit die Matrix-Vektor-Multiplikation Ax_k in (i) und (iii), welche n^2 Multiplikationen und n^2 Additionen benötigt

$$Ax_k = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^n a_{1j} x_j^{(k)} \\ \sum_{j=1}^n a_{2j} x_j^{(k)} \\ \vdots \\ \sum_{j=1}^n a_{nj} x_j^{(k)} \end{pmatrix}$$

Es ergibt sich ein Gesamtaufwand von $n^2 + \mathcal{O}(n)$ Multiplikationen/Divisionen und $n^2 + \mathcal{O}(n)$ Additionen/Subtraktionen. Die Anzahl der benötigten Iterationen c_{it} bis zum Erreichen der Toleranz ist problemabhängig. Bei einer guten Konditionierung der Systemmatrix $\kappa(A) = \mathcal{O}(1)$ (unabhängig von n) ist c_{it} aber in der Regel unabhängig von n. Die Gesamtzahl der benötigten Multiplikationen/Divisionen beträgt dann $c_{it} n^2 + \mathcal{O}(n)$ (im Gegensatz zu $\mathcal{O}(n^3)$ für das Gaußsche Eliminationsverfahren).

Alternative Formulierung Zur Implementierung des Jacobie-Verfahrens wird häufig ein alternativer Algorithmus verwendet. Schritt (ii) im obigen Algorithmus

$$x_k = x_{k-1} + D^{-1}(b - Ax_{k-1})$$

kann alternativ auch geschrieben werden als

$$Dx_k = b + (D - A)x_{k-1} = b - (R + L)x_{k-1}$$

bzw. in Komponentenschreibweise für $i = 1, ..., n$

$$\underline{a}_{ii}x_i^{(k)} = b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k-1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}.$$

Nach Teilen durch a_{ii} ergibt sich die Iterationsvorschrift für $i = 1, ..., n$:

$$x_i^{(k)} = \underline{a}_{ii}^{-1}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k-1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right)$$

Gauß-Seidel-Verfahren Für das Gauß-Seidel-Verfahren formuliert man Schritt (ii) um, da die Berechnung der Inversen $(D + L)^{-1}$ teuer wäre. Nach (4.34) lautet die Iterationsvorschrift

$$x_k = (I - (D + L)^{-1}A)x_{k-1} + (D + L)^{-1}b.$$

Die erste Matrix lässt sich in diesem Fall vereinfachen zu

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

Nun multipliziert man (4.34) von links mit $(D+L)$ und löst ein lineares Gleichungssystem (LGS) durch Vorwärtseinsetzen

$$x_k = -(D + L)^{-1}Rx_{k-1} + (D + L)^{-1}b \quad \Leftrightarrow \quad (D + L)x_k = -Rx_{k-1} + b.$$

Gauß-Seidel-Verfahren

Eingabe: Startwert $x_0 \in \mathbb{R}^n$

(i) Löse das LGS $(D + L)x_1 = -Rx_0 + b$, $k = 1$

Solange $\|x_k - x_{k-1}\| >$TOL

(ii) Löse das LGS $(D + L)x_k = -Rx_{k-1} + b$, $k \leftarrow k + 1$

Ausgabe: Iterierte x_k

Schritt (i) und (ii) benötigen je $\frac{n^2}{2} + \mathcal{O}(n)$ Multiplikationen/Divisionen zur Berechnung der rechten Seite und weitere $\frac{n^2}{2} + \mathcal{O}(n)$ Multiplikationen/Divisionen zur Lösung des LGS durch Vorwärtseinsetzen (siehe Kapitel 2). Zusammen beträgt der Aufwand pro Schritt beim Gauß-Seidel-Verfahren also $n^2 + \mathcal{O}(n)$ Mult./Div, wie beim Jacobi-Verfahren. Wählt man das Abbruchkriterium allerdings basierend auf $\|r_k\| = \|b - Ax_k\|$, wären weitere $n^2 + \mathcal{O}(n)$ Mult./Div. notwendig. Allerdings sind beim Gauß-Seidel-Verfahren im Vergleich zum Jacobi-Verfahren meist weniger Iteration c_{it} bis zum Erreichen der Toleranz TOL notwendig.

Alternative Formulierung Schritt (ii)

$$(D + L)x_k = -Rx_{k-1} + b$$

schreibt sich komponentenweise für $i = 1, ..., n$ als

$$\sum_{j=1}^{i} a_{ij} x_j^{(k)} = b_i - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}$$

bzw.

$$a_{ii} x_i^{(k)} = b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}$$

Auch diese Relation kann sukzessive zur Berechnung von $x_i^{(k)}$ für $i = 1, ..., n$ genutzt werden, da $x_j^{(k)}$ rechts für $j < i$ dann schon berechnet ist.

Der Vorteil dieser Formulierung ist, dass bei dieser Formulierung kein lineares Gleichungssystem zu lösen ist. Der Aufwand liegt jedoch auch hier bei je $n^2 + \mathcal{O}(n)$ Multiplikationen und Additionen.

4.7.1 Konvergenzresultate

Konvergenz der Iteration ist nach obigen Überlegungen für beliebige Startwerte $x_0 \in \mathbb{R}^n$ sichergestellt, wenn

$$|||I - CA||| < 1.$$

Dann gilt nämlich (Wiederholung)

$$\begin{aligned} \|x_k - z\| = \|g(x_{k-1}) - g(z)\| &\leq |||I - CA||| \, \|x_{k-1} - z\| \\ &\leq |||I - CA|||^k \, \|x_0 - z\| \to 0 \quad (k \to \infty). \end{aligned}$$

Aufgrund von Normäquivalenz im $\mathbb{R}^n$ genügt es dies für eine beliebige Norm $\|\cdot\|_*$ nachzuweisen. Dann folgt nämlich für eine beliebige andere Norm $\|\cdot\|$

$$\|x_k - z\| \leq c\|x_k - z\|_* \to 0 \quad (k \to \infty).$$

Zum Nachweis einer Schranke für $|||I - CA|||$ kann der Spektralradius einer Matrix spr(A) einer Matrix $A \in \mathbb{R}^{n \times n}$ nützlich sein. Wir definieren

$$\text{spr(A)} := \max\{|\lambda|, \, \lambda \in \mathbb{C} \text{ ist Eigenwert von } A\}.$$

Das folgende Lemma zeigt, dass es genügt den Spektralradius von $I - CA$ zu betrachten:

Lemma 4.13. *Zu jeder Matrix $A \in \mathbb{R}^{n \times n}$ und zu jedem $\epsilon > 0$ gibt es eine von einer Vektornorm $\|\cdot\|_\epsilon$ erzeugte Matrixnorm $|||\cdot|||_\epsilon$, so dass*

$$spr(A) \leq |||A|||_\epsilon \leq spr(A) + \epsilon.$$

Beweis. Siehe [4]. □

Konvergenz der Fixpunktiteration

$$x_k = x_{k-1} + C(b - Ax_{k-1})$$

für beliebige Startwerte x_0 liegt also genau dann vor, wenn

$$\text{spr}(I - CA) < 1. \tag{4.35}$$

Zwei wichtige Kriterien zum Nachweis von (4.35) basieren auf dem Begriff der Diagonaldominanz von Matrizen.

Definition 4.14. *Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt stark diagonaldominant, wenn*

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}| \quad \textit{für alle } i = 1, \ldots, n.$$

Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt schwach diagonaldominant, wenn

$$|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}| \quad \textit{für alle } i = 1, \ldots, n.$$

und

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}| \quad \textit{für mindestens ein } i \in \{1, \ldots, n\}.$$

Für strikt diagonaldominante Matrizen kann Konvergenz für Jacobi- und Gauß-Seidel-Verfahren gezeigt werden:

Lemma 4.15. *Sei $A \in \mathbb{R}^{n\times n}$ eine strikt diagonaldominante Matrix. Dann gilt sowohl für das Jacobi-Verfahren ($C = D^{-1}$) als auch für das Gauß-Seidel-Verfahren ($C = (D + L)^{-1}$)*

$$spr(I - CA) < 1,$$

d.h. beide Verfahren konvergieren für beliebige Startwerte $x_0 \in \mathbb{R}^n$.

Beweis. Übung □

In Anwendungen liegt häufig nur schwache Diagonaldominanz vor. Diese ist alleine noch nicht hinreichend für Konvergenz der beiden Verfahren. Wir benötigen noch den Begriff der *Irreduzibilität*:

Definition 4.16 (Irreduzibilität)**.** *Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt irreduzibel, wenn es zu jedem Indexpaar (i, j) einen Pfad*

$$i_0, i_1, \ldots i_m$$

gibt mit $i = i_0, j = i_m$ und Matrixelementen

$$a_{i_k, i_{k+1}} \neq 0, \qquad k = 0, \ldots, m - 1.$$

Wir betrachten ein Beispiel zur Illustration des Begriffs. Die Matrix

$$A = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{n\times n}$$

ist schwach diagonaldominant und irreduzibel. Für das Indexpaar $(i, j) = (1, m)$ wählt man zum Beispiel den Pfad:

$$i_0 = 1, \quad i_1 = 2, \quad i_2 = 3, \ldots, \quad i_{m-1} = m.$$

Analog kann man beliebige andere Indexpaare (i, j) behandeln.

Anmerkung: Setzt man in diesem Beispiel dagegen einen Nebendiagonaleintrag auf Null, so ist die Matrix nicht mehr irreduzibel.

Das folgende Lemma sichert Konvergenz für irreduzible, schwach diagonaldominante Matrizen:

Lemma 4.17. *Sei $A \in \mathbb{R}^{n\times n}$ eine irreduzible und schwach diagonaldominante Matrix. Dann gilt sowohl für das Jacobi-Verfahren ($C = D^{-1}$) als auch für das Gauß-Seidel-Verfahren ($C = (D + L)^{-1}$)*

$$spr(I - CA) < 1,$$

d.h. beide Verfahren konvergieren für beliebige Startwerte $x_0 \in \mathbb{R}^n$.

Beweis. Siehe [4]. □

Als drittes wichtiges Kriterium kann man für das Gauß-Seidel-Verfahren (im Gegensatz zum Jacobi-Verfahren) zeigen, dass es für symmetrisch positiv definite Matrizen konvergiert. Es gibt viele weitere iterative Verfahren zur Lösung von linearen Gleichungssystemen. Als Beispiel nennen wir noch eine Erweiterung des Gauß-Seidel-Verfahrens. Sei dazu die k-te Iterierte des Gauß-Seidel-Verfahrens bezeichnet mit $x^{(k),GS}$. Das SOR-Verfahren (*successive over-relaxation*) definiert Iterierte mit einem Relaxationsparameter $\omega \in (0, 2)$

$$x^{(k),SOR} = \underline{\omega} x^{(k),GS} + \underline{(}1 - \omega) x^{(k-1)}$$

Ist der optimale Relaxationsparameter ω_{opt} bekannt, so kann die Konvergenzrate gegenüber dem Gauß-Seidel-Verfahren deutlich verbessert werden.

Die iterative Lösung von linearen Gleichungssystemen ist ein Forschungsgebiet für sich, welches gegebenenfalls in weiterführenden Numerikvorlesung behandelt wird. Wir wenden abschließend Jacobi- und Gauß-Seidel-Verfahren in einem Beispiel an.

Beispiel Wir lösen das Problem $Ax = b$ für

$$A = \begin{pmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{pmatrix} \in \mathbb{R}^{n\times n}, \quad b = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^n,$$

n	Jacobi	Gauß-Seidel	LR
16	$2.1 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$
32	$8.9 \cdot 10^{-3}$	$4.1 \cdot 10^{-3}$	$1.3 \cdot 10^{-2}$
64	$3.2 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$6.6 \cdot 10^{-2}$
128	$1.1 \cdot 10^{-1}$	$4.6 \cdot 10^{-2}$	$5.1 \cdot 10^{-1}$
256	$4.3 \cdot 10^{-1}$	$1.8 \cdot 10^{-1}$	3.9
512	1.8	$7.7 \cdot 10^{-1}$	36.4
	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^3)$

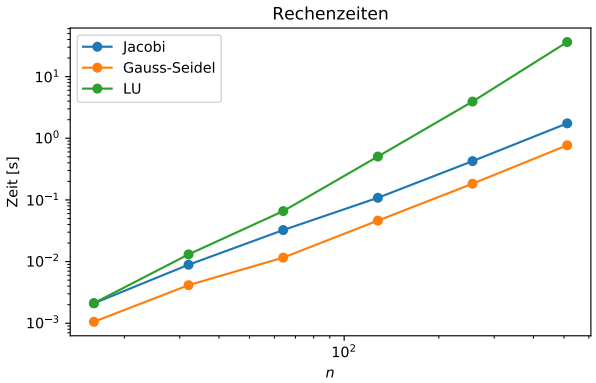

Abbildung 4.5: Laufzeit (in Sekunden) von Jacobi- und Gauß-Seidel-Verfahren sowie Gauß-Elimination (LR) zur Lösung von Beispiel 1 für verschiedene Matrixgrößen n auf einem Laptop. Bei Jacobi- und Gauß-Seidel-Verfahren wurde das relative Abbruchkriterium $r_k \leq 10^{-4} r_0$ verwendet.

für verschiedene n mit Jacobi und Gauß- Seidel-Verfahren sowie mit Gauß-Elimination.

Die Matrix A ist sowohl stark diagonaldominant als auch symmetrisch, negativ definit. Sowohl für das Jacobi- als auch das Gauß-Seidel-Verfahren gilt

$$|||I - CA||| < q < 1$$

und damit

$$\|x_k - z\| \leq q\|x_{k-1} - z\| \leq q^k \|x_0 - z\|.$$

Zum Erreichen einer relativen Toleranz TOL bzgl. $\|x_k - z\|$ sind daher unabhängig von n

$$c_{\text{it}} \approx \log_q(TOL)$$

Iterationen notwendig. Ähnliche Resultate gelten auch für andere Abbruchkriterien.

In Abbildung 4.5 werden die Laufzeiten der beiden Verfahren für verschiedene n mit dem Laufzeiten der Gauß-Elimination (inkl. Rückwärtseinsetzen) verglichen. Wie erwartet, wachsen die Laufzeiten mit $\mathcal{O}(n^3)$ bei der Gauß-Elimination und $\mathcal{O}(n^2)$ bei Jacobi- und Gauß-Seidel-Verfahren, wobei das Gauß-Seidel-Verfahren um etwas mehr als einen Faktor 2 schneller ist. Für größere n unterscheiden sich die Laufzeiten von Jacobi- und Gauß-Seidel-Verfahren auf der einen Seite und Gauß-Elimination auf der anderen Seite schon recht deutlich.

4.8 Newton-Verfahren im $\mathbb{R}^n$

Auch im mehrdimensionalen kann das Newton-Verfahren über die lokale Näherung von f durch Taylorentwicklung um x_k hergeleitet werden. Es gilt mit Taylorentwicklung

$$\begin{aligned} f(x) &= f(x_k) + \sum_{j=1}^{n} \partial_j f(x_k)(x - x_k)_j + \mathcal{O}(\|x - x_k\|^2) \\ &= f(x_k) + J_f(x_k)(x - x_k) + \mathcal{O}(\|x - x_k\|^2). \end{aligned}$$

Wir definieren die folgende Itererierte x_{k+1} wieder als Nullstelle des Taylorpolynoms erster Ordnung

$$x_{k+1} = x_k - J_f(x_k)^{-1} f(x_k).$$

Da das Invertieren der $n \times n$-Matrix $J_f(x_k)$ in der Regel recht teuer ist ($\mathcal{O}(n^3)$ a.Op.), berechnet man x_{k+1} durch Lösen des Gleichungssystems

$$J_f(x_k) \underbrace{(x_{k+1} - x_k)}_{\delta x_k} = -f(x_k).$$

Dies kann wieder mithilfe von iterativen Methoden erfolgen. Es ergibt sich folgender Algorithmus.

Newton-Verfahren im $\mathbb{R}^n$
Eingabe: Startwert $x_0 \in \mathbb{R}^n$ (Setze $k = 0$)
Solange $\|f(x_k)\| >$TOL

(i) Berechne die Jacobi-Matrix $J_f(x_k)$

(ii) Löse das LGS $J_f(x_k)\delta x_k = -f(x_k)$

(iii) Update $x_{k+1} = x_k + \delta x_k$

(iv) Berechne $f(x_{k+1})$, $k \leftarrow k + 1$

Ausgabe: Iterierte x_k

In jedem Newton-Schritt müssen zum einen n^2 Ableitungen der Jacobi-Matrix berechnet werden (i). Es ergibt sich ein Aufwand von $\mathcal{O}(n^2)$. Zum anderen muss in (ii) ein LGS gelöst werden. Geschieht dies mit einem direkten Verfahren (z.B. Gauß-Elimination) sind $\mathcal{O}(n^3)$ a.Op. notwendig. Sind die Voraussetzungen für die Konvergenz von Jacobi- oder Gauß-Seidel-Verfahren erfüllt und ist $J_f(x_k)$ gut konditioniert, reduziert sich der Aufwand mit diesen Verfahren auf $\mathcal{O}(n^2)$ und liegt damit im Bereich des Matrixaufbaus. Alle anderen Schritte haben nur lineare Komplexität $\mathcal{O}(n)$.

Wir formulieren nun das Hauptresultat dieses Abschnitts, den Konvergenzsatz von Newton-Kantorovich. Dessen Resultate sind auch in 1 Dimension gültig und damit eine Verallgemeinerung von Satz 4.1. Der Beweis des folgenden Satzes ist allerdings deutlich aufwändiger.

Satz 4.18. *(Newton-Kantorovich) Es sei $D \subset \mathbb{R}^n$ und $f : D \to \mathbb{R}^n$ stetig differenzierbar. Weiter seien die folgenden Voraussetzungen an f und den Startpunkt x_0 erfüllt:*

(i) Die Jacobi-Matrix J_f sei gleichmäßig Lipschitz-stetig auf D, d.h. es gelte für alle $x, y \in D$ mit einer Konstanten $L < \infty$

$$|||J_f(x) - J_f(y)||| \leq L\|x - y\|. \tag{4.36}$$

(ii) Die Jacobi-Matrix habe auf D eine gleichmäßig beschränkte Inverse

$$|||J_f(x)^{-1}||| \leq \beta, \quad x \in D \tag{4.37}$$

mit einer Konstanten $\beta < \infty$.

(iii) Für den Startpunkt $x_0 \in D$ gelte

$$q := \alpha\beta L < \frac{1}{2}, \quad \text{wobei } \alpha := \|J_f(x_0)^{-1} f(x_0)\|.$$

(iv) Die abgeschlossene Kugel $K_{2\alpha}(x_0)$ sei ganz in D enthalten.

Dann besitzt f genau eine Nullstelle $z \in K_{2\alpha}(x_0)$ und es gilt die a priori Fehlerabschätzung

$$\|x_k - z\| \leq 2\alpha q^{2^k - 1}, \quad k = 0, 1, ...$$

Bemerkung 4.19. *Satz 4.18 gilt auch in 1 Dimension und stellt damit eine Verallgemeinerung von Satz 4.1 dar. Der Hauptunterschied liegt darin, dass Satz 4.18 auch die* ***Existenz einer eindeutigen Nullstelle*** *$z \in K_{2\alpha}(x_0)$ sichert, deren Existenz in Satz 4.1 angenommen wurde. Wir vergleichen die übrigen Voraussetzungen für $D = I = [a, b]$*

(i) Statt 2-maliger stetiger Differenzierbarkeit wird in Satz 4.18 nur ***Lipschitz-Stetigkeit der ersten Ableitung*** *gefordert. Auf der abgeschlossenen Menge I ist das Maximum der zweiten Ableitungen eine obere Schranke für L, da (vgl Lemma 4.11)*

$$|f'(x) - f'(y)| \leq \max_{\xi \in I} |f''(\xi)||x - y| \qquad \Rightarrow \; M := \max_I |f''| \geq L$$

(ii) In 1 Dimension gilt

$$\beta = \max_I |(f'(x))^{-1}| = \left(\min_I |f'(x)| \right)^{-1} =: m^{-1},$$

so dass die Bedingungen $\beta < \infty$ (Satz 4.18) und $m > 0$ (Satz 4.1) identisch sind.

(iii) Der Rolle von ρ in Satz 4.1 ($x_0 \in K_\rho(z)$) entspricht in etwa 2α in Satz 4.18 ($z \in K_{2\alpha}(x_0)$). Mit $\beta = m^{-1}$ und $M \approx L$ gilt

$$q := \alpha\beta L \approx \frac{\rho M}{2m}.$$

In Satz 4.1 wurde $q < 1$ gefordert, in Satz 4.18 die stärkere Voraussetzung $q < \frac{1}{2}$. Tatsächlich gilt Satz 4.18 sogar für $q < 2$ (siehe [4]). Da der Beweis dafür allerdings wesentlich aufwändiger ist als der oben gezeigte, haben wir hier auf die schwächere Voraussetzung verzichtet.

(iv) Auch die vierte Voraussetzung $K_{2\alpha}(x_0) \subset I$ wurde analog in Satz 4.1 gefordert.

Bemerkung 4.20. *Wie im eindimensionalen ist das Newton-Verfahren auch im $\mathbb{R}^n$ das Standard-Verfahren zur Lösung von nichtlinearen Gleichungen. Probleme bestehen wie bereits im $\mathbb{R}^1$ diskutiert darin, dass der Konvergenzbereich je nach Funktion f recht klein sein kann.*

Zur Vergrößerung des Konvergenzradius kann auch hier ein ***gedämpftes Newton-Verfahren*** *eingesetzt werden, mit analogen Konvergenzaussagen wie im Eindimensionalen. Zur Vermeidung der (möglicherweise recht teuren) n^2 Ableitungsberechnungen können* ***vereinfachte Newton-Verfahren*** *definiert werden. Auch hierfür gelten analoge Konvergenzaussagen wie im $\mathbb{R}^1$.*

Beweis von Satz 4.18. Wir unterteilen den Beweis in 5 Teilschritte:

(I) Herleitung einer Hilfsabschätzung

(II) Wir zeigen: Die Iterierten x_k bleiben in der Kugel $K_{2\alpha}(x_0)$ und $\|x_{k+1} - x_k\| \leq \alpha q^{(2^k-1)}$

(III) Wir zeigen: $(x_k)_{k\geq 0}$ ist eine Cauchy-Folge und konvergiert gegen einen Grenzwert $z \in K_{2\alpha}(x_0)$. Es ergibt sich direkt die *a priori* Fehlerabschätzung.

(IV) Wir zeigen: Der Grenzwert z ist eine Nullstelle von f.

(V) Eindeutigkeit der Nullstelle in $K_{2\alpha}(x_0)$

(I) Wir beginnen mit der Herleitung eines Hilfsresultats. Seien dazu $x, y, w \in K_{2\alpha}(x_0)$. Für die Funktion

$$h : [0,1] \to \mathbb{R}^n, \quad h(s) := f(y + s(x-y))$$

gilt

$$h(1) - h(0) = \int_0^1 h'(s)\,\mathrm{d}s.$$

Dies ist gleichbedeutend mit

$$f(x) - f(y) = \int_0^1 J_f(y + s(x-y))\,(x-y)\,\mathrm{d}s.$$

Subtraktion von $J_f(w)(x-y)$ ergibt

$$f(x) - f(y) - J_f(w)(x-y) = \int_0^1 \left(J_f(y + s(x-y)) - J_f(w)\right)\,(x-y)\,\mathrm{d}s.$$

Mithilfe der Lipschitz-Stetigkeit von J_f schätzen wir ab

$$\begin{aligned} \|f(x) - f(y) - J_f(w)(x-y)\| &\leq L\|x-y\| \int_0^1 \|y + s(x-y) - w\|\,\mathrm{d}s \\ &\leq L\|x-y\| \int_0^1 s\|x-w\| + (1-s)\|y-w\|\,\mathrm{d}s \\ &\leq \frac{L}{2}\|x-y\|\left(\|x-w\| + \|y-w\|\right). \end{aligned} \tag{4.38}$$

Für $w = x$ vereinfacht sich diese Abschätzung noch zu

$$\|f(x) - f(y) - J_f(x)(x-y)\| \leq \frac{L}{2}\|x-y\|^2. \tag{4.39}$$

(II) Wir zeigen nun induktiv für $k = 0, 1, ...,$ dass

$$\|x_{k+1} - x_0\| \leq 2\alpha, \tag{4.40}$$

$$\|x_{k+1} - x_k\| \leq \alpha q^{(2^k-1)}. \tag{4.41}$$

Die erste Abschätzung ist gleichbedeutend mit $x_{k+1} \in K_{2\alpha}(x_0)$. Für $k = 0$ gilt

$$\|x_1 - x_0\| = \|J_f(x_0)^{-1} f(x_0)\| = \alpha,$$

so dass beide Bedingungen erfüllt sind. Wir nehmen nun an, dass (4.40)-(4.41) für $k = 0, ..., m-1$ erfüllt ist und schließen auf $k = m$. Es gilt nach Definition der Newton-Iterierten x_{m+1} und x_m und (4.37)

$$\begin{aligned} \|x_{m+1} - x_m\| &\overset{x_{m+1}}{=} \|J_f(x_m)^{-1} f(x_m)\| \\ &\overset{(4.37)}{\leq} \beta \|f(x_m)\| \\ &\overset{x_m}{=} \beta \|f(x_m) \underbrace{- f(x_{m-1}) - J_f(x_{m-1})(x_m - x_{m-1})}_{=0}\|. \end{aligned} \tag{4.42}$$

Nun wenden wir das Hilfsresultat (4.39) und die Induktionsvoraussetzung (4.41) an und erhalten

$$\begin{aligned} \|x_{m+1} - x_m\| \overset{(4.39)}{\leq} \frac{\beta L}{2} \|x_m - x_{m-1}\|^2 &\overset{(4.41)}{\leq} \frac{\beta L}{2} \left(\alpha q^{(2^{m-1}-1)}\right)^2 \\ &= \underbrace{\alpha \beta L}_{=q} \frac{\alpha}{2} q^{(2^m-2)} < \alpha q^{(2^m-1)}. \end{aligned} \tag{4.43}$$

Dies zeigt (4.41) für $k = m$. Um (4.40) zu zeigen, nutzen wir die Dreiecksungleichung und (4.43)

$$\begin{aligned} \|x_{m+1} - x_0\| &\leq \|x_{m+1} - x_m\| + \|x_m - x_{m-1}\| + ... + \|x_1 - x_0\| \\ &\leq \alpha \left(q^{(2^m-1)} + ... + q + 1\right) \\ &\leq \alpha \left(\sum_{k=0}^{\infty} q^k\right) = \frac{\alpha}{1-q} \leq 2\alpha, \end{aligned}$$

da $q \leq \frac{1}{2}$ nach Voraussetzung.

(III) (4.41) ist bereits ausreichend, um zu zeigen, dass $(x_k)_{k\geq 0}$ eine Cauchy-Folge bildet. Es gilt nämlich

$$\begin{aligned} \|x_{k+m} - x_k\| &\leq \|x_{k+m} - x_{k+m-1}\| + \|x_{k+m-1} - x_{k+m-2}\| + ... + \|x_{k+1} - x_k\| \\ &\leq \alpha \left(q^{(2^{k+m-1}-1)} + q^{(2^{k+m-2}-1)} + ... + q^{(2^k-1)}\right) \\ &= \alpha q^{(2^k-1)} \left(1 + q^{(2^k)} + ... + \left(q^{(2^k)}\right)^{2^m}\right) \\ &\leq \frac{\alpha q^{(2^k-1)}}{1-q} \leq 2\alpha q^{(2^k-1)} \to 0 \quad (k \to \infty). \end{aligned} \tag{4.44}$$

Die Folge $(x_k)_{k\geq 0}$ ist also eine Cauchy-Folge und hat einen Grenzwert z im (Banachraum) $\mathbb{R}^n$. Da nach (4.40) für alle k

$$\|x_k - x_0\| \leq 2\alpha$$

bleibt diese Beziehung im Grenzwert erhalten, d.h. es gilt $x_k \to z \in K_{2\alpha}(x_0)$. Übergang zum Limes $m \to \infty$ in (4.44) ergibt die *a priori* Abschätzung

$$\|x_k - z\| \leq 2\alpha q^{(2^k-1)}, \quad k = 0, 1, ...$$

(IV) Wir müssen noch zeigen, dass $f(z) = 0$ gilt. Dazu haben wir in (II) (4.42)-(4.43) bereits gezeigt, dass

$$\beta\|f(x_m)\| \leq \beta\frac{\alpha^2 L}{2}q^{(2^m-2)}.$$

Mit Grenzübergang $m \to \infty$ folgt $f(z) = 0$.

(V) Abschließend zeigen wir, dass f unter den gegebenen Voraussetzungen nur eine Nullstelle z in $K_{2\alpha}(x_0)$ hat. Dazu zeigen wir, dass die Fixpunktiteration (vereinfachte Newtoniteration)

$$g(x) = x - J_f(x_0)^{-1}f(x)$$

eine Kontraktion auf $K_{2\alpha}(x_0)$ ist.

Seien dazu $x, y \in K_{2\alpha}(x_0)$. Es gilt nach Definition von g

$$\begin{aligned}\|g(x) - g(y)\| &= \|x - y - J_f(x_0)^{-1}(f(x) - f(y))\| \\ &= \|J_f(x_0)^{-1}\left(J_f(x_0)(x - y) - f(x) + f(y)\right)\| \\ &\leq \underbrace{|||J_f(x_0)^{-1}|||}_{\leq\beta}\, \|J_f(x_0)(x - y) - f(x) + f(y)\|\|\end{aligned}$$

Mit dem Hilfsresultat (4.38) aus (I) für $w = x_0$ folgt daraus

$$\begin{aligned}\|g(x) - g(y)\| &\leq \beta\frac{L}{2}\|x - y\|\left(\underbrace{\|x - x_0\|}_{\leq 2\alpha} + \underbrace{\|y - x_0\|}_{\leq 2\alpha}\right) \\ &\leq 2L\alpha\beta\|x - y\| = 2q\|x - y\|.\end{aligned}$$

Da $q < \frac{1}{2}$, ist g eine Kontraktion auf $K_{2\alpha}(x_0)$. Daraus folgt direkt die Eindeutigkeit des Fixpunktes in $K_{2\alpha}(x_0)$. Gäbe es nämlich 2 Fixpunkte $z_1, z_2 \in K_{2\alpha}(x_0)$, so folgt

$$\|z_1 - z_2\| = \|g(z_1) - g(z_2)\| \leq 2q\|z_1 - z_2\|$$

und damit $z_1 = z_2$.

□

5 Polynominterpolation

In diesem Kapitel beschäftigen wir uns mit der **Approximation** einer Reihe von Daten (x_i, y_i) oder einer möglicherweise komplizierten Funktion f durch ein Element p einer einfacheren Funktionenklasse, insbesondere durch Polynome $p \in P_n$. Ähnliche Aufgabenstellungen wurden bereits in Kapitel 3 im Rahmen der *Linearen Ausgleichungsrechnung* diskutiert.

Im Gegensatz zur *Ausgleichsrechnung* fordert man bei der **Interpolation**, dass die *interpolierende Funktion* p Werte y_i (z.B. Funktionswerte $f(x_i)$) in gewissen Punkten x_i exakt annimmt

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

In der Numerik ist die **Interpolation mit Polynomen** besonders interessant. Das liegt daran, dass z.B. Integrale oder Ableitungen von Polynomen sehr einfach und effizient berechnet werden können. Die Polynominterpolation hat große Bedeutung in den weiterführenden Numerikvorlesungen, insbesondere in der *Numerik partieller Differentialgleichungen.*

Der Weierstraßsche Approximationssatz besagt, dass man jede Funktion $f \in C[a, b]$ beliebig gut durch ein Polynom **approximieren** kann: *Für jedes* $\epsilon > 0$ *gibt es ein Polynom* p*, so dass*

$$|f(x) - p(x)| < \epsilon \quad \forall x \in [a, b].$$

Leider liefert er allerdings kein Konstruktionsprinzip für solche Polynome p.

5.1 Lagrange-Interpolation

Wir beschäftigen uns zunächst mit der sogenannten **Lagrangeschen Interpolationsaufgabe**.

Definition 5.1. *(Lagrangesche Interpolationsaufgabe) Zu (n+1) paarweise verschiedenen Stützstellen* $x_0, \ldots, x_n \in [a, b]$ *und Stützwerten* $y_0, \ldots, y_n$ *ist ein Polynom* p *vom Grad* n *(*$p \in P_n$*) so zu bestimmen, dass*

$$p(x_i) = y_i, \quad i = 0, \ldots, n. \tag{5.1}$$

Die Interpolationsaufgabe ist wohlgestellt:

Satz 5.2. *Die Lagrangesche Interpolationsaufgabe ist eindeutig lösbar.*

Beweis. Wir zeigen zunächst die **Eindeutigkeit** von Lösungen. Dazu nehmen wir an es gäbe 2 Lösungen $p_1, p_2 \in P_n$, die (5.1) erfüllen. Für $p = p_1 - p_2$ gilt

$$p(x_i) = p_1(x_i) - p_2(x_i) = 0, \quad i = 0, \ldots, n.$$

Das Polynom $p \in P_n$ hat also $n + 1$ Nullstellen. Das ist nach dem Fundamentalsatz der Algebra nur möglich, wenn $p \equiv 0$ ist. Damit folgt die Eindeutigkeit einer Lösung.

Die **Existenz** von Lösungen kann man nun mit Argumenten der linearen Algebra aus der Eindeutigkeit folgern. Für ein beliebiges Polynom $p \in P_n$ gilt

$$p(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n.$$

Wir betrachten (5.1) als lineares Gleichungsssystem zur Berechnung der Koeffizienten $a_i, i = 0, ..., n$:

$$\underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & ... & x_0^n \\ 1 & x_1 & x_1^2 & ... & x_1^n \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & ... & x_n^n \end{pmatrix}}_{=:V_n} \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix}}_{=:w} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{5.2}$$

mit der sogenannten *Vandermonde*-Matrix $V_n \in \mathbb{R}^{(n+1)\times(n+1)}$. Aus dem Eindeutigkeitsbeweis oben folgt, dass die zur Matrix V_n gehörende lineare Abbildung

$$g : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}, \quad g(w) = V_n w$$

injektiv (Aus $V_n w_1 = V_n w_2$ folgt $w_1 = w_2$). Damit folgt, dass V_n Vollrang hat. Daraus folgt aber auch die Surjektivität von g und damit die Existenz einer Lösung $(a_0, ..., a_n)$ für beliebige rechte Seiten $(y_0, ..., y_n)$. □

(5.2) könnte nun direkt zur Berechnung der Koeffizienten $(a_0, \ldots, a_n)$ angewendet werden. Leider ist die *Vandermonde*-Matrix aber für wachsendes n sehr schlecht konditioniert. Bei äquidistanter Verteilung der Stützstellen über das Intervall $[1, 2]$, d.h. $x_i = 1 + \frac{i}{n}$, $i = 0, ..., n$ gilt beispielsweise (siehe [1])

$$\kappa_2(V_6) \approx 2.0 \cdot 10^7, \qquad \kappa_2(V_8) \approx 1.1 \cdot 10^{10}, \qquad \kappa_2(V_{10}) \approx 6.5 \cdot 10^{12}.$$

Das bedeutet, dass bei der Lösung des Gleichungssystems (5.2) mit einer extremen Fehlerverstärkung zu rechnen ist.

Für stabilere und effizientere Wege zur Berechnung des Lagrangeschen Interpolationspolynom p_n betrachten wir statt der **Monombasis** $(1, x, \ldots, x^n)$ nun alternative Basen des P_n.

5.1.1 Lagrangesche Darstellung des Interpolationspolynoms

Zu $(n+1)$ Stützstellen paarweise verschiedenen $x_0, ..., x_n$ definieren wir die $n+1$ **Lagrangeschen Basispolynome**

$$L_i^n(x) := \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \in P_n, \qquad i = 0, ..., n. \tag{5.3}$$

Nach Konstruktion gilt mit dem *Kronecker-Delta* δ_{ij}

$$L_i^n(x_j) = \delta_{ij} := \begin{cases} 1, \, i = j \\ 0, \, i \neq j \end{cases}. \tag{5.4}$$

Lemma 5.3. *Die in* (5.3) *definierten Polynome* $L_i^n, i = 0, ..., n$ *bilden eine Basis des Polynomraums* P_n.

Beweis. Es ist zu zeigen, dass die Polynome $L_i^n, i = 0, ..., n$ linear unabhängig sind. Seien also $\alpha_i, i = 0, ..., n$ reelle Zahlen, so dass

$$\sum_{i=0}^{n} \alpha_i L_i^n \equiv 0.$$

Mit der Eigenschaft (5.4) folgt für $j = 0, ..., n$

$$0 = \sum_{i=0}^{n} \alpha_i L_i^n(x_j) = \alpha_j$$

und damit die lineare Unabhängigkeit der L_i^n. □

Aufgrund der Eigenschaft (5.4) ist die Bestimmung der Koeffizienten des Interpolationspolynoms zur Lagrangeschen Basis trivial. Es gilt nämlich

$$p(x) := \sum_{i=0}^{n} y_i L_i^n(x). \tag{5.5}$$

Man überpüft leicht, dass dieses Polynom die Gleichungen (5.1) erfüllt. Die Darstellung (5.5) wird die *Lagrangesche Darstellung* des Interpolationspolynoms zu den Stützstellen $(x_0, y_0), \ldots, (x_n, y_n)$ genannt.

Der Vorteil der Lagrangeschen Darstellung des Interpolationspolynom ist ihre Einfachheit. Sie ist daher insbesondere in der Theorie sehr nützlich. Ihr Nachteil besteht darin, dass jedes Basispolynom $L_i^n(x)$ von allen Stützstellen $x_0, \ldots, x_n$ abhängt. Wird ein neuer Stützpunkt x_{n+1} hinzugenommen, z.B. um die Genauigkeit zu erhöhen, müssen alle Basispolynome neu berechnet werden. Dieser Nachteil der Lagrangeschen Darstellung motiviert die Betrachtung einer weiteren Basis von P_n, welche uns auf die **Newtonsche Darstellung** des Interpolationspolynoms führt.

5.1.2 Newtonsche Darstellung des Interpolationspolynoms

Wir definieren nun eine Basis, welche sich rekursiv aufbauen lässt

$$N_0(x) := 1, \quad N_i(x) := \prod_{j=0}^{i-1} (x - x_j), \quad i = 1, \ldots, n. \tag{5.6}$$

Es gilt

$$N_i(x) := N_{i-1}(x) \cdot (x - x_{i-1}).$$

Lemma 5.4. *Seien* $x_0, ..., x_n \in \mathbb{R}$ *paarweise verschieden. Die in* (5.6) *definierten Polynome* $N_i, i = 0, ..., n$ *bilden eine Basis des Polynomraums* P_n.

Beweis. Wir zeigen wieder die lineare Unabhängigkeit der $N_i, i = 0, ..., n$. Seien dazu $\alpha_i \in \mathbb{R}, i = 0, ..., n$ so, dass

$$\sum_{i=0}^{n} \alpha_i N_i \equiv 0.$$

Da N_i für $i > j$ den Faktor $(x - x_j)$ enthält, gilt $N_i(x_j) = 0$ für $j < i$. Es folgt zunächst

$$0 = \sum_{i=0}^{n} \alpha_i N_i(x_0) = \alpha_0.$$

Und dann induktiv

$$\begin{aligned}
0 &= \sum_{i=1}^{n} \alpha_i N_i(x_1) = \alpha_1(x_1 - x_0) \quad \Rightarrow \alpha_1 = 0, \\
0 &= \sum_{i=2}^{n} \alpha_i N_i(x_2) = \alpha_2(x_2 - x_0)(x_2 - x_1) \quad \Rightarrow \alpha_2 = 0, \\
&\vdots
\end{aligned}$$

Es folgt die lineare Unabhängigkeit der Polynome $N_i, i = 0, ..., n$. □

Die Koeffizienten a_i bzgl dieser Basis könnte man nun rekursiv aus dem folgenden Gleichungssystem berechnen

$$\begin{aligned}
y_0 &= p(x_0) = a_0 \\
y_1 &= p(x_1) = a_0 + a_1(x_1 - x_0) \quad \Rightarrow a_1 = \frac{y_1 - a_0}{x_1 - x_0}, \\
y_2 &= p(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \quad \Rightarrow a_2 = \frac{y_2 - a_0 - a_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}, \\
&\vdots \\
y_n &= p(x_n) = a_0 + \sum_{i=1}^{n-1} a_i N_i(x_n) + a_n N_n(x_n) \quad \Rightarrow a_n = \frac{y_n - \sum_{i=1}^{n-1} a_i N_i(x_n)}{N_n(x_n)}.
\end{aligned}$$

Ist das Interpolationspolynom p_n zu n Stützstellen gegeben und wird noch eine Stützstelle (x_{n+1}, y_{n+1}) hinzugefügt, so muss nur der neue *führende* Koeffizient a_{n+1} berechnet werden, die übrigen Bedingungen behalten ihre Gültigkeit. Es gilt

$$y_{n+1} = p(x_{n+1}) = p_n(x_{n+1}) + a_{n+1} N_{n+1}(x_{n+1}) \quad \Rightarrow a_{n+1} = \frac{y_{n+1} - p_n(x_{n+1})}{N_{n+1}(x_{n+1})}. \tag{5.7}$$

Dividierte Differenzen Effizienter und numerisch stabiler ist die Berechnung der Koeffizienten mit Hilfe der sogenannten *dividierten Differenzen*. Wir definieren $y[x_k, ..., x_{k+l}]$ für $k, l \geq 0$ und

$k + l \leq n$ als

$$\begin{aligned} y[x_k] &:= y_k, \qquad k = 0, ..., n \\ y[x_m, ..., x_{m+l}] &:= \frac{y[x_{m+1}, ..., x_{m+l}] - y[x_m, ..., x_{m+l-1}]}{x_{m+l} - x_m} \qquad l = 1, ..., n,\ m = 0, ..., n - l. \end{aligned} \tag{5.8}$$

Der folgenden Satz zeigt, dass die Koeffizienten a_i der Newton-Darstellung durch die dividierten Differenzen $y[x_0, ..., x_i]$ gegeben sind.

Satz 5.5. *(Newton-Darstellung) Seien $y_0, ..., y_n \in \mathbb{R}$ (n+1) Stützwerte zu (n+1) paarweise verschiedene Stützstellen $x_0, ..., x_n \in \mathbb{R}$. Das zugehörige Lagrangesche Interpolationspolynom $p \in P_n$ hat die Darstellung*

$$p(x) = \sum_{k=0}^{n} y[x_0, ..., x_k] N_k(x) \tag{5.9}$$

mit den in (5.8) *definierten dividierten Differenzen und den in* (5.6) *definierten Newtonschen Basispolynomen des P_n.*

Beweis. Wir bezeichnen mit $p_{k,k+l} \in P_l$ das Interpolationspolynom durch die Stützstellen $(x_k, y_k), ..., (x_{k+l}, y_{k+l})$. Das gesuchte Polynom hat dann die Darstellung $p = p_{0,n}$.

Wir zeigen nun für alle $l = 0, ..., n$ und $k = 0, ..., n - l$, dass

$$p_{k,k+l} = y[x_k] + y[x_k, x_{k+1}](x - x_k) + ... + y[x_k, ..., x_{k+l}](x - x_k) \cdots (x - x_{k+l-1}). \tag{5.10}$$

Für $k = 0$ und $l = n$ folgt die Behauptung des Satzes.

Wir führen den Beweis induktiv nach dem Polynomgrad l.
Induktionsanfang: Für $l = 0$ gilt

$$p_{k,k} = y[x_k] = y_k.$$

Dies ist offensichtlich das konstante Polynom $p_{k,k} \in P_0$ durch die Stützstelle (x_k, y_k).

Induktionsschritt: Nun sei die Behauptung erfüllt für $l - 1 \geq 0$. Damit gilt insbesondere, dass die Interpolationspolynome $p_{k,k+l-1}$ durch die Punkte $(x_k, y_k), ..., (x_{k+l-1}, y_{k+l-1})$ und $p_{k+1,k+l}$ durch die Punkte $(x_{k+1}, y_{k+1}), ..., (x_{k+l}, y_{k+l})$ die Darstellung (5.10) erfüllen.

Nach (5.7) besitzt das Interpolationspolynom bei Hinzunahme einer Stützstelle (x_{k+l}, y_{k+l}) die Darstellung

$$p_{k,k+l}(x) = p_{k,k+l-1}(x) + a(x - x_k)(x - x_{k+1}) \cdots (x - x_{k+l-1}) \quad \text{mit } a \in \mathbb{R}. \tag{5.11}$$

Da $p_{k,k+l-1}(x)$ die Darstellung (5.10) erfüllt, ist nur zu zeigen, dass

$$a = y[x_k, ..., x_{k+l}].$$

Hierzu definieren wir

$$q(x) := \frac{(x - x_k)p_{k+1,k+l}(x) - (x - x_{k+l})p_{k,k+l-1}(x)}{x_{k+l} - x_k}. \tag{5.12}$$

Man rechnet nach, dass dieses Polynom die Punkte $(x_k, y_k), ..., (x_{k+l}, y_{k+l})$ interpoliert

$$q(x_k) = \frac{-(x_k - x_{k+l})p_{k,k+l-1}(x_k)}{x_{k+l} - x_k} = p_{k,k+l-1}(x_k) = y_k$$
$$q(x_{k+l}) = \frac{(x_{k+l} - x_k)p_{k+1,k+l}(x_{k+l})}{x_{k+l} - x_k} = p_{k+1,k+l}(x_{k+l}) = y_{k+l},$$

sowie für $i = 1, ..., l-1$

$$q(x_{k+i}) = \frac{(x_{k+i} - x_k)\overbrace{p_{k+1,k+l}(x_{k+i})}^{=y_{k+i}} - (x_{k+i} - x_{k+l})\overbrace{p_{k,k+l-1}(x_{k+i})}^{=y_{k+i}}}{x_{k+l} - x_k}$$
$$= \frac{(x_{k+i} - x_k)y_{k+i} - (x_{k+i} - x_{k+l})y_{k+i}}{x_{k+l} - x_k} = y_{k+i}.$$

Es gilt also $p_{k,k+l} = q$. Vergleicht man den führenden Koeffizienten bzgl. x^l in (5.11) mit dem in (5.12), so folgt unter Anwendung der Induktionsvoraussetzung (5.10)

$$a = \frac{y[x_{k+1}, ..., x_{k+l}] - y[x_k, ..., x_{k+l-1}]}{x_{k+l} - x_k} = y[x_k, ..., x_{k+l}].$$

□

Bemerkung 5.6. *Aus der Newtondarstellung* (5.9) *mit dividierten Differenzen lässt sich noch eine Eigenschaft der dividierten Differenzen herleiten, welche wir später benötigen werden. Der führenden Koeffizient bzgl. der Monombasis ist nach* (5.9) *gegeben durch* $y[x_0, ..., x_n]$

Das Interpolationspolynom ist sicherlich invariant gegenüber einer Permutation der Stützstellen und damit auch sein führender Koeffizient bzgl. der Monombasis. Daher gilt

$$y[x_0, ..., x_n] = y[x_{i_0}, ..., x_{i_n}]$$

für eine beliebige Permutation $i_0, ..., i_n$ *der Indizes* $0, ..., n$.

Aufwandsanalyse Als nächsten analysieren wir den Aufwand zur Berechnung der Koeffizienten des Interpolationspolynoms in Newton-Darstellung und dessen anschließende Auswertung.

Dividierte Differenzen

Eingabe: Stützstellen $x_0, ..., x_n$, Stützwerte $y_0, ..., y_n$

Für $k = 0, ..., n$:
  $y[x_k] := y_k$

Für $l = 1, ..., n$:
  Für $m = 0, ..., n - l$:
    $y[x_m, ..., x_{m+l}] := \frac{y[x_{m+1}, ..., x_{m+l}] - y[x_m, ..., x_{m+l-1}]}{x_{m+l} - x_m}$

Ausgabe: Koeffizienten $a_i = y[x_0, ..., x_i]$, $i = 0, ..., n$

Die Anzahl der zu berechnenden Divisionen in (5.8) summieren sich zu

$$\sum_{l=1}^{n}\sum_{m=0}^{n-l} 1 = \sum_{l=1}^{n} n-l+1 = \sum_{j=1}^{n} j \;=\; \frac{n(n-1)}{2} \;=\; \frac{n^2}{2} + \mathcal{O}(n).$$

Außerdem sind doppelt so viele Subtraktionen zu berechnen.

Sind die Koeffizienten $a_i, i = 0, ..., n$ einmal berechnet, kann das Interpolationspolynom in Newtonscher Darstellung

$$p(x) = \sum_{i=0}^{n} a_i N_i(x)$$

effizient in $\mathcal{O}(n)$ Operationen an einer Stelle $\xi \in [a, b]$ ausgewertet werden. Am effizientesten und stabilsten geschieht das mit dem Horner-Schema (siehe Übung).

Auswertung eines Polynoms in Newton-Darstellung mit Horner-Schema

Eingabe: Koeffizienten $a_0, ..., a_n$, Stützstellen $x_0, ..., x_n$, Stelle $\xi \in [a, b]$

Setze $b_n = a_n$
Für $k = n-1, ..., 0$:
$\quad b_k = a_k + (\xi - x_k) b_{k+1}$

Ausgabe: $p(\xi) = b_0$.

Es sind n Multiplikationen und $2n$ Additionen notwendig.

Auch die Hinzunahme einer Stützstelle kann in der Newtondarstellung effizient ausgeführt werden. Ist nämlich das Interpolationspolynom

$$p_n(x) = \sum_{i=0}^{n} a_i N_i(x)$$

zu $n+1$ Stützstellen gegeben und wird noch eine Stützstelle (x_{n+1}, y_{n+1}) hinzugefügt, so muss nur der neue *führende* Koeffizient $a_{n+1} = y[x_0, ..., x_{n+1}]$ ($\mathcal{O}(n)$ a.Op.) und das Basispolynom $N_{n+1}(x) = N_n(x)(x - x_n)$ ($\mathcal{O}(1)$ a.Op.) neu berechnet werden.

Neville-Schema Schließlich kann die Relation (5.12) zu einer effizienten Auswertung des Interpolationspolynoms p an einer einzelnen Stelle $\xi \in [a, b]$ verwendet werden. Es gilt nämlich

$$\begin{aligned} p_{k,k+l}(\xi) &= \frac{(\xi - x_k) p_{k+1,k+l}(\xi) - \overbrace{(\xi - x_{k+l})}^{=\xi - x_k + x_k - x_{k+l}} p_{k,k+l-1}(\xi)}{x_{k+l} - x_k} \\ &= p_{k,k+l-1}(\xi) + (\xi - x_k)\frac{p_{k+1,k+l}(\xi) - p_{k,k+l-1}(\xi)}{x_{k+l} - x_k}. \end{aligned}$$

Dies führt auf das sogenannte Neville-Schema:

Neville-Schema zur Berechnung von $p(\xi)$

Eingabe: Stützstellen $x_0, ..., x_n$, Stützwerte $y_0, ..., y_n$, Stelle $\xi \in [a, b]$

Für $k = 0, ..., n$:

$p_{k,k} := y_k$

Für $l = 1, ..., n$

Für $k = 0, ..., n - l$:

$$p_{k,k+l} := p_{k,k+l-1} + (\xi - x_k)\frac{p_{k+1,k+l} - p_{k,k+l-1}}{x_{k+l} - x_k}$$

Ausgabe: $p(\xi) = p_{0,n}$

Der Wert $p_{0,n}$ ist (bei exakter Arithmetik) dann der Wert des Polynoms $p(\xi)$. Das Neville-Schema ist sehr gut geeignet, wenn man das Interpolationspolynom nur an einzelnen Stellen ξ auswerten möchte. Bei jeder Auswertung sind $\frac{n(n-1)}{2}$ Multiplikationen und genau so viele Divisionen notwendig. Ist man an einer Auswertung an vielen Stellen interessiert, ist die Berechnung der Koeffizienten mithilfe von dividierten Differenzen effizienter.

Beispiel Wir betrachten die Stützstellenpaare

x_i	0	1	2	3
y_i	0	1	8	27

welche von der Funktion $f(x) = x^3$ abgegriffen werden. Wir berechnen den Wert des Interpolationspolynoms $p \in P_3$ an der Stelle $\xi = 0.5$. Die Zwischenwerte $p_{k,k+l}$ sind von links nach rechts in im folgenden Neville-Tableau gegeben.

$p_{00} = 0$	$p_{01} = 0.5$	$p_{02} = -0.25$	$p_{03} = 0.125$
$p_{11} = 1$	$p_{12} = -2.5$	$p_{13} = 2$	
$p_{22} = 8$	$p_{23} = -20.5$		
$p_{33} = 27$			

Der Endwert p_{03} ist genau der Wert von $f(0.5) = p(0.5)$, da Funktion und Interpolationspolynom in diesem Beispiel übereinstimmen. Interessant ist aber, dass die Zwischenwerte, welche die Werte $p_{k,k+l}(x)$ der Interpolationspolynome zu einem Teil der Stützstellen sind, teilweise weit von $p(0.5)$ abweichen. Dies ist besonders dann der Fall, wenn die Stelle $x = 0.5$ außerhalb des von den Stützpunkten aufgespannten Intervalls liegt. In die Approximation p_{23} gehen z.B. nur die weit entfernten Werten $(2, 8)$ und $(3, 27)$ ein.

5.1.3 Fehlerabschätzung bei der Interpolation von Funktionen

Wir nehmen nun an, dass die zu interpolierenden Stützwerte von einer Funktion f abgegriffen sind, d.h. dass die Stützstellenpaare

$$(x_0, f(x_0)), (x_1, f(x_1)), ..., (x_n, f(x_n)) \tag{5.13}$$

zu interpolieren sind.

Es gilt die folgende *a priori* Fehlerabschätzung:

Satz 5.7. *Sei $f \in C^{n+1}[a,b]$ und sei $p_n \in P_n$ das zugehörige Interpolationspolynom durch die Stützstellenpaare* (5.13) *mit paarweise verschiedenen Stützstellen $x_i \in [a,b], i = 0,...,n$. Dann gibt es zu jedem $x \in [a,b]$ ein $\xi_x \in (a,b)$, so dass gilt*

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{j=0}^{n} (x - x_j). \tag{5.14}$$

Insbesondere gilt

$$|f(x) - p_n(x)| \leq \frac{\sup_{\xi \in (a,b)} |f^{(n+1)}(\xi)|}{(n+1)!} \prod_{j=0}^{n} |x - x_j|. \tag{5.15}$$

Beweis. Im Fall $x = x_j$ für eine Stützstelle x_j verschwinden beide Seiten in (5.14) und das Resultat folgt trivialerweise. Sei also $x \neq x_j, j = 0,...,n$. Wir definieren die Funktion $F \in C^{n+1}[a,b]$ durch die Zuordnung

$$F(t) := f(t) - p_n(t) - K(x) \prod_{j=0}^{n} (t - x_j), \qquad K(x) = \frac{f(x) - p_n(x)}{\prod_{j=0}^{n} (x - x_j)}.$$

$K(x)$ ist dabei so bestimmt, dass die Funktion $F(t)$ eine Nullstelle in x hat

$$F(x) = 0. \tag{5.16}$$

Damit besitzt F mindestens die (n+2) Nullstellen $x_0, ..., x_n, x$ in $[a,b]$. Nach dem Satz von Rolle hat die Ableitung F' dann mindestens $(n+1)$ Nullstellen in $[a,b]$ und die $(n+1)$-te Ableitung $F^{(n+1)}$ noch mindestens eine Nullstelle $\xi_x \in (a,b)$. Für diese gilt

$$0 = F^{(n+1)}(\xi_x) = f^{(n+1)}(\xi_x) - \underbrace{p_n^{(n+1)}(\xi_x)}_{=0} - K(x)(n+1)!$$

und damit

$$K(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}.$$

Mit (5.16) gilt

$$0 = F(x) = f(x) - p_n(x) - \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{j=0}^{n} (x - x_j).$$

□

Wir wollen die Fehlerabschätzung (5.15), d.h.

$$|f(x) - p_n(x)| \leq \frac{\sup_{\xi \in (a,b)} |f^{(n+1)}(\xi)|}{(n+1)!} \prod_{j=0}^{n} |x - x_j|.$$

kurz diskutieren. Für wachsendes n fällt der Term $\frac{1}{(n+1)!}$ sehr schnell. Der Term $\prod_{j=0}^{n}|x-x_j|$ fällt, wenn Stützstellen x_n nahe bei x hinzugefügt werden. In jedem Fall gilt

$$\prod_{j=0}^{n}|x-x_j| \le |b-a|^{n+1}.$$

Bei gleichmäßig beschränkten Ableitungen $f^{(n+1)}$ liegt wegen $\frac{|b-a|^{n+1}}{(n+1)!} \to 0\,(n\to\infty)$ unabhängig von der Stützstellenwahl Konvergenz vor

$$|f(x)-p_n(x)| \to 0 \qquad (n\to\infty).$$

Sind die Ableitungen von f dagegen nicht gleichmäßig beschränkt, etwa bei der Funktion

$$f(x)=\frac{1}{1-x^2}, \quad |f^{(n)}(x)| \sim \left(\frac{2|x|}{|1-x^2|}\right)^n n!$$

oder ist die Funktion nicht genügend regulär, so kann für $n\to\infty$ keine Konvergenz erwartet werden.

Wir betrachten als Spezialfall noch eine *äquidistante* Wahl der Stützstellen, d.h. $x_i = a+\frac{b-a}{n}i$. Für das Produkt $\prod_{j=0}^{n}|x-x_j|$ gilt dann mit $h := \frac{b-a}{n}$

$$\prod_{j=0}^{n}|x-x_j| \le h\cdot h\cdot 2h\cdot 3h\cdots nh = h^{n+1}n! = \left(\frac{b-a}{n}\right)^{n+1} n!\,.$$

Mit der Fehlerabschätzung (5.15) gilt dann sogar

$$|f(x)-p_n(x)| \le \sup_{\xi\in(a,b)} \frac{|f^{(n+1)}(\xi)|}{n+1}\left(\frac{b-a}{n}\right)^{n+1}.$$

Auch hier ist Konvergenz für $n\to\infty$ aber nur sichergestellt, wenn die Funktion glatt ist ($f\in C^\infty[a,b]$) und die Ableitungen für $n\to\infty$ beschränkt sind oder nur "moderat anwachsen".

Beispiel 5.8. *Wir interpolieren die Funktion $f(x)=|x|$ auf dem Intervall $[a,b]=[-1,1]$ mit den äquidistanten Stützstellen $x_i=-1+\frac{2i}{n}$, $i=0,...,n$. In Abb. 5.1 zeigen wir die Interpolationspolynome p_4, p_8, p_{16} und p_{32}. In diesem Fall divergiert das Interpolationspolynom p_n für $n\to\infty$ und es entstehen starke Oszillationen in der Nähe der Intervallenden für wachsendes n. Bei Funktionen, die nicht regulär sind, oder deren Ableitungen nicht gleichmäßig beschränkt ist, ist also die Verwendung eines Interpolationspolynoms hoher Ordnung nicht zu empfehlen.*

Als nächstes geben wir noch eine Fehlerschätzung mit alternativem Restglied an. Dazu führen wir für die dividierten Differenzen, die zu Stützstellen $(x_i, f(x_i)), i=0,...,n$, gehören (d.h. $y_i=f(x_i)$)) folgende Notation ein

$$\begin{aligned}
f[x] &:= f(x), \qquad f[x_i] := f(x_i),\ i=0,...,n,\\
f[x_k,...,x_{k+l}] &:= \frac{f[x_{k+1},...,x_{k+l}]-f[x_k,...,x_{k+l-1}]}{x_{k+l}-x_k}, \quad 0\le k<k+l\le n,\\
f[x,x_k,...,x_{k+l}] &:= \frac{f[x_k,...,x_{k+l}]-f[x,x_k,...,x_{k+l-1}]}{x_{k+l}-x}, \quad 0\le k\le k+l\le n.
\end{aligned}$$

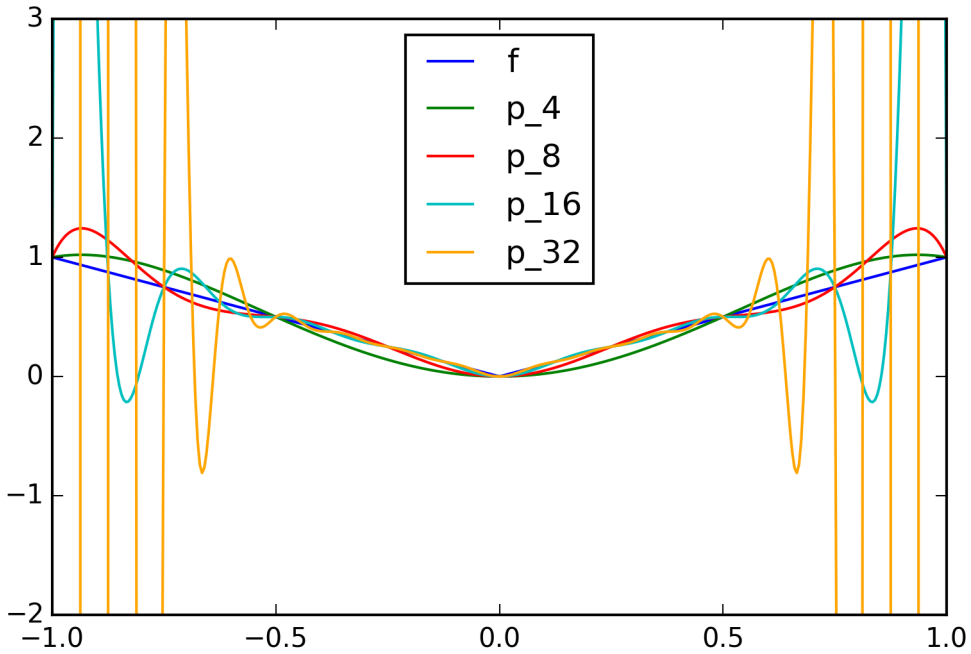

Abbildung 5.1: Polynominterpolation der Funktion $f(x) = |x|$ mit Polynomgrad $n = 4, 8, 16, 32$.

Satz 5.9. *Unter den Voraussetzungen von Theorem 5.7 gilt für $x \in [a, b], x \neq x_j, j = 0, ..., n$ die Fehlerdarstellung*

$$f(x) - p_n(x) = f[x, x_0, ..., x_n] \prod_{j=0}^{n} (x - x_j). \tag{5.17}$$

Beweis. Wir zeigen die Aussage des Satzes induktiv nach Anzahl der Stützstellen. Für $n = 0$ gilt $p_0(x) = f(x_0)$ und es gilt nach Definition der Dividierten Differenzen

$$f(x) - p_0(x) = f(x) - f(x_0) = f[x, x_0](x - x_0).$$

Sei die Behauptung nun richtig für $n - 1$, d.h. es gelte

$$f(x) - p_{n-1}(x) = f[x, x_0, ..., x_{n-1}] \prod_{j=0}^{n-1} (x - x_j) \tag{5.18}$$

für

$$p_{n-1}(x) = \sum_{i=0}^{n-1} f[x_0, ..., x_i] \prod_{j=0}^{i-1} (x - x_j).$$

Nach Definition von p_n und der Induktionsvoraussetzung (5.18) gilt dann

$$\begin{aligned} f(x) - p_n(x) &= f(x) - \sum_{i=0}^{n} f[x_0, ..., x_i] \prod_{j=0}^{i-1}(x - x_j) \\ &= f(x) - p_{n-1}(x) - f[x_0, ..., x_n] \prod_{j=0}^{n-1}(x - x_j) \\ &\overset{(5.18)}{=} f[x, x_0, ..., x_{n-1}] \prod_{j=0}^{n-1}(x - x_j) - f[x_0, ..., x_n] \prod_{j=0}^{n-1}(x - x_j) \\ &= (f[x, x_0, ..., x_{n-1}] - f[x_0, ..., x_n]) \prod_{j=0}^{n-1}(x - x_j) \\ &= \frac{f[x, x_0, ..., x_{n-1}] - f[x_0, ..., x_n]}{x - x_n} \prod_{j=0}^{n}(x - x_j). \end{aligned}$$

Die Induktionsbehauptung für n folgt mittels der Definition der Dividierten Differenzen. □

Vergleich der beiden Fehlerdarstellungen aus Satz 5.7 und 5.9 ergibt den folgenden Ausdruck für die dividierte Differenz

$$f[x, x_0, ..., x_n] = \frac{f^{(n+1)}(\xi)}{(n+1)!}, \quad \xi \in (a, b).$$

Die dividierten Differenzen sind also Näherungen der Taylor-Koeffizienten einer Funktion f.

5.1.4 Fehlerempfindlichkeit der Interpolationsaufgabe

Die Lagrangesche Interpolationsaufgabe ist sehr sensitiv bezüglich kleiner Störungen der Daten. Gestörte Daten $\tilde{y}_i = y_i + \epsilon_i$ wirken sich global auf das Interpolationspolynom p_n aus und können dieses dramatisch verändern, insbesondere für großes n.

Beispiel 5.10. *Wir betrachten die Polynominterpolation der Funktion $f \equiv 0$ auf dem Intervall $[-1, 1]$ mit äquidistanten Stützstellen $x_i = -1 + \frac{2i}{n}, i = 0, ..., n$ für gerades n. Das zugehörige Interpolationspolynom ist trivialerweise $p_n = f \equiv 0$.*

*Nun betrachten wir eine **Störung** des Stützwertes zur Stützstelle $x_{n/2} = 0$ um $\epsilon = 0.01$. Wir berechnen das Interpolationspolynom zu den Stützwerten $y_{n/2} = \epsilon = 0.01$ und $y_i = f(x_i) = 0$ sonst. Dieses lautet in Lagrangescher Darstellung*

$$p_n(x) = \epsilon L^n_{n/2}(x) = \epsilon \prod_{j=0, j\neq n/2}^{n} \frac{x - x_j}{x_j}.$$

Dieses wächst für $x \neq 0$ sehr schnell für $n \to \infty$ (siehe Abb. 5.2). Wir sehen, dass die Störung in $x = 0$ Einfluss auf die Interpolation im kompletten Intervall hat, insbesondere in der Nähe der Intervallenden. Wieder divergiert die Polynominterpolation für $n \to \infty$.

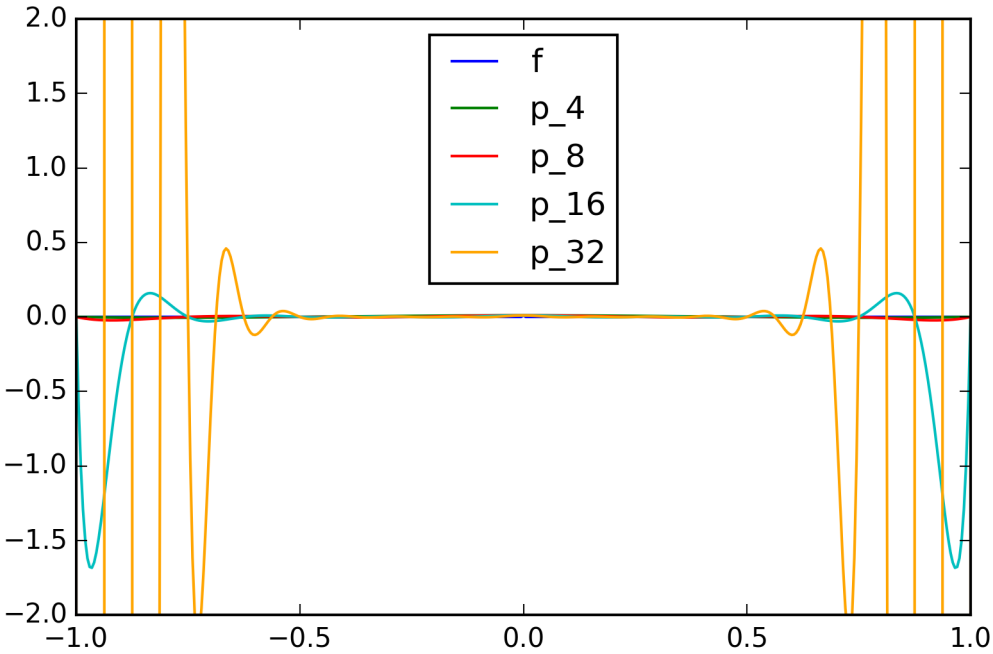

Abbildung 5.2: Polynominterpolation der Funktion $f(x) = 0$ mit einem gestörten Stützwert bei verschiedenem Polynomgrad n.

Das beobachtete Verhalten der Polynominterpolation an den Intervallrändern wird *Runge's Phänomen* genannt. Dieses lässt sich dadurch erklären, dass jedes Polynom $p \neq 0$ für $x \to \infty$ gegen $\pm\infty$ strebt. Anstatt eine Funktion mit f durch ein Polynom p_n mit großem Polynomgrad n anzunähern, ist in der Praxis daher meist eine *stückweise Interpolation* mit Polynomen kleineren Grades sinnvoll, siehe Abschnitt 5.4 unten.

5.2 Hermite-Interpolation

Die Hermitesche Interpolationsaufgabe ist eine Verallgemeinerung der Lagrangeschen Interpolationsaufgabe, bei der in den Stützpunkten x_i zusätzlich gefordert werden kann, dass das Interpolationspolynom neben den Funktionswerten $f(x_i)$ auch Ableitungswerte $f'(x_i), ..., f^{(\mu_i)}(x_i)$ annimmt ($\mu_i \geq 0$).

Definition 5.11. *(Hermite-Interpolationsaufgabe) Zu (m+1) paarweise verschiedenen Stützstellen* $x_0, \ldots, x_m \in [a, b]$, *Ableitungsgraden* $\mu_0, \ldots, \mu_m \geq 0$ *und Stützwerten* $y_{i,k}, i = 0, ..., m, k = 0, ..., \mu_i$ *ist ein Polynom* $p \in P_n$ *mit* $n + 1 = \sum_{i=0}^{m}(\mu_i + 1)$ *so zu bestimmen, dass*

$$p^{(k)}(x_i) = y_{i,k}, \quad i = 0, \ldots, m,\ k = 0, ..., \mu_i. \tag{5.19}$$

Analog zu den Sätzen 5.1 und 5.7 beweist man

Satz 5.12. *Die Hermite-Interpolationsaufgabe hat eine eindeutig bestimmte Lösung $p \in P_n$. Sind die Stützwerte $y_{i,k}, i = 0, ..., m, k = 0, ..., \mu_i$ von einer Funktion $f \in C^{n+1}[a, b]$ abgegriffen, d.h. es gilt*

$$y_{i,k} = f^{(k)}(x_i),\ i = 0, ..., m,\ k = 0, ..., \mu_i,$$

dann gilt die Fehlerdarstellung

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^{m} (x - x_i)^{\mu_i+1}$$

mit einem $\xi_x \in [a, b]$.

Beweis. [4], Aufgabe 2.5. □

Das Hermite-Interpolationspolynom lässt sich am einfachsten in der Newton-Basis angeben. Es gilt

$$p(x) = \sum_{i=0}^{m} \sum_{k=1}^{\mu_i+1} y[\underbrace{x_0, ..., x_0}_{\mu_0+1 \text{ mal}}, ..., \underbrace{x_{i-1}, ..., x_{i-1}}_{\mu_{i-1}+1 \text{ mal}}, \underbrace{x_i, ..., x_i}_{k \text{ mal}}] \cdot \left(\prod_{j=0}^{i-1} (x - x_j)^{\mu_j+1} \right) (x - x_i)^{k-1},$$

wobei die Dividierten Differenzen bei gleichen Indizes folgendermaßen modifiziert werden, damit mehrfache Stützstellen überhaupt möglich sind

$$y[\underbrace{x_i, ..., x_i}_{k \text{ mal}}] = \frac{y_{i,k-1}}{(k-1)!}.$$

Die übrigen Dividierten Differenzen definiert man dann wie in Kapitel 5.1.2, wobei auf der rechten Seite immer der erste bzw. der letzte Stützpunkt in der Klammer wegfallen

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

5.3 Extrapolation

In vielen Anwendungen ist man nicht an Werten $f(\xi)$ interessiert, bei denen die Stelle ξ zwischen den Stützpunkten $x_0, ..., x_n$ liegt, sondern außerhalb, d.h.

$$\xi < \min_{i=0,...,n} x_i, \qquad \text{oder } \xi > \max_{i=0,...,n} x_i.$$

Möglicherweise ist der Wert $f(\xi)$ auch gar nicht definiert und man sucht eine Näherung für $\lim_{x \to \xi} f(x)$. In diesem Fall spricht man von einer *Extrapolation* der Stützwerte $y_0, ..., y_n$. Die obigen Beispiele zeigen, dass eine Extrapolation nicht unbedingt zu sinnvollen Ergebnissen führt (siehe z.B. die Oszillationen an den Intervallenden bei Beispiel 5.10). In diesem Abschnitt werden wir Beispiele kennenlernen, bei denen die Konvergenz der Extrapolation sichergestellt ist.

Ein wichtiges Beispiel ist die numerische Differentiation. Ist f differenzierbar in x, so ist nach Definition

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

Kann der Differenzenquotient

$$a_1(h) := \frac{f(x+h) - f(x)}{h}$$

auf der rechten Seite für verschiedene $h > 0$ ausgewertet werden, so kann die Extrapolation genutzt werden, um den Wert für $h \to 0$ zu approximieren.

5.3.1 Numerische Differentiation

Zunächst beschäftigen wir uns aber mit der Frage, wie gut der oben genannte einseitige Differenzenquotient und der zentrale Differenzenquotient

$$a_2(h) := \frac{f(x+h) - f(x-h)}{2h}$$

die erste Ableitung approximieren. Es gilt folgender Satz.

Satz 5.13. *Sei $f \in C^3[a,b]$. Es gilt für den einseitigen Differenzenquotienten*

$$\frac{f(x+h) - f(x)}{h} = f'(x) + \frac{h}{2} f''(x) + \frac{h^2}{6} f'''(\xi_{x,h}) \tag{5.20}$$

mit einer Zwischenstelle $\xi_{x,h} \in [x, x+h]$, die von x und h abhängt. Für den zentralen Differenzenquotienten gilt bei $f \in C^5[a,b]$

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{6} f'''(x) + \mathcal{O}(h^4). \tag{5.21}$$

Ist f analytisch, so gelten für genügend kleines h die Reihenentwicklungen

$$\frac{f(x+h) - f(x)}{h} = f'(x) + \sum_{k=2}^{\infty} \frac{f^{(k)}(x)}{k!} h^{k-1}, \tag{5.22}$$

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \sum_{k=1}^{\infty} \frac{f^{(2k+1)}(x)}{(2k+1)!} h^{2k}. \tag{5.23}$$

Beweis. Übung. □

Definition 5.14. *Eine Approximation $a(h)$ konvergiert für $h \to 0$ mit Ordnung k gegen $a \in \mathbb{R}$, wenn mit einer Konstante $c \in \mathbb{R}$ asymptotisch für $h \to 0$ gilt*

$$|a(h) - a| \leq c h^k.$$

Im Falle $k = 1$ reden wir von linearer Konvergenz in h, für $k = 2$ von quadratischer und für $k = 3$ von kubischer Konvergenz.

Der einseitige Differenzenquotient konvergiert also *linear* gegen $f'(x)$, der zentrale Differenzenquotient *quadratisch*.

Wir bemerken noch, dass $a_1(h)$ und $a_2(h)$ die ersten Ableitungen des linearen Interpolationspolynoms p durch die Stützstellen $(x, f(x))$ und $(x+h, f(x+h))$ bzw. $(x-h, f(x-h))$ und $(x+h, f(x+h))$ sind. Mithilfe der Polynominterpolation können weitere Approximationen an die Ableitung $f'(x) \approx p'(x)$ konstruiert werden. Wir wollen diesen Ansatz hier aber nicht weiter diskutieren und stattdessen die Extrapolation der beiden wichtigsten Differenzenquotienten $a_1(h)$ und $a_2(h)$ im Limes $h \to 0$ untersuchen.

5.3.2 Richardson-Extrapolation zum Limes

Die Idee bei der Richardson-Extrapolation zum Limes liegt darin ein Interpolationspolynom durch die Stützwerte $(h_0, a(h_0)), ..., (h_n, a(h_n))$ zu legen und dieses für $h = 0$ auszuwerten. Wir untersuchen diesen Prozess zunächst anhand eines einfachen Beispiels.

Beispiel 5.15. *Wir berechnen den Wert der Ableitung von $f(x) = \tanh(x)$ an der Stelle $x = 0.5$. Es gilt $f'(0.5) \approx 0.78644773$. Auswertung des Differenzenquotienten*

$$a_1(h) = \frac{\tanh(x+h) - \tanh(x)}{h}$$

für $h = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$ ergibt

$$a\left(\frac{1}{2}\right) \approx 0.5989, \quad a\left(\frac{1}{4}\right) \approx 0.6921, \quad a\left(\frac{1}{8}\right) \approx 0.7399, \quad a\left(\frac{1}{16}\right) \approx 0.7634.$$

Das Neville-Schema zur Auswertung des Interpolationspolynoms im Punkt $\xi = 0$ lautet

$$p_{k,k} := a(h_k), \qquad p_{k,k+l} := p_{k,k+l-1} - h_k \frac{p_{k+1,k+l} - p_{k,k+l-1}}{h_{k+l} - h_k} \quad \text{für } l > 0$$

Die Berechnung mittels des Interpolationspolynom $p \in P_3$ zu den 4 Stützpunkten führt dann zu folgendem Tableau

Wir sehen, dass die extrapolierten Werte viel näher an der exakten Lösung sind als der Wert $a\left(\frac{1}{16}\right)$ für das kleinste h. Schon der Wert p_{23} des linearen Polynoms durch die beiden Stützstellen $h_2 = \frac{1}{8}$ und $h_3 = \frac{1}{16}$ liefert eine große Verbesserung. In diesem Beispiel zahlt sich die Extrapolation also aus.

Wie wollen nun die Konvergenzbeschleunigung bei der Approximation mit dem *rechtsseitigen* Differenzenquotienten von Funktionen $f \in C^3[a, b]$ analytisch untersuchen. Nach (5.13) gilt

$$a_1(h) = \frac{f(x+h) - f(x)}{h} = f'(x) + \frac{h}{2} f''(x) + \frac{h^2}{6} f'''(\xi_{x,h}) \tag{5.24}$$

mit einer Zwischenstelle $\xi_{x,h} \in [x, x+h]$, die sowohl von x als auch von h abhängt. Die Approximation mit dem Differenzenquotienten $a_1(h)$ ist also wie oben bereits bemerkt von erster Ordnung in h (lineare Konvergenz).

$p_{00} = \underline{0}.5989$	$p_{01} = \underline{0.78}53$	$p_{02} = \underline{0.78}836$	$p_{03} = \underline{0.786}5$
$p_{11} = \underline{0}.6921$	$p_{12} = \underline{0.7}976$	$p_{13} = \underline{0.786}73$	
$p_{22} = \underline{0.7}399$	$p_{23} = \underline{0.786}9$		
$p_{33} = \underline{0.7}634$			

$e_{00} = 1.9 \cdot 10^{-1}$	$e_{01} = 1.1 \cdot 10^{-3}$	$e_{02} = 1.9 \cdot 10^{-3}$	$e_{04} = 5.9 \cdot 10^{-4}$
$e_{11} = 8.6 \cdot 10^{-2}$	$e_{12} = 1.2 \cdot 10^{-3}$	$e_{13} = 2.9 \cdot 10^{-4}$	
$e_{22} = 4.6 \cdot 10^{-2}$	$e_{23} = 4.5 \cdot 10^{-4}$		
$e_{33} = 2.3 \cdot 10^{-2}$			

Tabelle 5.1: *Oben*: Neville-Tableau der Extrapolation zur Berechnung der Ableitung $\tanh'(0.5)$. Korrekte Dezimalstellen sind unterstrichen. *Unten*: Fehler $e_{ij} = |p_{ij} - \tanh'(0.5)|$

Sie nun $p_1 \in P_1$ das lineare Interpolationspolynom durch die Stützstellen $(h, a_1(h))$ und $(\frac{h}{2}, a_1(\frac{h}{2}))$. In Lagrangescher Darstellung lautet dieses

$$p(t) = \frac{t - \frac{h}{2}}{h - \frac{h}{2}} a_1(h) + \frac{t - h}{\frac{h}{2} - h} a_1\left(\frac{h}{2}\right).$$

Die Approximation $p(0)$ an $a_1(0)$ lautet dann

$$p(0) = -a_1(h) + 2a_1\left(\frac{h}{2}\right).$$

Mit der Taylorentwicklung (5.24) für h und $\frac{h}{2}$ gilt dann

$$\begin{aligned} p(0) &= -\left(f'(x) + \frac{h}{2} f''(x) + \frac{h^2}{6} f'''(\xi_{x,h})\right) + 2\left(f'(x) + \frac{h}{4} f''(x) + \frac{h^2}{24} f'''(\xi_{x,h/2})\right) \\ &= f'(x) + \mathcal{O}(h^2). \end{aligned}$$

Die Approximation $p(0)$ konvergiert also quadratisch (in h) gegen $f'(x)$. Wir werden weiter unten im Rahmen eines allgemeinen Resultates sehen, dass die Approximationsordnung bei Verwendung von weiteren Stützstellen weiter gesteigert werden kann.

Zunächst untersuchen wir allerdings noch das Konvergenzverhalten der Extrapolation bei Verwendung des zentralen Differenzenquotienten

$$a_2(h) = \frac{f(x + h) - f(x - h)}{2h} = f'(x) + \mathcal{O}(h^2). \tag{5.25}$$

Dieser weist bereits ohne Extrapolation eine Konvergenzordnung von 2 auf. Für analytisches f gilt nach Satz 5.13 sogar die Reihenentwicklung

$$a_2(h) = f'(x) + \sum_{i=1}^{\infty} \frac{f^{(2i+1)}(x)}{(2i)!} h^{2i}. \tag{5.26}$$

a_2 ist also eine *gerade* Funktion in h. In diesem Fall macht es Sinn für das Interpolationspolynom auch nur gerade Polynome in h zu verwenden. Das folgende Lemma zeigt, dass die Interpolationsaufgabe auch in diesem Fall wohldefiniert ist.

Lemma 5.16. *Sei $q > 0$ und seien $h_0, ..., h_n$ paarweise verschiedene positive Stützstellen. Ein Polynom $p_n \in P_n$ ist durch die Vorgabe von Werten*

$$p_n(h_i^q) = a(h_i), \quad i = 0, ..., n$$

eindeutig bestimmt. In "Lagrangescher" Darstellung lässt sich das Interpolationspolynom folgendermaßen schreiben

$$p_n(h^q) = \sum_{i=0}^{n} a(h_i) \prod_{j=0, j\neq i}^{n} \frac{h^q - h_j^q}{h_i^q - h_j^q}.$$

Beweis. Durch die Substitutionen $z_i := h_i^q$ und $z := h^q$ ergibt sich eine Lagrangesche Interpolationsaufgabe mit Stützwerten $y_i = a(h_i)$, $i = 0, ..., n$. Die Aussagen folgen aus Satz 5.2 und (5.5). □

Besitzt $a(h)$ z.B. eine gerade Reihenentwicklung, so ist $q = 2$ zu wählen.

Beispiel 5.17. *(Zentraler Differenzenquotient) Wir approximieren wieder den Wert der Ableitung von $f(x) = \tanh(x)$ an der Stelle $x = 0.5$, dieses Mal mit dem zentralen Differenzenquotienten*

$$a_2(h) := \frac{f(x+h) - f(x-h)}{2h}.$$

Es gilt $f'(0.5) \approx 0.78644773$.

(i) Wir ignorieren zunächst die Tatsache, dass $a_2(h)$ eine Reihenentwicklung in geraden Potenzen besitzt (5.23) *und bilden das Interpolationspolynom zu den Stützstellen $(h_i, a_2(h_i))$ mit $h_i = 2^{-i-1}$, $i = 0, ..., 3$. Das Neville-Schema führt zum Tableau in Tabelle 5.2.*

$p_{00} = \underline{0.7}616$	$p_{01} = \underline{0.7}993$	$p_{02} = \underline{0.786}20$	$p_{03} = \underline{0.7864}593$
$p_{11} = \underline{0.78}05$	$p_{12} = \underline{0.78}95$	$p_{13} = \underline{0.7864}3$	
$p_{22} = \underline{0.78}50$	$p_{23} = \underline{0.78}72$		
$p_{33} = \underline{0.786}1$			

Tabelle 5.2: Extrapolation zur Berechnung der Ableitung $\tanh'(0.5)$ mithilfe des zentralen Differenzenquotienten ohne Berücksichtigung der Reihenentwicklung in "gerade" Potenzen. Korrekte Dezimalstellen sind unterstrichen.

Wir sehen, dass die Werte schon deutlich besser sind als bei Verwendung des einseitigen Differenzenquotienten. Allerdings liefert die zweite und die vierte Spalte keine Verbesserung gegenüber der ersten und dritten. Dies liegt daran, dass wir hier von einem geraden Polynom der Ordnung 0

zu einem ungeraden der Ordnung 1 bzw. von der Ordnung 2 zur Ordnung 3 übergehen. Ungerade Polynomgrade sind in der Entwicklung (5.23) *aber nicht vorhanden.*

(ii) Als nächstes bilden wir nun das Interpolationspolynom $p \in P_n$ zu den Stützstellen $(h_i^2, a_2(h_i))$. Das Neville-Schema lautet in diesem Fall

$$p_{kk} = a(h_k), \quad k = 0, ..., n$$
$$p_{k,k+l} = p_{k,k+l-1} - h_k^2 \frac{p_{k+1,k+l} - p_{k,k+l-1}}{h_{k+l}^2 - h_k^2}, \quad 0 \le k < k + l \le n.$$

Das zugehörige Neville-Tableau ist in Tabelle 5.3 gegeben.

$p_{00} = \underline{0.7}616$	$p_{01} = \underline{0.786}7494$	$p_{02} = \underline{0.7864}537$	$p_{03} = \underline{0.786447}44$
$p_{11} = \underline{0.78}05$	$p_{12} = \underline{0.7864}722$	$p_{13} = \underline{0.78644}783$	
$p_{22} = \underline{0.78}50$	$p_{23} = \underline{0.7864}493$		
$p_{33} = \underline{0.786}1$			
$e_{00} = 2.4 \cdot 10^{-2}$	$e_{01} = 3.0 \cdot 10^{-4}$	$e_{02} = 6.0 \cdot 10^{-6}$	$e_{03} = 1.4 \cdot 10^{-8}$
$e_{11} = 5.9 \cdot 10^{-3}$	$e_{12} = 2.5 \cdot 10^{-5}$	$e_{13} = 1.0 \cdot 10^{-7}$	
$e_{22} = 1.4 \cdot 10^{-3}$	$e_{23} = 1.6 \cdot 10^{-6}$		
$e_{33} = 3.5 \cdot 10^{-4}$			

Tabelle 5.3: Extrapolation zur Berechnung der Ableitung $\tanh'(0.5)$ mithilfe des zentralen Differenzenquotienten mit Berücksichtigung der Reihenentwicklung in "gerade" Potenzen ($q = 2$). *Oben*: Neville-Tableau. Korrekte Dezimalstellen sind unterstrichen. *Unten*: Fehler $e_{ij} = |p_{ij} - \tanh'(0.5)|$.

Nun sehen wir wesentliche Verbesserungen in jeder Spalte. Nach Satz 5.13 konvergiert die erste Spalte mit 2-ter Ordnung für $h \to 0$. In Satz 5.18 werden wir zeigen, dass sich die Konvergenzordnung in jeder Spalte um $q = 2$ erhöht. Der Wert p_{03} ist also von der Ordnung $\mathcal{O}(h_0^8)$.

Der folgende Satz liefert die Grundlage dafür, dass die Konvergenzordnung durch Extrapolation gesteigert werden kann.

Satz 5.18. *Für die Funktion $a(h) : (0, 1) \to \mathbb{R}$ gelte asymptotisch für $h \to 0$ eine Reihenentwicklung der Form*

$$a(h) = a_0 + \sum_{j=1}^{n} a_j h^{jq} + a_{n+1}(h) h^{(n+1)q} \tag{5.27}$$

mit einem $q > 0$, Koeffizienten $a_j \in \mathbb{R}, j = 0, ..., n$ und einem nach oben beschränkten Restglied $|a_{n+1}(h)| < c$ für $h \le h_0$. Weiter sei $(h_k)_{k \ge 0}$ eine monoton fallende Folge positiver Zahlen

$$0 < \frac{h_{k+1}}{h_k} \le 1 - \epsilon$$

mit $\epsilon > 0$. Für das Interpolationspolynom $p_{k,k+n} \in P_n$ durch $(h_k^q, a(h_k)), ..., (h_{k+n}^q, a(h_{k+n}))$ gilt dann

$$|p_{k,k+n}(0) - a(0)| = \mathcal{O}\left(h_k^{(n+1)q}\right).$$

Beweis. Wir schreiben $z = h^q$ und $z_k = h_k^q$. Das Interpolationspolynom durch $(z_k, a(h_k)), ...,$ $(z_{k+n}, a(h_{k+n}))$ hat in Lagrangescher Darstellung die Form

$$p_{k,k+n}(z) = \sum_{i=0}^{n} a(h_{k+i}) L_{k+i}^n(z), \qquad L_{k+i}^n(z) = \prod_{l=0, l\neq i}^{n} \frac{z - z_{k+l}}{z_{k+i} - z_{k+l}}.$$

Wie setzen die vorausgesetzte Reihendarstellung (5.27) ein und werten das Polynom am Punkt $z = 0$ aus

$$\begin{aligned} p_{k,k+n}(0) &= \sum_{i=0}^{n} \left(a_0 + \sum_{j=1}^{n} a_j h_{k+i}^{jq} + a_{n+1}(h_{k+i}) h_{k+i}^{(n+1)q} \right) L_{k+i}^n(0) \\ &= a_0 \sum_{i=0}^{n} L_{k+i}^n(0) + \sum_{j=1}^{n} \left(a_j \sum_{i=0}^{n} z_{k+i}^j L_{k+i}^n(0) \right) + \sum_{i=0}^{n} a_{n+1}(h_{k+i}) z_{k+i}^{n+1} L_{k+i}^n(0). \end{aligned} \tag{5.28}$$

Wir analysieren nun die Terme

$$q_j(z) := \sum_{i=0}^{n} z_{k+i}^j L_{k+i}^n(z), \quad j = 0, ..., n.$$

Diese stellen gerade die Interpolationspolynome $q_j(z) \in P_n$ zur Funktion $f_j(z) = z^j$ und den Stützstellen $z_{k+i}, i = 0, ..., n$ dar. Da $j \leq n$ werden die Funktionen f_j exakt interpoliert, d.h. $q_j \equiv f_j$. Damit gilt insbesondere

$$1 = f_0(0) = q_0(0) = \sum_{i=0}^{n} L_{k+i}^n(0), \qquad 0 = f_j(0) = q_j(0) = \sum_{i=0}^{n} z_{k+i}^j L_{k+i}^n(0).$$

Setzen wir dies in (5.28) ein, so folgt

$$p_{k,k+n}(0) = a_0 + \sum_{i=0}^{n} a_{n+1}(h_{k+i}) z_{k+i}^{n+1} L_{k+i}^n(0). \tag{5.29}$$

Mit der Voraussetzung $0 < \frac{h_{k+1}}{h_k} \leq 1 - \epsilon$ für $k \in \mathbb{N}$ folgt auch $0 < \frac{z_{k+1}}{z_k} = \frac{h_{k+1}^q}{h_k^q} \leq 1 - \epsilon_2$ für ein $\epsilon_2 > 0$ und $\frac{z_k}{z_{k+1}} \geq 1 + \epsilon_3$ für ein $\epsilon_3 > 0$. Weiter gilt für beliebiges $j \neq k$, dass

$$\left| \frac{z_j}{z_k} - 1 \right| \geq \min\left\{ \left| \frac{z_{k+1}}{z_k} - 1 \right|, \left| \frac{z_{k-1}}{z_k} - 1 \right| \right\} \geq \min\{\epsilon_2, \epsilon_3\} =: \epsilon_4 > 0.$$

Für den letzten Term in (5.29) folgt

$$|L_{k+i}^n(0)| = \prod_{l=0, l\neq i}^{n} \left| \frac{z_{k+l}}{z_{k+i} - z_{k+l}} \right| = \prod_{l=0, l\neq i}^{n} \underbrace{\frac{1}{|z_{k+i}/z_{k+l} - 1|}}_{>\epsilon_4} \leq \epsilon_4^{-n} = c(\epsilon, n).$$

Dieser Term ist also unabhängig von h beschränkt. Mit der Voraussetzung $|a_{n+1}(h_{k+i})| \leq c$ und $|z_{k+i}^{n+1}| \leq |z_k^{n+1}| = h_k^{q(n+1)}$ folgt die Behauptung

$$|p_{k,k+n}(0) - a_0| \;\leq\; \sum_{i=0}^{n} |a_{n+1}(h_{k+i})|\,|z_{k+i}^{n+1}|\,|L_{k+i}^{n}(0)| \;\leq\; c_2(\epsilon, n) h_k^{q(n+1)}.$$

□

Für den rechtsseitigen Grenzwert gilt für eine analytische Funktion f nach (5.22)

$$a_1(h) = f'(x) + \sum_{k=2}^{\infty} \frac{f^{(k)}(x)}{k!} h^k.$$

Die Extrapolation mit einem Interpolationspolynom $p_n \in P_n$ konvergiert also nach Satz 5.18 mit $(n+1)$-ter Ordnung in h, während der rechtsseitige Differenzenquotienten alleine nur Konvergenz erster Ordnung aufweist. Für die Werte in Tabelle 5.1 gilt in der n-ten Spalte, die zu einem Interpolationspolynom der Ordnung $n-1$ korrespondiert, eine Abschätzung der Form $\mathcal{O}(h_k^n)$, wobei h_k die größte Stützstelle ist, welche verwendet wird. In der ersten Spalte beobachten wir folglich lineare Konvergenz, der Fehler halbiert sich von Zeile zu Zeile, aufgrunde von $h_{k+1} = \frac{1}{2} h_k$. In der zweiten Spalte beobachten wir quadratische Konvergenz, da sich der Fehler von Zeile zu Zeile etwa um einen Faktor 4 verringert ($h_{k+1}^2 = \frac{1}{4} h_k^2$).

Ist die Funktion f dagegen nur 3-mal stetig differenzierbar, so gilt nach (5.20) immerhin noch

$$a_1(h) = f'(x) + \frac{h}{2} f''(x) + \frac{h^2}{6} f'''(\xi_{x,h}) \tag{5.30}$$

mit einer Zwischenstelle $\xi_{x,h}$. Eine Extrapolation der Ordnung $n = 1$, d.h. zu 2 Stützstellen liefert hier die maximale Konvergenzordnung $n + 1 = 2$. Eine Hinzunahme weiterer Stützstellen bringt dagegen aufgrund des Restterms zweiter Ordnung in (5.30) keine weitere Erhöhung der Konvergenzordnung bei der Extrapolation.

Beim zentralen Differenzenquotienten sichert Satz 5.18 bei Verwendung der Stützstellen $(h_i^2, a_2(h_i))$ Konvergenz der Ordnung $(2n + 2)$, da

$$a_2(h) = f'(x) + \sum_{k=1}^{\infty} \frac{f^{(2k+1)}(x)}{(2k+1)!} h^{2k}.$$

Für die Werte in Tabelle 5.3 gilt in der n-ten Spalte, die zu einem Interpolationspolynom der Ordnung $n-1$ in h^2 korrespondiert, eine Abschätzung der Form $\mathcal{O}(h_k^{2n})$. In den ersten beiden Spalten kann man wieder anhand der Abnahme der Fehler um Faktoren von etwa $4 = 2^2$ bzw. $16 = 2^4$ von oben nach unten die Konvergenzordnungen 2 bzw. 4 beobachten. Auch hier hängt die maximal mögliche Konvergenzordnung allerdings von der Regularität der Funktion f ab.

Entwicklungen dieser Formen sind die Voraussetzung, dass die Extrapolation vom Limes eine höhere Konvergenzordnung ergibt. Diese lassen sich allerdings in vielen praktischen Fällen anwenden, etwa bei der numerischen Integration oder bei der Diskretisierung von Differentialgleichungen mit Diskretisierungsweite $h > 0$ (siehe Beispiel 2.1 aus dem ersten Teil der Vorlesung).

Bemerkung 5.19. *Die Schrittweitenbedingung* $\frac{h_{k+1}}{h_k} \leq 1 - \epsilon$ *ist wesentlich. Sie ist notwendig, um eine Schranke für* $|L^n_{k+i}(0)|$ *sicherzustellen. Eine erlaubte Schrittfolge ist z.B.* $h_k = \frac{h_0}{2^k}$*. Nicht erlaubt ist dagegen* $h_k = \frac{h_0}{k}$*, da in diesem Fall* $\lim_{k\to\infty} \frac{h_{k+1}}{h_k} = 1$*.*

5.4 Stückweise Interpolation

Aufgrund der oben genannten Probleme beim Übergang zu großen Polynomgraden n ist die Konvergenz der Polynominterpolation für $n \to \infty$ eher ein theoretisches Resultat. In der Praxis führt man die Polynominterpolation in der Regel *stückweise* durch. Dazu wird das Intervall $[a, b]$ zunächst in Teilintervalle $I_j = [x_{j-1}, x_j], j = 1, ..., N$ unterteilt mit

$$a = x_0 < x_1 < ... < x_{N-1} < x_N = b.$$

Auf jedem der Teilintervalle werden die Stützwerte nun mithilfe von Polynomen niederen Grades interpoliert.

5.4.1 Stückweise lineare Interpolation

Der einfachste Fall ist die stückweise lineare Interpolation mit Polynomen $p_j \in P_1(I_j)$. Wir suchen ein stückweise lineares Polynom im Raum

$$S^{1,0}_h[a, b] := \left\{ p \in C[a, b],\, p_{|I_j} \in P_1(I_j),\, j = 1, ..., N \right\},$$

welches die *Lagrangesche Interpolationsaufgabe* erfüllt, d.h.

$$p(x_i) = y_i, \qquad i = 0, ..., N. \tag{5.31}$$

Der Index h steht hierbei für die maximale Größe eines Teilintervalls $h := \max_{j=1,...,N} |x_j - x_{j-1}|$. Das lineare Polynom $p_j := p_{|I_j}$ ist natürlich durch die Vorgabe der beiden Werte (x_{j-1}, y_{j-1}) und (x_j, y_j) eindeutig bestimmt. Die Stetigkeit der so zusammengesetzten linearen Polynomen folgt aufgrund von $p_j(x_j) = y_j = p_{j+1}(x_j)$, siehe auch Abb. 5.3.

Mithilfe der Theorie aus Abschnitt 5.1 können wir für den Fall, dass die Stützwerte $y_i = f(x_i)$ zu einer Funktion $f \in C^2[a, b]$ gehören, direkt eine Fehlerabschätzung angeben.

Satz 5.20. *Sei* $f \in C^2[a, b]$ *und* $y_i := f(x_i)$*,* $i = 0, ..., N$ *zu* $(N + 1)$ *paarweise verschiedenen Stützstellen* $x_i, i = 0, ..., N$*. Für die eindeutig bestimmte Lösung* $p \in S^{1,0}_h[a, b]$ *der Lagrangeschen Interpolationsaufgabe* (5.31) *gilt für* $x \in [a, b]$

$$|f(x) - p(x)| \leq \frac{h^2}{2} \max_{\xi\in[a,b]} |f''(\xi)|.$$

Beweis. Wir wenden Theorem 5.7 stückweise auf jedem Intervall I_j an. Für $x \in I_j$ gilt

$$|f(x) - p(x)| \;\leq\; \max_{\xi\in I_j} \frac{|f''(\xi)|}{2} \underbrace{|x - x_{j-1}|}_{\leq h} \underbrace{|x - x_j|}_{\leq h} \leq \frac{h^2}{2} \max_{\xi\in I_j} |f''(\xi)|. \tag{5.32}$$

□

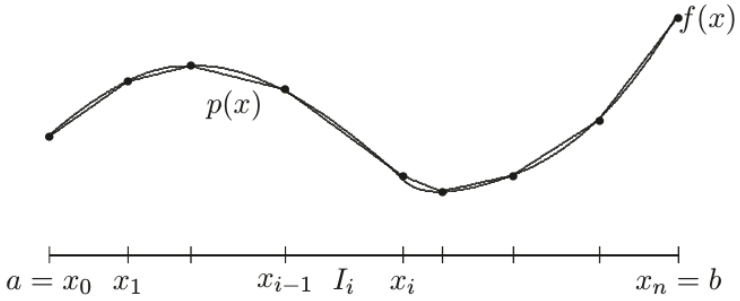

Abbildung 5.3: Stückweise lineare Interpolation einer Funktion f (Quelle:[4])

Wir erhalten quadratische Konvergenz für $h \to 0$ (d.h. $N \to \infty$). Hierzu ist "nur" die Beschränktheit der zweiten Ableitungen notwendig. Genau genommen muss die Funktion f sogar nur stückweise in jedem Teilintervall I_j regulär sein. Die Abschätzung (5.32) gilt dann trotzdem auf jedem Teilintervall separat. An einem Stützpunkt x_i muss die Funktion dagegen nur stetig sein.

5.4.2 Stückweise Interpolation vom Grad n

Anstatt linearer Polynome $p_j \in P_1(I_j)$ können auch Polynome vom Grad n angesetzt werden. Wir definieren den Raum

$$S_h^{n,0}[a,b] := \{p \in C[a,b],\, p_{|I_j} \in P_n(I_j),\, j = 1, ..., N\}.$$

Zur Formulierung einer wohlgestellten *Lagrangeschen Interpolationsaufgabe* zum Polynomgrad n müssen nun in jedem Intervall $I_j = [x_{j-1}, x_j]$ $n+1$ Bedingungen (für die $n+1$ Koeffizienten) gestellt werden. Dazu definieren wir $n+1$ Stützstellen in I_j

$$x_{j-1} = z_{j,0} < z_{j,1} < ... < z_{j,n-1} < z_{j,n} = x_j \tag{5.33}$$

und die Lagrangesche Interpolationsaufgabe

$$p(z_{j,k}) = y_{j,k} \quad j = 1, ..., N,\, k = 0, ..., n. \tag{5.34}$$

Auch diese ist wieder eindeutig lösbar (Satz 5.2 auf jedem Teilintervall). Mit Satz 5.7 gilt folgende Fehlerabschätzung.

Satz 5.21. *Es sei eine Zerlegung des Intervalls $[a,b]$ in N Teilintervalle $I_j = [x_{j-1}, x_j]$ gegeben sowie paarweise verschiedenen Stützstellen $z_{j,k} \in I_j$, $j = 1, ..., N, k = 0, ..., n$, welche laut (5.33) angeordnet sind. Weiter seien die Stützwerte $y_{j,k} = f(z_{j,k})$ von einer Funktion $f \in C^{n+1}[a,b]$ abgegriffen. Die eindeutig bestimmte Lösung $p \in S_h^{n,0}$ der Lagrangeschen Interpolationsaufgabe (5.34) erfüllt die Fehlerabschätzung*

$$|f(x) - p(x)| \leq \frac{h^{n+1}}{(n+1)!} \max_{\xi \in [a,b]} |f^{(n+1)}(\xi)|, \quad x \in [a,b].$$

Beweis. Anwendung von Satz 5.7 auf jedem Teilintervall I_j. □

Wieder ist es ausreichend, dass die Funktion f nur lokal regulär und global stetig ist. Es ergibt sich Konvergenz $(n+1)$-ter Ordnung für $h \to 0$, für eine stückweise kubische Interpolation in $S_h^{3,0}[a,b]$ also beispielsweise die Konvergenzordnung 4.

Bei der praktischen Berechnung der stückweisen Interpolation kann man intervallweise vorgehen, d.h. die jeweiligen Interpolationspolynome $p_j = p|_{I_j}$ für $j = 1, ..., N$ mit den Algorithmen aus Abschnitt 5.1 berechnen. Ist man nur an einzelnen Werten $p(x)$ des Interpolationspolynoms genügt natürlich das Berechnen des Polynoms p_j für den Index j mit $x \in I_j$ (z.B. mit Hilfe des Neville-Schemas).

5.4.3 Spline-Interpolation

Die Spline-Interpolation stellt eine Verallgemeinerung der oben betrachteten stückweisen Interpolation dar, bei der wir zusätzlich fordern, dass die stückweisen Polynome auch an den Stützstellen x_i gewisse Regularitätseigenschaften besitzen, d.h. $p \in C^r[a,b]$ für $r \geq 0$. Wir suchen den interpolierenden *Spline* im *Ansatzraum*

$$S_h^{n,r}[a,b] := \{p \in C^r[a,b],\, p_{|I_j} \in P_n(I_j),\, j = 1, ..., N\}.$$

Soll beispielsweise die Fahrbahn eines autonomen Fahrzeugs berechnet werden, so soll diese für einen gewissen Fahrkomfort keine *Zickzack*-Bewegungen beinhalten, weswegen die Stetigkeit der Ableitung gefordert wird. Darüber hinaus hat die Spline-Interpolation z.B. auch große Bedeutung in der Computergraphik bei der Erzeugung von *glatten* Kurven.

Diese Betrachtungen motivieren die folgenden Bedingungen für die sogenannte Spline-Interpolation.

Definition 5.22. *Seien $x_0, ..., x_N$ $(N+1)$ paarweise verschiedene Stützstellen und $y_0, ..., y_N$ die zugehörigen Stützwerte. Bei der Spline-Interpolation ist ein stückweises Polynom $p \in S_h^{n,r}[a,b]$ gesucht, welches die Lagrangesche Interpolationsaufgabe*

$$p(x_i) = y_i, \quad i = 0, ..., N \tag{5.35}$$

erfüllt.

Bemerkung 5.23. *Die Voraussetzung $p \in S_h^{n,r}[a,b]$ ist äquivalent dazu, dass ein stückweises Polynom*

$$p \in P_h^n[a,b] := \{p : [a,b] \to \mathbb{R},\, p_{|I_j} \in P_n(I_j), j = 1, ..., N\}$$

die Zusatzbedingungen

$$p_j(x_j) = p_{j+1}(x_j), \quad p_j'(x_j) = p_{j+1}'(x_j), \quad ... \quad p_j^{(r)}(x_j) = p_{j+1}^{(r)}(x_j) \tag{5.36}$$

für die Polynome $p_j := p_{|I_j} \in P_n(I_j)$ erfüllt.

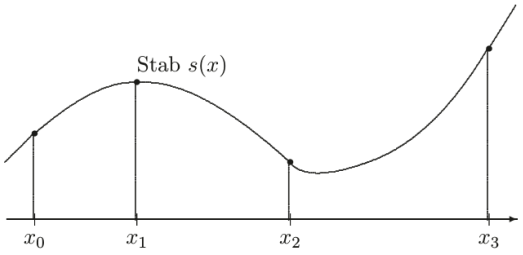

Abbildung 5.4: Spline-Interpolation (Quelle:[4])

Die Bezeichnung *spline* bezeichnet im Englischen einen *Biegestab*. Man stelle sich dazu einen elastischen Stab vor, der in $N+1$ Punkten fixiert ist, siehe auch Abb. 5.4. Neben der Fixierung in diesen Punkten kann man den Stab auch nicht beliebig knicken, ohne dass er bricht. Dies führt in der Mechanik unter gewissen Voraussetzungen zu den Bedingungen (5.36) für $r = 2$.

Wie bemerken, dass die Bedingungen (5.35) im Allgemeinen nicht ausreichen, um das stückweise Polynom $p \in S_h^{n,r}$ eindeutig zu bestimmen. Wir werden die Wohlgestelltheit der Spline-Interpolation im nächsten Abschnitt für den wichtigen Spezialfall von kubischen Splines diskutieren.

5.4.4 Kubische Splines

Von besonderer Bedeutung in den Anwendungen sind die sogenannten *kubischen Splines* aus dem Raum $S_h^{3,2}[a,b]$. Eine notwendige Voraussetzung für die Wohlgestelltheit der Lagrangeschen Interpolationsaufgabe (5.35) durch Splines $p \in S_h^{3,2}[a,b]$ ist, dass die Anzahl der (linear unabhängigen) Bedingungen und die Anzahl der Freiheitsgrade übereinstimmen. Wir betrachten dazu wieder die in Bemerkung 5.23 eingeführte äquivalente Aufgabe ein Polynom $p \in P_h^3[a,b]$ zu bestimmen, so dass die Bedingungen (5.36) und (5.35) erfüllt sind.

Ein Polynom $p \in P_h^3[a,b]$ kann in jedem Teilintervall I_j folgendermaßen dargestellt werden

$$p_{|I_j} = p_j(x) = a_{j0} + a_{j1}(x - x_j) + a_{j2}(x - x_j)^2 + a_{j3}(x - x_j)^3. \tag{5.37}$$

Dieser Raum hat also $4N$ **Freiheitsgrade** (4 Koeffizienten in jedem Teilintervall).

Für den in Definition 5.22 definierten Spline gelten folgende Bedingungen

- $2N$ **Bedingungen** $p_j(x_j) = f(x_j)$, $j = 1, ..., N$ und $p_{j+1}(x_j) = f(x_j)$, $j = 0, ..., N-1$ (Dies impliziert schon die globale Stetigkeit von p)
- $2N - 2$ **Bedingungen** für die globale Differenzierbarkeit erster und zweiter Ordnung

$$p_j'(x_j) = p_{j+1}'(x_j), \quad p_j''(x_j) = p_{j+1}''(x_j), \quad j = 1, ..., N-1$$

Es fehlen 2 Bedingungen, um eine wohlgestellte Aufgabe erhalten zu können. Eine genaue Betrachtung ergibt, dass die Koeffizienten in den äußeren Intervallen I_1 und I_N unterbestimmt sind. In jedem inneren Punkt x_j sind 4 Bedingungen gestellt, in den Punkten x_0 und x_N aber nur eine.

Mögliche Zusatzbedingungen sind

$$p''(x_0) = y_a, \quad p''(x_N) = y_b. \tag{5.38}$$

Für $y_a = y_b = 0$ werden diese auch *natürliche* Randbedingungen genannt, da sie sich bei einem elastischen Biegestab in Abwesenheit von Kräften einstellen. Den resultierenden *Spline* nennt man auch den **natürlichen kubischen Spline**.

Satz 5.24. *Die in Definition 5.22 definierte Spline-Interpolation besitzt für $n = 3$ und $r = 2$ unter den Zusatzbedingungen (5.38) eine eindeutige Lösung $s \in S_h^{3,2}[a,b]$. Weiter gilt für den natürlichen Spline s_n und jede Funktion $g \in C^2[a,b]$ mit $g(x_j) = y_j, j = 0, ..., N$, dass*

$$\int_a^b |s_n''(x)|^2 \, dx \leq \int_a^b |g''(x)|^2 \, dx. \tag{5.39}$$

Bemerkung 5.25. *Aus der Energieminimierungseigenschaft (5.39) folgt eine wichtige Stabilitätseigenschaft des natürlichen kubischen Splines s_n. Die zweiten Ableitungen einer Funktion beschreiben die Krümmung einer Funktion g. Oszillationen wie in Beispiel 5.10 führen automatisch zu großen Werten von g''. Diese sind beim natürlichen kubischen Spline aufgrund von (5.39) minimal. Man beachte, dass das Minimum bzgl aller Funktionen $g \in C^2[a,b]$, die die Stützwerte annehmen, angenommen wird. Die Energie ist also insbesondere auch kleiner als bei allen anderen Splines mit $r \geq 2$.*

Beweis. (i) *Existenz und Eindeutigkeit*: Wir betrachten wieder die äquivalente Aufgabe, dass ein Polynom $p \in P_h^3[a,b]$ die folgenden $4N$ Bedingungen erfüllt

$$\begin{aligned} &p_j(x_j) = f(x_j),\ j = 1, ..., N, && p_{j+1}(x_j) = f(x_j), && j = 0, ..., N-1 \\ &p_j'(x_j) = p_{j+1}'(x_j), && p_j''(x_j) = p_{j+1}''(x_j), && j = 1, ..., N-1 \\ &p_1''(x_0) = y_a, && p_N''(x_N) = y_b. \end{aligned} \tag{5.40}$$

Wir stellen die Interpolationsaufgabe als ein lineares Gleichungssystem für die in (5.37) definierten Koeffizienten $a_{jk}, j = 1, ..., N, k = 0, ..., 3$ dar

$$Au = b,$$

wobei u der von allen Koeffizienten a_{jk} gebildete Vektor ist, $A \in \mathbb{R}^{4N \times 4N}$ und $b \in \mathbb{R}^{4N}$. Mit den im Beweis von Satz 5.2 benutzten Argumenten der Linearen Algebra folgt die Existenz einer Lösung aus der Eindeutigkeit.

Seien also s_1, s_2 2 Lösungen von (5.40) und $s = s_1 - s_2$. Diese liegt insbesondere im Raum

$$\mathcal{V} := \{\phi \in C^2[a,b],\ \phi(x_i) = 0,\ i = 0, ..., N\}.$$

Wir zeigen zunächst, dass die Form

$$(v, w)_{\mathcal{V}} := \int_a^b v''(x) w''(x) \, dx$$

ein Skalarprodukt auf $\mathcal{V}$ darstellt. Linearität, Symmetrie und Positivität für $v = w$ folgen sofort. Wir müssen noch zeigen, dass $(v, v)_{\mathcal{V}} = 0 \Rightarrow v \equiv 0$ impliziert. Verschwindet $(v, v)_{\mathcal{V}}$, so gilt $v''(x) = 0 \, \forall x \in [a, b]$, d.h. v ist ein lineares Polynom auf $[a, b]$. Da $v(x_i) = 0$ in $x_i = 0, ..., N$ folgt $v \equiv 0$.

Der Spline $s = s_1 - s_2$ ist in jedem Teilintervall I_j ein Polynom dritten Grades und damit insbesondere glatt innerhalb jeden Teilintervalls. Weiter gilt $s''(x_0) = s_1''(x_0) - s_2''(x_0) = 0$ und analog $s''(x_N) = 0$. Für ein beliebiges $w \in \mathcal{V}$ gilt dann mit zweifacher partieller Integration

$$\begin{aligned}
(s, w)_{\mathcal{V}} &= \int_a^b s''(x) w''(x) \, dx = \sum_{j=1}^{N} \int_{I_j} s''(x) w''(x) \, dx \\
&= \sum_{j=1}^{N} \left\{ s''(x_j) w'(x_j) - s''(x_{j-1}) w'(x_{j-1}) - \int_{I_j} s'''(x) w'(x) \, dx \right\} \\
&= \underbrace{s''(x_N)}_{=0} w'(x_N) - \underbrace{s''(x_0)}_{=0} w'(x_0) - \sum_{j=1}^{N} \int_{I_j} s'''(x) w'(x) \, dx \\
&= - \sum_{j=1}^{N} \left\{ s'''(x_j) \underbrace{w(x_j)}_{=0} - s'''(x_{j-1}) \underbrace{w(x_{j-1})}_{=0} - \int_{I_j} \underbrace{s^{(4)}(x)}_{=0} w(x) \, dx \right\} = 0.
\end{aligned} \tag{5.41}$$

Es folgt insbesondere, dass $(s, s)_{\mathcal{V}} = 0$. Nach den obigen Vorarbeiten folgt daraus, dass $s = 0$, d.h. die Eindeutigkeit.

(ii) *Energieminimum*: Zunächst bemerken wir, dass die Argumentation (5.41) auch für den natürlichen Spline s_n anstelle von s gilt, d.h. es gilt

$$(s_n, w)_{\mathcal{V}} = 0 \quad \forall w \in \mathcal{V}. \tag{5.42}$$

In (5.41) wurde nämlich nur benutzt, dass s ein stückweise Polynom dritten Grades ist und dass $s''(x_0) = s''(x_N) = 0$.

Weiter gilt für jede Funktion $g \in C^2[a, b]$ mit $g(x_j) = y_j, j = 0, ..., N$, dass

$$s_n - g \in \mathcal{V}.$$

Aufgrund der Orthogonalitätsbeziehung (5.42) gilt also insbesondere, dass

$$0 = (s_n, s_n - g)_{\mathcal{V}} = (s_n, s_n)_{\mathcal{V}} - (s_n, g)_{\mathcal{V}} = \int_a^b |s_n''(x)|^2 \, dx - \int_a^b s_n''(x) \cdot g''(x) \, dx.$$

Mithilfe der Cauchy-Schwarz-Ungleichung folgt daraus

$$\int_a^b |s_n''(x)|^2 \, dx = \int_a^b s_n''(x) \cdot g''(x) \, dx \leq \left(\int_a^b |s_n''(x)|^2 \, dx \right)^{1/2} \left(\int_a^b |g''(x)|^2 \, dx \right)^{1/2}.$$

Division durch den ersten Term auf der rechten Seite und anschließendes Quadrieren ergibt die Behauptung. □

Schließlich gilt die folgende Fehlerabschätzung.

Satz 5.26. *Sei $f \in C^4[a,b]$ und seien $N+1$ paarweise verschiedene Stützstellen $x_j, j = 0, ..., N$ gegeben. Für den durch die Stützwerte $(x_0, f(x_0)), ..., (x_N, f(x_N))$ und die Zusatzbedingungen $s''(a) = f''(a)$ und $s''(b) = f''(b)$ definierten kubischen Spline $s \in S_h^{3,2}[a,b]$ gilt die Fehlerabschätzung*

$$\max_{x\in[a,b]} |f(x) - s(x)| \leq \frac{1}{2} h^4 \max_{x\in[a,b]} |f^{(4)}(x)|.$$

Beweis. Für den technischen Beweis dieser Aussage verweisen wir auf [6]. □

Praktische Berechnung Zur praktischen Berechung der Spline-Interpolation ist das lineare Gleichungssystem (5.40) für die Koeffizienten a_{ji} von

$$p_j(x) = a_{j0} + a_{j1}(x - x_j) + a_{j2}(x - x_j)^2 + a_{j3}(x - x_j)^3, \tag{5.43}$$

$j = 1, ..., N$ zu lösen. Dies ist zunächst ein $(4N \times 4N)$-System mit folgenden Bedingungen $(h_j := x_j - x_{j-1})$

$$\begin{aligned}
&(1)\ p_j(x_j) = f(x_j) && \Rightarrow\ a_{j0} = f(x_j), \quad j = 1, ..., N,\\
&(2)\ p_{j+1}(x_j) = f(x_j) && \Rightarrow\ \underbrace{a_{j+1,0}}_{=f(x_{j+1})} - h_{j+1}a_{j+1,1} + h_{j+1}^2 a_{j+1,2} - h_{j+1}^3 a_{j+1,3} = f(x_j),\ j = 0, ..., N-1,\\
&(3)\ p_j'(x_j) = p_{j+1}'(x_j) && \Rightarrow\ a_{j1} = a_{j+1,1} - 2h_{j+1}a_{j+1,2} + 3h_{j+1}^2 a_{j+1,3}, \quad j = 1, ..., N-1,\\
&(4)\ p_j''(x_j) = p_{j+1}''(x_j) && \Rightarrow\ 2a_{j2} = 2a_{j+1,2} - 6h_{j+1}a_{j+1,3}, \quad j = 1, ..., N-1,\\
&(5)\ p_1''(x_0) = y_a && \Rightarrow 2a_{12} - 6a_{13}h_1 = y_a,\\
&(6)\ p_N''(x_N) = y_b && \Rightarrow 2a_{N2} = y_b.
\end{aligned} \tag{5.44}$$

Dies kann nun weiter vereinfacht werden. Die Koeffizienten a_{j0} sind mit (1) direkt bestimmt, der Koeffizient a_{2N} durch (6). Die Koeffizienten a_{j1} und a_{j3} können mithilfe der a_{j2} und $a_{j+1,2}$ dargestellt werden. Dazu ersetzt man zunächst mithilfe von (4) und (5) a_{j3} durch a_{j2} und $a_{j+1,2}$ und anschließend a_{j1} mithilfe von (2).

Es ergibt sich schließlich folgendes Tridiagonalsystem für die $(N-1)$ Koeffizienten $a_{j2}, j = 1, ..., N-1$, welches effizient und numerisch stabil mit direkten Methoden in $\mathcal{O}(N)$ arithmetischen Operationen gelöst werden kann.

$$\underbrace{\begin{pmatrix}
2(h_1+h_2) & h_2 & & & & 0\\
h_2 & 2(h_2+h_3) & \ddots & & & \\
& \ddots & \ddots & \ddots & & \\
& & \ddots & 2(h_{N-2}+h_{N-1}) & h_{N-1} & \\
0 & & & h_{N-1} & 2(h_{N-1}+h_N)
\end{pmatrix}}_{=:A}
\begin{pmatrix} a_{12} \\ a_{22} \\ \vdots \\ \\ a_{N-1,2} \end{pmatrix} = b$$

mit einer $(N-1) \times (N-1)$-Matrix A und einem Vektor $b \in \mathbb{R}^{N-1}$. Anschließend berechnet man a_{j1} und a_{j3} mithilfe von (2), (4) und (5).

6 Numerische Quadratur

Die Aufgabe der numerischen Quadratur besteht darin Integrale näherungsweise zu berechnen. Als Beispiele seien die Integrale

$$\int_a^b \frac{\sin(x)}{x}\,dx, \quad \int_a^b \exp(-x^2)\,dx$$

genannt, für die keine Stammfunktionen bekannt sind.

Definition 6.1. *Eine Quadraturformel ist eine Approximation der Form*

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

mit Stützpunkten $x_i \in \mathbb{R}, i = 0, ..., n$ und Quadraturgewichten $\alpha_i \in \mathbb{R}, i = 0, ..., n$.

6.1 Interpolatorische Quadraturformeln

Die wichtigsten Quadraturformeln lassen sich über die Approximation von f mittels der Interpolationspolynome $p_n \in P_n$ herleiten. Wir approximieren

$$\int_a^b f(x)\,dx \approx \int_a^b p_n(x)\,dx = \sum_{i=0}^{n} f(x_i) \underbrace{\int_a^b L_i^n(x)\,dx}_{=:\alpha_i},$$

wobei L_i^n die Lagrangeschen Basispolynome zu Stützstellen $a \le x_0 < x_1 < ... < x_n \le b$ bezeichen (siehe Abschnitt 5.1)

$$L_i^n(x) := \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}.$$

Dabei lassen wir hier nun auch $a < x_0$ und $x_n < b$ zu. Als erstes einfaches Beispiel sei hier die Interpolation mit einem konstanten Polynom $p_0 \in P_0$ zur Stützstelle $x_0 = a$ genannt. Es gilt

$$\alpha_0 := \int_a^b \underbrace{L_0^0(x)}_{=1}\,dx = b - a.$$

Es ergibt sich die sogenannte *Boxregel*

$$I^{(0)}_{[a,b],\mathrm{box}}(f) := \int_a^b p_0(x)\,dx = (b-a)f(a). \tag{6.1}$$

Bei der Interpolation mit einem linearen Polynom $p_1 \in P_1$ mit Stützstelle $x_0 = a, x_1 = b$ ergibt sich

$$\alpha_0 := \int_a^b L_0^1(x)\,dx = \int_a^b \frac{x-b}{a-b}\,dx = \frac{1}{2}(b-a),$$
$$\alpha_1 := \int_a^b L_1^1(x)\,dx = \int_a^b \frac{x-a}{b-a}\,dx = \frac{1}{2}(b-a).$$

Dies führt auf die sogenannte Trapezregel

$$I^{(1)}_{[a,b]}(f) := \frac{1}{2}(b-a)(f(a)+f(b)) \qquad \left(= \int_a^b p_1(x)\,dx\right). \tag{6.2}$$

Die zugehörigen allgemeinen Fehlerdarstellungen und -abschätzungen können direkt aus der Theorie zur Polynominterpolation aus Abschnitt 5.1 hergeleitet werden.

Satz 6.2. *Sei $f \in C^{n+1}[a,b]$ und $p_n \in P_n$ das zugehörige Interpolationspolynom zu $(n+1)$ paarweise verschiedenen Stützstellen $x_0, ..., x_n$. Es gelten die Fehlerdarstellungen*

$$\int_a^b f(x)\,dx - \int_a^b p_n(x)\,dx = \int_a^b \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\prod_{j=0}^{n}(x-x_j)\,dx \tag{6.3}$$

mit von x abhängigen Stellen $\xi_x \in (a,b)$. Weiter gilt die Fehlerabschätzung

$$\Big|\int_a^b f(x)\,dx - \int_a^b p_n(x)\,dx\Big| \le \sup_{\xi\in(a,b)} |f^{(n+1)}(\xi)|(b-a)h^{n+1} \tag{6.4}$$

wobei $h := \max\{x_0 - a, x_1 - x_0, ..., x_n - x_{n-1}, b - x_n\}$ den maximalen Abstand zwischen benachbarten Rand- bzw. Stützstellen bezeichne.

Beweis. Die Fehlerdarstellung folgt direkt aus Satz 5.7. Aus (6.3) folgt weiter

$$\begin{aligned}
\Big|\int_a^b f(x) - p_n(x)\,dx\Big| &\le \sup_{\xi\in(a,b)} \frac{|f^{(n+1)}(\xi)|}{(n+1)!}\Big|\int_a^b \prod_{j=0}^{n}(x-x_j)\,dx\Big| \\
&\le \sup_{\xi\in(a,b)} \frac{|f^{(n+1)}(\xi)|}{(n+1)!}\int_a^b \underbrace{(h\cdot(2h)\cdot ...\cdot(nh)\cdot((n+1)h))}_{=((n+1)!)h^{n+1}}\,dt \\
&\le \sup_{\xi\in(a,b)} |f^{(n+1)}(\xi)|h^{n+1}(b-a).
\end{aligned}$$

□

Wie bei der Polynominterpolation erhält man Konvergenz für $h \to 0$ (bzw. $n \to \infty$), wenn die Ableitungen gleichmäßig beschränkt bleiben.

Wir betrachten als nächstes die interpolatorischen Quadraturformeln zu äquidistanten Stützstellen, die sogenannten *Newton-Cotes*-Formeln.

6.1.1 Newton-Cotes-Formeln

Wir unterteilen das Intervall $[a, b]$ wieder in gleich große Teilintervalle. Man unterscheidet zwischen den *abgeschlossenen* Newton-Cotes-Formeln, bei denen auch die Intervallenden a und b Stützstellen sind, und den *offenen* Newton-Cotes-Formeln, bei denen dies nicht der Fall ist.

Bei einer *abgeschlossenen* Newton-Cotes-Formel lauten die Stützstellen

$$x_i = a + \frac{b-a}{n} i, \quad i = 0, ..., n.$$

Wir definieren den Abstand der Stützstellen als

$$h := \frac{b-a}{n}.$$

Bei einer *offenen* Newton-Cotes-Formel lauten die Stützstellen

$$x_i = a + \frac{b-a}{n+2}(i+1), \quad i = 0, ..., n$$

und deren Abstand ist

$$h := \frac{b-a}{n+2}.$$

Zur Berechnung der Quadraturgewichte nutzen wir die Integraltransformation $x(t) = a + th, t \in [0, n]$. Für die abgeschlossenen Newton-Cotes-Formeln gilt für $i = 0, ..., n$

$$\alpha_i := \int_a^b \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}\, dx = \int_a^b \prod_{j=0, j\neq i}^{n} \frac{(a+th) - (a+jh)}{(a+ih) - (a+jh)} \underbrace{dx}_{=h\,dt} = h \int_0^n \prod_{j=0, j\neq i}^{n} \frac{t-j}{i-j}\, dt.$$

Das letzte Integral ist nun unabhängig vom Intervall $[a, b]$ und kann ein für alle mal berechnet werden. Für die offenen Newton-Cotes-Formeln gilt analog

$$\begin{aligned}\alpha_i = \int_a^b \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}\, dx &= \int_a^b \prod_{j=0, j\neq i}^{n} \frac{(a+th) - (a+(j+1)h)}{(a+(i+1)h) - (a+(j+1)h)} \underbrace{dx}_{=h\,dt} \\ &= h \int_0^{n+2} \prod_{j=0, j\neq i}^{n} \frac{t-j-1}{i-j}\, dt.\end{aligned}$$

Die abgeschlossene Newton-Cotes-Formel niedrigsten Grades ($n = 1$) ist die Trapezregel (6.2), für die die Gewichte bereits oben berechnet wurden. Die abgeschlossene Formel der Ordnung 2 wird *Simpson-Regel* genannt. Die Gewichte berechnen sich wie folgt

$$\begin{aligned}\alpha_0 &= h \int_0^2 \frac{t-1}{0-1}\frac{t-2}{0-2}\, dt = \frac{h}{2}\int_0^2 t^2 - 3t + 2\, dt = \frac{h}{2}\left(\frac{8}{3} - 6 + 4\right) = \frac{h}{3}, \\ \alpha_1 &= h \int_0^2 \frac{t-0}{1-0}\frac{t-2}{1-2}\, dt = -h\int_0^2 t^2 - 2t\, dt = -h\left(\frac{8}{3} - 4\right) = \frac{4h}{3}, \\ \alpha_2 &= h \int_0^2 \frac{t-0}{2-0}\frac{t-1}{2-1}\, dt = \frac{h}{2}\int_0^2 t^2 - t\, dt = \frac{h}{2}\left(\frac{8}{3} - 2\right) = \frac{h}{3}.\end{aligned}$$

Es ergibt sich die Simpson-Regel

$$I^{(2)}_{[a,b]}(f) := \frac{h}{3}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) \qquad \left(= \int_a^b p_2(x)\,dx\right).$$

Wir geben noch die ersten 4 abgeschlossenen Newton-Cotes-Formeln an ($h = \frac{b-a}{n}$)

$$\begin{aligned}
I^{(1)}_{[a,b]}(f) &:= \frac{b-a}{2}\left(f(a) + f(b)\right) && \text{(Trapezregel)}\\
I^{(2)}_{[a,b]}(f) &:= \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) && \text{(Simpson-Regel)}\\
I^{(3)}_{[a,b]}(f) &:= \frac{b-a}{8}\left(f(a) + 3f(a+h) + 3f(b-h) + f(b)\right) && (\tfrac{3}{8}\text{-Regel})\\
I^{(4)}_{[a,b]}(f) &:= \frac{b-a}{90}\left(7f(a) + 32f(a+h) + 12f\left(\frac{a+b}{2}\right) + 32f(b-h) + 7f(b)\right).
\end{aligned} \tag{6.5}$$

Für die ersten offenen Formeln gilt ($h = \frac{b-a}{n+2}$)

$$\begin{aligned}
\tilde{I}^{(0)}_{[a,b]}(f) &:= (b-a)f\left(\frac{a+b}{2}\right) && \text{(Mittelpunktsregel)}\\
\tilde{I}^{(1)}_{[a,b]}(f) &:= \frac{b-a}{2}\left(f(a+h) + f(b-h)\right)\\
\tilde{I}^{(2)}_{[a,b]}(f) &:= \frac{b-a}{3}\left(2f(a+h) - f\left(\frac{a+b}{2}\right) + 2f(b-h)\right)\\
\tilde{I}^{(3)}_{[a,b]}(f) &:= \frac{b-a}{24}\left(11f(a+h) + f(a+2h) + f(b-2h) + 11f(b-h)\right).
\end{aligned} \tag{6.6}$$

Man beachte, dass bei den offenen Formeln ab $n = 2$ negative Gewichte α_i auftreten können. Dasselbe gilt bei den abgeschlossenen Formeln ab $n \geq 7$. Die Newton-Cotes-Formeln werden daher für großes n sehr rundungsanfällig. Selbst für eine rein positive Funktion f kann *Auslöschung* auftreten.

Wir analysieren nun für einige wichtige Formeln, ob die allgemeine Fehlerabschätzung (6.4) weiter verbessert werden kann. Dazu führen wir zunächst den Begriff der Ordnung einer Quadraturformel ein.

Definition 6.3. *Eine Quadraturformel $I(\cdot)$ heißt (mindestens) von der Ordnung n*

$$ord(I) = n$$

wenn durch sie alle Polynome aus P_{n-1} exakt integriert werden, d.h. wenn gilt

$$I_{[a,b]}(p) = \int_a^b p(x)\,dx \quad \forall p \in P_{n-1}, \quad a, b \in \mathbb{R},\ a \leq b.$$

Bemerkung 6.4. *Eine interpolatorische Quadraturformel $I^{(n)}_{[a,b]}$ ist damit nach Konstruktion immer (mindestens) von der Ordnung $n+1$, da die Polynominterpolation mit Polynomgrad n für Polynome $p \in P_n$ exakt ist.*

Satz 6.5. *Sei $f \in C^1[a,b]$. Für die in* (6.1) *definierte Boxregel gilt mit einer Zwischenstelle $\xi \in [a,b]$*

$$\int_a^b f(x)\,dx - I^{(0)}_{[a,b],box}(f) = \frac{(b-a)^2}{2} f'(\xi).$$

Bei $f \in C^2[a,b]$ gilt für die in (6.6) *definierte Mittelpunktsregel*

$$\int_a^b f(x)\,dx - \tilde{I}^{(0)}_{[a,b]}(f) = \frac{(b-a)^3}{24} f''(\xi_0)$$

und für die in (6.5) *definierte Trapezegel*

$$\int_a^b f(x)\,dx - I^{(1)}_{[a,b]}(f) = -\frac{(b-a)^3}{12} f''(\xi_1)$$

mit Zwischenstellen $\xi_0, \xi_1 \in [a,b]$. Ist $f \in C^4[a,b]$ so gilt für die in (6.5) *definierte Simpson-Regel mit einem $\xi_2 \in [a,b]$*

$$\int_a^b f(x)\,dx - I^{(2)}_{[a,b]}(f) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi_2).$$

Beweis. (i) Boxregel: Mit Taylorentwicklung gilt

$$\int_a^b f(x)\,dx - I^{(0)}_{[a,b],\mathrm{box}}(f) = \int_a^b f(x) - f(a)\,dx = \int_a^b f'(\xi_x)(x-a)\,dx$$

mit einer x-abhängigen Zwischenstelle $\xi_x \in [a,x]$. Wir wollen den Mittelwertsatz der Integralrechnung anwenden und $f'(\xi_x)$ vor das Integral ziehen. Dazu müssen wir zeigen, dass $g(x) := f'(\xi_x)$ eine stetige Funktion auf $[a,b]$ ist. Es gilt nach Definition der Zwischenstelle ξ_x

$$g(x) = f'(\xi_x) = \begin{cases} \frac{f(x)-f(a)}{x-a}, & x \neq a, \\ f'(a), & x = a. \end{cases}$$

Wegen $\lim\limits_{x\to a} \frac{f(x)-f(a)}{x-a} = f'(a)$ ist g stetig auf ganz $[a,b]$. Da $(x-a) \geq 0$ auf $[a,b]$ folgt mit dem Mittelwertsatz der Integralrechnung mit einem $\xi_0 \in [a,b]$

$$\int_a^b f(x)\,dx - I^{(0)}_{\mathrm{box},[a,b]}(f) = f'(\xi_0) \int_a^b x - a\,dx = f'(\xi_0)\frac{(b-a)^2}{2}.$$

(ii) Für die Mittelpunktsregel ergibt Taylorentwicklung um $x_m = \frac{a+b}{2}$

$$f(x) - f(x_m) = f'(x_m)(x-x_m) + \frac{f''(\xi_x)}{2}(x-x_m)^2 \tag{6.7}$$

mit einer von x abhängigen Zwischenstelle $\xi_x \in [x_m, x]$ oder $\xi_x \in [x, x_m]$. Es folgt

$$\begin{aligned}\int_a^b f(x)\,dx - \tilde{I}^{(0)}_{[a,b]}(f) &= \int_a^b f(x) - f(x_m)\,dx \\ &= f'(x_m)\underbrace{\int_a^b (x - x_m)\,dx}_{=0} + \frac{1}{2}\int_a^b f''(\xi_x)(x - x_m)^2\,dx.\end{aligned}$$

Wir zeigen wieder die Stetigkeit der Funktion $g(x) := f''(\xi_x)$, um den Mittelwertsatz der Integralrechnung anwenden zu können. Es gilt mit (6.7)

$$g(x) = f''(\xi_x) = \begin{cases} 2\left(\frac{f(x)-f(x_m)}{(x-x_m)^2} - \frac{f'(x_m)}{x-x_m}\right), & x \neq x_m, \\ f''(x_m), & x = x_m. \end{cases}$$

Die Stetigkeit für $x \neq x_m$ ist klar. Für $x \to x_m$ gilt wegen $\xi_x \in [x, x_m]$ notwendig $\xi_x \to x_m$ und daher

$$\lim_{x \to x_m} g(x) = \lim_{x \to x_m} f''(\xi_x) = f''(x_m) = g(x_m).$$

Mit dem Mittelwertsatz der Integralrechnung folgt aufgrund von $(x - x_m)^2 \geq 0$ für ein $\xi \in [a, b]$

$$\begin{aligned}\int_a^b f(x)\,dx - \tilde{I}^{(0)}_{[a,b]}(f) = \frac{f''(\xi)}{2}\int_a^b (x - x_m)^2\,dx &= \frac{f''(\xi)}{6}\left((\underbrace{b - x_m}_{=(b-a)/2})^3 - (\underbrace{a - x_m}_{=-(b-a)/2})^3 \right) \\ &= \frac{(b-a)^3}{24} f''(\xi_0).\end{aligned}$$

(iii) Zur Herleitung einer optimalen Fehlerabschätzung für die Trapezregel führen wir die Funktion

$$g(x) = \frac{1}{2}(x - a)(x - b)$$

ein. Es gilt $g'(x) = x - \frac{a+b}{2}$ und $g'' \equiv 1$. Mit zweimaliger partieller Integration erhalten wir die Beziehung

$$\begin{aligned}\int_a^b f(x)\,dx &= \int_a^b f(x)g''(x)\,dx = f(b)g'(b) - f(a)g'(a) - \int_a^b f'(x)g'(x)\,dx \\ &= f(b)\underbrace{g'(b)}_{=(b-a)/2} - f(a)\underbrace{g'(a)}_{=(a-b)/2} - f'(b)\underbrace{g(b)}_{=0} + f'(a)\underbrace{g(a)}_{=0} + \int_a^b f''(x)g(x)\,dx \\ &= \underbrace{\frac{b-a}{2}\left(f(b) + f(a)\right)}_{=I^{(1)}_{[a,b]}(f)} + \frac{1}{2}\int_a^b f''(x)(x - a)(x - b)\,dx.\end{aligned}$$

Mithilfe des Mittelwertsatzes der Integralrechung ($(x-a)(x-b) \leq 0$) folgt für ein $\xi_1 \in (a,b)$

$$\int_a^b f(x)\,dx - I^{(1)}_{[a,b]}(f) = \frac{f''(\xi_1)}{2} \underbrace{\int_a^b (x-a)(x-b)\,dx}_{=-\frac{1}{6}(b-a)^3} = -f''(\xi_1)\frac{(b-a)^3}{12}.$$

(iv) Simpson-Regel: Der Beweis für die Restglieddarstellung der Simpson-Regel ist etwas aufwändiger. Wir verweisen hier auf die Literatur (z.B. [3]).

□

Bemerkung 6.6. *Aus der Restglieddarstellung kann man sofort die Ordnung der Formeln ablesen (für ein Polynom n-ten Grades verschwindet die $(n+1)$-te Ableitung auf der rechten Seite). Es folgt, dass die Boxregel die für eine interpolatorische Quadraturformel vom Grad $n = 0$ minimale Ordnung $n+1 = 1$ hat, die Mittelpunktregel dagegen die Ordnung $2 = n+2$. Die Trapezregel weist wieder die minimale Ordnung $n+1$ auf, die Simpsonregel ($n = 2$) dagegen wieder eine höhere Ordnung ($4 = n+2$). Dieses Superapproximationsprinzip lässt sich verallgemeinern: Für gerades n weisen die Newton-Cotes-Formeln die Ordnung $n+2$ auf, da aus Symmetriegründen Integrale über ungerade Anteile der Taylorentwicklung verschwinden.*

Wie bei der Polynominterpolation ist die Konvergenz für $n \to \infty$ von geringer praktischer Relevanz, da sie gleichmäßig beschränkte Ableitungen für $n \to \infty$ voraussetzt. Hinzu kommt die Störungsanfälligkeit der Polynominterpolation für großes n und die Gefahr der Auslöschung aufgrund negativer Gewichte. Daher wendet man in der Praxis auch die Quadraturformeln in der Regel stückweise an.

6.2 Stückweise interpolatorische Quadratur

Wir unterteilen das Intervall $[a,b]$ wieder in Teilintervalle $I_j = [x_{j-1}, x_j], j = 1, ..., N$, so dass

$$a = x_0 < x_1 < ... < x_N = b.$$

Die Intervalllänge sei mit $H_j = x_j - x_{j-1}$ bezeichnet und wir setzen $H := \max\limits_{j=1,...,N} H_j$. Bei einer stückweise interpolatorischen Quadraturformel wendet man dann in jedem Teilintervall eine der obigen Quadraturformeln an

$$I^{(n)}_H(f) := \sum_{j=1}^{N} I^{(n)}_{I_j}(f), \tag{6.8}$$

wobei $I^{(n)}_{I_j}(f)$ die Formel bezüglich des Integrationsintervalls I_j bezeichne. Dabei ist zu beachten, dass das stückweise Interpolationspolynom innerhalb jedes Teilintervalls $I_j, j = 1, ..., N$ durch die Stützstellen $z_{j,k} \in [x_{j-1}, x_j], k = 0, ..., n$

$$x_{j-1} \leq z_{j,0} < z_{j,1} < ... < z_{j,n} \leq x_j$$

definiert ist. Wir definieren den maximalen Abstand zwischen benachbarten Rand- bzw. Stützstellen in I_j als

$$h_j := \max\{z_{j,0} - x_{j-1}, z_{j,1} - z_{j0}, ..., z_{j,n} - z_{j,n-1}, x_j - z_{j,n}\}.$$

Aus Satz 6.2 erhalten wir zunächst folgende allgemeine Fehlerabschätzung.

Satz 6.7. *Sei $f \in C[a,b]$ und stückweise regulär $f \in C^{n+1}(I_j), j = 1, ..., N$. Für eine stückweise interpolatorische Quadraturformel der Form* (6.8) *mit Polynomgrad n und N Teilintervallen gilt die Fehlerabschätzung*

$$\Big| \int_a^b f(x)\,dx - I_H^{(n)}(f) \Big| \le H^{n+1}(b-a) \sup_{\xi \in [a,b]\setminus\{x_0,...,x_N\}} \big| f^{(n+1)}(\xi) \big|.$$

Beweis. Es gilt mit der stückweisen Fehlerabschätzung aus Satz 6.2

$$\begin{aligned} \Big| \int_a^b f(x)\,dx - I_H^{(n)}(f) \Big| &\le \sum_{j=1}^N \Big| \int_{I_j} f(x)\,dx - I_{I_j}^{(n)}(f) \Big| \le \sum_{j=1}^N \sup_{\xi_j \in (x_{j-1},x_j)} |f^{(n+1)}(\xi_j)| h_j^{n+1} H_j \\ &\le \sup_{\xi \in [a,b]\setminus\{x_0,...,x_N\}} |f^{(n+1)}(\xi)| H^{n+1} \underbrace{\sum_{j=1}^N H_j}_{=(b-a)}. \end{aligned}$$

□

6.2.1 Summierte Newton-Cotes-Formeln

Wir wenden uns nun wieder dem Spezialfall zu, dass die Stützstellen äquidistant gewählt wurden (sowohl $x_j, j = 1, ..., N$ als auch $z_{j,k}$, $j = 1, ..., N$, $k = 0, ..., n$)

$$\begin{aligned} x_j &= a + j \underbrace{\frac{b-a}{N}}_{=H}, \qquad j = 0, ..., N \\ z_{jk} &= x_j + k\frac{H}{n}, \quad k = 0, ..., n \quad \text{(abgeschlossene Formeln)} \\ \text{bzw. } z_{jk} &= x_j + (k+1)\frac{H}{n+2}, \quad k = 0, ..., n \quad \text{(offene Formeln)}. \end{aligned}$$

Die stückweisen Varianten der oben analysierten Newton-Cotes-Formeln haben folgende Form

- Summierte Boxregel

$$I_{H,\text{box}}^{(0)}(f) := \sum_{j=1}^N I_{I_j,\text{box}}^{(0)}(f) = H \sum_{j=0}^{N-1} f(x_j). \tag{6.9}$$

- Summierte Mittelpunktregel

$$\tilde{I}_H^{(0)}(f) := \sum_{j=1}^N \tilde{I}_{I_j}^{(0)}(f) = H \sum_{j=1}^N f\left(\frac{x_j + x_{j-1}}{2}\right). \tag{6.10}$$

- Summierte Trapezregel

$$I_H^{(1)}(f) := \sum_{j=1}^{N} I_{I_j}^{(1)}(f) = \frac{H}{2} \sum_{j=1}^{N} f(x_j) + f(x_{j-1}) = H \left(\frac{f(a)}{2} + \sum_{j=1}^{N-1} f(x_j) + \frac{f(b)}{2} \right). \tag{6.11}$$

- Summierte Simpsonregel

$$\begin{aligned} I_H^{(2)}(f) := \sum_{j=1}^{N} I_{I_j}^{(2)}(f) &= \frac{H}{6} \sum_{j=1}^{N} f(x_{j-1}) + 4f\left(\frac{x_{j-1} + x_j}{2} \right) + f(x_j) \\ &= \frac{H}{6} \left(f(a) + 2 \sum_{j=1}^{N-1} f(x_j) + 4 \sum_{j=1}^{N} f\left(\frac{x_{j-1} + x_j}{2} \right) + f(b) \right). \end{aligned} \tag{6.12}$$

Die summierte Box- und Mittelpunktregel brauchen dabei jeweils N, die Trapezregel $N + 1$ und die Simpsonregel $2N + 1$ Funktionsauswertungen.

Wir wenden uns nun der Fehleranalyse für die 4 aufgeführten Formeln zu. Die Fehlerabschätzungen folgen dabei direkt aus Satz 6.5.

Satz 6.8. *Für die in* (6.9) *definierte summierte Boxregel gilt unter der Voraussetzung, dass* $f \in C^1[a, b], j = 1, ..., N$

$$\int_a^b f(x)\,dx - I_{H,box}^{(0)}(f) = \frac{H}{2}(b - a)f'(\xi)$$

mit einer Zwischenstelle $\xi \in [a, b]$*. Gilt zusätzlich* $f \in C^2[a, b]$*, dann folgt für die in* (6.10) *definierte summierte Mittelpunktsregel*

$$\int_a^b f(x)\,dx - \tilde{I}_H^{(0)}(f) = \frac{H^2}{24}(b - a)f''(\xi_0)$$

und die in (6.11) *definierte summierte Trapezegel*

$$\int_a^b f(x)\,dx - I_H^{(1)}(f) = -\frac{H^2}{12}(b - a)f''(\xi_1)$$

mit Zwischenstellen $\xi_0, \xi_1 \in [a, b]$*. Ist* $f \in C^4[a, b]$ *so gilt für die in* (6.12) *definierte summierte Simpson-Regel mit einem* $\xi_2 \in [a, b]$

$$\int_a^b f(x)\,dx - I_H^{(2)}(f) = -\frac{H^4}{2880}(b - a)f^{(4)}(\xi_2).$$

Beweis. Nach Satz 6.5 gelten in jedem Teilintervall I_j Fehlerdarstellungen der Form

$$\int_{I_j} f(x)\,dx - I_{I_j}^{(n)}(f) = w_n f^{(m+1)}(\xi_j) H^{m+2}$$

mit einem $m \geq n, w_n \in \mathbb{R}$ und $\xi_j \in I_j$. Summation über alle Teilintervalle I_j ergibt unter Anwendung des Zwischenwertsatzes

$$\begin{aligned}\int_a^b f(x)\,dx - I_H^{(n)}(f) = w_n \sum_{j=1}^{N} f^{(m+1)}(\xi_j) H^{m+2} &= w_n f^{(m+1)}(\xi) N H^{m+2} \\ &= w_n H^{m+1} f^{(m+1)}(\xi)(b-a)\end{aligned}$$

mit einer Zwischenstelle $\xi \in [a,b]$. Dabei haben wir ausgenutzt, dass $NH = b - a$. Damit folgen alle vier Fehlerdarstellungen aus Satz 6.5. □

Man beachte, dass ähnliche Fehlerdarstellungen gelten, wenn nur die stückweise Regularität $f \in C^{m+1}(I_j)$ und globale Stetigkeit vorausgesetzt wird.

Wir wollen die summierten Quadraturformeln noch vergleichen und nehmen dazu an, dass $f \in C^4[a,b]$. Bei vergleichbarem Aufwand (N bzw. $N+1$ Funktionsauswertungen) konvergieren die summierte Mittelpunkts- und Trapezregel für $N \to \infty$ (bzw. $H \to 0$) deutlich schneller gegen das zu berechnende Integral als die summierte Boxregel. Bei der summierten Simpsonregel sind etwa doppelt so viele Funktionsauswertungen nötig, die Konvergenzordnung verdoppelt sich.

Für einen fairen Vergleich zwischen der Simpsonregel und der Trapezregel wählen wir bei der Simpsonregel halb so viele Stützstellen N_S wie bei der Trapezregel $N_\mathrm{S} = \frac{N_\mathrm{T}}{2}$, so dass ungefähr gleich viele Funktionsauswertungen notwendig sind. Es gilt dann $H_\mathrm{S} = 2H_\mathrm{T}$ für die Teilintervallgrößen und für den Fehler mit gewissen Konstanten c_S und c_T (die unter anderem von den Ableitungen von f abhängen)

$$c_\mathrm{S} H_\mathrm{S}^4 = c_\mathrm{S}(2H_\mathrm{T})^4 = 16 c_\mathrm{S} H_\mathrm{T}^4 \ll c_\mathrm{T} H_\mathrm{T}^2$$

für kleines H_T.

Beispiel Wir approximieren das Integral

$$\int_0^1 \exp(-x^2)\,dx$$

mut der summierten Box-, Mittelpunkt-, Trapez und Simpsonregel für verschiedene N (Anzahl der Teilintervalle). Die Ergebnisse sind in Tabelle 6.1 gegeben. Die beobachteten Konvergenzordnungen entsprechen den theoretisch gezeigten, die Simpsonregel konvergiert mit Abstand am schnellsten. Die Fehler von Mittelpunkt- und Trapezregel unterscheiden sich wie in Theorem 6.8 gezeigt etwa um den Faktor (-2).

N	Boxregel	Mittelpunktregel	Trapezregel	Simpsonregel
2	1.43e-01	7.77e-03	-1.55e-02	3.12e-05
4	7.52e-02	1.92e-03	-3.84e-03	1.99e-06
8	3.85e-02	4.79e-04	-9.59e-04	1.25e-07
16	1.95e-02	1.20e-04	-2.40e-04	7.79e-09
32	9.82e-03	2.99e-05	-5.99e-05	4.87e-10
64	4.92e-03	7.48e-06	-1.50e-05	3.05e-11
128	2.47e-03	1.87e-06	-3.74e-06	1.93e-12
256	1.23e-03	4.68e-07	-9.36e-07	1.46e-13
512	6.17e-04	1.17e-07	-2.34e-07	3.43e-14
1024	3.09e-04	2.92e-08	-5.85e-08	2.71e-14
	$\mathcal{O}(H)$	$\mathcal{O}(H^2)$	$\mathcal{O}(H^2)$	$\mathcal{O}(H^4)$

Tabelle 6.1: Fehler $\int_0^1 f(x)\,dx - I_H(f)$ bei der Approximation des Integrals über die Funktion $f(x) = \exp(-x^2)$ mit verschiedenen summierten Quadraturformeln.

6.3 Gauß-Quadratur

Wir haben oben bereits gesehen, dass die **Wahl der Stützstellen** in einem Intervall Einfluss auf die Ordnung der Quadraturformeln haben kann. Bei den beiden Integrationsformeln der Ordnung 0 (konstante Polynome) weist die Boxregel die minimale Ordnung ($1 = n + 1$) einer interpolatorischen Quadraturformel auf, während die Mittelpunktregel die Ordnung $2 > n + 1$ hat.

Es stellt sich also die Frage, wie die Stützstellen **optimal** zu wählen sind. Wir rücken hier von der äquidistanten Wahl der Newton-Cotes-Formeln ab. Dabei betrachten wir zunächst im Gegensatz zum vorherigen Abschnitt keine stückweisen Quadraturformeln, sondern untersuchen interpolatorische Formeln auf ganz $[a, b]$. Als erstes zeigen wir eine obere Schranke für die Ordnung einer interpolatorischen Quadraturformel mit $n + 1$ Stützstellen.
Wir benutzen dabei die Notation

$$L^n_{x_0,\dots,x_n} f$$

für das Lagrangesche Interpolationspolynom vom Grad n zu den Stützstellen $x_0, ..., x_n$ und

$$I^{n,[a,b]}_{x_0,\dots,x_n} f = \int_a^b (L^n_{x_0,\dots,x_n} f)(x)\,dx \tag{6.13}$$

für die zugehörige Quadraturformel.

Satz 6.9. *Eine interpolatorische Quadraturformel $I^{n,[a,b]}_{x_0,\dots,x_n}$ zu $n + 1$ Stützstellen $x_0, ..., x_n$ kann maximal die Ordnung $2n + 2$ haben.*

Beweis. Wir führen den Beweis per Widerspruch. Wir nehmen also an, es gäbe eine Formel $I^{n,[a,b]}_{x_0,\dots,x_n}$ der Ordnung $2n + 3$, d.h. diese würde alle Polynome $p \in P_{2n+2}$ exakt integrieren. Diese müsste also insbesondere das Polynom

$$q(x) := \prod_{j=0}^{n} (x - x_j)^2 \quad \in P_{2n+2}$$

exakt integrieren.

Nun gilt aber nach Definition für das zu q gehörige Interpolationspolynom $L^n_{x_0,\dots,x_n}q$, dass

$$(L^n_{x_0,\dots,x_n}q)(x_j) = q(x_j) = 0 \quad (j = 0, \dots, n).$$

Daraus folgt, dass $L^n_{x_0,\dots,x_n}q$ mindestens $(n+1)$ Nullstellen hat. Da $L^n_{x_0,\dots,x_n}q$ aber nur ein Polynom n-ten Grades ist, folgt

$$L^n_{x_0,\dots,x_n}q \equiv 0.$$

Nach der Definition (6.13) gilt dann auch

$$I^{n,[a,b]}_{x_0,\dots,x_n}q = \int_a^b (L^n_{x_0,\dots,x_n}q)(x)\,dx = 0.$$

Andererseits ist das Integral über q aber positiv

$$\int_a^b q(x)\,dx = \int_a^b \underbrace{\prod_{j=0}^{n}(x-x_j)^2}_{\geq 0}\,dx > 0.$$

Daraus folgt, dass dieses Polynom nicht exakt integriert wird. □

Wir werden im nächsten Abschnitt sehen, dass es für jedes n durch geschickte Wahl der Stützstellen $x_0, \dots, x_n$ möglich ist, eine Quadraturformel $I^{n,[a,b]}_{x_0,\dots,x_n}$ der Ordnung $2n+2$ zu erzeugen.

Definition 6.10. *Eine interpolatorische Quadraturformel $I^{n,[a,b]}_{x_0,\dots,x_n}$ zu $(n+1)$ Stützstellen $x_0, \dots, x_n$ wird Gauß-Quadraturformel genannt, wenn sie die Ordnung $2n+2$ hat, d.h. wenn für alle Polynome $p \in P_{2n+1}$ gilt, dass*

$$\int_a^b p(x)\,dx = I^{n,[a,b]}_{x_0,\dots,x_n}p.$$

6.3.1 Konstruktion von Gauß-Quadraturformeln

Die Konstruktion von Gauß-Formeln basiert auf folgendem Resultat.

Satz 6.11. *Eine interpolatorische Quadraturformel $I^{n,[a,b]}_{x_0,\dots,x_n}$ zu $(n+1)$ paarweise verschiedenen Stützstellen $x_0, \dots, x_n$ ist genau dann eine Gauß-Formel, wenn gilt*

$$\int_a^b \left(\prod_{j=0}^{n}(x-x_j)\right) \cdot q(x)\,dx = 0 \quad \forall q \in P_n. \tag{6.14}$$

Beweis. (i) Wir beginnen mit einer Vorbemerkung. Sei $p \in P_{2n+1}$ ein beliebiges Polynom vom Grad $2n+1$ und $x_0, \dots, x_{2n+1} \in [a,b]$ beliebige paarweise verschiedene Punkte. Dieses Polynom wird vom Lagrangeschen Interpolationspolynom $L^{2n+1}_{x_0,\dots,x_{2n+1}}$ exakt dargestellt, d.h. es gilt

$$p(x) = (L^{2n+1}_{x_0,\dots,x_{2n+1}}p)(x) = \sum_{i=0}^{2n+1} p[x_0,\dots,x_i]\prod_{j=0}^{i-1}(x-x_j). \tag{6.15}$$

Dabei haben wir die Newtonsche Darstellung des Interpolationspolynoms verwendet.

(ii) "⇐": Wir gehen nun davon aus, dass die Stützstellen $x_0, ..., x_n$ die Orthogonalitätsbedingung (6.14) erfüllen. Aus der Darstellung (6.15) folgt, dass

$$\begin{aligned}\int_a^b p(x)\,dx - I^{n,[a,b]}_{x_0,...,x_n} p &= \int_a^b (L^{2n+1}_{x_0,...,x_{2n+1}} p)(x) - (L^n_{x_0,...,x_n} p)(x)\,dx \\ &= \int_a^b \left(\sum_{i=0}^{2n+1} p[x_0,...,x_i] \prod_{j=0}^{i-1} (x - x_j) \right) - \left(\sum_{i=0}^{n} p[x_0,...,x_i] \prod_{j=0}^{i-1} (x - x_j) \right) dx \\ &= \int_a^b \sum_{i=n+1}^{2n+1} p[x_0,...,x_i] \prod_{j=0}^{i-1} (x - x_j)\,dx.\end{aligned} \tag{6.16}$$

Aus der letzten Darstellung können wir noch den Faktor $\prod_{j=0}^{n}(x - x_j)$ ausklammern und es folgt mit der Voraussetzung (6.14)

$$\int_a^b p(x)\,dx - I^{n,[a,b]}_{x_0,...,x_n} p = \int_a^b \prod_{j=0}^{n} (x - x_j) \cdot \underbrace{\left(\sum_{i=n+1}^{2n+1} p[x_0,...,x_i] \prod_{j=n+1}^{i-1} (x - x_j) \right)}_{\in P_n} dx = 0.$$

Polynome $p \in P_{2n+1}$ werden also exakt integriert.

(iii) "⇒": Ist $I^{n,[a,b]}_{x_0,...,x_n}$ eine Gauß-Formel, so gilt für beliebige $p \in P_{2n+1}$ und beliebige $x_{n+1}, ..., x_{2n+1}$

$$\int_a^b p(x)\,dx - I^{n,[a,b]}_{x_0,...,x_n} p = 0.$$

Mit der Rechnung in (6.16) folgt

$$\int_a^b \prod_{j=0}^{n} (x - x_j) \cdot \underbrace{\left(\sum_{i=n+1}^{2n+1} p[x_0,...,x_i] \prod_{j=n+1}^{i-1} (x - x_j) \right)}_{=:q \in P_n} dx = 0.$$

Da p beliebig ist, können die dividierten Differenzen $(p[x_0], ..., p[x_0, ..., x_m])$ beliebige Werte im $\mathbb{R}^m$ annehmen (Beweis per Induktion nach m). Es folgt

$$\int_a^b \prod_{j=0}^{n} (x - x_j) \cdot q(x)\,dx = 0 \qquad \forall q \in P_n.$$

□

Es bleibt zu zeigen, dass die Stützstellen $x_0, ..., x_n$ so gewählt werden können, dass die Orthogonalitätsbedingung (6.14) gilt, d.h. für

$$N_{n+1}(x) := \prod_{j=0}^{n} (x - x_j) \tag{6.17}$$

soll gelten, dass

$$(N_{n+1}, q)_{(a,b)} = \int_a^b N_{n+1}(x)q(x)\,dx = 0 \quad \forall q \in P_n.$$

Wir suchen also ein Polynom $N_{n+1} \in P_{n+1}$ der Form (6.17), das bezüglich des L^2-Skalarprodukts orthogonal auf dem Raum P_n steht. Dieses kann mithilfe des **Gram-Schmidt**-Algorithmus zur Orthogonalisierung konstruiert werden. Dazu gehen wir von der Monombasis $\{1, x, ..., x^n\}$ aus und orthogonalisieren diese bzgl. des $L^2(a,b)$-Skalarprodukts.

Gram-Schmidt-Algorithmus zur Berechnung einer Orthogonalbasis bzgl $L^2(a,b)$

$$\begin{aligned} N_0 &\equiv 1 \\ N_n(x) &= x^n - \sum_{i=0}^{n-1} \frac{(x^n, N_i(x))_{(a,b)}}{\|N_i\|^2_{L^2(a,b)}} N_i(x), \quad n \geq 1. \end{aligned} \tag{6.18}$$

Dieser Algorithmus liefert Polynome N_n der Form (6.17) (mit führendem Koeffizienten 1), welche orthogonal aufeinander stehen. Dies sieht man induktiv: Gilt $(N_j, N_i) = 0$ für $i \neq j$, $i, j \leq n-1$, so folgt

$$\begin{aligned} (N_n, N_j)_{(a,b)} &= (x^n, N_j)_{(a,b)} - \sum_{i=0}^{n-1} (x^n, N_i)_{(a,b)} \frac{(N_i, N_j)_{(a,b)}}{\|N_i\|^2_{L^2(a,b)}} \\ &= (x^n, N_j)_{(a,b)} - (x^n, N_j)_{(a,b)} = 0. \end{aligned}$$

Da $n+1$ orthogonale Polynome $N_j, j = 0, ..., n$ den ganzen P_n aufspannen, gilt

$$(N_{n+1}, q)_{(a,b)} = 0 \quad \forall q \in P_n.$$

Um die Nullstellen von N_{n+1} als optimale Stützstellen x_i, $i = 0, ..., n$ bei der Gauß-Quadratur verwenden zu können, müssen wir noch zeigen, dass diese reell und einfach sind und in $[a, b]$ liegen.

Satz 6.12. *Es existiert genau ein Polynom der Form*

$$N_{n+1}(x) := \prod_{j=0}^{n} (x - x_j), \tag{6.19}$$

welches die Orthogonalitätsbedingung

$$(N_{n+1}, q)_{(a,b)} = 0 \quad \forall q \in P_n. \tag{6.20}$$

erfüllt. Dessen Nullstellen $x_0, ..., x_n$ sind reell und einfach und liegen in $[a, b]$.

Beweis. (i) Die Existenz eines solchen Polynoms haben wir bereits oben konstruktiv gezeigt. Die Eindeutigkeit folgt mithilfe einer einfachen Betrachtung. Seien 2 Polynome p_1, p_2 der Form

(6.19) gegeben. Dann ist die Differenz $p := p_1 - p_2$ ein Polynom n-ten Grades (Der führende Koeffizient x^{n+1} verschwindet). Aufgrund der Orthogonalitätsbeziehung (6.20) gilt

$$(p, q)_{(a,b)} = 0 \qquad \forall q \in P_n.$$

Wir wählen $q = p$ und erhalten

$$\int_a^b p^2(x)\, dx = 0 \quad \Rightarrow p \equiv 0.$$

(ii) Nach dem Fundamentalsatz der Algebra hat N_{n+1} $(n+1)$ komplexe Nullstellen (wenn mehrfache Nullstellen mehrfach gezählt werden). Wir zeigen zunächst per Widerspruch, dass diese reell und einfach sind. Sei also eine Nullstelle λ nicht reell oder mehrfach. Im ersten Fall ist dann auch $\overline{\lambda}$ eine Nullstelle. In beiden Fällen gilt

$$N_{n+1}(x) = p_{n-1}(x)|x - \lambda|^2$$

mit $p_{n-1} \in P_{n-1}$. Es gilt dann nach (6.20)

$$0 = (N_{n+1}, p_{n-1})_{(a,b)} = \int_a^b p_{n-1}(x)^2\, |x - \lambda|^2\, dx.$$

Da der Integrand nicht-negativ ist, muss gelten $p_{n-1} \equiv 0$ und damit $N_{n+1} \equiv 0$. Dies ist ein Widerspruch zu Darstellung (6.19).

(iii) Schließlich zeigen wir noch, dass die Nullstellen in $[a, b]$ liegen. Auch hierfür benutzen wir ein Widerpruchsargument und nehmen an, dass es eine Nullstelle $\lambda \notin [a, b]$ gibt. Dann gilt die Darstellung

$$N_{n+1}(x) = p_n(x)(x - \lambda)$$

mit $p_n \in P_n$ und damit wegen (6.20)

$$0 = (N_{n+1}, p_n)_{(a,b)} = \int_a^b \frac{N_{n+1}(x)^2}{x - \lambda}\, dx.$$

Wenn $\lambda \notin [a, b]$ hat der Nenner keinen Vorzeichenwechsel in $[a, b]$ und wir können wieder folgern, dass $N_{n+1} \equiv 0$, was einen Widerspruch zu (6.19) ergibt. □

Für den Spezialfall $[a, b] = [-1, 1]$ heißen die Polynome N_i, $i = 0, ..., n$ *Legendre-Polynome*.

Berechnung der Stützstellen Wir berechnen nun noch die Stützstellen $x_0, ..., x_n$ der ersten Gauß-Formeln für das Intervall $[a, b] = [-1, 1]$. Die Stützstellen im allgemeinen Fall $[a, b]$ ergeben sich dann mithilfe der linearen Transformation (siehe unten).

$$\tilde{x}_i = \phi(x_i) = \frac{1}{2}(a + b) + \frac{1}{2}(b - a)x_i.$$

Dazu führen wir den Gram-Schmidt-Algorithmus (6.18) durch. Es ergibt sich

$$\begin{aligned}
N_0(x) &= 1 \\
N_1(x) &= x - \underbrace{(x,1)_{L^2(-1,1)}}_{=0} \frac{1}{\|1\|^2_{L^2(-1,1)}} = x \\
N_2(x) &= x^2 - (x^2,1)_{L^2(-1,1)} \frac{1}{\|1\|^2_{L^2(-1,1)}} - \underbrace{(x^2,x)_{(-1,1)}}_{=0} \frac{x}{\|x\|^2_{L^2(-1,1)}} \\
&= x^2 - \frac{2}{3}\cdot\frac{1}{2} = (x-\sqrt{1/3})(x+\sqrt{1/3}).
\end{aligned}$$

Die Gauß-Formel für $n = 0$ hat also die Stützstelle $x_0 = 0$, die Formel für $n = 1$ die beiden Stützstellen $x_0 = -\sqrt{1/3}$ und $x_1 = \sqrt{1/3}$. Wie hier gilt ganz allgemein, dass die Stützstellen symmetrisch zum Mittelpunkt (hier $x = 0$) liegen. Für $n = 0$ ergibt sich die bereits oben analysierte **Mittelpunktregel**, welche die Ordnung $2 = 2n + 2$ aufweist.

Die Gewichte α_i zu den Stützstellen berechnen sich dann wieder wie oben

$$I^{n,[-1,1]}_{x_0,\dots,x_n} f = \int_{-1}^{1} (L^n_{x_0,\dots,x_n} f)(x)\,dx = \sum_{i=0}^{n} f(x_i) \underbrace{\int_{-1}^{1} \prod_{j=0, j\neq i}^{n} \frac{x-x_j}{x_i-x_j}\,dx}_{=:\alpha_i}.$$

Für die Mittelpunktregel auf $[-1,1]$ haben wir

$$\alpha_0 = \int_{-1}^{1} 1\,dx = 2 \qquad \Rightarrow I^{0,[-1,1]}_G f = 2f(0).$$

Für die Gauß-Formel erster Ordnung ergibt sich

$$\begin{aligned}
\alpha_0 &= \int_{-1}^{1} \frac{x-\sqrt{1/3}}{-\sqrt{1/3}-\sqrt{1/3}}\,dx = 1 \\
\alpha_1 &= \int_{-1}^{1} \frac{x+\sqrt{1/3}}{\sqrt{1/3}+\sqrt{1/3}}\,dx = 1 \qquad \Rightarrow I^{1,[-1,1]}_G f = f(-\sqrt{1/3}) + f(\sqrt{1/3}).
\end{aligned}$$

Diese hat nach Konstruktion die Ordnung $2 \cdot 1 + 2 = 4$, während die Newton-Cotes-Formel zu 2 Stützstellen (die Trapezregel) nur die Ordnung 2 aufweist. Wir geben noch die Gauß-Formel für $n = 2$ auf $[-1,1]$ an

$$I^{2,[-1,1]}_G f = \frac{1}{9}\left(5f(-\sqrt{3/5}) + 8f(0) + 5f(\sqrt{3/5})\right).$$

Mithilfe der Transformation

$$\phi(x) = \underbrace{\frac{1}{2}(a+b)}_{=x_m} + \underbrace{\frac{b-a}{2}}_{=\theta} x = x_m + \theta x$$

ergeben sich die folgenden Gauß-Formeln auf $[a,b]$,

$$
\begin{aligned}
I_G^{0,[a,b]} f &= (b-a) f(x_m), \\
I_G^{1,[a,b]} f &= \frac{b-a}{2} f(x_m - \sqrt{1/3}\,\theta) + f(x_m + \sqrt{1/3}\,\theta), \\
I_G^{2,[a,b]} f &= \frac{b-a}{18} \left(5f(x_m - \sqrt{3/5}\,\theta) + 8f(x_m) + 5f(x_m + \sqrt{3/5})\,\theta \right).
\end{aligned}
$$

Fehlerabschätzung Abschließend zeigen wir noch folgende Fehlerabschätzung für die Gauß-Quadraturformeln.

Satz 6.13. *Sei $f \in C^{2n+2}[a,b]$. Für die eindeutige bestimmte Gauß-Formel $I_G^{n,[a,b]}$ vom Grad n mit zugehörigen Stützstellen $x_0, ..., x_n$ gilt die Fehlerdarstellung*

$$
\int_a^b f(x)\,dx - I_G^{n,[a,b]} f = \int_a^b \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} \prod_{j=0}^{n} (x - x_j)^2\,dx \tag{6.21}
$$

mit von x abhängigen Stellen $\xi_x \in [a,b]$. Weiter gilt die Fehlerabschätzung

$$
\Big| \int_a^b f(x)\,dx - I_G^{n,[a,b]} f \Big| \leq \max_{\xi \in [a,b]} \frac{|f^{(2n+2)}(\xi)|}{2^{n+1}} (b-a) h^{2n+2} \tag{6.22}
$$

wobei $h := \max\{x_0 - a, x_1 - x_0, ..., x_n - x_{n-1}, b - x_n\}$ den maximalen Abstand zwischen benachbarten Rand- bzw. Stützstellen bezeichne.

Beweis. Wir betrachten die folgende Hermite-Interpolationsaufgabe: Finde $q \in P_{2n+1}$, so dass

$$
q(x_i) = f(x_i), \quad q'(x_i) = f'(x_i), \qquad i = 0, ..., N.
$$

Diese hat nach Satz 5.12 eine eindeutig bestimmte Lösung. Da $q \in P_{2n+1}$ wird diese von der Gauß-Quadraturformel $I_G^{n,[a,b]}$ exakt integriert. Es gilt also

$$
\int_a^b f(x)\,dx - I_G^{n,[a,b]}(f) = \int_a^b f(x) - q(x)\,dx - I_G^{n,[a,b]}(f - q)
$$

Der zweite Term auf der rechten Seite verschwindet, da

$$
I_G^{n,[a,b]}(f - q) = \sum_{i=0}^{n} \alpha_i \big(\underbrace{f(x_i) - q(x_i)}_{=0} \big) = 0.
$$

Mithilfe der Fehlerdarstellung der Hermite-Interpolation aus Satz 5.12 folgt

$$
\int_a^b f(x)\,dx - I_G^{n,[a,b]}(f) = \int_a^b f(x) - q(x)\,dx = \int_a^b \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} \prod_{i=0}^{n} (x - x_i)^2\,dx
$$

N	Mittelpunktregel	Trapezregel	Simpsonregel	Gauß $(n=1)$	Gauß $(n=2)$
2	7.77e-03	-1.55e-02	3.12e-05	2.08e-05	3.61e-08
4	1.92e-03	-3.84e-03	1.99e-06	1.33e-06	4.02e-10
8	4.79e-04	-9.59e-04	1.25e-07	8.31e-08	5.72e-12
16	1.20e-04	-2.40e-04	7.79e-09	5.20e-09	6.06e-14
32	2.99e-05	-5.99e-05	4.87e-10	3.25e-10	2.52e-14
64	7.48e-06	-1.50e-05	3.05e-11	2.03e-11	2.68e-14
128	1.87e-06	-3.74e-06	1.93e-12	1.24e-12	2.68e-14
256	4.68e-07	-9.36e-07	1.46e-13	5.25e-14	2.75e-14
512	1.17e-07	-2.34e-07	3.43e-14	2.12e-14	2.73e-14
	$\mathcal{O}(H^2)$	$\mathcal{O}(H^2)$	$\mathcal{O}(H^4)$	$\mathcal{O}(H^4)$	$\mathcal{O}(H^6)$

Tabelle 6.2: Fehler von summierten Newton-Cotes und Gauß-Quadraturformeln bei der Approximation des Integrals $\int_0^1 \exp(-x^2)\,dx$.

und damit (6.21). Die Fehlerabschätzung (6.22) folgt wie im Beweis von Satz 6.2 aus der Abschätzung

$$\Big|\prod_{j=0}^{n}(x-x_j)^2\Big| \le h^2\cdot(2h)^2\cdot((n+1)h)^2 = ((n+1)!)^2 h^{2n+2}$$

und

$$\frac{((n+1)!)^2}{(2n+2)!} = \frac{1\cdot 2\cdot \ldots\cdot(n+1)}{(n+2)\cdot(n+3)\cdot\ldots\cdot(2n+2)} < \left(\frac{1}{2}\right)^{n+1}.$$

□

Schließlich bemerken wir, dass die Gauß-Quadraturformeln wie die Newton-Cotes-Formeln in der Praxis oft **stückweise** angewendet werden. Wie bei den stückweisen Newton-Cotes-Formeln (Satz 6.7) zeigt man auch hier, dass sich die Ordnung in (6.22) auf die Konvergenzordnung $\mathcal{O}(H^{2n+2})$ der stückweisen Gauß-Quadraturformeln überträgt.

Numerisches Beispiel Wir vergleichen die summierten Gauß-Formeln für $n=0$ (Mittelpunktregel), $n=1$ und $n=2$ mit den Newton-Cotes-Formeln für $n=1$ (Trapezregel) und $n=2$ (Simpsonregel), anhand des Integrals

$$\int_0^1 \exp(-x^2)\,dx.$$

Die Ergebnisse sind in Tabelle 6.2 gegeben. Auch bei den Gauß-Formeln stimmen die beobachteten Konvergenzordnungen mit den zu erwartenden ($\mathcal{O}(H^{2n+2})$) überein. Wir sehen, dass die Gauß-Formel für $n=1$ schon leicht bessere Werte liefert als die Simpsonregel (Newton-Cotes-Formel mit $n=2$). Die Gauß-Formel für $n=2$ erreicht bereits bei $N=16$ Teilintervallen die Fehlerordnung $\mathcal{O}(10^{-14})$. Anschließend stagniert der Fehler wie bei den anderen Verfahren für größere N auch bei $\mathcal{O}(10^{-14})$ aufgrund von Rundungsfehlern.

Summierte Gauß-Quadratur Auch die Gauß-Quadratur erbt die Nachteile der Polynominterpolation für großes n. Daher unterteilen wir $[a, b]$ wieder in N Teilintervalle I_j $(j = 1, ..., N)$ und wenden die Gauß-Quadratur stückweise an:

$$I_{H,G}^{(n)}(f) := \sum_{j=1}^{N} I_G^{n,[x_{j-1},x_j]}(f), \qquad H := \max_{j=1,...,N}(x_j - x_{j-1}).$$

Es gilt folgendes Konvergenzresultat:

Satz 6.14. *Sei $f \in C[a, b]$ und stückweise regulär $f \in C^{n+1}(I_j), j = 1, ..., N$. Für die Gauß-Quadraturformel mit Polynomgrad n und N Teilintervallen gilt die Fehlerabschätzung*

$$\Big| \int_a^b f(x)\, dx - I_{H,G}^{(n)}(f) \Big| \leq \frac{H^{2n+2}}{2^{n+1}}(b - a) \sup_{\xi \in [a,b]\setminus\{x_0,...,x_N\}} \big|f^{(2n+2)}(\xi)\big|,$$

Beweis. Anwendung von Satz 6.13 in jedem Teilintervall. □

Auch hier überträgt sich die Ordnung der Quadraturformel $(2n + 2)$ also wieder auf die Konvergenzordnung der summierten Formel.

Literaturverzeichnis

[1] W. Dahmen, A. Reusken: *Numerik für Ingenieure und Naturwissenschaftler*, Springer-Verlag, 2006.

[2] R. Denk, R. Racke: *Kompendium der Analysis*, Vieweg+ Teubner Verlag, 2012.

[3] G. Hämmerlin, K.-H. Hoffmann: *Numerische Mathematik*, Vol. 7. Springer-Verlag, 2013.

[4] R. Rannacher: Vorlesungsskriptum *Numerik 0: Einführung in die Numerische Mathematik*, Heidelberg University Publishing, 2017. `https://doi.org/10.17885/heiup.206.281`

[5] T. Richter, T. Wick: *Einführung in die Numerische Mathematik: Begriffe, Konzepte und zahlreiche Anwendungsbeispiele*, Springer-Verlag, 2017.

[6] H. Werner, R. Schaback: *Praktische Mathematik II*, Springer (1972).

[7] J. Stoer, R. Bulirsch: *Introduction to numerical analysis*, Vol. 12, Springer, 2013.