Einführung in die Numerik

Skript zur Vorlesung
im
Frühjahrsemester 2024

Helmut Harbrecht

Stand: 1. April 2026

Vorwort

Dieses Vorlesungsskript kann und soll nicht ganz den Wortlaut der Vorlesung wiedergeben. Es soll das Nacharbeiten des Inhalts der Vorlesung erleichtern. Speziell enthält das Skript auch weiterführende Anmerkungen und ergänzende Beispiele.

Im Inhaltsverzeichnis findet folgende Farbkodierung verwendet:

- Rot markiert ist der Vorlesungsstoff, der für *alle Studierende* relevant ist. Dieser Stoff wird montags unterrichtet wird.
- Grün markiert ist der Vorlesungsstoff, der nur in *Einführung in die Numerik* behandelt wird, also dienstags unterrichtet wird.
- Blau markiert sind weiterführende Kapitel, die *nicht* in der Vorlesung behandelt werden. Deren Inhalt hat ergänzenden Charakter und darf durchaus auch durchgearbeitet werden.

Literatur zur Vorlesung:

— M. Hanke-Bourgeois: *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*, Teubner-Verlag

— H. Harbrecht und M. Multerer: *Algorithmische Mathematik: Graphen, Numerik und Probabilistik*, Springer-Verlag

— R. Schaback und H. Wendland: *Numerische Mathematik*, Springer-Verlag

— J. Stoer und R. Bulirsch: *Numerische Mathematik I+II*, Springer-Verlag

Inhaltsverzeichnis

1 Grundlagen **5**
1.1 Einführung . . . 5
1.2 Gleitkommazahlen . . . 7
1.3 Rundung . . . 8
1.4 Kondition und Stabilität . . . 9

2 Lineare Gleichungssysteme **13**
2.1 Vektor- und Matrixnormen . . . 13
2.2 Fehlerbetrachtungen . . . 18
2.3 Gauß-Algorithmus revisited . . . 20
2.4 Block-Gauß-Elimination . . . 25
2.5 LR-Zerlegung mit Pivotisierung . . . 26
2.6 Cholesky-Zerlegung . . . 30

3 Polynominterpolation **36**
3.1 Lagrange-Interpolation . . . 36
3.2 Neville-Schema . . . 40
3.3 Newtonsche Interpolationsformel . . . 42
3.4 Tschebyscheff-Interpolation . . . 45

4 Trigonometrische Interpolation **48**
4.1 Theoretische Grundlagen . . . 48
4.2 Schnelle Fourier-Transformation . . . 52
4.3 Zirkulante Matrizen . . . 56

5 Splines **59**
5.1 Spline-Räume . . . 59
5.2 Kubische Splines . . . 61
5.3 B-Splines . . . 63
5.4 Interpolationsfehler . . . 67

6 Numerische Quadratur **70**
6.1 Trapezregel . . . 70
6.2 Newton-Cotes-Formeln . . . 72
6.3 Adaptive Quadratur . . . 75
6.4 Euler-Maclaurinsche Summenformel . . . 77
6.5 Romberg-Verfahren . . . 79
6.6 Quadratur periodischer Funktionen . . . 82
6.7 Orthogonalpolynome . . . 84

6.8 Gauß-Quadratur 89

7 Lineare Ausgleichsprobleme 92
7.1 Normalengleichungen 92
7.2 QR-Zerlegung 96
7.3 Methode der Orthogonalisierung 102

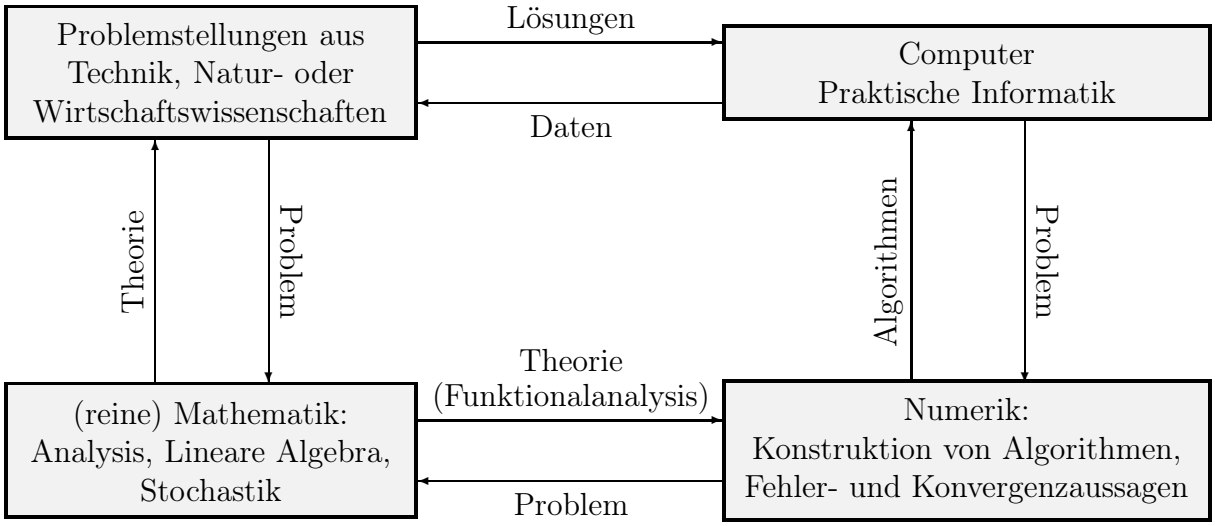
8 Iterative Lösungsverfahren 105
8.1 Fixpunktiterationen 105
8.2 Iterationsverfahren für lineare Gleichungsysteme 110
8.3 Newton-Verfahren 113
8.4 Verfahren der konjugierten Gradienten 117

1. Grundlagen

1.1 Einführung

Die Aufgabe der Numerik ist die Konstruktion und Analyse von Algorithmen zur Lösung mathematischer Aufgaben. Solche Aufgaben stammen zum Beispiel aus der Technik, den Naturwissenschaften, den Wirtschaftswissenschaften oder den Sozialwissenschaften. Mathematische Methoden sind häufig auf ein spezielles Anwendungsgebiet zugeschnitten. Sobald Zahlenwerte erlaubt sind, treten jedoch überall ähnliche Probleme auf. Beispielsweise treten in 70% aller Anwendungen lineare Gleichungssysteme auf.

Beziehung der Numerik zu anderen Bereichen:



Ziel der Numerik: Das Ziel der Numerik ist die Konstruktion ökonomischer und stabiler Algorithmen. Speziell gilt es, mögliche Fehlerquellen zu berücksichtigen. Diese ergeben sich durch Modellierungsfehler, durch Fehler in den Eingangsdaten und durch Fehler im Algorithmus. Mit den letztgenannten werden wir uns in diesem Abschnitt befassen. Zum Einstieg in dieses Thema betrachten wir zunächst ein konkretes Beispiel.

Beispiel 1.1 (Algorithmen zur Berechnung von e) Es gibt viele verschiedene Möglichkeiten, um die Eulersche Zahl e numerisch zu approximieren. Nachfolgendend führen wir vier verschiedene Varianten an.

1. Einerseits erhält man e, indem man den Grenzwert

$$e^x = \lim_{n\to\infty} \left(1 + \frac{x}{n}\right)^n$$

für $x = 1$ bildet.

2. Eine andere Möglichkeit ist, die Funktion

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = \lim_{m\to\infty} \sum_{n=0}^{m} \frac{x^n}{n!}$$

für $x = 1$ auszuwerten.

3. Da $\ln(x)$ Umkehrfunktion von e^x ist, ist e die Nullstelle von $g(x) = \ln(x) - 1$, das heißt, wir suchen ein $x^\star$ mit

$$g(x^\star) \stackrel{!}{=} 0.$$

Dabei kann

$$\ln(x) = \int_1^x \frac{1}{t}\,\mathrm{d}t$$

verwendet werden.

4. Schließlich kann e aus dem Anfangswertproblem

$$y' = y, \quad y(0) = 1 \quad \Longrightarrow \quad y(1) = e$$

bestimmt werden, da $y(x) = e^x$ die eindeutige Lösung dieses Problems ist.

Wir wollen unsere Diskussion hier an dieser Stelle auf die ersten beiden Varianten beschränken. Die Varianten 3 und 4 können erst später behandelt werden, wenn die zugrundeliegende Theorie bereitgestellt wurde.

m	n	$e_m^{(1)}$	$\Delta_m^{(1)}$	$e_m^{(2)}$	$\Delta_m^{(2)}$
$m = 1$	$n = 10^1$	2.593742	$1.24 \cdot 10^{-1}$	2.000000	$7.18 \cdot 10^{-1}$
$m = 2$	$n = 10^2$	2.704814	$1.35 \cdot 10^{-2}$	2.500000	$2.18 \cdot 10^{-1}$
$m = 3$	$n = 10^3$	2.716924	$1.36 \cdot 10^{-3}$	2.666667	$5.16 \cdot 10^{-2}$
$m = 4$	$n = 10^4$	2.718146	$1.36 \cdot 10^{-4}$	2.708333	$9.95 \cdot 10^{-3}$
$m = 5$	$n = 10^5$	2.718268	$1.36 \cdot 10^{-5}$	2.716667	$1.62 \cdot 10^{-3}$
$m = 6$	$n = 10^6$	2.718280	$1.36 \cdot 10^{-6}$	2.718056	$2.26 \cdot 10^{-4}$
$m = 7$	$n = 10^7$	2.718282	$1.34 \cdot 10^{-7}$	2.718254	$2.79 \cdot 10^{-5}$
$m = 8$	$n = 10^8$	2.718282	$3.01 \cdot 10^{-8}$	2.718279	$3.06 \cdot 10^{-6}$
$m = 9$	$n = 10^9$	2.718282	$2.24 \cdot 10^{-7}$	2.718282	$3.03 \cdot 10^{-7}$
$m = 10$	$n = 10^{10}$	2.718282	$2.25 \cdot 10^{-7}$	2.718282	$2.73 \cdot 10^{-8}$
$m = 11$	$n = 10^{11}$	2.718282	$2.25 \cdot 10^{-7}$	2.718282	$2.26 \cdot 10^{-9}$
$m = 12$	$n = 10^{12}$	2.718523	$2.42 \cdot 10^{-4}$	2.718282	$1.73 \cdot 10^{-10}$
$m = 13$	$n = 10^{13}$	2.716110	$2.17 \cdot 10^{-3}$	2.718282	$1.23 \cdot 10^{-11}$
$m = 14$	$n = 10^{14}$	2.716110	$2.17 \cdot 10^{-3}$	2.718282	$8.15 \cdot 10^{-13}$
$m = 15$	$n = 10^{15}$	3.035035	$3.17 \cdot 10^{-1}$	2.718282	$5.06 \cdot 10^{-14}$
$m = 16$	$n = 10^{16}$	1.000000	$1.72 \cdot 10^{0}$	2.718282	$2.66 \cdot 10^{-15}$

Tabelle 1.1: Approximation der Eulerschen Zahl e.

Für die erste Variante berechnen wir den Ausdruck

$$e \approx e_m^{(1)} = \left(1 + \frac{1}{n}\right)^n \quad \text{für } n = 10^m,\ m = 1, 2, 3, \ldots. \tag{1.1}$$

Bei der zweiten Variante bestimmen wir

$$e \approx e_m^{(2)} = \sum_{n=0}^{m} \frac{1}{n!} \quad \text{für } m = 1, 2, 3, \ldots. \tag{1.2}$$

Beide Varianten können einfach mit Hilfe der vier Grundoperationen numerisch umgesetzt werden und liefern uns somit Algorithmen. Die Resultate und die zugehörigen Approximationsfehler $\Delta^{(i)} = |e - e_m^{(i)}|$ für $i = 1, 2$ sind in Tabelle 1.1 aufgeführt.
Bei Variante 1 stellen wir fest, dass der Fehler $\Delta^{(1)}$ ab $m = 9$ wieder wächst. Ab $m = 16$ erhalten wir überhaupt keine Approximation mehr. Variante 2 hingegen liefert sehr schnell eine gute Approximation an e, welche gleichmäßig in m konvergiert. Um zu verstehen, was bei der ersten Variante für $m = 16$ passiert, müssen wir uns die Zahlendarstellung im Computer näher ansehen. △

1.2 Gleitkommazahlen

Bekanntlich lässt sich jede Zahl $x \in \mathbb{R} \setminus \{0\}$ in normalisierter Dezimaldarstellung beschreiben:

$$x = \pm \underbrace{a}_{\text{Mantisse}} \cdot 10 \overbrace{e}^{\text{Exponent}}, \qquad 0.1 \le a < 1, \qquad e \in \mathbb{Z}.$$

Allgemein kann jede Zahl zu einer Basis $b \in \mathbb{N}$ geschrieben werden gemäß

$$x = \pm \left(\sum_{j=1}^{\infty} d_j \cdot b^{-j} \right) \cdot b^e, \qquad e = \pm \sum_{j=0}^{M} c_j \cdot b^j \tag{1.3}$$

mit $c_j, d_j \in \{0, 1, \ldots, b-1\}$ und $d_1 \neq 0$.[1)] Da Computer nur endliche Mantissenlängen m und Exponentenlängen n besitzen, das heißt

$$x = \pm \left(\sum_{j=1}^{m} d_j \cdot b^{-j} \right) \cdot b^e, \qquad e = \pm \sum_{j=0}^{n-1} c_j \cdot b^j, \tag{1.4}$$

ist die Menge $\mathcal{A}$ der Maschinenzahlen *endlich*. Insbesondere gilt

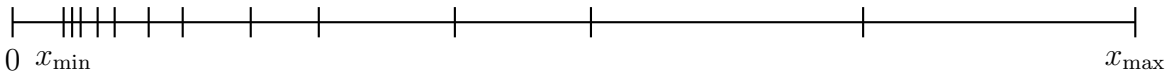
$$x \in \mathcal{A} \quad \Longrightarrow \quad |x| \in \{0\} \cup [x_{\min}, x_{\max}]$$

mit

$$x_{\min} := b^{-1} \cdot b^{-\sum_{j=0}^{n-1}(b-1)\cdot b^j},$$
$$x_{\max} := \left(\sum_{j=1}^{m} (b-1) \cdot b^{-j} \right) \cdot b^{\sum_{j=0}^{n-1}(b-1)\cdot b^j}.$$

[1)] Die Zahlendarstellung ist in dieser Form nicht eindeutig. Die Eindeutigkeit erhält man nur, wenn in der Mantisse für jedes $i \in \mathbb{N}$ der Fall $d_i = d_{i+1} = d_{i+2} = \cdots = b - 1$ ausgeschlossen wird.

Beachte: Maschinenzahlen sind nicht gleichverteilt auf dem Zahlenstrahl angeordnet, sondern wie folgt:



Beispiel 1.2 Für $x = 123.75$ gilt im sechsstelligen dezimalen Gleitkomma-Zahlensystem ($b = 10,\ m = 6$)

$$0.123\,750 \cdot 10^3,$$

während sich im 12-stelligen binären Gleitkomma-Zahlensystem ($b = 2,\ m = 12$)

$$0.111\,101\,111\,000 \cdot 2^{111}$$

ergibt. △

1.3 Rundung

Problem: Approximation von $x \in \mathbb{R}$ mit $x \notin \mathcal{A}$ durch $y \in \mathcal{A}$.
Diese Approximation heißt *Rundung*. Die Rundung ist eine Abbildung

$$\mathrm{rd} : \mathbb{R} \to \mathcal{A} \qquad x \mapsto y = \mathrm{rd}(x),$$

die folgende Eigenschaften erfüllen soll:

- $\forall a \in \mathcal{A} : \mathrm{rd}(a) = a$
- $|x - \mathrm{rd}(x)| = \min_{a \in \mathcal{A}} |x - a| \quad \rightsquigarrow$ Rundung zur nächstgelegenen Maschinenzahl

Realisierung von rd: Gegeben sei $x \in \mathbb{R}$ mit Darstellung (1.3). Gilt $|x| \in [x_{\min}, x_{\max}]$, so erhalten wir die Maschinenzahl (1.4) vermittels

$$\mathrm{rd}(x) = \begin{cases} \pm \left(\sum_{j=1}^{m} d_j \cdot b^{-j} \right) \cdot b^e & \text{falls } d_{m+1} < \frac{b}{2} \\ \pm \left(\sum_{j=1}^{m} d_j \cdot b^{-j} + b^{-m} \right) \cdot b^e & \text{falls } d_{m+1} \geq \frac{b}{2} \end{cases}$$

Bemerkung 1.3 Abschneiden der Mantisse verletzt die zweite Eigenschaft! △

Gilt $|x| < x_{\min}$ oder $|x| > x_{\max}$, so ergibt sich ein Underflow ($\mathrm{rd}(x) = 0$) beziehungsweise ein Overflow ($\mathrm{rd}(x) = \pm\texttt{inf}$).
Für den relativen Rundungsfehler gilt

$$\frac{|x - \mathrm{rd}(x)|}{|x|} \leq \frac{\frac{b}{2} \cdot b^{-(m+1)} \cdot b^e}{\underbrace{b^{-1} \cdot b^e}_{\text{untere Schranke für die Mantisse}}} = \frac{b}{2} \cdot b^{-m} =: \texttt{eps}$$

Die Zahl `eps` heißt *Maschinengenauigkeit* (round-off unit). Bei heutigen Computern gilt im allgemeinen $\texttt{eps} \approx 1.1 \cdot 10^{-16}$.

Beispiel 1.4 Die Zahl $x = 0.2$ besitzt im Binärsystem die exakte Darstellung $0.\overline{0011}$. Im Fall $m = 6$ ergibt sich gerundet $0.110\,011 \cdot 2^{-10}$. Rückkonvertiert ins Dezimalsystem bedeutet dies $\frac{51}{256} = 0.199\,218\,75$. △

Die Arithmetik von heutigen Rechnern genügt im allgemeinem dem IEEE-754-Standard. Das Ergebnis einer Gleitkommaoperation $\{\boxplus, \boxminus, \boxdot, \boxslash\}$ entspricht dann dem gerundeten Wert des exakten Rechenergebnisses. Für $x, y \in \mathcal{A}$ gilt also

$$\begin{aligned} x \boxplus y &:= \mathrm{rd}(x + y), \\ x \boxminus y &:= \mathrm{rd}(x - y), \\ x \boxdot y &:= \mathrm{rd}(x \cdot y), \\ x \boxslash y &:= \mathrm{rd}(x / y). \end{aligned}$$

Auch die Berechnung der Quadratwurzel $\boxed{\sqrt{}}\, x$ liefert den gerundeten Wert $\mathrm{rd}\left(\sqrt{x}\right)$ des exakten Rechenergebnisses $\sqrt{x}$.

Wir wollen nun unsere Beobachtungen aus Beispiel 1.1 erklären. Wegen $10^{-16} < \texttt{eps}$ folgt

$$1 \boxplus 10^{-16} = \mathrm{rd}\big(1 + 10^{-16}\big) = 1,$$

weshalb sich in (1.1) für $m = 16$

$$e_{16}^{(1)} = \mathrm{rd}\Big(\big(1 \boxplus 10^{-16}\big)^{10^{16}}\Big) = 1$$

ergibt. Die Berechnung mittels der Reihenentwicklung (1.2) funktioniert hingegen prima, da wir nur positive Zahlen aufsummieren. Da die Summanden immer kleiner werden, ändert sich dann das Ergebnis irgendwann einfach nicht mehr.

1.4 Kondition und Stabilität

Gegeben sei eine differenzierbare Funktion

$$f : \mathbb{R} \to \mathbb{R}, \qquad x \mapsto y = f(x).$$

Für fehlerbehaftete Daten $x + \Delta x$ gilt

$$\frac{f(x + \Delta x) - f(x)}{\Delta x} \approx f'(x).$$

Folglich gilt für den absoluten Datenfehler

$$\Delta y = f(x + \Delta x) - f(x) \approx f'(x)\Delta x$$

und für den relativen Datenfehler

$$\frac{\Delta y}{y} \approx \frac{f'(x)\Delta x}{f(x)} = \frac{f'(x) \cdot x}{f(x)} \frac{\Delta x}{x}.$$

Definition 1.5 Die Zahl

$$\kappa_{\mathrm{abs}} = |f'(x)|$$

heißt **absolute Konditionszahl** des Problems $x \mapsto f(x)$. Für $x \cdot f(x) \neq 0$ ist

$$\kappa_{\mathrm{rel}} = \left| \frac{f'(x) \cdot x}{f(x)} \right|$$

die entsprechende **relative Konditionszahl**. Ein Problem heißt **schlecht konditioniert**, falls eine der Konditionszahlen deutlich größer ist als 1, ansonsten heißt es **gut konditioniert**.

Beispiel 1.6

- Im Fall der Addition $f(x) = x + a$ haben wir
$$\kappa_{\text{rel}} = \left| \frac{f'(x) \cdot x}{f(x)} \right| = \left| \frac{x}{x+a} \right|.$$
Dies bedeutet, die relative Konditionszahl ist groß, wenn $|x + a| \ll |x|$. Letzteres gilt speziell dann, wenn $x \approx -a$ ist. Man spricht in diesem Fall von *Auslöschung*.
- Im Fall der Multiplikation $f(x) = ax$ gilt
$$\kappa_{\text{abs}} = |f'(x)| = |a|.$$
Die absolute Kondition ist also schlecht, falls $1 \ll a$. Wegen
$$\kappa_{\text{rel}} = \left| \frac{f'(x) \cdot x}{f(x)} \right| = \left| \frac{ax}{ax} \right| = 1$$
ist die relative Kondition jedoch immer gut.

△

Definition 1.7 Erfüllt die Implementierung eines Algorithmus $\boxed{f}$ zur Lösung eines Problems $x \mapsto f(x)$
$$\left| \frac{\boxed{f}(x) - f(x)}{f(x)} \right| \leq C_V \kappa_{\text{rel}} \, \texttt{eps}$$
mit einem mäßig großen $C_V > 0$, so wird der Algorithmus $\boxed{f}$ **vorwärtsstabil** genannt. Ergibt die Rückwärtanalyse $\boxed{f}(x) = f(x + \Delta x)$ mit
$$\left| \frac{\Delta x}{x} \right| \leq C_R \, \texttt{eps}$$
und $C_R > 0$ ist nicht zu groß, so ist der Algorithmus $\boxed{f}$ **rückwärtsstabil**.

Für die Vorwärtsstabilität wird gemessen, wie weit das Resultat des Algorithmus vom exakten Rechenergebnis abweicht. Bei der Rückwärtsstabilität interpretiert man das Resultat des Algorithmus als exaktes Rechenergebnis zu entsprechend geänderten Daten. In Abbildung 1.1 findet man eine Veranschaulichung dieses Sachverhaltes.

Bemerkung 1.8 Rückwärtsstabile Algorithmen sind auch vorwärtsstabil, denn für ein geeignetes Δx gilt
$$\left| \frac{\boxed{f}(x) - f(x)}{f(x)} \right| = \left| \frac{f(x + \Delta x) - f(x)}{f(x)} \right| \approx \kappa_{\text{rel}} \left| \frac{\Delta x}{x} \right| \leq C_R \kappa_{\text{rel}} \, \texttt{eps} \, .$$

△

Wir können nun die auftretenden Fehler bei der numerischen Lösung eines Problems charakterisieren. Es gilt mit der Dreiecksungleichung
$$\left| f(x) - \boxed{f}(x + \Delta x) \right| \leq \underbrace{\left| f(x) - f(x + \Delta x) \right|}_{\text{Fehler in den Daten (Kondition)}} + \underbrace{\left| f(x + \Delta x) - \boxed{f}(x + \Delta x) \right|}_{\text{Fehler im Algorithmus (Stabilität)}}.$$
Diese Erkenntnis liefert uns die folgende Faustregel.

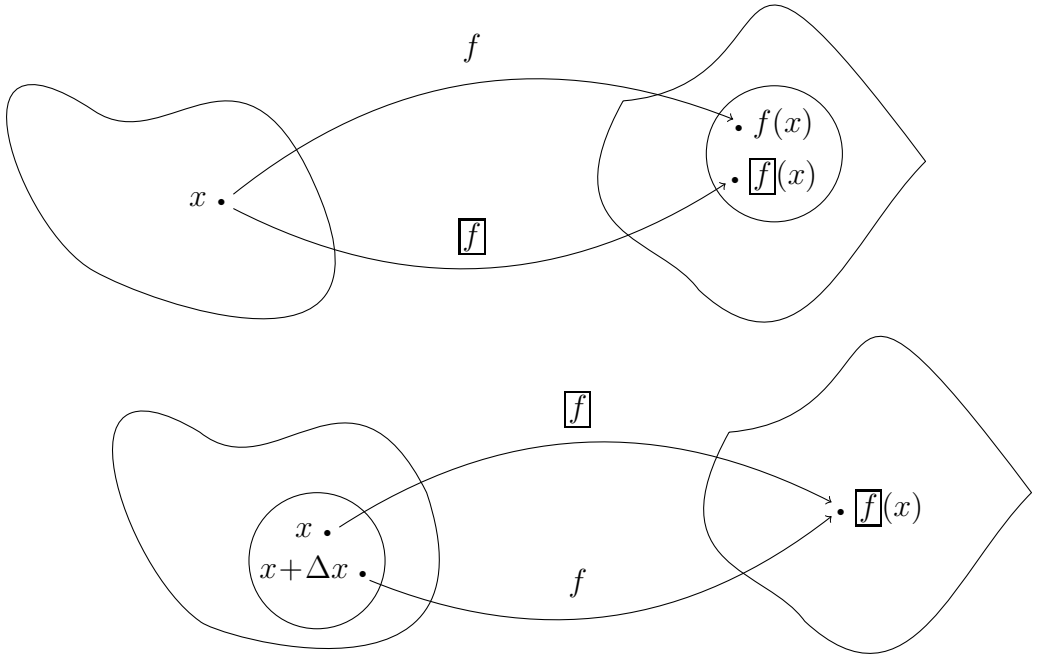


Abbildung 1.1: Veranschaulichung der Vorwärtsstabilität (oben) und der Rückwärtsstabilität (unten).

Faustregel: Ein gut konditioniertes Problem in Verbindung mit einem stabilen Algorithmus liefert gute numerische Ergebnisse. Ein schlecht konditioniertes Problem oder ein instabiler Algorithmus liefern fragwürdige Ergebnisse.

Beispiel 1.9 Die quadratische Gleichung

$$x^2 - 2px + q = 0$$

besitzt die Lösungen

$$x_{1/2} = p \pm \sqrt{p^2 - q}.$$

Diese lassen sich berechnen gemäß

```
d = sqrt(p*p-q);
x1 = p+d;
x2 = p-d;
```

Ein numerisches Beispiel mit den Werten $p = 100$ und $q = 1$, ausgeführt mit einer dreistelligen dezimalen Rechnerarithmetik, ergibt:

$$d = \tilde{\sqrt{}}(p \cdot p \ominus q) = \tilde{\sqrt{}}(10000 \ominus 1) = \tilde{\sqrt{}}(\underbrace{9999}_{=0.100\cdot 10^5}) = 0.100 \cdot 10^3,$$

$$x_1 = p \oplus d = 0.100 \cdot 10^3 \oplus 0.100 \cdot 10^3 = 0.200 \cdot 10^3 = 200,$$

$$x_2 = p \ominus d = 0.100 \cdot 10^3 \ominus 0.100 \cdot 10^3 = 0.$$

Die exakten Werte in dreistelliger Arithmetik lauten jedoch $x_1 = 200$ und $x_2 = 0.005$. In Anbetracht der Rechnergenauigkeit

$$\texttt{eps} = \frac{1}{2} \cdot 10^{-2} = 0.005$$

muss die Abweichung des errechneten Ergebnisses von der exakten Lösung als inakzeptabel betrachtet werden. Das Ergebnis $x_2 = 0$ ist schlicht falsch.
Für

$$f(x) = p - \sqrt{p^2 - x}$$

mit $|x| \ll 1 < p$ gilt

$$\kappa_{\text{abs}} = |f'(x)| = \frac{1}{2\sqrt{p^2 - x}} < 1$$

und

$$\begin{aligned}
\kappa_{\text{rel}} &= \left| \frac{f'(x) \cdot x}{f(x)} \right| = \left| \frac{x}{2\sqrt{p^2 - x}(p - \sqrt{p^2 - x})} \right| \\
&= \frac{1}{2} \left| \frac{x(p + \sqrt{p^2 - x})}{\sqrt{p^2 - x}\underbrace{(p - \sqrt{p^2 - x})(p + \sqrt{p^2 - x})}_{=x}} \right| \\
&= \frac{1}{2} \left| \frac{p + \sqrt{p^2 - x}}{\sqrt{p^2 - x}} \right| \approx 1.
\end{aligned}$$

Weil die Nullstellenberechnung gut konditioniert ist, muss folglich der Algorithmus instabil sein.
Das Problem ist die *Auslöschung* bei der Berechnung von x_2. Sie kann allerdings mithilfe des *Wurzelsatzes von Vieta*

$$x_1 x_2 = q \tag{1.5}$$

vermieden werden. Es wird lediglich die betragsgrößere Nullstelle berechnet, die zweite wird dann via (1.5) berechnet:

```
d = sqrt(p*p-q);
if (p >= 0) x1 = p+d;
else        x1 = p-d;
x2 = q/x1;
```

Konkret erhält man nun ein erheblich verbesserte Ergebnis:

$$\begin{aligned}
d &= 0.100 \cdot 10^3, \\
x_1 &= p \boxplus d = 200, \\
x_2 &= 1 \boxslash 200 = 0.005.
\end{aligned}$$

△

2. Lineare Gleichungssysteme

2.1 Vektor- und Matrixnormen

Oftmals werden wir im folgenden Normen für Vektoren und Matrizen benötigen. Dazu bezeichne $\mathbb{R}^n$ den Raum der reellwertigen Vektoren

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad x_i \in \mathbb{R}$$

und $\mathbb{R}^{m\times n}$ den Raum der reellwertigen Matrizen

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & & a_{2,n} \\ \vdots & & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{bmatrix}, \quad a_{i,j} \in \mathbb{R}.$$

Definition 2.1 Sei $X = \mathbb{R}^n$ oder $X = \mathbb{R}^{m\times n}$. Eine Abbildung

$$\|\cdot\| : X \to \mathbb{R}_{\geq 0}$$

heißt **Norm** auf X, wenn gilt

1. $\|\mathbf{x}\| > 0$ für alle $\mathbf{x} \in X \setminus \{\mathbf{0}\}$
2. $\|\alpha\mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$ für alle $\mathbf{x} \in X$ und $\alpha \in \mathbb{R}$
3. $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ für alle $\mathbf{x}, \mathbf{y} \in X$

Bemerkung 2.2 Wegen $\mathbf{x} = \mathbf{x} - \mathbf{0}$, kann $\|\mathbf{x}\|$ als Abstand von $\mathbf{x}$ zum Nullpunkt in X interpretiert werden. In der Tat hat $\mathrm{dist}(\mathbf{x}, \mathbf{y}) := \|\mathbf{x}-\mathbf{y}\|$ die Eigenschaften einer *Distanz* von zwei Elementen. Der Begriff Distanz ist allerdings allgemeiner, und nicht nur auf (normierte) Vektorräume beschränkt. Insofern liefern Normen spezielle Distanzbegriffe. △

Häufig verwendete Normen für Vektoren im Raum $X = \mathbb{R}^n$ sind:

$$\text{Betragssummennorm:} \qquad \|\mathbf{x}\|_1 := \sum_{i=1}^{n} |x_i|$$

$$\text{Euklid-Norm:} \qquad \|\mathbf{x}\|_2 := \sqrt{\sum_{i=1}^{n} |x_i|^2} = \sqrt{\mathbf{x}^\intercal \cdot \mathbf{x}}$$

$$\text{Maximumsnormorm:} \qquad \|\mathbf{x}\|_\infty := \max_{1 \le i \le n} |x_i|.$$

Dagegen sind für Matrizen im Raum $X = \mathbb{R}^{m \times n}$ nachfolgende Normen sehr gebräuchlich:

$$\text{Spaltensummennorm:} \qquad \|\mathbf{A}\|_1 := \max_{1 \le j \le n} \sum_{i=1}^{m} |a_{i,j}|$$

$$\text{Zeilensummennorm:} \qquad \|\mathbf{A}\|_\infty := \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{i,j}|$$

$$\text{Frobenius-Norm:} \qquad \|\mathbf{A}\|_F := \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{i,j}|^2}$$

Beispiel 2.3 Für die Matrix

$$\mathbf{A} = \begin{bmatrix} 2 & -3 \\ 1 & 1 \end{bmatrix}$$

gilt

$$\|\mathbf{A}\|_1 = 4, \qquad \|\mathbf{A}\|_\infty = 5, \qquad \|\mathbf{A}\|_F = \sqrt{15}.$$

△

Satz 2.4 Alle Normen auf $\mathbb{R}^n$ sind äquivalent, das heißt, für zwei Normen $\|\cdot\|_a$ und $\|\cdot\|_b$ auf $\mathbb{R}^n$ gibt es positive Konstanten $c, C > 0$ mit

$$c\|\mathbf{x}\|_a \le \|\mathbf{x}\|_b \le C\|\mathbf{x}\|_a \quad \text{für alle } \mathbf{x} \in \mathbb{R}^n.$$

Beweis. Es genügt, die Behauptung für $\|\cdot\|_a = \|\cdot\|_\infty$ zu zeigen. Dazu seien $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ beliebig und $\|\cdot\|$ eine Norm im $\mathbb{R}^n$. Wegen

$$\mathbf{x} - \mathbf{y} = \sum_{i=1}^{n} (x_i - y_i)\mathbf{e}_i$$

folgt

$$\big|\|\mathbf{x}\| - \|\mathbf{y}\|\big| \le \|\mathbf{x} - \mathbf{y}\| \le \sum_{i=1}^{n} |x_i - y_i| \|\mathbf{e}_i\| \le \|\mathbf{x} - \mathbf{y}\|_\infty \sum_{i=1}^{n} \|\mathbf{e}_i\|.$$

Folglich ist $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ eine Lipschitz-stetige Funktion mit Lipschitz-Konstante $L := \sum_{i=1}^{n} \|\mathbf{e}_i\|$. Als solche nimmt $\|\cdot\|$ auf der kompakten Einheitssphäre $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_\infty = 1\}$

sowohl ihr Maximum C als auch ihr Minimum c an. Wegen der ersten Normeigenschaft aus Definition 2.1 ist insbesondere $c > 0$. Daher folgt für beliebiges $\mathbf{z} \in \mathbb{R}^n$, dass

$$c \leq \left\| \frac{\mathbf{z}}{\|\mathbf{z}\|_\infty} \right\| \leq C,$$

beziehungsweise

$$c\|\mathbf{z}\|_\infty \leq \|\mathbf{z}\| \leq C\|\mathbf{z}\|_\infty.$$

□

Beispiel 2.5 Für $\mathbf{x} \in \mathbb{R}^n$ folgt aus

$$\max_{1\leq i\leq n} |x_i|^2 \leq \sum_{k=1}^{n} |x_k|^2 \leq n \cdot \max_{1\leq i\leq n} |x_i|^2$$

sofort die Ungleichung

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{n} \cdot \|\mathbf{x}\|_\infty.$$

△

Bemerkung 2.6 Satz 2.4 gilt auch im Fall von Matrixnormen, sprich im Fall des $\mathbb{R}^{m\times n}$. △

Der Vektorraum $\mathbb{R}^{n\times n}$ unterscheidet sich von allen anderen genannten Räumen dadurch, dass eine weitere Operation definiert ist, nämlich die Multiplikation $\mathbf{A}\cdot\mathbf{B}$ mit $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$.

Definition 2.7 Eine Matrixnorm $\|\cdot\|_M$ auf $\mathbb{R}^{n\times n}$ heißt **submultiplikativ**, falls gilt

$$\|\mathbf{A}\cdot\mathbf{B}\|_M \leq \|\mathbf{A}\|_M \cdot \|\mathbf{B}\|_M \quad \text{für alle } \mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}.$$

Eine Matrixnorm $\|\cdot\|_M$ auf $\mathbb{R}^{n\times n}$ heißt **verträglich** mit einer Vektornorm $\|\cdot\|_V$ auf $\mathbb{R}^n$, wenn gilt

$$\|\mathbf{A}\cdot\mathbf{x}\|_V \leq \|\mathbf{A}\|_M \cdot \|\mathbf{x}\|_V \quad \text{für alle } \mathbf{A} \in \mathbb{R}^{n\times n} \text{ und } \mathbf{x} \in \mathbb{R}^n.$$

Beispiel 2.8

1. $\|\mathbf{A}\| := \max_{1\leq i,j\leq n} |a_{i,j}|$ ist eine Matrixnorm auf $\mathbb{R}^{n\times n}$, aber nicht submultiplikativ:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \|\mathbf{A}\| = 1,$$
$$\mathbf{A}^2 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}, \quad \|\mathbf{A}^2\| \not\leq 1 = \|\mathbf{A}\|^2.$$

2. Mit der Cauchy-Schwarzschen Ungleichung folgt für die i-te Komponente $[\mathbf{Ax}]_i$ des Vektors $\mathbf{Ax}$ die Abschätzung

$$[\mathbf{Ax}]_i^2 = \left(\sum_{j=1}^{n} a_{i,j}\cdot x_j\right)^2 \leq \left(\sum_{j=1}^{n} |a_{i,j}|^2\right)\cdot\left(\sum_{j=1}^{n} |x_j|^2\right) = \left(\sum_{j=1}^{n} |a_{i,j}|^2\right)\cdot\|\mathbf{x}\|_2^2.$$

Somit ist die Frobenius-Norm mit der Euklid-Norm verträglich:

$$\|\mathbf{Ax}\|_2^2 = \sum_{i=1}^{n} [\mathbf{Ax}]_i^2 \leq \underbrace{\sum_{i=1}^{n} \left(\sum_{j=1}^{n} |a_{i,j}|^2 \right)}_{=\|\mathbf{A}\|_F^2} \cdot \|\mathbf{x}\|_2^2 = \|\mathbf{A}\|_F^2 \cdot \|\mathbf{x}\|_2^2.$$

△

Definition 2.9 Sei $\|\cdot\|_V$ eine Vektornorm auf $\mathbb{R}^n$. Dann ist

$$|||\mathbf{A}||| := \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{Ax}\|_V}{\|\mathbf{x}\|_V} = \max_{\|\mathbf{x}\|_V=1} \|\mathbf{Ax}\|_V$$

eine Norm auf $\mathbb{R}^{n\times n}$, die sogenannte **induzierte Norm** von $\|\cdot\|_V$. (Die Normeigenschaften sind leicht nachzuprüfen.)

Bemerkung 2.10 Die Spaltensummennorm ist von der Betragssummennorm induziert, die Zeilensummennorm ist von der Maximumsnorm induziert. Denn für die Spaltensummennorm gilt einerseits

$$\begin{aligned}\max_{\|\mathbf{x}\|_1=1} \|\mathbf{Ax}\|_1 &= \max_{\|\mathbf{x}\|_1=1} \sum_{i=1}^{n} \left| \sum_{j=1}^{n} a_{i,j} x_j \right| \leq \max_{\|\mathbf{x}\|_1=1} \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{i,j}||x_j| \\ &= \max_{\|\mathbf{x}\|_1=1} \sum_{j=1}^{n} |x_j| \sum_{i=1}^{n} |a_{i,j}| \leq \underbrace{\max_{\|\mathbf{x}\|_1=1} \sum_{j=1}^{n} |x_j|}_{=1} \left(\max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}| \right) = \|\mathbf{A}\|_1.\end{aligned}$$

Umgekehrt sei $1 \leq k \leq n$ der Index der größten Spaltensumme. Damit ergibt sich

$$\|\mathbf{A}\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{i,j}| = \sum_{i=1}^{n} |a_{i,k}| = \|\mathbf{Ae}_k\|_1 \leq \max_{\|\mathbf{x}\|_1=1} \|\mathbf{Ax}\|_1.$$

Mit einer ähnlichen Argumention zeigt man die Aussage für die Zeilensummennorm. △

Lemma 2.11 Die von $\|\cdot\|_V$ induzierte Norm $|||\cdot|||$ ist submultiplikativ und mit der Ausgangsnorm verträglich. Ist $\|\cdot\|_M$ eine mit $\|\cdot\|_V$ verträgliche Norm, dann gilt

$$|||\mathbf{A}||| \leq \|\mathbf{A}\|_M \quad \text{für alle } \mathbf{A} \in \mathbb{R}^{n\times n}.$$

Beweis. 1. Sei $\mathbf{B} \neq \mathbf{0}$, dann gilt

$$\begin{aligned}|||\mathbf{AB}||| &= \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{ABx}\|_V}{\|\mathbf{x}\|_V} = \sup_{\mathbf{Bx}\neq\mathbf{0}} \frac{\|\mathbf{ABx}\|_V}{\|\mathbf{x}\|_V} = \sup_{\mathbf{Bx}\neq\mathbf{0}} \left(\frac{\|\mathbf{ABx}\|_V}{\|\mathbf{Bx}\|_V} \cdot \frac{\|\mathbf{Bx}\|_V}{\|\mathbf{x}\|_V} \right) \\ &\leq \sup_{\mathbf{Bx}\neq\mathbf{0}} \frac{\|\mathbf{ABx}\|_V}{\|\mathbf{Bx}\|_V} \cdot \sup_{\mathbf{Bx}\neq\mathbf{0}} \frac{\|\mathbf{Bx}\|_V}{\|\mathbf{x}\|_V} \leq \sup_{\mathbf{y}\neq\mathbf{0}} \frac{\|\mathbf{Ay}\|_V}{\|\mathbf{y}\|_V} \cdot \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{Bx}\|_V}{\|\mathbf{x}\|_V} \\ &= |||\mathbf{A}||| \cdot |||\mathbf{B}|||.\end{aligned}$$

2. Wegen
$$|||\mathbf{A}||| = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_V}{\|\mathbf{x}\|_V} \geq \frac{\|\mathbf{A}\mathbf{x}\|_V}{\|\mathbf{x}\|_V} \quad \forall \mathbf{x} \in \mathbb{R}^n \backslash \{\mathbf{0}\},$$
ergibt sich
$$\|\mathbf{A}\mathbf{x}\|_V \leq |||\mathbf{A}||| \cdot \|\mathbf{x}\|_V \quad \forall \mathbf{x} \in \mathbb{R}^n \backslash \{\mathbf{0}\}.$$
Im Fall $\mathbf{x} = \mathbf{0}$ folgt sofort
$$0 = \|\mathbf{A}\mathbf{x}\|_V \leq |||\mathbf{A}||| \cdot \underbrace{\|\mathbf{x}\|_V}_{=0} = 0.$$

3. Die Behauptung ergibt sich aus
$$|||\mathbf{A}||| = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_V}{\|\mathbf{x}\|_V} \leq \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\|_M \cdot \|\mathbf{x}\|_V}{\|\mathbf{x}\|_V} = \|\mathbf{A}\|_M.$$

□

Bemerkung 2.12 Verallgemeinerungen von Definitionen 2.7 und 2.9 auf Matrixnormen im $\mathbb{R}^{m\times n}$ gelten entsprechend; in diesem Fall müssen dann Vektornormen sowohl für den $\mathbb{R}^m$ als auch für den $\mathbb{R}^n$ spezifiziert werden. △

Wir wollen zum Abschluss dieses Abschnitts noch klären, welche Matrixnorm durch die Euklid-Norm induziert wird. Dazu beachten wir, dass gilt
$$\|\mathbf{A}\|_2 := \max_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2 = \max_{\|\mathbf{x}\|_2=1} \sqrt{(\mathbf{A}\mathbf{x})^\intercal(\mathbf{A}\mathbf{x})} = \max_{\|\mathbf{x}\|_2=1} \sqrt{\mathbf{x}^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{x}}.$$

Das nachfolgende Resultat begründet, warum man diese Matrixnorm auch *Spektralnorm* nennt.

Satz 2.13 Es gilt
$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}^\intercal\mathbf{A})} = \sqrt{\max\{\lambda : \lambda \text{ ist Eigenwert von } \mathbf{A}^\intercal\mathbf{A}\}}.$$

Beweis. $\mathbf{A}^\intercal\mathbf{A}$ ist symmetrisch, das heisst, es ist $\mathbf{B}^\intercal = \mathbf{B}$ für $\mathbf{B} := \mathbf{A}^\intercal\mathbf{A}$. Außerdem ist $\mathbf{A}^\intercal\mathbf{A}$ auch positiv semidefinit, das heißt, alle Eigenwerte sind nichtnegativ. Folglich hat $\mathbf{A}^\intercal\mathbf{A}$ n nichtnegative Eigenwerte $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ und zugehörige, paarweise orthonormale Eigenvektoren $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$. Jeder Vektor $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|_2 = 1$, lässt sich entwickeln
$$\mathbf{x} = \sum_{i=1}^n \xi_i \cdot \mathbf{v}_i,$$
woraus folgt
$$1 = \|\mathbf{x}\|_2^2 = \mathbf{x}^\intercal\mathbf{x} = \left(\sum_{i=1}^n \xi_i\mathbf{v}_i^\intercal\right) \cdot \left(\sum_{j=1}^n \xi_j\mathbf{v}_j\right) = \sum_{i,j=1}^n \xi_i\xi_j \cdot \underbrace{\mathbf{v}_i^\intercal\mathbf{v}_j}_{=\delta_{i,j}} = \sum_{i=1}^n |\xi_i|^2.$$

Einerseits ergibt sich nun

$$\mathbf{x}^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{x} = \left(\sum_{i=1}^{n}\xi_i\mathbf{v}_i^\intercal\right)\cdot\underbrace{\mathbf{A}^\intercal\mathbf{A}\cdot\left(\sum_{j=1}^{n}\xi_j\mathbf{v}_j\right)}_{=\sum_{j=1}^{n}\xi_j\mathbf{A}^\intercal\mathbf{A}\mathbf{v}_j} = \sum_{i,j=1}^{n}\xi_i\xi_j\lambda_j\cdot\underbrace{\mathbf{v}_i^\intercal\mathbf{v}_j}_{=\delta_{i,j}}$$
$$= \sum_{i=1}^{n}\xi_i^2\cdot\lambda_i \le \lambda_1\cdot\underbrace{\sum_{i=1}^{n}\xi_i^2}_{=1} = \lambda_1.$$

Andererseits gilt aber auch

$$\max_{\|\mathbf{x}\|_2=1}\mathbf{x}^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{x} \ge \mathbf{v}_1^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{v}_1 = \lambda_1\cdot\mathbf{v}_1^\intercal\mathbf{v}_1 = \lambda_1.$$

Dies bedeutet aber

$$\max_{\|\mathbf{x}\|_2=1}\mathbf{x}^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{x} = \lambda_1,$$

woraus die Behauptung folgt. □

Beispiel 2.14 (Fortsetzung von Beispiel 2.3) Wir wollen für

$$\mathbf{A} = \begin{bmatrix}2 & -3\\ 1 & 1\end{bmatrix}$$

die Spektralnorm berechnen. Die Eigenwerte der Matrix

$$\mathbf{A}^\intercal\mathbf{A} = \begin{bmatrix}5 & -5\\ 5 & 10\end{bmatrix}$$

kann man mit Hilfe der Regel von Sarrus über

$$\det(\mathbf{A}^\intercal\mathbf{A}-\lambda\mathbf{I}) = \begin{vmatrix}5-\lambda & -5\\ 5 & 10-\lambda\end{vmatrix} = \underbrace{(5-\lambda)(10-\lambda)}_{=\lambda^2-15\lambda+50}+25 \stackrel{!}{=} 0$$

bestimmen. Es folgt

$$\lambda_{1/2} = \frac{15\pm 5\sqrt{5}}{2}$$

und damit

$$\|\mathbf{A}\|_2 = \sqrt{\frac{1}{2}\cdot(15+5\sqrt{5})}.$$

△

2.2 Fehlerbetrachtungen

Für eine nichtsinguläre Matrix $\mathbf{A}\in\mathbb{R}^{n\times n}$ wollen wir das lineare Gleichungssystem $\mathbf{A}\mathbf{x}=\mathbf{b}$ lösen. Offensichtlich ist bei Eingangsfehler $\Delta\mathbf{b}$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}, \qquad \mathbf{x}+\Delta\mathbf{x} = \mathbf{A}^{-1}(\mathbf{b}+\Delta\mathbf{b}) = \mathbf{A}^{-1}\mathbf{b}+\mathbf{A}^{-1}\Delta\mathbf{b},$$

das heißt

$$\Delta\mathbf{x} = \mathbf{A}^{-1}\Delta\mathbf{b}.$$

Für ein verträgliches Matrix-/Vektornormpaar ergibt sich somit

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} = \frac{\|\mathbf{A}^{-1}\Delta\mathbf{b}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}^{-1}\|\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}\frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}^{-1}\|\|\mathbf{A}\|\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}. \tag{2.1}$$

Definition 2.15 Der Faktor

$$\mathrm{cond}_M(\mathbf{A}) := \|\mathbf{A}^{-1}\|_M\|\mathbf{A}\|_M$$

wird als **Kondition** der Matrix $\mathbf{A}$ bezüglich der Matrixnorm $\|\cdot\|_M$ bezeichnet.

Wie in Abschnitt 1.4 beschreibt die Kondition die relative Fehlerverstärkung in diesem Problem, diesmal allerdings normweise für den schlimmstmöglichen Fall. Ist die Matrixnorm $\|\cdot\|_M$ durch eine Vektornorm induziert, so kann man einfache Beispiele für $\mathbf{b}$ und $\Delta\mathbf{b}$ konstruieren, für die diese Fehlerverstärkung exakt ist, also (2.1) mit Gleichheitszeichen gilt.

Beispiel 2.16 Sei

$$\mathbf{A} = \begin{bmatrix} 10^{-3} & 1 \\ 1 & 2 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}.$$

Die Matrix $\mathbf{A}$ ist gut konditioniert, denn mit

$$\mathbf{A}^{-1} \approx \begin{bmatrix} -2.004 & 1.002 \\ 1.002 & -0.001 \end{bmatrix}$$

folgt $3 = \|\mathbf{A}\|_\infty \approx \|\mathbf{A}^{-1}\|_\infty$ und daher ist

$$\mathrm{cond}_\infty(\mathbf{A}) \approx 9.$$

Das Lösen des linearen Gleichungssytems $\mathbf{A}\mathbf{x} = \mathbf{b}$ ist also gut konditioniert, wobei die Lösung

$$\mathbf{x} \approx \begin{bmatrix} 1.002\ldots \\ 0.9989\ldots \end{bmatrix}$$

lautet. Mit dem Gauß-Algorithmus und dreistelliger Gleitkommaarithmetik ergibt sich jedoch bei kleiner Datenstörung ein völlig falsches Resultat:

$$\left[\begin{array}{cc|c} 0.001 & 1 & 1.01 \\ 1 & 2 & 3.01 \end{array}\right] \begin{array}{c} \\ -1000 \\ \hookleftarrow \end{array} \longrightarrow \left[\begin{array}{cc|c} 0.001 & 1 & 1.01 \\ 0 & -998 & -1010 \end{array}\right]$$

$$\Longrightarrow \quad x_2 = 1010 \boxslash 998 = 1.01$$
$$x_1 = 1000 \boxdot (1.01 \boxminus 1.01) = 0$$

Der Grund für dieses unbefriedigende Resultat ist der folgende: Das kleine (1,1)-Element bewirkt einen großen Faktor (nämlich -1000) im Gauß-Eliminationsschritt und damit eine große Fehlerverstärkung, das heißt, Instabilität. △

Die Diagonalelemente, die bei der Gauß-Elimination im i-ten Schritt an Position (i,i) auftreten, werden *Pivotelemente* genannt. Zur Stabilisierung der Gauß-Elimination vertauscht man nun vor jedem Eliminierungsschritt die i-te und die k-te Zeile derart, dass das Pivotelement am betragsgrößten ist ("Spaltenpivotsuche" oder "partial pivoting"). Am exakten Resultat ändert das nichts!

Beispiel 2.17 (Fortsetzung von Beispiel 2.16) In unserem Fall würde man also die beiden Zeilen vertauschen, da $1 > 0.001$:

$$\left[\begin{array}{cc|c} 1 & 2 & 3.01 \\ 0.001 & 1 & 1.01 \end{array}\right] \begin{array}{c} -\frac{1}{1000} \\ \hookleftarrow \end{array} \longrightarrow \left[\begin{array}{cc|c} 1 & 2 & 3.01 \\ 0 & 0.998 & 1.01 \end{array}\right]$$

$$\Longrightarrow \quad x_2 = 1.01 \boxslash 0.998 = 1.01$$

$$x_1 = 3.01 \boxminus (2 \boxdot 1.01) = 3.01 \boxminus 2.02 = 0.99$$

△

Die Auswahl des Pivotelements hängt stark von der Skalierung des linearen Gleichungssystems ab. Beispielsweise könnte man auch einfach die erste Gleichung mit 1000 multiplizieren und das (1,1)-Element als Pivot behalten. Das Ergebnis wäre dann wieder so verheerend wie im Beispiel 2.16. Ein Ausweg wäre daher, im i-ten Teilschritt dasjenige Element als Pivotelement auszuwählen, welches am betragsgrößten ist. Dieses Vorgehen nennt sich *totale Pivotisierung* oder *total pivoting*. Es liefert die *stabilste* Variante des Gauß-Algorithmus.

2.3 Gauß-Algorithmus revisited

Vorgelegt seien $\mathbf{A} \in \mathbb{R}^{n\times n}$ und $\mathbf{b} \in \mathbb{R}^n$. Zunächst erinnern wir daran, dass man im i-ten Teilschritt mit $1 \le i \le n-1$ des Gauß-Algorithmus zur Lösung des linearen Gleichungssystems $\mathbf{Ax} = \mathbf{b}$ wie folgt vorgeht. Für alle $i < j \le n$ zieht man von der j-ten Zeile der Matrix $\mathbf{A}_i$ das jeweils $\tau_j^{(i)}$-fache der i-ten Zeile ab mit dem Ziel, die Null im (j,i)-ten Eintrag der Matrix zu erzwingen. Symbolisch führt dies auf

$$\underbrace{\left[\begin{array}{ccccccc} \star & \cdots & \star & \star & \cdots & \star & \star \\ & \ddots & \vdots & \vdots & & \vdots & \vdots \\ & & \star & \star & \cdots & \star & \star \\ & & & a_{i,i}^{(i)} & \cdots & a_{i,n}^{(i)} & b_i^{(i)} \\ & \mathbf{0} & & a_{i+1,i}^{(i)} & \cdots & a_{i+1,n}^{(i)} & b_{i+1}^{(i)} \\ & & & \vdots & & \vdots & \vdots \\ & & & a_{n,i}^{(i)} & \cdots & a_{n,n}^{(i)} & b_n^{(i)} \end{array}\right]}_{=\mathbf{A}_i \quad\quad = \mathbf{b}_i} \begin{array}{c} \\ \\ \\ -\tau_{i+1}^{(i)} \\ \hookleftarrow \\ \\ \\ \end{array} \cdots \begin{array}{c} \\ \\ \\ -\tau_n^{(i)} \\ \\ \\ \hookleftarrow \end{array} \tag{2.2}$$

Dabei muss $\tau_j^{(i)}$ wie folgt gewählt werden, um das Gewünschte zu erzielen:

$$\tau_j^{(i)} = \frac{a_{j,i}^{(i)}}{a_{i,i}^{(i)}}, \qquad i < j \le n$$

Der gemeinsame Nenner $a_{i,i}^{(i)}$ der Faktoren $\tau_j^{(i)}$ wird *Pivotelement* genannt. Der obige Eliminationsschritt kann in Matrixnotation wie folgt geschrieben werden:

$$\underbrace{\begin{bmatrix} 1 & & & & & \\ & \ddots & & & \mathbf{0} & \\ & & 1 & & & \\ & & & 1 & & \\ & & & -\tau_{i+1}^{(i)} & 1 & \\ & \mathbf{0} & & \vdots & & \ddots \\ & & & -\tau_n^{(i)} & & 1 \end{bmatrix}}_{=\mathbf{L}_i} \left[\begin{array}{c|c} \mathbf{A}_i & \mathbf{b}_i \end{array}\right] = \left[\begin{array}{c|c} \mathbf{A}_{i+1} & \mathbf{b}_{i+1} \end{array}\right],$$

wobei

$$\mathbf{L}_i = \mathbf{I} - \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ \tau_{i+1}^{(i)} \\ \vdots \\ \tau_n^{(i)} \end{bmatrix}}_{=:\boldsymbol{\ell}_i} \underbrace{\begin{bmatrix} 0 & \cdots & 0 & \overbrace{1}^{\text{Stelle } i} & 0 & \cdots & 0 \end{bmatrix}}_{=:\mathbf{e}_i^\mathsf{T}} = \mathbf{I} - \boldsymbol{\ell}_i \mathbf{e}_i^\mathsf{T}.$$

Mit $\mathbf{A}_1 = \mathbf{A}$ und $\mathbf{b}_1 = \mathbf{b}$ ergibt sich durch Auflösen der Rekursion

$$\mathbf{L}_{n-1}\mathbf{L}_{n-2}\cdots\mathbf{L}_1 \left[\begin{array}{c|c} \mathbf{A} & \mathbf{b} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{A}_n & \mathbf{b}_n \end{array}\right] = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{c} \end{array}\right] = \left[\begin{array}{ccc|c} \star & \cdots & \star & \star \\ & \ddots & \vdots & \vdots \\ \mathbf{0} & & \star & \star \end{array}\right]$$

mit der rechten oberen Dreiecksmatrix $\mathbf{R}$. Speziell gilt

$$\mathbf{L}_{n-1}\mathbf{L}_{n-2}\cdots\mathbf{L}_1\mathbf{A} = \mathbf{R},$$

das heißt, wir erhalten die Faktorisierung

$$\mathbf{A} = \underbrace{\mathbf{L}_1^{-1}\mathbf{L}_2^{-1}\cdots\mathbf{L}_{n-1}^{-1}}_{=:\mathbf{L}}\mathbf{R} = \mathbf{L}\mathbf{R}.$$

Die inversen Matrizen $\mathbf{L}_i^{-1}$ sowie $\mathbf{L}$ lassen sich explizit angeben:

Lemma 2.18

1. Die Inverse von $\mathbf{L}_i = \mathbf{I} - \boldsymbol{\ell}_i\mathbf{e}_i^\mathsf{T}$ berechnet sich gemäß

$$\mathbf{L}_i^{-1} = \mathbf{I} + \boldsymbol{\ell}_i\mathbf{e}_i^\mathsf{T} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & \mathbf{0} & \\ & & 1 & & & \\ & & & 1 & & \\ & & & \tau_{i+1}^{(i)} & 1 & \\ & \mathbf{0} & & \vdots & & \ddots \\ & & & \tau_n^{(i)} & & 1 \end{bmatrix}.$$

2. Die Matrix $\mathbf{L}$ erfüllt

$$\mathbf{L} = \mathbf{I} + \boldsymbol{\ell}_1\mathbf{e}_1^\mathsf{T} + \boldsymbol{\ell}_2\mathbf{e}_2^\mathsf{T} + \cdots + \boldsymbol{\ell}_{n-1}\mathbf{e}_{n-1}^\mathsf{T} = \begin{bmatrix} 1 & & & & \\ \tau_2^{(1)} & 1 & & \mathbf{0} & \\ \tau_3^{(1)} & \tau_3^{(2)} & \ddots & & \\ \vdots & \vdots & & 1 & \\ \tau_n^{(1)} & \tau_n^{(2)} & \cdots & \tau_n^{(n-1)} & 1 \end{bmatrix}. \tag{2.3}$$

Beweis. Aufgrund der Nulleinträge in $\boldsymbol{\ell}_i$ und $\mathbf{e}_i$ ist $\mathbf{e}_i^\mathsf{T}\boldsymbol{\ell}_j = 0$ für $i \leq j$. Daraus folgt

$$(\underbrace{\mathbf{I} - \boldsymbol{\ell}_i\mathbf{e}_i^\mathsf{T}}_{=\mathbf{L}_i})(\mathbf{I} + \boldsymbol{\ell}_i\mathbf{e}_i^\mathsf{T}) = \mathbf{I} - \boldsymbol{\ell}_i\mathbf{e}_i^\mathsf{T} + \boldsymbol{\ell}_i\mathbf{e}_i^\mathsf{T} - \boldsymbol{\ell}_i \underbrace{\mathbf{e}_i^\mathsf{T}\boldsymbol{\ell}_i}_{=0}\mathbf{e}_i^\mathsf{T} = \mathbf{I},$$

dies bedeutet, $\mathbf{L}_i^{-1} = \mathbf{I} + \boldsymbol{\ell}_i\mathbf{e}_i^\mathsf{T}$. Weiter ergibt sich induktiv aus

$$\mathbf{L}_1^{-1}\mathbf{L}_2^{-1}\cdots\mathbf{L}_i^{-1} = \mathbf{I} + \boldsymbol{\ell}_1\mathbf{e}_1^\mathsf{T} + \boldsymbol{\ell}_2\mathbf{e}_2^\mathsf{T} + \cdots + \boldsymbol{\ell}_i\mathbf{e}_i^\mathsf{T},$$

dass

$$\begin{aligned} \mathbf{L}_1^{-1}\mathbf{L}_2^{-1}\cdots\mathbf{L}_{i+1}^{-1} &= (\mathbf{I} + \boldsymbol{\ell}_1\mathbf{e}_1^\mathsf{T} + \boldsymbol{\ell}_2\mathbf{e}_2^\mathsf{T} + \cdots + \boldsymbol{\ell}_i\mathbf{e}_i^\mathsf{T})\mathbf{L}_{i+1}^{-1} \\ &= (\mathbf{I} + \boldsymbol{\ell}_1\mathbf{e}_1^\mathsf{T} + \boldsymbol{\ell}_2\mathbf{e}_2^\mathsf{T} + \cdots + \boldsymbol{\ell}_i\mathbf{e}_i^\mathsf{T})(\mathbf{I} + \boldsymbol{\ell}_{i+1}\mathbf{e}_{i+1}^\mathsf{T}) \\ &= \mathbf{I} + \boldsymbol{\ell}_1\mathbf{e}_1^\mathsf{T} + \boldsymbol{\ell}_2\mathbf{e}_2^\mathsf{T} + \cdots + \boldsymbol{\ell}_i\mathbf{e}_i^\mathsf{T} + \boldsymbol{\ell}_{i+1}\mathbf{e}_{i+1}^\mathsf{T} + \sum_{j=1}^{i} \boldsymbol{\ell}_j \underbrace{\mathbf{e}_j^\mathsf{T}\boldsymbol{\ell}_{i+1}}_{=0}\mathbf{e}_{i+1}^\mathsf{T}. \end{aligned}$$

□

Wird im Verlauf des Gauß-Algorithmus ein Pivotelement $a_{i,i}^{(i)}$ Null, dann bricht das Verfahren in dieser Form zusammen. Sind hingegen alle Pivotelemente für $i = 1, 2, \ldots, n$ von Null verschieden, so haben wir das folgende Resultat bewiesen.

Satz 2.19 Falls kein Pivotelement Null wird, bestimmt der Gauß-Algorithmus neben der Lösung $\mathbf{x}$ von $\mathbf{Ax} = \mathbf{b}$ eine LR-Zerlegung $\mathbf{A} = \mathbf{LR}$ in eine linke untere und eine rechte obere Dreiecksmatrix. Die Matrix $\mathbf{L}$ ist dabei durch (2.3) gegeben.

Beispiel 2.20 Für die Matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 10 \end{bmatrix}$$

werden im Gauß-Algorithmus die folgenden Eleminationsschritte vollzogen:

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 10 \end{bmatrix} \begin{matrix} -2 \\ \hookleftarrow \\ \\ \end{matrix} \begin{matrix} -3 \\ \\ \hookleftarrow \end{matrix} \longrightarrow \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{bmatrix} \begin{matrix} \\ -2 \\ \hookleftarrow \end{matrix} \longrightarrow \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{bmatrix}.$$

Deshalb erhalten wir

$$\mathbf{R} = \begin{bmatrix} 1 & 4 & 7 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}.$$

△

Bemerkung 2.21 Bei der Realisierung der LR-Zerlegung am Computer überschreibt man die ursprünglichen Einträge $a_{i,j}^{(1)} = a_{i,j}$ der Matrix $\mathbf{A}$ mit den jeweils aktuellen Einträgen $a_{i,j}^{(i)}$. Die Matrix $\mathbf{L}$ lässt sich sukzessive in die nicht mehr benötigte untere Hälfte von $\mathbf{A}$ schreiben. Damit wird kein zusätzlicher Speicherplatz für die LR-Zerlegung gebraucht. Man spricht daher auch von einem *Auf-dem-Platz-Algorithmus*. △

Die Lösung des linearen Gleichungssystems $\mathbf{Ax} = \mathbf{b}$ wird mit Hilfe der LR-Zerlegung wie folgt berechnet:

① zerlege $\mathbf{A} = \mathbf{LR}$ mit dem Gauß-Algorithmus

② löse $\mathbf{Ax} = \mathbf{LRx} = \mathbf{b}$ in zwei Schritten:

- löse $\mathbf{Ly} = \mathbf{b}$ durch Vorwärtssubstitution
- löse $\mathbf{Rx} = \mathbf{y}$ durch Rückwärtssubstitution

Vorwärtssubstitution und Rückwartssubstitution sind in Algorithmus 2.22 zu finden.

Algorithmus 2.22 (Vorwärts- und Rückwärtssubstitution)

input: linke untere Dreiecksmatrix $\mathbf{L} = [\ell_{i,j}] \in \mathbb{R}^{n\times n}$, rechte obere Dreiecksmatrix $\mathbf{R} = [r_{i,j}] \in \mathbb{R}^{n\times n}$ und Vektor $\mathbf{b} = [b_i] \in \mathbb{R}^n$

output: Lösung $\mathbf{x} = [x_i] \in \mathbb{R}^n$ des linearen Gleichungssystems $\mathbf{LRx} = \mathbf{b}$

① für alle $i = 1, 2, \ldots, n$ berechne

$$y_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{i,j} y_j \right) \Big/ \ell_{i,i}$$

$$\underbrace{\begin{bmatrix} * & & \\ \vdots & \ddots & \\ * & \cdots & * \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}}_{\mathbf{b}}$$

② für alle $i = n, n-1, \ldots, 1$ berechne

$$x_i = \left(y_i - \sum_{j=i+1}^{n} r_{i,j} x_j \right) \Big/ r_{i,i}$$

$$\underbrace{\begin{bmatrix} * & \cdots & * \\ & \ddots & \vdots \\ & & * \end{bmatrix}}_{\mathbf{R}} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}}$$

Beispiel 2.23 (Fortsetzung von Beispiel 2.20) Für $\mathbf{b} = [1, 1, 1]^\intercal$ wollen wir das lineare Gleichungssystem $\mathbf{Ax} = \mathbf{b}$ lösen:

1. bestimme $\mathbf{y}$ mit $\mathbf{Ly} = \mathbf{b}$ durch Vorwärtssubstitution:

$$\left[\, \mathbf{L} \,\middle|\, \mathbf{b} \,\right] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 1 \end{array}\right] \quad \Longrightarrow \quad \begin{aligned} y_1 &= 1 \\ y_2 &= 1 - 2 = -1 \\ y_3 &= 1 - 3 + 2 = 0 \end{aligned}$$

2. bestimme $\mathbf{x}$ mit $\mathbf{Rx} = \mathbf{y}$ durch Rückwärtssubstitution:

$$\left[\begin{array}{c|c} \mathbf{R} & \mathbf{y} \end{array}\right] = \left[\begin{array}{ccc|c} 1 & 4 & 7 & 1 \\ 0 & -3 & -6 & -1 \\ 0 & 0 & 1 & 0 \end{array}\right] \quad \Longrightarrow \quad \begin{array}{l} x_3 = 0 \\ x_2 = (-1+0)/(-3) = 1/3 \\ x_1 = 1 - 4/3 - 0 = -1/3 \end{array}$$

△

Aufwand: Wir wollen nun den Aufwand zum Lösen eines linearen Gleichungssytems mit der LR-Zerlegung abschätzen. Dazu betrachten wir die beiden Teilschritte des obigen Verfahrens getrennt:

① Gemäß (2.2) werden im i-ten Teilschritt des Gauß-Algorithmus

$$\underbrace{(n-i)}_{\substack{\text{Berechnung} \\ \text{der } \{\tau_j^{(i)}\}_{j=i+1}^n}} + \underbrace{(n-i)}_{\substack{\text{Anzahl} \\ \text{der Zeilen}}} \underbrace{(n-i)}_{\substack{\text{Anzahl} \\ \text{der Spalten}}} = (n-i)^2 + (n-i)$$

Multiplikationen durchgeführt. Für die Berechnung der LR-Zerlegung werden daher insgesamt

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

Multiplikationen verwendet.

② Sowohl die Vorwärtssubstitution als auch die Rückwärtssubstitution haben den gleichen Aufwand. Da jeder Eintrag der zugrundeliegenden Dreiecksmatrizen jeweils einmal multipliziert wird, werden hier

$$2\sum_{i=1}^{n} i = n(n+1) = \mathcal{O}(n^2)$$

Multiplikationen ausgeführt.

Demnach werden also insgesamt zum Lösen eines linearen Gleichungssystems mit Hilfe der LR-Zerlegung $n^3/3 + \mathcal{O}(n^2)$ Multiplikationen (und, wie man leicht nachrechnet, nochmals ebensoviele Additionen) benötigt. Der Speicherplatzbedarf ist dabei allerdings nur von der Ordnung $\mathcal{O}(n^2)$.

Bemerkung 2.24 Eine weitere Möglichkeit, das lineare Gleichungssystem $\mathbf{Ax} = \mathbf{b}$ zu lösen, bietet bekanntlich die Cramersche Regel. Danach lautet die i-te Komponente x_i der Lösung

$$x_i = \frac{\det \mathbf{A}_i}{\det \mathbf{A}},$$

wobei hier $\mathbf{A}_i \in \mathbb{R}^{n\times n}$ diejenige Matrix ist, die aus $\mathbf{A}$ entsteht, wenn man die i-te Spalte durch $\mathbf{b}$ ersetzt. Berechnet man die Determinante nach dem Laplaceschen Entwicklungssatz, so benötigt man i.a. $n!$ Operationen. Bei Verwendung eines Rechners mit 10^8 Gleitkommaoperationen pro Sekunde (100 Megaflops) ergäben sich dann die folgenden Rechenzeiten:

n	10	12	14	16	18	20
Rechenzeit	0.4 s	1 min	3.6 h	41 Tage	38 Jahre	16 000 Jahre

△

2.4 Block-Gauß-Elimination

Die Verwendung des Gauß-Algorithmus zur Bestimmung einer LR-Zerlegung lässt sich auch auf *Blockmatrizen* anwenden. Zu einer gegebenen Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ betrachten wir die Blockpartitionierung

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix} \tag{2.4}$$

mit $\mathbf{A}_{1,1} \in \mathbb{R}^{p\times p}$ und $\mathbf{A}_{2,2} \in \mathbb{R}^{(n-p)\times(n-p)}$, wobei $1 \leq p < n$. Ist $\mathbf{A}_{1,1}$ nichtsingulär, so kann man das lineare Gleichungssystem

$$\mathbf{A}\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} \tag{2.5}$$

vermittels Block-Gauß-Elimination lösen:

$$\left[\begin{array}{cc|c} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \mathbf{b} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \mathbf{c} \end{array}\right] \begin{array}{c} -\mathbf{A}_{2,1}\mathbf{A}_{1,1}^{-1} \\ \hookleftarrow \end{array} \longrightarrow \left[\begin{array}{cc|c} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \mathbf{b} \\ \mathbf{0} & \mathbf{A}_{2,2} - \mathbf{A}_{2,1}\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,2} & \mathbf{c} - \mathbf{A}_{2,1}\mathbf{A}_{1,1}^{-1}\mathbf{b} \end{array}\right]$$

Definition 2.25 Die Matrix

$$\mathbf{S} := \mathbf{A}_{2,2} - \mathbf{A}_{2,1}\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,2} \in \mathbb{R}^{(n-p)\times(n-p)} \tag{2.6}$$

heißt **Schur-Komplement** der Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ bezüglich $\mathbf{A}_{1,1} \in \mathbb{R}^{p\times p}$.

Für die Lösung von (2.5) folgt nun

$$\begin{aligned} \mathbf{y} &= \mathbf{S}^{-1}\big(\mathbf{c} - \mathbf{A}_{2,1}\mathbf{A}_{1,1}^{-1}\mathbf{b}\big), \\ \mathbf{x} &= \mathbf{A}_{1,1}^{-1}(\mathbf{b} - \mathbf{A}_{1,2}\mathbf{y}). \end{aligned}$$

Wir wollen dies allerdings an dieser Stelle nicht weiter vertiefen. Vielmehr bemerken wir, dass die Block-Gauß-Elimination auf eine Block-LR-Zerlegung führt. Es gilt

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{L}_{2,1} & \mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{0} & \mathbf{S} \end{bmatrix} \quad \text{mit} \quad \mathbf{L}_{2,1} = \mathbf{A}_{2,1}\mathbf{A}_{1,1}^{-1}.$$

Wenden wir diese Block-LR-Zerlegung für $p = 1$ rekursiv auf das Schur-Komplement an, dann ergibt sich die *LR-Zerlegung mit Rang-1-Updates*:

Algorithmus 2.26 (LR-Zerlegung mit Rang-1-Updates)
input: reguläre Matrix $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{n\times n}$
output: LR-Zerlegung $\mathbf{A} = \mathbf{LR} = [\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_n][\mathbf{r}_1, \ldots, \mathbf{r}_n]^\intercal$

① setze $\mathbf{A}_1 = \big[a_{i,j}^{(1)}\big] = \mathbf{A}$

② für alle $i = 1, 2, \ldots, n$ bilde

$$\boldsymbol{\ell}_i = \frac{1}{a_{i,i}^{(i)}} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_{i,i}^{(i)} \\ a_{i+1,i}^{(i)} \\ \vdots \\ a_{n,i}^{(i)} \end{bmatrix} \in \mathbb{R}^n \quad \text{und} \quad \mathbf{r}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_{i,i}^{(i)} \\ a_{i,i+1}^{(i)} \\ \vdots \\ a_{i,n}^{(i)} \end{bmatrix} \in \mathbb{R}^n$$

und berechne $\mathbf{A}_{i+1} = \left[a_{i,j}^{(i+1)}\right] = \mathbf{A}_i - \boldsymbol{\ell}_i \mathbf{r}_i^\mathsf{T}$

Dieser Algorithmus setzt im i-ten Schritt die i-te Zeile und i-te Spalte von $\mathbf{A}_i$ auf Null. Insbesondere gilt $\mathbf{A}_{n+1} = \mathbf{0}$ und somit durch Auflösen der Rekursion

$$\mathbf{A} = \boldsymbol{\ell}_1 \mathbf{r}_1^\mathsf{T} + \boldsymbol{\ell}_2 \mathbf{r}_2^\mathsf{T} + \cdots + \boldsymbol{\ell}_n \mathbf{r}_n^\mathsf{T} = [\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_n][\mathbf{r}_1, \ldots, \mathbf{r}_n]^\mathsf{T}.$$

Algorithmus 2.26 hat gegenüber einer naiven Implementierung des Gauß-Algorithmus den Vorteil, dass er vektorisiert und damit im allgemeinen sehr effizient ist.

2.5 LR-Zerlegung mit Pivotisierung

Wir wollen nun die Matrixformulierung des Gauß-Algorithmus mit Spaltenpivotsuche herleiten. Hier wird im i-ten Schritt der betragsgrößte Eintrag $|a_{k,i}^{(i)}| = \max_{i \le \ell \le n} |a_{\ell,i}^{(i)}|$ aus den zu eleminierenden Einträgen in der i-ten Spalte als Pivotelement ausgewählt. Das Vertauschen der i-ten und der k-ten Zeile der Matrix $\mathbf{A}_i$ kann durch die zugehörige Permutationsmatrix

$$\mathbf{P}_i = \begin{array}{l} \left[\begin{array}{ccccccc} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ \hline & & 0 & & 1 & & \\ \hline & & & 1 & & & \\ & & & & \ddots & & \\ & & & & & 1 & \\ \hline & & 1 & & 0 & & \\ \hline & & & & & 1 & \\ & & & & & & \ddots \\ & & & & & & 1 \end{array}\right] \\ \begin{array}{ccccccc} & & \uparrow & & & \uparrow & \\ & & i\text{-te Spalte} & & & k\text{-te Spalte} & \end{array} \end{array} \begin{array}{l} \\ \\ \\ \leftarrow i\text{-te Zeile} \\ \\ \\ \\ \leftarrow k\text{-te Zeile} \\ \\ \\ \\ \end{array} \tag{2.7}$$

beschrieben werden. Es gelten nämlich die folgenden Rechenregeln:

Rechenregeln:

1. Multiplikation mit $\mathbf{P}_i$ von *links* $\longrightarrow$ vertausche *Zeilen* i und k
2. Multiplikation mit $\mathbf{P}_i$ von *rechts* $\longrightarrow$ vertausche *Spalten* i und k
3. Insbesondere gilt $\mathbf{P}_i^2 = \mathbf{I}$.

Folglich wird aus der Matrix $\mathbf{A}_i$ durch Vertauschung die Matrix $\mathbf{P}_i\mathbf{A}_i$. Diese Matrix wird nun im Eliminationsschritt weiter reduziert. Der vollständige i-te Teilschritt des Gauß-Algorithmus mit Spaltenpivotsuche transformiert $\mathbf{A}_i$ also wie folgt:

$$\mathbf{A}_i \mapsto \mathbf{A}_{i+1} = \mathbf{L}_i\mathbf{P}_i\mathbf{A}_i. \tag{2.8}$$

Wir benötigen folgendes Lemma:

Lemma 2.27 Sei $j < i$ und $\mathbf{P}_i$ durch (2.7) gegeben. Dann ist $\mathbf{P}_i\mathbf{L}_j = \widetilde{\mathbf{L}}_j\mathbf{P}_i$, wobei $\widetilde{\mathbf{L}}_j$ wieder die gleiche Form hat wie $\mathbf{L}_j$, außer dass $\tau_k^{(j)}$ und $\tau_i^{(j)}$ vertauscht sind.

Beweis. Wegen $\mathbf{P}_i^2 = \mathbf{I}$ gilt

$$\mathbf{P}_i\mathbf{L}_j = \mathbf{P}_i\mathbf{L}_j\mathbf{P}_i^2 = (\mathbf{P}_i\mathbf{L}_j\mathbf{P}_i)\mathbf{P}_i,$$

dies bedeutet

$$\widetilde{\mathbf{L}}_j = \mathbf{P}_i\mathbf{L}_j\mathbf{P}_i.$$

Da die Multiplikation mit $\mathbf{P}_i$ von links die Zeilen i und k vertauscht, folgt

$$\mathbf{P}_i\mathbf{L}_j = \begin{array}{r} \\ \\ \\ \\ \\ i\text{-te Zeile} \rightarrow \\ \\ \\ \\ k\text{-te Zeile} \rightarrow \\ \\ \\ \\ \end{array} \left[\begin{array}{cccc|c|ccc|c|ccc}
1 &&&&&&&&&&& \\
& \ddots &&&&&&&&&& \\
&& 1 &&&&&&&&& \\
&& -\tau_{j+1}^{(j)} & \ddots &&&&&&&& \\
&& \vdots && 1 &&&&&&& \\
\hline
&& -\tau_k^{(j)} &&& 0 &&&& 1 && \\
\hline
&&&&&& 1 &&&&& \\
&& \vdots &&&&& \ddots &&&& \\
&&&&&&&& 1 &&& \\
\hline
&& -\tau_i^{(j)} &&& 1 &&&& 0 && \\
\hline
&&&&&&&&&& 1 & \\
&& \vdots &&&&&&&&& \ddots \\
&& -\tau_n^{(j)} &&&&&&&&& 1
\end{array}\right].$$

$\uparrow$ i-te Spalte $\uparrow$ k-te Spalte

Schließlich vertauscht die Multiplikation mit $\mathbf{P}_i$ von rechts noch die Spalten i und k:

$$
\mathbf{P}_i\mathbf{L}_j\mathbf{P}_i = \begin{array}{r}
\\ \\ \\ \\ \\ i\text{-te Zeile} \rightarrow \\ \\ \\ \\ k\text{-te Zeile} \rightarrow \\ \\ \\ \\
\end{array}
\left[\begin{array}{ccc|c|ccc|c|ccc}
1 & & & & & & & & & & \\
 & \ddots & & & & & & & & & \\
 & & 1 & & & & & & & & \\
 & & -\tau_{j+1}^{(j)} & \ddots & & & & & & & \\
 & & \vdots & & 1 & & & & & & \\
\hline
 & & -\tau_k^{(j)} & & 1 & & & 0 & & & \\
\hline
 & & & & & 1 & & & & & \\
 & & \vdots & & & & \ddots & & & & \\
 & & & & & & & 1 & & & \\
\hline
 & & -\tau_i^{(j)} & & 0 & & & 1 & & & \\
\hline
 & & & & & & & & 1 & & \\
 & & \vdots & & & & & & & \ddots & \\
 & & -\tau_n^{(j)} & & & & & & & & 1
\end{array}\right].
$$

$$
\begin{array}{cc}
\uparrow & \uparrow \\
i\text{-te Spalte} & k\text{-te Spalte}
\end{array}
$$

Damit ist $\widetilde{\mathbf{L}}_j$ von der behaupteten Form. □

Mit Hilfe von Lemma 2.27 können wir den folgenden Satz für den Gauß-Algorithmus mit Spaltenpivotsuche beweisen:

Satz 2.28 Ist $\mathbf{A}$ nichtsingulär, dann bestimmt der Gauß-Algorithmus mit Spaltenpivotsuche eine Zerlegung der Matrix $\mathbf{PA} = \widetilde{\mathbf{L}}\mathbf{R}$, wobei $\mathbf{R}$ wie zuvor die rechte obere Dreiecksmatrix $\mathbf{A}_n$, $\mathbf{P} = \mathbf{P}_{n-1}\mathbf{P}_{n-2}\cdots\mathbf{P}_1$ eine Permutationsmatrix und

$$\widetilde{\mathbf{L}} = \widetilde{\mathbf{L}}_1^{-1}\widetilde{\mathbf{L}}_2^{-1}\cdots\widetilde{\mathbf{L}}_{n-1}^{-1}$$

eine linke untere Dreiecksmatrix ist mit

$$
\begin{aligned}
\widetilde{\mathbf{L}}_{n-1} &= \mathbf{L}_{n-1}, \\
\widetilde{\mathbf{L}}_{n-2} &= \mathbf{P}_{n-1}\mathbf{L}_{n-2}\mathbf{P}_{n-1}, \\
\widetilde{\mathbf{L}}_{n-3} &= \mathbf{P}_{n-1}\mathbf{P}_{n-2}\mathbf{L}_{n-3}\mathbf{P}_{n-2}\mathbf{P}_{n-1}, \\
&\vdots \\
\widetilde{\mathbf{L}}_1 &= \mathbf{P}_{n-1}\mathbf{P}_{n-2}\cdots\mathbf{P}_2\mathbf{L}_1\mathbf{P}_2\cdots\mathbf{P}_{n-2}\mathbf{P}_{n-1}.
\end{aligned}
$$

Beweis. Nehmen wir zunächst an, dass der Gauß-Algorithmus mit Spaltenpivotsuche nicht zusammenbricht. Dann ergibt sich aus (2.8) durch sukzessive Anwendung von Lem-

ma 2.27, dass

$$\begin{aligned}\mathbf{R} = \mathbf{A}_n &= \mathbf{L}_{n-1}\mathbf{P}_{n-1}\mathbf{A}_{n-1}\\ &= \widetilde{\mathbf{L}}_{n-1}\mathbf{P}_{n-1}\mathbf{L}_{n-2}\mathbf{P}_{n-2}\mathbf{A}_{n-2}\\ &= \widetilde{\mathbf{L}}_{n-1}\widetilde{\mathbf{L}}_{n-2}\mathbf{P}_{n-1}\mathbf{P}_{n-2}\mathbf{L}_{n-3}\mathbf{P}_{n-3}\mathbf{A}_{n-3}\\ &\;\;\vdots\\ &= \widetilde{\mathbf{L}}_{n-1}\widetilde{\mathbf{L}}_{n-2}\cdots\widetilde{\mathbf{L}}_1\mathbf{P}_{n-1}\mathbf{P}_{n-2}\cdots\mathbf{P}_1\mathbf{A}.\end{aligned}$$

Zu klären bleibt schließlich der Punkt, dass der Gauß-Algorithmus mit Spaltenpivotsuche nicht abbricht, also dass alle Pivotelemente nach der Spaltenpivotsuche von Null verschieden sind. Wäre das Pivotelement im i-ten Teilschritt tatsächlich Null, dann gälte zwangsläufig

$$\mathbf{A}_i = \begin{array}{c}\left\lceil\mathbf{B}\right\rfloor\end{array}\left[\begin{array}{ccc|ccc} \star & \cdots & \star & & & \\ & \ddots & \vdots & & \star & \\ \mathbf{0} & & \star & & & \\ \hline & & & 0 & \star & \cdots & \star \\ & \mathbf{0} & & \vdots & \vdots & & \vdots \\ & & & 0 & \star & \cdots & \star \end{array}\right].$$

Daraus folgt jedoch

$$\det\mathbf{A}_i = \det\mathbf{B}\cdot\det\begin{bmatrix}0 & \star & \cdots & \star\\ \vdots & \vdots & & \vdots\\ 0 & \star & \cdots & \star\end{bmatrix} = 0$$

und weiter

$$0 = \det\mathbf{A}_i = \det(\mathbf{L}_{i-1}\mathbf{P}_{i-1}\cdots\mathbf{L}_1\mathbf{P}_1\mathbf{A}) = \prod_{j=1}^{i-1}\underbrace{\det\mathbf{L}_j}_{=1}\cdot\prod_{j=1}^{i-1}\underbrace{\det\mathbf{P}_j}_{=\pm1}\cdot\det\mathbf{A}.$$

Dies impliziert $\det\mathbf{A} = 0$ im Widerspruch zur Voraussetzung. □

Beispiel 2.29

$$\mathbf{A} = \begin{bmatrix}1 & 1 & 0 & 2\\ 1/2 & 1/2 & 2 & -1\\ -1 & 0 & -1/8 & -5\\ 1 & -7 & 9 & 10\end{bmatrix}\begin{matrix}-1/2\\ \hookleftarrow\end{matrix}\begin{matrix}+1\\ \hookleftarrow\end{matrix}\begin{matrix}-1\\ \hookleftarrow\end{matrix}\longrightarrow\begin{bmatrix}1 & 1 & 0 & 2\\ 0 & 0 & 2 & -2\\ 0 & 1 & -1/8 & -3\\ 0 & -8 & 9 & 8\end{bmatrix}\begin{matrix}\\ \hookleftarrow\\ \\ \hookleftarrow\end{matrix}$$

$$\xrightarrow{\mathbf{P}_2}\begin{bmatrix}1 & 1 & 0 & 2\\ 0 & -8 & 9 & 8\\ 0 & 1 & -1/8 & -3\\ 0 & 0 & 2 & -2\end{bmatrix}\begin{matrix}\\ +1/8\\ \hookleftarrow\\ \\ \end{matrix}\longrightarrow\begin{bmatrix}1 & 1 & 0 & 2\\ 0 & -8 & 9 & 8\\ 0 & 0 & 1 & -2\\ 0 & 0 & 2 & -2\end{bmatrix}\begin{matrix}\\ \\ \hookleftarrow\\ \hookleftarrow\end{matrix}$$

$$\xrightarrow{\mathbf{P}_3}\begin{bmatrix}1 & 1 & 0 & 2\\ 0 & -8 & 9 & 8\\ 0 & 0 & 2 & -2\\ 0 & 0 & 1 & -2\end{bmatrix}\begin{matrix}\\ \\ -1/2\\ \hookleftarrow\end{matrix}\longrightarrow\begin{bmatrix}1 & 1 & 0 & 2\\ 0 & -8 & 9 & 8\\ 0 & 0 & 2 & -2\\ 0 & 0 & 0 & -1\end{bmatrix}.$$

Damit ergibt sich (beachte: $\mathbf{P}_1 = \mathbf{I}$)

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & 0 & 2 \\ 0 & -8 & 9 & 8 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \qquad \widetilde{\mathbf{L}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1/2 & 0 & 1 & 0 \\ -1 & -1/8 & 1/2 & 1 \end{bmatrix},$$

$$\mathbf{PA} = \begin{bmatrix} 1 & 1 & 0 & 2 \\ 1 & -7 & 9 & 10 \\ 1/2 & 1/2 & 2 & -1 \\ -1 & 0 & -1/8 & -5 \end{bmatrix}.$$

△

Faustregel: Um auf $\widetilde{\mathbf{L}}$ zu kommen, erstellt man zunächst eine Matrix $\mathbf{L}$ wie gewohnt, und führt dann in jeder Spalte *alle* Vertauschungen (in der Reihenfolge $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{n-1}$) durch, bei denen nur Elemente unterhalb der Diagonalen betroffen sind.

Totale Pivotisierung: Zum Schluss wollen wir noch die LR-Zerlegung mit totaler Pivotisierung betrachten. Hier wird im i-ten Teilschritt dasjenige Element $a_{k,m}^{(i)}$ $(i \le k, m \le n)$ als Pivotelement gewählt, das in der gesamten verbliebenen Restmatrix betragsmäßig am größten ist:

$$|a_{k,m}^{(i)}| = \max_{i \le \ell, j \le n} |a_{\ell,j}^{(i)}|.$$

Da man hierzu Zeilen und Spalten tauschen muss, benötigt man formal zwei Permutationsmatrizen $\mathbf{P}_i$ und $\mathbf{\Pi}_i$:

$$\mathbf{A}_i \mapsto \mathbf{A}_{i+1} = \mathbf{L}_i \mathbf{P}_i \mathbf{A}_i \mathbf{\Pi}_i.$$

Man erhält so schließlich eine LR-Zerlegung der Matrix $\mathbf{PA\Pi}$ mit $\mathbf{\Pi} = \mathbf{\Pi}_1 \mathbf{\Pi}_2 \cdots \mathbf{\Pi}_{n-1}$.

Wird die Totalpivotsuche bei der Lösung eines linearen Gleichungssystems eingesetzt, dann entsprechen Spaltenvertauschungen Permutationen der Lösung $\mathbf{x}$. Der Ergebnisvektor ist also nicht mehr in der richtigen Reihenfolge.

Die totale Pivotisierung ist stabil, wird aber in der Praxis nur selten eingesetzt, da die Suche nach dem betragsgrößten Element im i-ten Schritt einem Aufwand $(n - i + 1)^2$ entspricht. Der Gesamtaufwand

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

ist folglich nicht mehr vernachlässigbar gegenüber der eigentlichen Rechnung. Speziell besitzt die *QR-Zerlegung* den selben Aufwand wie die LR-Zerlegung mit totaler Pivotisierung und wird dann generell bevorzugt.

2.6 Cholesky-Zerlegung

Im Fall symmetrischer Matrizen, welche auch positiv definit sind, können wir eine Dreieckszerlegung durch ein Verfahren gewinnen, das nur halb so teuer ist wie die LR-Zerlegung.

Definition 2.30 Eine symmetrische Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ heißt **positiv definit**, falls gilt

$$\mathbf{x}^\intercal \mathbf{A}\mathbf{x} > 0 \text{ für alle } \mathbf{x} \neq \mathbf{0}. \tag{2.9}$$

Bemerkung 2.31 Eine symmetrische und positiv definite Matrix ist regulär, da ihr Kern aufgrund von (2.9) nur trivial sein kann. Speziell ist jede symmetrische Matrix diagonalisierbar. Dies bedeutet, es gilt $\mathbf{A} = \mathbf{T}\mathbf{D}\mathbf{T}^\intercal$ mit einer orthogonalen Matrix $\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_n]$ und einer Diagonalmatrix $\mathbf{D} = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$. Dabei sind die $\{(\lambda_i, \mathbf{t}_i)\}_{i=1}^n$ die Eigenpaare von $\mathbf{A}$. Setzen wir in (2.9) $\mathbf{x} = \mathbf{t}_i$ ein, so folgt wegen $\mathbf{t}_i^\intercal \mathbf{t}_j = \delta_{i,j}$ sofort

$$\mathbf{t}_i^\intercal \mathbf{A}\mathbf{t}_i = \mathbf{t}_i^\intercal \mathbf{T}\mathbf{D}\mathbf{T}^\intercal \mathbf{t}_i = \lambda_i > 0.$$

Folglich ist eine symmetrische Matrix $\mathbf{A}$ genau dann positiv definit, wenn alle ihre Eigenwerte positiv sind. △

Wir partitionieren die Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ wieder gemäß

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix}$$

mit $\mathbf{A}_{1,1} \in \mathbb{R}^{p\times p}$ und $\mathbf{A}_{2,2} \in \mathbb{R}^{(n-p)\times(n-p)}$. Aufgrund der Symmetrie

$$\begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix} = \mathbf{A} = \mathbf{A}^\intercal = \begin{bmatrix} \mathbf{A}_{1,1}^\intercal & \mathbf{A}_{2,1}^\intercal \\ \mathbf{A}_{1,2}^\intercal & \mathbf{A}_{2,2}^\intercal \end{bmatrix}$$

gelten dann für die einzelnen Blockmatrizen die Beziehungen

$$\mathbf{A}_{1,1} = \mathbf{A}_{1,1}^\intercal, \quad \mathbf{A}_{2,2} = \mathbf{A}_{2,2}^\intercal, \quad \mathbf{A}_{1,2} = \mathbf{A}_{2,1}^\intercal. \tag{2.10}$$

Lemma 2.32 Sei $\mathbf{A} \in \mathbb{R}^{n\times n}$ symmetrisch und positiv definit. Dann ist das Schur-Komplement $\mathbf{S} = \mathbf{A}_{2,2} - \mathbf{A}_{2,1}\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,2} \in \mathbb{R}^{(n-p)\times(n-p)}$ wohldefiniert und sowohl $\mathbf{A}_{1,1}$ als auch $\mathbf{S}$ sind symmetrisch und positiv definit.

Beweis. Sei $\left[\begin{smallmatrix}\mathbf{x}\\ \mathbf{y}\end{smallmatrix}\right]$ entsprechend zu $\mathbf{A}$ partitioniert. Es gilt

$$0 \leq \begin{bmatrix}\mathbf{x}\\ \mathbf{0}\end{bmatrix}^\intercal \mathbf{A} \begin{bmatrix}\mathbf{x}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{x}\\ \mathbf{0}\end{bmatrix}^\intercal \begin{bmatrix}\mathbf{A}_{1,1}\mathbf{x}\\ \mathbf{A}_{2,1}\mathbf{x}\end{bmatrix} = \mathbf{x}^\intercal \mathbf{A}_{1,1}\mathbf{x},$$

wobei sich Gleichheit nur für $\mathbf{x} = \mathbf{0}$ ergibt. Daher ist auch $\mathbf{A}_{1,1}$ symmetrisch und positiv definit, weshalb nach Bemerkung 2.31 $\mathbf{A}_{1,1}^{-1}$ existiert. $\mathbf{S}$ ist somit wohldefiniert und

$$\mathbf{S}^\intercal = \mathbf{A}_{2,2}^\intercal - \mathbf{A}_{1,2}^\intercal \mathbf{A}_{1,1}^{-\intercal} \mathbf{A}_{2,1}^\intercal \overset{(2.10)}{=} \mathbf{A}_{2,2} - \mathbf{A}_{2,1}\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,2} = \mathbf{S}.$$

Schließlich betrachten wir $\left[\begin{smallmatrix}\mathbf{x}\\ \mathbf{y}\end{smallmatrix}\right]$ mit $\mathbf{x} = -\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,2}\mathbf{y}$:

$$\begin{aligned} 0 \leq \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}^\intercal \mathbf{A} \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} &= \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}^\intercal \begin{bmatrix}\mathbf{A}_{1,1}\mathbf{x} + \mathbf{A}_{1,2}\mathbf{y}\\ \mathbf{A}_{2,1}\mathbf{x} + \mathbf{A}_{2,2}\mathbf{y}\end{bmatrix} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}^\intercal \begin{bmatrix} -\mathbf{A}_{1,2}\mathbf{y} + \mathbf{A}_{1,2}\mathbf{y}\\ -\mathbf{A}_{2,1}\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,2}\mathbf{y} + \mathbf{A}_{2,2}\mathbf{y}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}^\intercal \begin{bmatrix}\mathbf{0}\\ \mathbf{S}\mathbf{y}\end{bmatrix} = \mathbf{y}^\intercal \mathbf{S}\mathbf{y}. \end{aligned}$$

Da Gleichheit nur im Fall $\mathbf{x} = \mathbf{0}$ und $\mathbf{y} = \mathbf{0}$ gilt, ist $\mathbf{S}$ ebenfalls positiv definit. □

Definition 2.33 Eine Zerlegung $\mathbf{A} = \mathbf{L}\mathbf{L}^\intercal$ mit unterer Dreiecksmatrix $\mathbf{L}$ mit positiven Diagonaleinträgen heißt **Cholesky-Zerlegung** von $\mathbf{A}$.

Proposition 2.34 Besitzt $\mathbf{A}$ eine Cholesky-Zerlegung, dann ist $\mathbf{A}$ symmetrisch und positiv definit.

Beweis. Aus $\mathbf{A} = \mathbf{L}\mathbf{L}^\intercal$ folgt

$$\mathbf{A}^\intercal = \left(\mathbf{L}^\intercal\right)^\intercal \mathbf{L}^\intercal = \mathbf{L}\mathbf{L}^\intercal = \mathbf{A},$$

das heißt, $\mathbf{A}$ ist symmetrisch. Wegen

$$\mathbf{x}^\intercal \mathbf{A}\mathbf{x} = \mathbf{x}^\intercal \mathbf{L}\mathbf{L}^\intercal \mathbf{x} = \left(\mathbf{L}^\intercal \mathbf{x}\right)^\intercal \mathbf{L}^\intercal \mathbf{x} = \|\mathbf{L}^\intercal \mathbf{x}\|_2^2 \geq 0,$$

ist $\mathbf{A}$ auch positiv semidefinit. Da $\mathbf{L}$ nach Voraussetzung nichtsingulär ist, impliziert $\mathbf{L}^\intercal \mathbf{x} = \mathbf{0}$ auch $\mathbf{x} = \mathbf{0}$. Damit ist $\mathbf{A}$ sogar definit. □

Satz 2.35 Ist $\mathbf{A}$ symmetrisch und positiv definit, dann existiert eine Cholesky-Zerlegung von $\mathbf{A}$.

Beweis. Wir beweisen diesen Satz mit Hilfe von vollständiger Induktion über n. Im Fall $n = 1$ ist $\mathbf{A} = [a_{1,1}]$ und, weil $\mathbf{A}$ positiv definit ist, gilt $a_{1,1} > 0$. Wegen

$$\mathbf{A} = [a_{1,1}] \overset{!}{=} [\ell_{1,1}] \cdot [\ell_{1,1}] = \mathbf{L} \cdot \mathbf{L}^\intercal$$

folgt damit $\ell_{1,1} = \sqrt{a_{1,1}} > 0$.
Wir wollen nun annehmen, dass die Aussage für $n-1$ bereits bewiesen ist und betrachten

$$\mathbf{A} = \left[\begin{array}{c|c} a_{1,1} & \mathbf{A}_{1,2} \\ \hline \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{array}\right] \in \mathbb{R}^{n \times n}$$

mit $\mathbf{A}_{2,1} = \mathbf{A}_{1,2}^\intercal$ und das Schur-Komplement

$$\mathbf{S} = \mathbf{A}_{2,2} - \frac{1}{a_{1,1}} \mathbf{A}_{2,1} \mathbf{A}_{1,2} \in \mathbb{R}^{(n-1)\times(n-1)} \tag{2.11}$$

von $\mathbf{A}$ bezüglich $a_{1,1}$. Nach Lemma 2.32 ist $a_{1,1} > 0$ und $\mathbf{S}$ symmetrisch und positiv definit. Also ist $\ell_{1,1} = \sqrt{a_{1,1}} > 0$ und aufgrund der Induktionsannahme besitzt $\mathbf{S}$ eine Cholesky-Zerlegung

$$\mathbf{S} = \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^\intercal.$$

Definiere damit

$$\mathbf{L} = \left[\begin{array}{c|c} \ell_{1,1} & \mathbf{0} \\ \hline \frac{1}{\ell_{1,1}} \mathbf{A}_{2,1} & \widetilde{\mathbf{L}} \end{array}\right], \qquad \mathbf{L}^\intercal = \left[\begin{array}{c|c} \ell_{1,1} & \frac{1}{\ell_{1,1}} \mathbf{A}_{1,2} \\ \hline \mathbf{0} & \widetilde{\mathbf{L}}^\intercal \end{array}\right].$$

Es ergibt sich

$$\mathbf{L}\mathbf{L}^\intercal = \left[\begin{array}{c|c} \ell_{1,1} & \mathbf{0} \\ \hline \frac{1}{\ell_{1,1}}\mathbf{A}_{2,1} & \widetilde{\mathbf{L}} \end{array}\right] \cdot \left[\begin{array}{c|c} \ell_{1,1} & \frac{1}{\ell_{1,1}}\mathbf{A}_{1,2} \\ \hline \mathbf{0} & \widetilde{\mathbf{L}}^\intercal \end{array}\right] = \left[\begin{array}{c|c} a_{1,1} & \mathbf{A}_{1,2} \\ \hline \mathbf{A}_{2,1} & \frac{1}{a_{1,1}}\mathbf{A}_{2,1}\mathbf{A}_{1,2} + \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^\intercal \end{array}\right].$$

Wegen

$$\frac{1}{a_{1,1}}\mathbf{A}_{2,1}\mathbf{A}_{1,2} + \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^\intercal = \frac{1}{a_{1,1}}\mathbf{A}_{2,1}\mathbf{A}_{1,2} + \mathbf{S} \overset{(2.11)}{=} \mathbf{A}_{2,2},$$

folgt hieraus der Induktionsschritt $n-1 \mapsto n$:

$$\mathbf{L}\mathbf{L}^\intercal = \left[\begin{array}{c|c} a_{1,1} & \mathbf{A}_{1,2} \\ \hline \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{array}\right] = \mathbf{A}.$$

□

Bemerkung 2.36 Durch Kombination von Proposition 2.34 und Satz 2.35 erhalten wir die Aussage, dass eine Cholesky-Zerlegung genau dann existiert, falls $\mathbf{A}$ symmetrisch und positiv definit ist. △

Die Berechnung von $\mathbf{L}$ ergibt sich durch Koeffizientenvergleich: Aus

$$\underbrace{\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} \ell_{1,1} & & & \mathbf{0} \\ \ell_{2,1} & \ell_{2,2} & & \\ \vdots & \vdots & \ddots & \\ \ell_{n,1} & \ell_{n,2} & \cdots & \ell_{n,n} \end{bmatrix}}_{\mathbf{L}} \cdot \underbrace{\begin{bmatrix} \ell_{1,1} & \ell_{2,1} & \cdots & \ell_{n,1} \\ & \ell_{2,2} & & \ell_{n,2} \\ & & \ddots & \vdots \\ \mathbf{0} & & & \ell_{n,n} \end{bmatrix}}_{\mathbf{L}^\intercal}$$

folgt

$$\begin{aligned}
a_{1,1} &= \ell_{1,1}^2 && \rightsquigarrow \ell_{1,1} = \sqrt{a_{1,1}} > 0 \\
a_{2,1} &= \ell_{2,1}\ell_{1,1} && \rightsquigarrow \ell_{2,1} = a_{2,1}/\ell_{1,1} \\
a_{3,1} &= \ell_{3,1}\ell_{1,1} && \rightsquigarrow \ell_{3,1} = a_{3,1}/\ell_{1,1} \\
&\vdots && \vdots \\
a_{n,1} &= \ell_{n,1}\ell_{1,1} && \rightsquigarrow \ell_{n,1} = a_{n1}/\ell_{1,1} \\
a_{2,2} &= \ell_{2,1}^2 + \ell_{2,2}^2 && \rightsquigarrow \ell_{2,2} = \sqrt{a_{2,2} - \ell_{2,1}^2} > 0 \\
a_{3,2} &= \ell_{3,1}\ell_{2,1} + \ell_{3,2}\ell_{2,2} && \rightsquigarrow \ell_{3,2} = (a_{3,2} - \ell_{3,1}\ell_{2,1})/\ell_{2,2} \\
&\vdots && \vdots \\
a_{n,2} &= \ell_{n,1}\ell_{2,1} + \ell_{n,2}\ell_{2,2} && \rightsquigarrow \ell_{n,2} = (a_{n,2} - \ell_{n,1}\ell_{2,1})/\ell_{2,2}
\end{aligned}$$

und allgemein

$$\begin{aligned}
\ell_{j,j} &= \sqrt{a_{j,j} - \sum_{k=1}^{j-1} \ell_{j,k}^2} > 0, \qquad 1 \le j \le n, \\
\ell_{i,j} &= \frac{1}{\ell_{j,j}}\left(a_{i,j} - \sum_{k=1}^{j-1} \ell_{i,k}\ell_{j,k}\right), \qquad j < i \le n.
\end{aligned} \tag{2.12}$$

Die Berechenbarkeit der Cholesky-Zerlegung von $\mathbf{A}$ vermittels (2.12) ist durch den Existenzbeweis (Satz 2.35) gewährleistet, das heißt, es gilt $\ell_{i,i} \neq 0$ für alle $i = 1, 2, \ldots, n$. Insbesondere ergibt sich aus (2.12) auch sofort die folgende Aussage:

Korollar 2.37 Die Cholesky-Zerlegung von einer symmetrischen und positiv definiten Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ ist eindeutig.

Aufwand: Für jeden festen Index j sind zur Berechnung der Koeffizienten $\ell_{i,j}$, $j \leq i \leq n$, der Cholesky-Zerlegung gemäß den Formeln (2.12) j Multiplikationen beziehungsweise Wurzeln auszuführen. Die Aufsummation über alle $1 \leq j \leq n$ ergibt somit insgesamt

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

Multiplikationen beziehungsweise Wurzeln. Der Aufwand ist demnach nur halb so groß wie für die LR-Zerlegung.

Beispiel 2.38 Für

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 10 \end{bmatrix}$$

ergibt sich wegen

$$\begin{aligned} \ell_{1,1} &= \sqrt{1} = 1 & \qquad \ell_{3,1} &= 1/1 = 1 \\ \ell_{2,1} &= 2/1 = 2 & \qquad \ell_{3,2} &= (2-2)/1 = 0 \\ \ell_{2,2} &= \sqrt{5-4} = 1 & \qquad \ell_{3,3} &= \sqrt{10-1} = 3 \end{aligned}$$

die Cholesky-Zerlegung $\mathbf{L}\mathbf{L}^\mathsf{T}$ mit

$$\mathbf{L} = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ 1 & 0 & 3 \end{bmatrix}, \qquad \mathbf{L}^\mathsf{T} = \begin{bmatrix} 1 & 2 & 1 \\ & 1 & 0 \\ & & 3 \end{bmatrix}.$$

△

Bemerkung 2.39 Im Gegensatz zur LR-Zerlegung ist die Cholesky-Zerlegung immer stabil. △

Analog zur LR-Zerlegung mit Rang-1-Updates, das ist Algorithmus 2.26, kann die Cholesky-Zerlegung mit Hilfe von Rang-1-Updates formuliert werden.

Algorithmus 2.40 (Cholesky-Zerlegung mit Rang-1-Updates)
input: reguläre Matrix $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{n\times n}$
output: Cholesky-Zerlegung $\mathbf{A} = \mathbf{L}\mathbf{L}^\mathsf{T} = [\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_n][\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_n]^\mathsf{T}$

① setze $\mathbf{A}_1 = \left[a_{i,j}^{(1)}\right] = \mathbf{A}$

② für alle $i = 1, 2, \ldots, n$ bilde

$$\boldsymbol{\ell}_i = \frac{1}{\sqrt{a_{i,i}^{(i)}}} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_{i,i}^{(i)} \\ a_{i+1,i}^{(i)} \\ \vdots \\ a_{n,i}^{(i)} \end{bmatrix} \in \mathbb{R}^n$$

und berechne $\mathbf{A}_{i+1} = \left[a_{i,j}^{(i+1)}\right] = \mathbf{A}_i - \boldsymbol{\ell}_i \boldsymbol{\ell}_i^\mathsf{T}$

Auf den ersten Blick sieht es so aus, als ob die Cholesky-Zerlegung in dieser Form dieselben Kosten verursacht wie die LR-Zerlegung. Man beachte allerdings, dass aufgrund der Symmetrie jeweils nur die halbe Matrix berechnet werden muss. Dies erklärt die Beschleunigung durch den Faktor 2.

3. Polynominterpolation

3.1 Lagrange-Interpolation

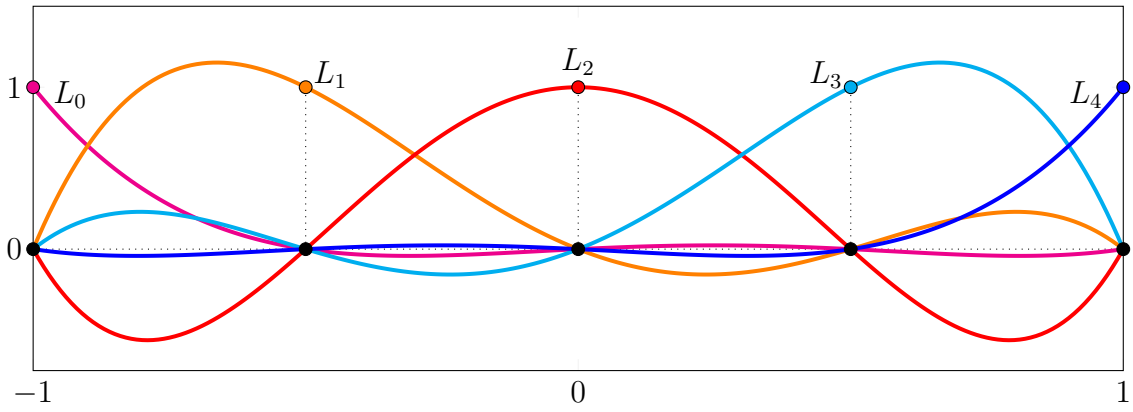


Abbildung 3.1: Lagrange-Polynome vom Grad 4.

In vielen Anwendungen sind nur diskrete Punktauswertungen einer Funktion gegeben. Diese können beispielsweise durch Sampling oder durch Messungen entstehen. Aufgabe ist es dann, auf eine sinnvolle Weise eine stetige Funktion durch diese Punktwerte zu legen. Dies kann mittels Polynominterpolation geschehen. Um deren Grundlagen zur Verfügung zu stellen, bezeichne

$$\Pi_n = \left\{ a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 : a_0, a_1, \ldots, a_n \in \mathbb{R} \right\}$$

den Raum der Polynome bis zum Grad n. Dann lautet die Aufgabenstellung wie folgt:

Problem 3.1 (Lagrangesche Interpolationsaufgabe) Gegeben seien $n+1$ paarweise verschiedene Knoten $x_0 < x_1 < \cdots < x_n$ sowie $n+1$ Werte $y_0, y_1, \ldots, y_n$, kurz $n+1$ Stützstellen $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$. Gesucht ist ein Polynom $p \in \Pi_n$ mit

$$p(x_i) \stackrel{!}{=} y_i \quad \text{für alle } i = 0, 1, \ldots, n. \tag{3.1}$$

Definition 3.2 Zu gegebenen $n+1$ Knoten $x_0 < x_1 < \cdots < x_n$ nennen wir

$$w(x) = \prod_{j=0}^{n} (x - x_j) \in \Pi_{n+1}$$

das **Knotenpolynom**. Die zugehörigen **Lagrange-Polynome** lauten

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \in \Pi_n, \quad i = 0, 1, \ldots, n.$$

Offensichtlich besitzen die Lagrange-Polynome gerade die Eigenschaft, dass gilt

$$L_i(x_j) = \delta_{i,j} \quad \text{für alle } i, j = 0, 1, \ldots, n, \tag{3.2}$$

vergleiche auch Abbildung 3.1 für eine Illustration der Lagrange-Polynome vom Grad 4. Daher kann man das *Interpolationspolynom* zu den Interpolationsbedingungen (3.1) leicht hinschreiben:

$$p(x) = \sum_{i=0}^{n} y_i L_i(x). \tag{3.3}$$

Denn wegen (3.2) ist

$$p(x_j) = \sum_{i=0}^{n} y_i L_i(x_j) = y_j \quad \text{für alle } j = 0, 1, \ldots, n.$$

Hiermit ist die Existenz einer Lösung gesichert. Dass (3.3) auch die einzige Lösung der Interpolationsaufgabe ist, besagt der nachfolgenden Satz.

Satz 3.3 Die Interpolationsaufgabe (3.1) ist eindeutig lösbar.

Beweis. Seien $p, q \in \Pi_n$ zwei Lösungen der Interpolationsaufgabe (3.1). Dann ist $p - q \in \Pi_n$, sprich ein Polynom vom Grad n, das aufgrund von

$$(p - q)(x_i) = 0 \quad \text{für alle } i = 0, 1, \ldots, n$$

$n+1$ Nullstellen besitzt. Daraus folgt jedoch $p(x) \equiv q(x)$. □

Bemerkung 3.4 Die Koeffizienten des Interpolationspolynoms lassen sich theoretisch auch mit Hilfe der *Vandermonde-Matrix*

$$\mathbf{V}(x_0, \ldots, x_n) := \begin{bmatrix} x_0^0 & x_0^1 & \cdots & x_0^n \\ x_1^0 & x_1^1 & \cdots & x_1^n \\ \vdots & \vdots & & \vdots \\ x_n^0 & x_n^1 & \cdots & x_n^n \end{bmatrix}$$

bestimmen. Die Koeffizienten $a_0, a_1, \ldots, a_n$ des Interpolationspolynoms $p(x)$ ergeben sich dann offenbar aus dem linearen Gleichungssystem

$$\begin{bmatrix} x_0^0 & x_0^1 & \cdots & x_0^n \\ x_1^0 & x_1^1 & \cdots & x_1^n \\ \vdots & \vdots & & \vdots \\ x_n^0 & x_n^1 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

Da für die Determinante der Vandermonde-Matrix gilt

$$\det \mathbf{V}(x_0, \ldots, x_n) = \prod_{0 \le i < j \le n} (x_j - x_i),$$

ergibt sich hieraus ebenfalls, dass die Interpolationsaufgabe (3.1) genau dann eindeutig eindeutig lösbar ist, wenn alle Stützstellen verschieden sind. Zur numerischen Lösung der Interpolationsaufgabe ist die Vandermonde-Matrix allerdings überhaupt nicht brauchbar. Man kann zeigen, dass ihre Kondition für gewisse Stützstellen exponentiell wächst. △

Wird eine hinreichend glatte Funktion $f(x)$ durch ein Polynom $p(x)$ interpoliert, dann ergibt sich die folgende Fehlerabschätzung für den Approximationsfehler $|f(x) - p(x)|$.

Satz 3.5 Seien f $(n+1)$-mal stetig differenzierbar und $p \in \Pi_n$ das Interpolationspolynom zu den Knoten $a = x_0 < x_1 < \cdots < x_n = b$ und Werten $y_i = f(x_i)$. Dann existiert zu jedem $x \in \mathbb{R}$ ein Wert ξ aus dem kleinsten Intervall I, das $[a, b]$ und x enthält, so dass

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} w(x). \tag{3.4}$$

Beweis. Fällt x für ein $0 \le i \le n$ mit einem Knoten x_i zusammen, so ist (3.4) gültig mit $\xi = x$. Sei also $x \notin \{x_i\}_{i=0}^n$. Dann besitzt die Funktion

$$h(t) := f(t) - p(t) - \frac{w(t)}{w(x)} \big\{ f(x) - p(x) \big\} \in C^{n+1}(I)$$

$n+2$ einfache Nullstellen, nämlich in $x_0, x_1, \ldots, x_n$ und für $t = x$. Folglich enthält jedes der $n+1$ Teilintervalle zwischen diesen Nullstellen nach dem Satz von Rolle eine einfache Nullstelle $t_i^{(1)}$ von h', $i = 1, 2, \ldots, n+1$. In jedem der n Teilintervalle zwischen den Nullstellen $\big\{t_i^{(1)}\big\}_{i=1}^{n+1}$ finden wir wiederum nach dem Satz von Rolle (vergleiche Abbildung 3.2) je eine einfache Nullstelle $t_i^{(2)}$ von h''. Ein wiederholtes Anwenden dieses Arguments liefert folglich $n+1, n, n-1, \ldots$ Nullstellen $\big\{t_i^{(k)}\big\}_{i=1}^{n+2-k}$ von $h^{(k)}$, $k = 1, 2, 3, \ldots$, im Intervall I. Schließlich ergibt sich eine Nullstelle $\xi = t_1^{(n+1)}$ von $h^{(n+1)}$ in I. Wegen $p \in \Pi_n$ ist $p^{(n+1)}(t) \equiv 0$, und wegen

$$w^{(n+1)}(t) = \left(\frac{\mathrm{d}}{\mathrm{d}t}\right)^{n+1} (t^{n+1} + \ldots) \equiv (n+1)!$$

folgt hieraus

$$0 = h^{(n+1)}(\xi) = f^{(n+1)}(\xi) - \frac{(n+1)!}{w(x)} \big\{ f(x) - p(x) \big\}.$$

Dies ist gleichbedeutend zum Ausdruck (3.4). □

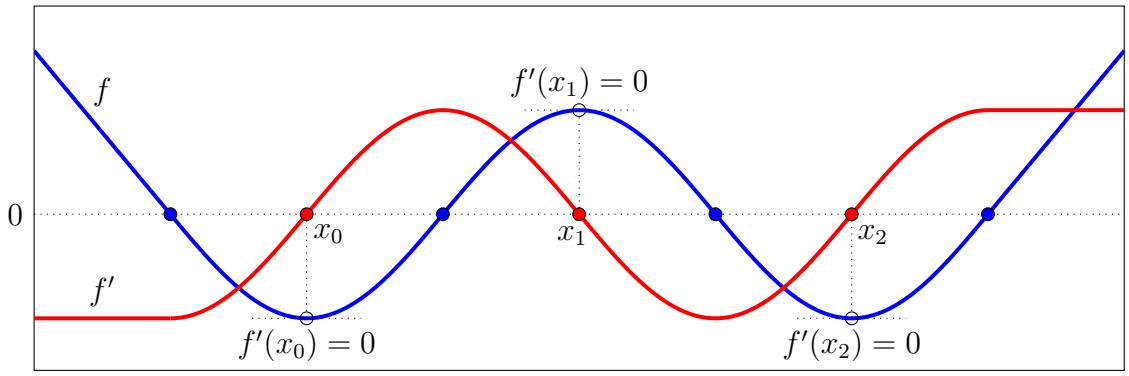


Abbildung 3.2: Illustration des Satzes von Rolle.

Bemerkung 3.6 Wegen $|w(x)| \leq (b-a)^{n+1}$, impliziert die punktweise Fehlerabschätzung (3.4) sofort die globale Fehlerschranke

$$\max_{x\in[a,b]} |f(x) - p(x)| \leq \frac{(b-a)^{n+1}}{(n+1)!} \left\| f^{(n+1)} \right\|_{C([a,b])},$$

wobei die $C([a,b])$-Norm definiert ist durch

$$\left\| f^{(n+1)} \right\|_{C([a,b])} := \max_{x\in[a,b]} \left| f^{(n+1)}(x) \right|.$$

Die verwendete Abschätzung für das Knotenpolynom ist aber je nach Knotenwahl recht grob. Abhängig von der speziellen Lage der Knoten ist eine verbesserte Fehlerabschätzung möglich. △

Als Verfahren der *numerischen Approximation* ist die Polynominterpolation nur mit großer Vorsicht anwendbar, weil große Oszillationen auftauchen können, wenn der Term $\max_{x\in[a,b]} |f^{(n+1)}(x)|/(n+1)!$ für $n \to \infty$ nicht beschränkt bleibt. Wenn man nichts genaues über die zu approximierende Funktion weiß, sollte deshalb der Grad des Interpolationspolynoms 6 nicht übersteigen.

Beispiel 3.7 (Runge) Die Folge von Interpolationspolynomen $p_n \in \Pi_{2n}$, die die Funktion

$$f(x) = \frac{1}{1+25x^2} \in C^\infty([-1,1])$$

in den äquidistanten Stützstellen $x_i = \pm i/n$ $(i = 0, 1, \ldots, n)$ interpolieren, konvergiert auf $[-1,1]$ nicht punktweise gegen f. Dies hat Carl Runge im Jahr 1901 nachgewiesen. Eine Visualisierung der Interpolationspolynome in den Fällen $n = 8$ und $n = 16$ ist in Abbildung 3.3 zu finden. △

Bemerkung 3.8 Bei der *Hermite-Interpolation* werden einzelne Knoten mehrfach zugelassen. Tritt beispielsweise der Knoten x_i k-mal auf, so werden neben dem Funktionswert y_i an der Stelle $x = x_i$ zusätzlich zum Funktionswert auch die ersten $k-1$ Ableitungen des Interpolationspolynoms vorgeschrieben:

$$p(x_i) = y_i,\ p'(x_i) = y_i',\ \ldots,\ p^{(k-1)}(x_i) = y_i^{(k-1)}.$$

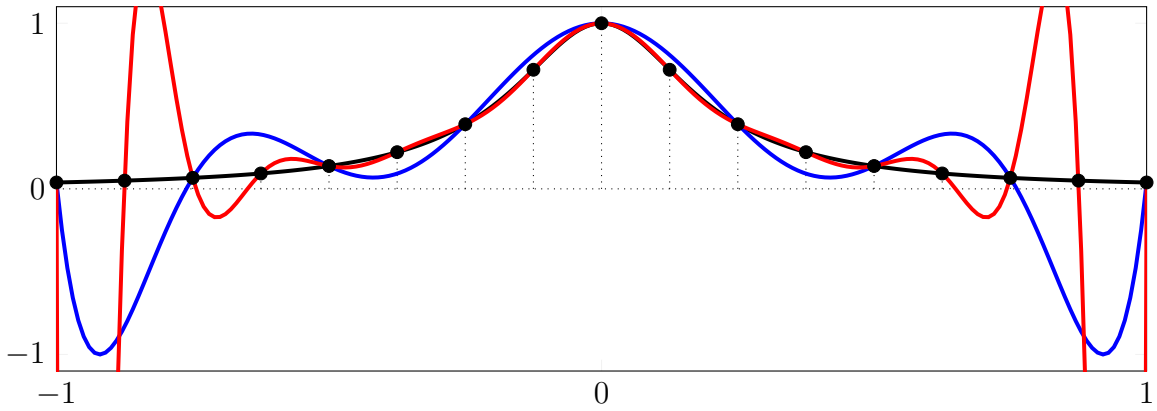


Abbildung 3.3: Runge-Funktion (schwarz) und Interpolationspolynome vom Grad 8 (blau) und vom Grad 16 (rot) zu äquidistanten Stützstellen.

Auch diese Interpolationsaufgabe ist eindeutig lösbar und Satz 3.5 gilt entsprechend: Im Knotenpolynom w treten mehrfache Knoten x_i dann mit entsprechender Vielfachheit auf. △

3.2 Neville-Schema

Die Bestimmung des Interpolationspolynoms mit Hilfe der Lagrange-Polynome ist instabil und teuer. Will man das Interpolationspolynom nur an einigen wenigen Stellen auswerten, so bietet sich das Neville-Schema an. Um es herzuleiten, bezeichnen wir für $0 \le i \le i+j \le n$ mit $f_{i,i+1,\dots,i+j}(x)$ dasjenige Polynom vom Grad $\le j$, das in den $n+1$ Stützstellen $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ die Interpolationsbedingungen

$$f_{i,i+1,\dots,i+j}(x_k) = y_k, \qquad k = i, i+1, \dots, i+j$$

erfüllt. Dieses Polynom ist gemäß Satz 3.3 eindeutig bestimmt.

Satz 3.9 Die Interpolationspolynome $f_{i,i+1,\dots,i+j}(x)$ mit $0 \le i \le i+j \le n$ genügen der Rekursionsformel

$$\begin{aligned} f_i(x) &\equiv y_i, \quad j = 0, \\ f_{i,i+1,\dots,i+j}(x) &= \frac{(x - x_i) f_{i+1,i+2,\dots,i+j}(x) - (x - x_{i+j}) f_{i,i+1,\dots,i+j-1}(x)}{x_{i+j} - x_i}, \quad j \ge 1. \end{aligned} \tag{3.5}$$

Beweis. Wir führen den Beweis mittels vollständiger Induktion über j. Für $j = 0$ ist die Aussage für alle i wegen $f_i \in \Pi_0$ und $f_i(x_i) = y_i$ offensichtlich richtig. Um den Induktionsschritt $j-1 \mapsto j$ nachzuweisen, bezeichnen wir mit $q(x)$ die rechte Seite von (3.5) und zeigen $p(x) \equiv f_{i,i+1,\dots,i+j}(x)$. Weil $f_{i+1,i+2,\dots,i+j}(x)$ und $f_{i,i+1,\dots,i+j-1}(x)$ Polynome

vom Grad $j-1$ sind, ist $p(x)$ ein Polynom vom Grad j. Weiter gilt

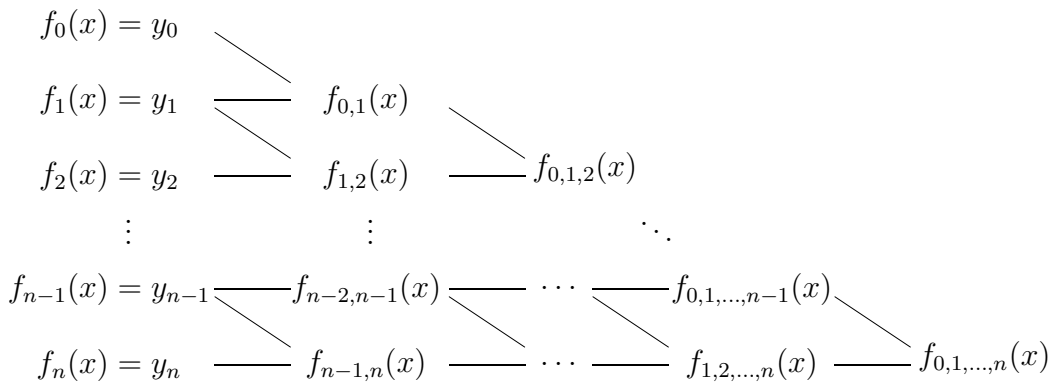
$$p(x_i) = \frac{0-(x_i-x_{i+j})f_{i,i+1,\ldots,i+j-1}(x_i)}{x_{i+j}-x_i} = \frac{0-(x_i-x_{i+j})y_i}{x_{i+j}-x_i} = y_i,$$
$$p(x_{i+j}) = \frac{(x_{i+j}-x_i)f_{i+1,i+2,\ldots,i+j}(x)-0}{x_{i+j}-x_i} = \frac{(x_{i+j}-x_i)y_{i+j}-0}{x_{i+j}-x_i} = y_{i+j},$$

und für alle $k = i+1, i+2, \ldots, i+j-1$ gilt

$$\begin{aligned} p(x_k) &= \frac{(x_k-x_i)f_{i+1,i+2,\ldots,i+j}(x_k)-(x_k-x_{i+j})f_{i,i+1,\ldots,i+j-1}(x_k)}{x_{i+j}-x_i} \\ &= \frac{(x_k-x_i)y_k-(x_k-x_{i+j})y_k}{x_{i+j}-x_i} = \frac{-x_iy_k+x_{i+j}y_k}{x_{i+j}-x_i} = y_k. \end{aligned}$$

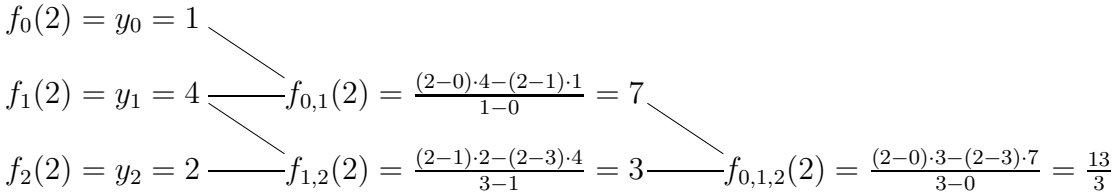
Aufgrund der Eindeutigkeit des Interpolationspolynoms müssen daher notwendigerweise $p(x)$ und $f_{i,i+1,\ldots,i+j}(x)$ identisch sein. □

Die sich für die Werte $f_{i,i+1,\ldots,i+j}(x)$ aus der Rekursionsformel (3.5) ergebenden Abhängigkeiten sind im nachfolgendem *Neville-Schema* dargestellt:



Die Einträge lassen sich spaltenweise jeweils von oben nach unten berechnen. Das resultierende Verfahren wird zur Auswertung des Interpolationspolynoms an einzelnen Stellen x verwendet. Wie man leicht nachzählt, fallen dabei jeweils $3n^2/2 + \mathcal{O}(n)$ Multiplikationen an.

Beispiel 3.10 Gegeben seien die Stützstellen $(0,1)$, $(1,4)$ und $(3,2)$. Für $x = 2$ ergibt die Auswertung des zugehörigen Interpolationspolynoms gemäß dem Neville-Schema:



△

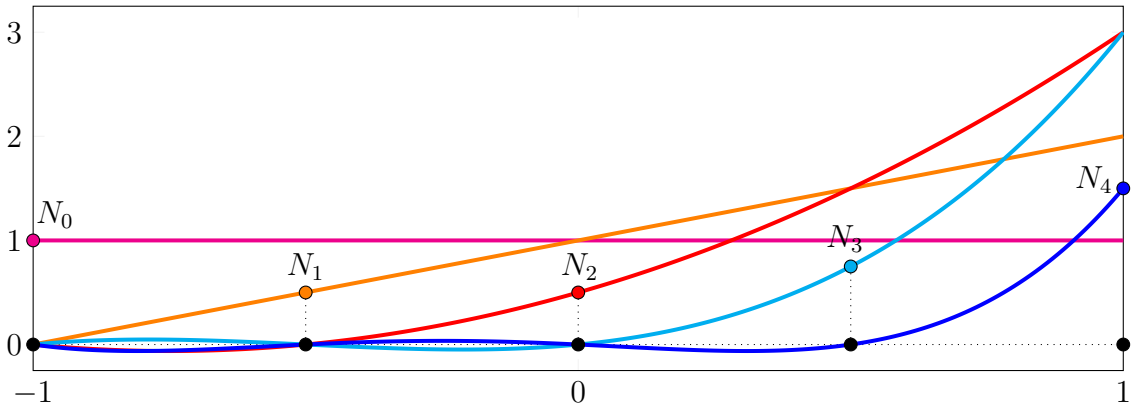


Abbildung 3.4: Die ersten vier Newton-Polynome.

3.3 Newtonsche Interpolationsformel

Im Neville-Schema wird das Interpolationspolynom nicht aufgestellt. Daher ist dieses Schema nur dann sinnvoll, wenn das Interpolationspolynom an einigen wenigen Stellen ausgewertet werden soll. Werden hingegen viele Auswertungen das Interpolationpolynoms benötigt, ist es besser, es explizit aufzustellen. Dies geschieht mit Hilfe des Newton-Schemas.

Definition 3.11 Zu gegebenen paarweise verschiedenen $n+1$ Knoten $x_0, x_1, \ldots, x_n \in \mathbb{R}$ sind die $n+1$ **Newton-Polynome** folgendermaßen erklärt:

$$N_0(x) = 1, \qquad N_i(x) = (x - x_{i-1})N_{i-1}(x), \quad 1 \le i \le n. \tag{3.6}$$

Offensichtlich gilt

$$N_i(x) = (x - x_{i-1})(x - x_{i-2}) \cdots (x - x_0),$$

das heißt, $N_i(x)$ stimmt mit dem Knotenpolynom zu den Knoten $x_0, x_1, \ldots, x_{i-1}$ überein, ist also ein Polynom vom Grad i. Folglich ist die Menge aller N_i linear unabhängig und das gesuchte interpolierende Polynom $p \in \Pi_n$ kann als Linearkombination der Newton-Polynome dargestellt werden:

$$p(x) = \sum_{i=0}^{n} a_i N_i(x). \tag{3.7}$$

Eine Illustration der Newton-Polynome findet sich in Abbildung 3.4.

Darstellung (3.7) ist die *Newtonsche Darstellung* des Interpolationspolynoms. Benutzt man die rekursive Struktur der Newton-Polynome,

$$a_i N_i(\xi) + a_{i-1} N_{i-1}(\xi) = \big(a_i(\xi - x_{i-1}) + a_{i-1}\big) N_{i-1}(\xi) \quad \text{für alle } i = 1, 2, \ldots, n,$$

so sieht man leicht ein, dass sich das Interpolationspolynom $p(x)$ für jedes $x = \xi$ effizient mit dem *Horner-Schema* auswerten lässt:

$$p(\xi) = \Big(\cdots\Big(\big(a_n(\xi - x_{n-1}) + a_{n-1}\big)(\xi - x_{n-2}) + a_{n-2}\Big)(\xi - x_{n-3}) + \cdots + a_1\Big)(\xi - x_0) + a_0.$$

Die Auswertung wird von links nach rechts durchgeführt, was auf den nachfolgenden Algorithmus führt:

Algorithmus 3.12 (Horner-Schema)
input: Koeffizienten $\{a_i\}_{i=0}^n$ und Stützstellen $\{x_i\}_{i=0}^n$ der Newton-Darstellung des Interpolationspolynoms und Auswertepunkt $\xi \in \mathbb{R}$
output: Funktionsauswertung $y = p(\xi)$ des Interpolationspolynoms

① Initialisierung: setze $y = a_n$
② für alle $i = n-1, n-2, \ldots, 0$ datiere $y = y \cdot (\xi - x_i) + a_i$ auf

Nachzählen bestätigt, dass bei jeder Auswertung des Interpolationspolynoms nur $3n$ Operationen anfallen. Das ist wesentlich weniger als beim quadratisch skalierenden Neville-Schema. Allerdings müssen wir das Interpolationspolynom in der Newton-Darstellung (3.7) erst noch aufstellen.
Weil $N_i(x_j) = 0$ für alle $j < i$ ist, können die Koeffizienten $a_i \in \mathbb{R}$ sukzessive aus den Interpolationsbedingungen gewonnen werden:

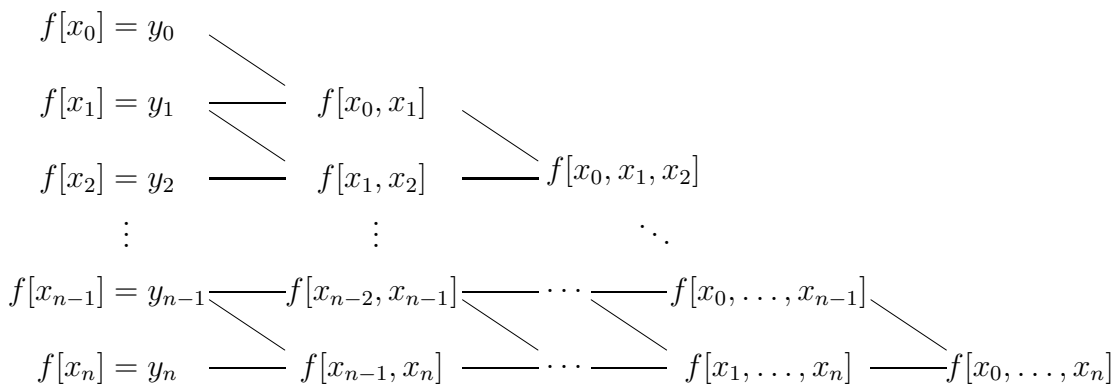
$$\begin{aligned} y_0 &= p(x_0) = a_0 N_0(x_0), \\ y_1 &= p(x_1) = a_0 N_0(x_1) + a_1 N_1(x_1), \\ y_2 &= p(x_2) = a_0 N_0(x_2) + a_1 N_1(x_2) + a_2 N_2(x_2), \\ &\vdots \end{aligned}$$

Bei einer naiven Berechnung fallen allerdings $n^3/6 + \mathcal{O}(n^2)$ Multiplikationen an. Daher werden wir im folgenden dividierte Differenzen verwenden, welche eine Berechnung der Koeffizienten in $\mathcal{O}(n^2)$ Operationen ermöglichen.

Definition 3.13 Zu gegebenen Stützstellen $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ sind die **dividierten Differenzen** erklärt gemäß

$$\begin{aligned} f[x_i] &\equiv y_i, \qquad i = 0, 1, 2, \ldots, n, \\ f[x_i, x_{i+1}, \ldots, x_{i+j}] &= \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+j}] - f[x_i, x_{i+1}, \ldots, x_{i+j-1}]}{x_{i+j} - x_i}, \\ & \qquad\qquad 0 \le i < i + j \le n. \end{aligned} \tag{3.8}$$

Für die Berechnung aller dividierten Differenzen zu den Stützstellen $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ sind lediglich $n(n+1)/2$ Multiplikationen erforderlich. Die Abhängigkeiten zwischen den dividierten Differenzen sind im folgendem *Newton-Schema* dargestellt:



Satz 3.14 (Newtonsche Interpolationsformel) Für das Interpolationspolynom $p \in \Pi_n$ zu gegebenen Stützstellen $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ gilt

$$p(x) = f[x_0]N_0(x) + f[x_0, x_1]N_1(x) + \cdots + f[x_0, x_1, \ldots, x_n]N_n(x). \tag{3.9}$$

Beweis. Wir bemerken zunächst, dass der führende Koeffizient des Polynoms $p(x)$ aus (3.9) durch $f[x_0, x_1, \ldots, x_n]$ gegeben ist:

$$p(x) = f[x_0, x_1, \ldots, x_n]x^n + \ldots.$$

Wir wollen den Beweis nun mit vollständiger Induktion führen. Im Fall $n = 0$ ist die Aussage klar. Wir nehmen daher an, dass die Darstellung (3.9) für $n - 1$ gilt. Für die Interpolationspolynome $f_{0,1,\ldots,n-1}, f_{1,2,\ldots,n} \in \Pi_{n-1}$ folgt dann nach Induktionsannahme

$$\begin{aligned} f_{0,1,\ldots,n-1}(x) &= f[x_0, x_1, \ldots, x_{n-1}]x^{n-1} + \ldots, \\ f_{1,2,\ldots,n}(x) &= f[x_1, x_2, \ldots, x_n]x^{n-1} + \ldots. \end{aligned}$$

Wegen $f_{0,1,\ldots,n} \in \Pi_n$, lässt sich das Interpolationspolynom $f_{0,1,\ldots,n}(x)$ schreiben als

$$f_{0,1,\ldots,n}(x) = c\underbrace{(x - x_0)(x - x_1)\cdots(x - x_{n-1})}_{=N_n(x)} + q(x), \quad c \in \mathbb{R}, \quad q \in \Pi_{n-1}.$$

Können wir zeigen, dass $c = f[x_0, x_1, \ldots, x_n]$ und $q(x) \equiv f_{0,1,\ldots,n-1}(x)$ gilt, dann ergibt sich die Behauptung aus $p(x) \equiv f_{0,1,\ldots,n}(x)$. Aus

$$y_i = f_{0,1,\ldots,n}(x_i) = c\underbrace{(x_i - x_0)(x_i - x_1)\cdots(x_i - x_{n-1})}_{=0} + q(x_i) \quad \text{für alle } i = 0, 1, \ldots, n-1$$

folgt tatsächlich $q(x) \equiv f_{0,1,\ldots,n-1}(x)$. Gemäß (3.5) gilt ferner

$$\begin{aligned} f_{0,1,\ldots,n}(x) &= \frac{(x - x_0)f_{1,2,\ldots,n}(x) - (x - x_n)f_{0,1,\ldots,n-1}(x)}{x_n - x_0} \\ &= \frac{(x - x_0)(f[x_1, \ldots, x_n]x^{n-1} + \ldots) - (x - x_n)(f[x_0, \ldots, x_{n-1}]x^{n-1} + \ldots)}{x_n - x_0} \\ &= \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0}x^n + \ldots, \end{aligned}$$

dies bedeutet

$$c = \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0} \overset{(3.8)}{=} f[x_0, x_1, \ldots, x_n].$$

Damit ist der Satz bewiesen. □

Beispiel 3.15 Zu den Stützstellen aus Beispiel 3.10 wollen wir das entsprechende Interpolationspolynom in der Newtonschen Darstellung bestimmen. Mit dem Newton-Schema erhalten wir

$$\begin{array}{lllll} f[x_0] = y_0 = 1 & & & & \\ & \searrow & & & \\ f[x_1] = y_1 = 4 & \text{——} & f[x_0, x_1] = \frac{4-1}{1-0} = 3 & \searrow & \\ & \searrow & & & \\ f[x_2] = y_2 = 2 & \text{——} & f[x_1, x_2] = \frac{2-4}{3-1} = -1 & \text{——} & f[x_0, x_1, x_2] = \frac{-1-3}{3-0} = -\frac{4}{3} \end{array}$$

Damit ergibt sich schließlich das Interpolationspolynom

$$p(x) = 1 + 3x - \frac{4}{3}x(x-1).$$

△

3.4 Tschebyscheff-Interpolation

In der Abschätzung (3.4) des Interpolationsfehlers taucht als einziger Stellhebel für die Genauigkeit das Knotenpolynom $w(x)$ auf, welches von der Wahl der Knoten abhängt. Daher wollen wir im folgenden herausfinden, wie die bestmögliche Wahl der Knoten aussieht. Wir suchen also Knoten $a \le x_0 < x_1 < \cdots < x_{n-1} < x_n \le b$, so dass der Ausdruck

$$\|w\|_{C([a,b])} = \max_{x \in [a,b]} \prod_{i=0}^{n} |x - x_i|$$

minimal wird.

Zur Klärung dieser Frage müssen wir Tschebyscheff-Polynome untersuchen. Diese sind auf dem Intervall $[-1, 1]$ definiert durch die Gleichung

$$T_n(x) = \cos(n \arccos x), \quad -1 \le x \le 1. \tag{3.10}$$

Obwohl T_n auf den ersten Blick nicht wie ein Polynom aussieht, überzeugt man sich leicht, dass

$$T_0(x) = 1, \quad T_1(x) = x,$$

und — mittels trigonometrischer Identitäten und der Substitution $\xi = \arccos x$ —

$$\begin{aligned} T_{n-1}(x) + T_{n+1}(x) &= \cos\big((n-1)\xi\big) + \cos\big((n+1)\xi\big) \\ &= \cos n\xi \cos\xi - \sin n\xi \sin(-\xi) + \cos n\xi \cos\xi - \sin n\xi \sin\xi \\ &= 2\cos(\arccos x)\cos(n \arccos x) \\ &= 2xT_n(x). \end{aligned}$$

Mit anderen Worten, es gilt die Dreitermrekursion

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n = 1, 2, \ldots, \tag{3.11}$$

und man erkennt, dass $T_n(x)$ ein Polynom vom Grad n ist. Ferner ergibt sich aus (3.10)

$$|T_n(x)| \leq 1 \text{ für alle } x \in [-1, 1],$$

während aus (3.11) folgt

$$T_n(x) = 2^{n-1}x^n + \ldots.$$

Die Tschebyscheff-Polynome haben eine Reihe von extremalen Eigenschaften, die sie für viele Anwendungen in der Numerik interessant machen. Die für unseren Zweck benötigte Eigenschaft ist in folgendem Satz formuliert:

Satz 3.16 Unter allen monischen Polynomen $p_n(x) = x^n + \ldots$ minimiert $2^{1-n}T_n$ die Maximumsnorm $\|p_n\|_{C([-1,1])}$ über $[-1, 1]$.

Beweis. Sei $s_n := 2^{n-1}T_n$ und p_n ein weiteres monisches Polynom vom Grad n. Dann hat $s_n - p_n$ höchstens den Grad $n - 1$. Wir nehmen nun an, dass gilt

$$\|p_n\|_{C([-1,1])} < 2^{1-n} = \|s_n\|_{C([-1,1])}.$$

Da s_n nach (3.10) an den $n + 1$ Stellen $x_k = \cos(k\pi/n)$, $0 \leq k \leq n$, alternierend die Extremalwerte $\pm 2^{1-n}$ annimmt, hat $s_n - p_n$ mindestens n Vorzeichenwechsel, also n Nullstellen. Da der Grad des Polynoms $s_n - p_n$ jedoch kleiner oder gleich $n - 1$ ist, muss daher $s_n - p_n \equiv 0$ gelten im Widerspruch zur gemachten Annahme. □

Dieser Satz besagt, dass auf dem Intervall $[-1, 1]$ das Polynom $2^{-n}T_{n+1}$ das optimale Knotenpolynom zu $n+1$ Stützstellen darstellt. Folglich müssen die Knoten als Nullstellen dieses Polynoms gewählt werden, die gegeben sind durch

$$x_k = \cos\left(\frac{2k+1}{2n+2}\pi\right), \quad k = 0, 1, \ldots, n.$$

Wie in Abbildung (3.5) dargestellt ist, entstehen diese Knoten also durch eine Projektion von äquidistant auf dem Einheitskreis verteilten Punkten auf die reelle Achse. Speziell liegen sie am Rand des Intervalls $[-1, 1]$ viel dichter als in dessen Inneren.

Für ein beliebiges Intervall haben wir folgende Aussage:

Korollar 3.17 Sei $f \in C^{n+1}([a, b])$. Löst $p \in \Pi_n$ die Interpolationsaufgabe (3.1) in den *Tschebyscheff-Knoten*

$$x_k = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{2k+1}{2n+2}\pi\right), \quad k = 0, 1, \ldots, n,$$

so gilt die Fehlerabschätzung

$$\max_{x\in[a,b]} |f(x) - p(x)| \leq \frac{(b-a)^{n+1}}{2^{2n+1}} \frac{\|f^{(n+1)}\|_{C([a,b])}}{(n+1)!}.$$

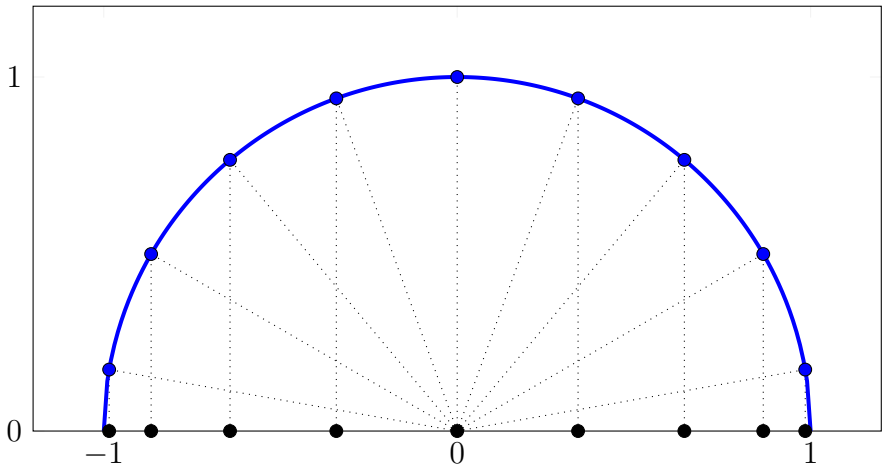


Abbildung 3.5: Die Stützstellen für die Tschebyscheff-Interpolation entstehen durch die Projektion von äquidistant auf dem Einheitskreis verteilten Punkten.

Beweis. Das Intervall $[a,b]$ wird durch die affine Transformation $x \mapsto (2x-b-a)/(b-a)$ auf das Intervall $[-1,1]$ abgebildet. Wie man leicht überprüft, sind die ausgewählten Knoten gerade die Nullstellen des damit transformierten Tschebyscheff-Polynoms $T_{n+1}\big((2x-b-a)/(b-a)\big)$. Wegen

$$2^{-n}T_{n+1}\left(\frac{2x-b-a}{b-a}\right) = \left(\frac{2x}{b-a}\right)^{n+1} + \cdots,$$

gilt für das Knotenpolynom

$$w(x) = \prod_{j=0}^{n}(x-x_j) = \left(\frac{b-a}{2}\right)^{n+1} 2^{-n}T_{n+1}\left(\frac{2x-b-a}{b-a}\right).$$

Hieraus folgt

$$\|w\|_{C([a,b])} \le \frac{(b-a)^{n+1}}{2^{2n+1}} \underbrace{\max_{x\in[a,b]}\left|T_{n+1}\left(\frac{2x-b-a}{b-a}\right)\right|}_{=1}.$$

Somit genügt das Interpolationspolynom $p \in \Pi_n$ der Fehlerabschätzung

$$\|f-p\|_{C([a,b])} \le \|w\|_{C([a,b])}\frac{\|f^{(n+1)}\|_{C([a,b])}}{(n+1)!} \le \frac{(b-a)^{n+1}}{2^{2n+1}}\frac{\|f^{(n+1)}\|_{C([a,b])}}{(n+1)!}.$$

□

4. Trigonometrische Interpolation

4.1 Theoretische Grundlagen

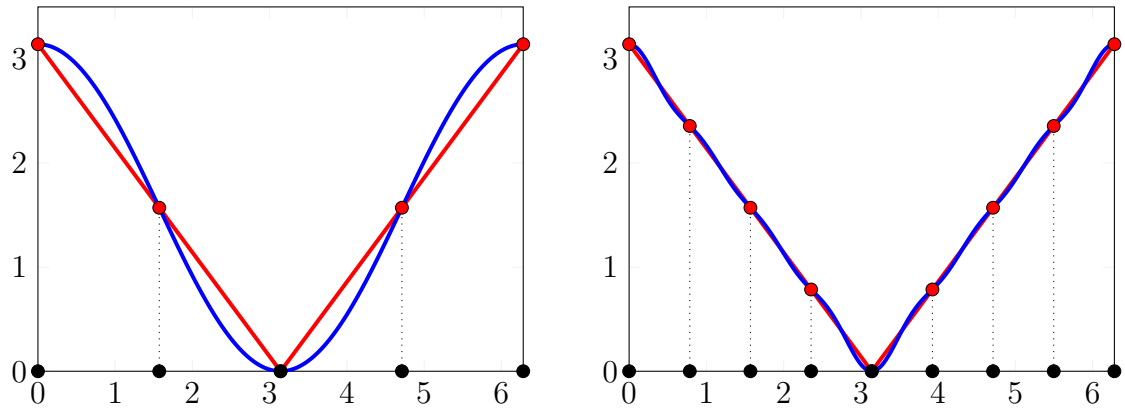


Abbildung 4.1: Trigonometrische Interpolation der 2π-periodischen Fortsetzung der Funktion $f(x) = |x - \pi|$ in $n = 4$ (links) and $n = 8$ (rechts) äquidistanten Stützstellen auf $[0, 2\pi)$.

Die trigonometrische Interpolation dient zur Analyse von 2π-periodischen Funktionen, das sind Funktionen, für die gilt

$$f(x) = f(x + 2\pi k) \quad \text{für alle } x \in [0, 2\pi) \text{ und } k \in \mathbb{Z}.$$

Gegeben seien n Stützstellen $(x_0, y_0), (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$, wobei wir der Einfachheit halber sogar annehmen, dass die Knoten $\{x_k\}_{k=0}^{n-1}$ äquidistant verteilt sind, also

$$x_k = \frac{2\pi k}{n}, \quad k = 0, 1, \ldots, n-1.$$

Da die trigonometrischen Funktionen $\cos(kx)$ und $\sin(kx)$ für ganzzahliges k alle die Periode 2π besitzen, liegt es nahe, aus n solchen Funktionen geeignete Linearkombinationen zur Interpolation der n Stützpunkte zu verwenden. Als zweckmäßig stellt sich ein Ansatz der Form

$$\begin{aligned} q(x) &= \frac{a_0}{2} + \sum_{\ell=1}^{m} \big\{a_\ell \cos(\ell x) + b_\ell \sin(\ell x)\big\}, && \text{falls } n = 2m+1 \\ q(x) &= \frac{a_0}{2} + \sum_{\ell=1}^{m-1} \big\{a_\ell \cos(\ell x) + b_\ell \sin(\ell x)\big\} + \frac{a_m}{2}\cos(mx), && \text{falls } n = 2m \end{aligned} \tag{4.1}$$

heraus, denn er entspricht gerade einer Zerlegung der 2π-periodischen Funktion f in die ersten n Frequenzen. In Abbildung 4.1 findet sich eine Illustration der trigonometrischen Interpolationspolynome im Fall einer Zick-Zack-Funktion.
Die Formeln werden bei komplexer Rechnung durchsichtiger:

Problem 4.1 (trigonometrische Interpolationsaufgabe) Gesucht ist ein *trigonometrisches Polynom* der Ordnung n

$$p(x) = \beta_0 + \beta_1 e^{ix} + \beta_2 e^{2ix} + \cdots + \beta_{n-1} e^{(n-1)ix} \tag{4.2}$$

mit

$$p(x_k) \stackrel{!}{=} y_k \quad \text{für alle } k = 0, 1, \ldots, n-1. \tag{4.3}$$

Dass (4.2) äquivalent zu den Ausdrücken (4.1) ist, sieht man leicht aus der Moivreschen Formel

$$e^{i\ell x} = \cos(\ell x) + i \sin(\ell x)$$

und der Definition der x_k,

$$e^{-i\ell x_k} = e^{-2\pi i \ell k/n} = e^{2\pi i (n-\ell) k/n} = e^{i(n-\ell)x_k},$$

so dass

$$\cos(\ell x_k) = \frac{e^{i\ell x_k} + e^{i(n-\ell)x_k}}{2}, \quad \sin(\ell x_k) = \frac{e^{i\ell x_k} - e^{i(n-\ell)x_k}}{2i}. \tag{4.4}$$

Ersetzt man in (4.1) für $x = x_k$ die Terme $\cos(\ell x_k)$ und $\sin(\ell x_k)$ durch (4.4) und ordnet man die Summanden nach Potenzen von e^{ix_k} um, findet man für $n = 2m + 1$ den Zusammenhang

$$\begin{aligned} &\beta_0 = \frac{a_0}{2}, \qquad \beta_k = \frac{a_k - ib_k}{2}, \qquad \beta_{n-k} = \frac{a_k + ib_k}{2}, \qquad k = 1, 2, \ldots, m, \\ &a_0 = 2\beta_0, \qquad a_\ell = \beta_\ell + \beta_{n-\ell}, \qquad b_\ell = i(\beta_\ell - \beta_{n-\ell}), \qquad \ell = 1, 2, \ldots, m, \end{aligned} \tag{4.5}$$

und für $n = 2m$

$$\begin{aligned} &\beta_0 = \frac{a_0}{2}, \quad \beta_k = \frac{a_k - ib_k}{2}, \quad \beta_{n-k} = \frac{a_k + ib_k}{2}, \quad k = 1, 2, \ldots, m-1, \quad \beta_m = \frac{a_m}{2}, \\ &a_0 = 2\beta_0, \quad a_\ell = \beta_\ell + \beta_{n-\ell}, \quad b_\ell = i(\beta_\ell - \beta_{n-\ell}), \quad \ell = 1, 2, \ldots, m-1, \quad a_m = 2\beta_m. \end{aligned} \tag{4.6}$$

Man beachte jedoch, dass entsprechend der Herleitung $p(x_k) = q(x_k)$ für alle $k = 0, 1, \ldots, n-1$ gilt, jedoch ist im allgemeinen nicht $p(x) = q(x)$ für $x \neq x_k$. Damit sind p und q nur in dem Sinne äquivalent, dass aus der Darstellung (4.2) sofort die Darstellung (4.1) folgt, und umgekehrt.

Satz 4.2 Zu beliebigen Werten $y_k \in \mathbb{C}$ gibt es genau ein trigonometrisches Polynom der Gestalt (4.2), das die Interpolationsaufgabe (4.3) löst.

Beweis. Substituieren wir in (4.2) $\omega = e^{ix}$ und $\omega_k = e^{ix_k}$, so stellen wir fest, dass die Interpolationsaufgabe (4.3) äquivalent ist zu: suche

$$r(\omega) = \sum_{\ell=0}^{n-1} \beta_\ell \omega^\ell \in \Pi_{n-1}$$

mit $r(\omega_k) \stackrel{!}{=} y_k$ für alle $k = 0, 1, \ldots, n-1$. Diese Aufgabe ist aber gemäß Satz 3.3 eindeutig lösbar. □

Die Koeffizienten β_ℓ können explizit angegeben werden. Eine wesentliche Rolle spielt dabei die n-te komplexe Einheitswurzel

$$\omega_n := e^{2\pi i/n}.$$

Speziell gelten für sie die Rechenregeln

$$\overline{\omega_n^k} = \overline{e^{2\pi ik/n}} = e^{-2\pi ik/n} = \omega_n^{-k} \tag{4.7}$$

und

$$\omega_n^k \omega_n^\ell = e^{2\pi ik/n} e^{2\pi i\ell/n} = e^{2\pi i(k+\ell)/n} = \omega_n^{k+\ell}. \tag{4.8}$$

Lemma 4.3 Die Vektoren

$$\boldsymbol{\omega}^k := \begin{bmatrix} \omega_n^0 \\ \omega_n^k \\ \omega_n^{2k} \\ \vdots \\ \omega_n^{(n-1)k} \end{bmatrix}, \qquad k = 0, 1, \ldots, n-1, \tag{4.9}$$

bilden eine Orthogonalbasis im $\mathbb{C}^n$ mit

$$(\boldsymbol{\omega}^\ell)^\star \boldsymbol{\omega}^k = \begin{cases} n, & \text{falls } k = \ell, \\ 0, & \text{falls } k \neq \ell. \end{cases} \tag{4.10}$$

Beweis. Der Beweis ergibt sich durch einfaches Nachrechnen. Für beliebiges $0 \leq k, \ell < n$ gilt wegen (4.7) und (4.8)

$$(\boldsymbol{\omega}^\ell)^\star \boldsymbol{\omega}^k = \sum_{j=0}^{n-1} \overline{\omega_n^{j\ell}} \omega_n^{jk} = \sum_{j=0}^{n-1} \omega_n^{-j\ell} \omega_n^{jk} = \sum_{j=0}^{n-1} \omega_n^{j(k-\ell)}.$$

Im Fall $k = \ell$ ergibt sich

$$(\boldsymbol{\omega}^\ell)^\star \boldsymbol{\omega}^k = \sum_{j=0}^{n-1} \omega_n^0 = \sum_{j=0}^{n-1} 1 = n.$$

Ist $k \neq \ell$, so folgt mit der Summenformel für die geometrische Reihe

$$(\boldsymbol{\omega}^\ell)^\star \boldsymbol{\omega}^k = \sum_{j=0}^{n-1} e^{2\pi ij(k-\ell)/n} = \frac{e^{2\pi in(k-\ell)/n} - 1}{e^{2\pi i(k-\ell)/n} - 1} = \frac{1-1}{e^{2\pi i(k-\ell)/n} - 1} = 0.$$

□

Mit Hilfe dieses Lemmas können wir nun die Koeffizienten $\beta_\ell \in \mathbb{C}$ des trigonometrischen Polynoms (4.2) ausrechnen.

Satz 4.4 Das trigonometrische Polynom $p(x) = \sum_{\ell=0}^{n-1} \beta_\ell e^{i\ell x}$ erfüllt die Interpolationsbedingung (4.3) genau dann, wenn

$$\beta_\ell = \frac{1}{n}\sum_{j=0}^{n-1} y_j \omega_n^{-j\ell} = \frac{1}{n}\sum_{j=0}^{n-1} y_j e^{-2\pi i j\ell/n}, \qquad \ell = 0, 1, \ldots, n-1.$$

Beweis. Definieren wir den Datenvektor

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} \in \mathbb{C}^n,$$

dann können wir im Hinblick auf (4.9) die Interpolationsaufgabe (4.3) schreiben als

$$\mathbf{y} = \sum_{k=0}^{n-1} \beta_k \boldsymbol{\omega}^k.$$

Zusammen mit (4.10) folgt hieraus das Behauptete:

$$\sum_{j=0}^{n-1} y_j \omega_n^{-j\ell} = (\boldsymbol{\omega}^\ell)^\star \mathbf{y} = (\boldsymbol{\omega}^\ell)^\star \left(\sum_{k=0}^{n-1} \beta_k \boldsymbol{\omega}^k \right) = \sum_{k=0}^{n-1} \beta_k \underbrace{(\boldsymbol{\omega}^\ell)^\star \boldsymbol{\omega}^k}_{=n\delta_{k,\ell}} = n\beta_\ell.$$

□

Wir kehren nun zu den trigonometrischen Ausdrücken (4.1) zurück.

Satz 4.5 Die trigonometrischen Ausdrücke (4.1) genügen den Interpolationsbedingungen $q(x_k) = y_k$ für alle $k = 0, 1, 2, \ldots, n-1$ genau dann, wenn für ihre Koeffizienten gilt

$$a_\ell = \frac{2}{n}\sum_{j=0}^{n-1} y_j \cos(jx_\ell) = \frac{2}{n}\sum_{j=0}^{n-1} y_j \cos\left(\frac{2\pi j\ell}{n}\right),$$
$$b_\ell = \frac{2}{n}\sum_{j=0}^{n-1} y_j \sin(jx_\ell) = \frac{2}{n}\sum_{j=0}^{n-1} y_j \sin\left(\frac{2\pi j\ell}{n}\right).$$

Beweis. Aufgrund der Umrechnungsformeln (4.5) und (4.6) können wir das Resultat von Satz 4.4 sofort auf die Koeffizienten $a_\ell \in \mathbb{R}$ und $b_\ell \in \mathbb{R}$ des reellen trigonometrischen Polynomes $q(x)$ aus (4.1) übersetzten. Für den Koeffizienten a_0 gilt

$$a_0 = 2\beta_0 = \frac{2}{n}\sum_{j=0}^{n-1} y_j \omega_n^{-j0} = \frac{2}{n}\sum_{j=0}^{n-1} y_j \cos(jx_0).$$

Benutzen wir die Identität $\beta_{n/2} = \beta_{n-n/2}$ falls n gerade ist, so folgt für alle anderen Koeffizienten a_ℓ

$$a_\ell = \beta_\ell + \beta_{n-\ell} = \frac{1}{n}\sum_{j=0}^{n-1} y_j\{\omega_n^{-j\ell} + \omega_n^{j(\ell-n)}\} = \frac{1}{n}\sum_{j=0}^{n-1} y_j\{e^{-2\pi ij\ell/n} + e^{2\pi ij(\ell-n)/n}\}$$
$$= \frac{2}{n}\sum_{j=0}^{n-1} y_j \frac{e^{-2\pi ij\ell/n} + e^{2\pi ij\ell/n}}{2} = \frac{2}{n}\sum_{j=0}^{n-1} y_j \cos(jx_\ell).$$

Ganz ähnlich ergibt sich die Behauptung auch im Fall der Koeffizienten b_ℓ:

$$b_\ell = i(\beta_\ell - \beta_{n-\ell}) = \frac{i}{n}\sum_{j=0}^{n-1} y_j\{\omega_n^{-j\ell} - \omega_n^{j(\ell-n)}\} = \frac{i}{n}\sum_{j=0}^{n-1} y_j\{e^{-2\pi ij\ell/n} - e^{2\pi ij(\ell-n)/n}\}$$
$$= -\frac{2}{n}\sum_{j=0}^{n-1} y_j \frac{e^{-2\pi ij\ell/n} - e^{2\pi ij\ell/n}}{2i} = \frac{2}{n}\sum_{j=0}^{n-1} y_j \sin(jx_\ell).$$

□

4.2 Schnelle Fourier-Transformation

Schreiben wir

$$\mathbf{T}_n = \left[\ \boldsymbol{\omega}^0 \,\middle|\, \boldsymbol{\omega}^1 \,\middle|\, \boldsymbol{\omega}^2 \,\middle|\, \ldots \,\middle|\, \boldsymbol{\omega}^{n-1}\ \right] = \begin{bmatrix} \omega_n^0 & \omega_n^0 & \omega_n^0 & \ldots & \omega_n^0 \\ \omega_n^0 & \omega_n^1 & \omega_n^2 & \ldots & \omega_n^{n-1} \\ \omega_n^0 & \omega_n^2 & \omega_n^4 & \ldots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega_n^0 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \ldots & \omega_n^{(n-1)(n-1)} \end{bmatrix}$$

und

$$\mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}, \qquad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{n-1} \end{bmatrix},$$

so bedeutet Satz 4.4, dass

$$\boldsymbol{\beta} = \frac{1}{n}\mathbf{T}_n^\star \mathbf{y}.$$

Da ferner Lemma 4.3 die Beziehung

$$\mathbf{T}_n^\star \mathbf{T}_n = n\mathbf{I}$$

impliziert, folgt umgekehrt

$$\mathbf{y} = \mathbf{T}_n \boldsymbol{\beta}.$$

Definition 4.6 Die Abbildung

$$\mathbf{y} \mapsto \boldsymbol{\beta} = \frac{1}{n}\mathbf{T}_n^\star \mathbf{y}$$

wird **diskrete Fourier-Transformation** genannt. Ihre Umkehrung

$$\boldsymbol{\beta} \mapsto \mathbf{y} = \mathbf{T}_n \boldsymbol{\beta}$$

heißt **Fourier-Synthese**.

Weil $\mathbf{T}_n$ symmetrisch ist, folgt

$$\boldsymbol{\beta} = \frac{1}{n}\mathbf{T}_n^\star \mathbf{y} = \frac{1}{n}\overline{\mathbf{T}_n}\mathbf{y} = \frac{1}{n}\overline{\mathbf{T}_n \overline{\mathbf{y}}}.$$

Daher ist die Fourier-Transformation mit Hilfe der Fourier-Synthese berechenbar, weshalb wir uns im folgenden auf letztere beschränken können. Mathematisch entspricht die Fourier-Synthese der Auswertung eines trigonometrischen Polynoms $p(x)$ an den Stellen $x_k = 2\pi k/n$, denn es ist

$$y_k = p(x_k) = \sum_{\ell=0}^{n-1} \beta_\ell e^{2\pi i k\ell/n} = \sum_{\ell=0}^{n-1} \beta_\ell \omega_n^{k\ell}, \qquad k = 0, 1, \ldots, n-1.$$

Bei einer naiven Anwendung von $\mathbf{T}_n$ sind n^2 Multiplikationen auszuführen. Die *schnelle Fourier-Transformation*, oftmals nur als *FFT* für *Fast Fourier Transform* bezeichnet, entwickelt von Cooley und Tukey (1965), benötigt hingegen nur $(n \log_2 n)/2$ Multiplikationen. Anhand eines Beispiels wollen wir die Vorgehensweise bei der schnellen Fourier-Transformation motivieren.

Beispiel 4.7 Es sei $n = 8$, dann berechnet sich die Fourier-Synthese gemäß $\mathbf{y} = \mathbf{T}_8\boldsymbol{\beta}$. Ausgeschrieben bedeutet dies

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_7 \end{bmatrix} = \begin{bmatrix} \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 \\ \omega_8^0 & \omega_8^1 & \omega_8^2 & \omega_8^3 & \omega_8^4 & \omega_8^5 & \omega_8^6 & \omega_8^7 \\ \omega_8^0 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8^0 & \omega_8^2 & \omega_8^4 & \omega_8^6 \\ \omega_8^0 & \omega_8^3 & \omega_8^6 & \omega_8^1 & \omega_8^4 & \omega_8^7 & \omega_8^2 & \omega_8^5 \\ \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 \\ \omega_8^0 & \omega_8^5 & \omega_8^2 & \omega_8^7 & \omega_8^4 & \omega_8^1 & \omega_8^6 & \omega_8^3 \\ \omega_8^0 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^0 & \omega_8^6 & \omega_8^4 & \omega_8^2 \\ \omega_8^0 & \omega_8^7 & \omega_8^6 & \omega_8^5 & \omega_8^4 & \omega_8^3 & \omega_8^2 & \omega_8^1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_7 \end{bmatrix}.$$

Wir ordnen nun die rechte Seite nach geraden und ungeraden Indizes:

$$\left[\begin{array}{c} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \hline y_4 \\ y_5 \\ y_6 \\ y_7 \end{array}\right] = \left[\begin{array}{cccc|cccc} \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 \\ \omega_8^0 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8^1 & \omega_8^3 & \omega_8^5 & \omega_8^7 \\ \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 & \omega_8^2 & \omega_8^6 & \omega_8^2 & \omega_8^6 \\ \omega_8^0 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^3 & \omega_8^1 & \omega_8^7 & \omega_8^5 \\ \hline \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^4 & \omega_8^4 & \omega_8^4 & \omega_8^4 \\ \omega_8^0 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8^5 & \omega_8^7 & \omega_8^1 & \omega_8^3 \\ \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 & \omega_8^6 & \omega_8^2 & \omega_8^6 & \omega_8^2 \\ \omega_8^0 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^7 & \omega_8^5 & \omega_8^3 & \omega_8^1 \end{array}\right] \left[\begin{array}{c} \beta_0 \\ \beta_2 \\ \beta_4 \\ \beta_6 \\ \hline \beta_1 \\ \beta_3 \\ \beta_5 \\ \beta_7 \end{array}\right].$$

Es ist $\omega_8^4 = e^{\pi i} = -1$ und $\omega_8^{2k} = e^{2\pi ik/4} = \omega_4^k$. Mit $\mathbf{D}_4 = \operatorname{diag}(\omega_8^0, \omega_8^1, \omega_8^2, \omega_8^3)$ können wir daher schreiben

$$\left[\begin{array}{c} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \hline y_4 \\ y_5 \\ y_6 \\ y_7 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{T}_4 & \mathbf{D}_4\mathbf{T}_4 \\ \hline \mathbf{T}_4 & -\mathbf{D}_4\mathbf{T}_4 \end{array}\right] \left[\begin{array}{c} \beta_0 \\ \beta_2 \\ \beta_4 \\ \beta_6 \\ \hline \beta_1 \\ \beta_3 \\ \beta_5 \\ \beta_7 \end{array}\right].$$

Setzen wir

$$\mathbf{c} = \mathbf{T}_4 \begin{bmatrix} \beta_0 \\ \beta_2 \\ \beta_4 \\ \beta_6 \end{bmatrix}, \qquad \mathbf{d} = \mathbf{D}_4\mathbf{T}_4 \begin{bmatrix} \beta_1 \\ \beta_3 \\ \beta_5 \\ \beta_7 \end{bmatrix},$$

so folgt

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \mathbf{c} + \mathbf{d}, \qquad \begin{bmatrix} y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \mathbf{c} - \mathbf{d}.$$

Dies bedeutet, die Fourier-Synthese für $n = 8$ Unbekannte lässt sich aus zwei Fourier-Synthesen für $n/2 = 4$ Unbekannte zusammensetzen. △

Satz 4.8 Zu gegebenem $\boldsymbol{\beta} \in \mathbb{C}^{2n}$ sei

$$\mathbf{D}_n = \operatorname{diag}(\omega_{2n}^0, \omega_{2n}^1, \ldots, \omega_{2n}^{n-1}), \qquad \mathbf{c} = \mathbf{T}_n \begin{bmatrix} \beta_0 \\ \beta_2 \\ \vdots \\ \beta_{2n-2} \end{bmatrix}, \qquad \mathbf{d} = \mathbf{D}_n\mathbf{T}_n \begin{bmatrix} \beta_1 \\ \beta_3 \\ \vdots \\ \beta_{2n-1} \end{bmatrix}.$$

Dann gilt für $\mathbf{y} = \mathbf{T}_{2n}\boldsymbol{\beta} \in \mathbb{C}^{2n}$ die Beziehung

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \mathbf{c} + \mathbf{d}, \qquad \begin{bmatrix} y_n \\ y_{n+1} \\ \vdots \\ y_{2n-1} \end{bmatrix} = \mathbf{c} - \mathbf{d}.$$

Beweis. Nachrechnen liefert für $\ell = 0, 1, 2, \ldots, n-1$

$$\begin{aligned} y_\ell &= \sum_{k=0}^{n-1} \beta_{2k}\omega_n^{k\ell} + \omega_{2n}^\ell \sum_{k=0}^{n-1} \beta_{2k+1}\omega_n^{k\ell} = \sum_{k=0}^{n-1} \beta_{2k}\omega_{2n}^{2k\ell} + \omega_{2n}^\ell \sum_{k=0}^{n-1} \beta_{2k+1}\omega_{2n}^{2k\ell} \\ &= \sum_{k=0}^{n-1} \beta_{2k}\omega_{2n}^{2k\ell} + \sum_{k=0}^{n-1} \beta_{2k+1}\omega_{2n}^{(2k+1)\ell} = \sum_{k=0}^{2n-1} \beta_k\omega_{2n}^{k\ell} \end{aligned}$$

und für $\ell = n, n+1, \ldots, 2n-1$

$$y_\ell = \sum_{k=0}^{n-1} \beta_{2k}\omega_n^{k\ell} - \omega_{2n}^{\ell-n} \sum_{k=0}^{n-1} \beta_{2k+1}\omega_n^{k\ell} = \sum_{k=0}^{n-1} \beta_{2k}\omega_n^{k\ell} + \omega_{2n}^{\ell} \sum_{k=0}^{n-1} \beta_{2k+1}\omega_n^{k\ell} = \sum_{k=0}^{2n-1} \beta_k \omega_{2n}^{k\ell}.$$

□

Zur Berechnung von $\mathbf{y} = \mathbf{T}_{2n}\boldsymbol{\beta} \in \mathbb{C}^{2n}$ werden demnach nur die Vektoren $\mathbf{c}, \mathbf{d} \in \mathbb{C}^n$ benötigt. Dies entspricht einer Divide-and-Conquer-Strategie, da die ursprüngliche Aufgabe für $2n$ Unbekannte durch geschicktes Unterteilen in zwei Unterprobleme mit nur noch n Unbekannten zerlegt wurde. Die vollständige schnelle Fourier-Synthese erhalten wir, wenn wir $\mathbf{c}$ und $\mathbf{d}$ rekursiv mit Hilfe desselben Algorithmus berechnen. Dazu seien $j \in \mathbb{N}$ und $n = 2^j$ eine Zweierpotenz.

Algorithmus 4.9 (schnelle Fourier-Synthese)
input: Koeffizienten $\boldsymbol{\beta} \in \mathbb{C}^n$ eines trigonometrischen Polynoms
output: Funktionsauswertungen $\mathbf{y} \in \mathbb{C}^n$ in den Stützstellen $\{2\pi k/n\}_{k=0}^{n-1}$

Ist $n = 2$, so bilde

$$\mathbf{y} = \mathbf{T}_2\boldsymbol{\beta} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \boldsymbol{\beta}.$$

Andernfalls setze $n := n/2$ und

① berechne

$$\mathbf{c} = \mathbf{T}_n \begin{bmatrix} \beta_0 \\ \beta_2 \\ \vdots \\ \beta_{2n-2} \end{bmatrix}, \qquad \mathbf{d} = \mathbf{T}_n \begin{bmatrix} \beta_1 \\ \beta_3 \\ \vdots \\ \beta_{2n-1} \end{bmatrix}$$

mit Algorithmus 4.9

② für alle $k = 0, 1, \ldots, n-1$ multipliziere $d_k := e^{\pi i k/n} d_k$

③ setze

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \mathbf{c} + \mathbf{d}, \qquad \begin{bmatrix} y_n \\ y_{n+1} \\ \vdots \\ y_{2n-1} \end{bmatrix} = \mathbf{c} - \mathbf{d}.$$

Der Aufwand dieses Algorithmus wird im nachfolgenden Satz ermittelt.

Satz 4.10 Der Aufwand zur Bewältigung der vollständigen schnellen Fourier-Synthese 4.9 lässt sich abschätzen durch $(n \log_2 n)/2$ Multiplikationen.

Beweis. Wir führen den Beweis vermittels Induktion über j. Für den Induktionsanfang $j = 1$ ist keine weitere Unterteilung möglich und wir müssen die volle (2×2)-Matrix anwenden

$$\mathbf{T}_2\boldsymbol{\beta} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \boldsymbol{\beta}.$$

Der Aufwand hierfür lässt sich durch $1/2 \cdot 2 \log_2 2 = 1$ Multiplikation abschätzen. Wir wollen nun annehmen, dass die Behauptung für j richtig ist, wir also nicht mehr als $1/2 \cdot 2^j \log_2 2^j = 1/2 \cdot 2^j j$ Multiplikationen zur Berechnung von $\mathbf{T}_{2^j}\boldsymbol{\beta}$ benötigen. Der Induktionsschritt $j \mapsto j+1$ ergibt sich nun (vgl. Satz 4.8) wie folgt:

$$2 \cdot \frac{1}{2} j 2^j + 2^j = (j+1)2^j = \frac{1}{2}(j+1)2^{j+1}.$$

□

Bemerkung 4.11 Die Reduktion der Komplexität von $\mathcal{O}(n^2)$ auf $\mathcal{O}(n \log_2 n)$ bewirkte eine technische Revolution in der Signalverarbeitung. Erst mit Hilfe der schnellen Fourier-Transformation wurde die digitale Signalverarbeitung überhaupt möglich. △

4.3 Zirkulante Matrizen

Eine wichtige Anwendung der schnellen Fourier-Transformation ergibt sich im Fall von Toeplitz-Matrizen, speziell zirkulanter Matrizen. Derartige Matrizen sind in der Praxis oft anzutreffen, etwa bei der Berechnung der *Faltung* zweier Vektoren. Für $\mathbf{a} = [a_k] \in \mathbb{R}^n$ und $\mathbf{b} = [b_k] \in \mathbb{R}^n$ ist diese definiert durch

$$[\mathbf{a} * \mathbf{b}]_k = \sum_{\ell=0}^{n-1} a_{(k+\ell) \bmod n} b_\ell.$$

Definition 4.12 Eine Matrix $\mathbf{T} \in \mathbb{R}^{n \times n}$ ist eine **Toeplitz-Matrix**, falls ein $\mathbf{t} = [t_{1-n}, t_{2-n}, \ldots, t_{n-2}, t_{n-1}]^T \in \mathbb{R}^{2n-1}$ existiert, so dass

$$\mathbf{T} = \begin{bmatrix} t_0 & t_1 & t_2 & \cdots & t_{n-1} \\ t_{-1} & t_0 & t_1 & \cdots & t_{n-2} \\ t_{-2} & t_{-1} & t_0 & \cdots & t_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{1-n} & t_{2-n} & t_{3-n} & \cdots & t_0 \end{bmatrix}.$$

Eine **zirkulante Matrix** $\mathbf{C} \in \mathbb{R}^{n \times n}$ ist eine Toeplitz-Matrix ist mit $t_k = t_{k-n}$ für alle $k = 1, 2, \ldots, n-1$. Dies bedeutet, es existiert ein $\mathbf{c} = [c_0, c_1, c_2, \ldots, c_{n-1}]^T \in \mathbb{R}^n$ existiert, so dass

$$\mathbf{C} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \cdots & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & \cdots & c_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & c_3 & \cdots & c_0 \end{bmatrix}.$$

Der nachfolgende Satz zeigt, dass sich die Eigenpaare von zirkulanten Matrizen $\mathbf{C}$ durch die Fourier-Transfomation ergeben.

Satz 4.13 Sei $\mathbf{C} \in \mathbb{R}^{n\times n}$ eine zirkulante Matrix und

$$\begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{n-1} \end{bmatrix} = \mathbf{T}_n \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix}.$$

Dann gilt für $\mathbf{D} = \operatorname{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{n-1})$

$$\mathbf{C} = \frac{1}{n}\mathbf{T}_n \mathbf{D} \mathbf{T}_n^\star,$$

dies bedeutet, die $(\lambda_k, \boldsymbol{\omega}^k)$ sind die Eigenpaare von $\mathbf{C}$.

Beweis. Für $j, k = 0, 1, 2, \ldots, n-1$ bezeichne $\left[\mathbf{C}\boldsymbol{\omega}^k\right]_j$ den j-ten Eintrag des Vektors $\mathbf{C}\boldsymbol{\omega}^k$. Setzen wir $c_{n+i} := c_i$, so gilt für diesen

$$\begin{aligned}
\left[\mathbf{C}\boldsymbol{\omega}^k\right]_j &= \sum_{i=0}^{n-1} c_{i-j}\omega_n^{ik} = \omega_n^{jk} \sum_{i=0}^{n-1} c_{i-j}\omega_n^{(i-j)k} \\
&= \omega_n^{jk}\left[\sum_{i=-j}^{-1} \underbrace{c_i\omega_n^{ik}}_{=c_{n+i}\omega_n^{(n+i)k}} + \sum_{i=0}^{n-1-j} c_i\omega_n^{ik}\right] \\
&= \omega_n^{jk} \underbrace{\sum_{i=0}^{n-1} c_i\omega_n^{ik}}_{=\lambda_k}.
\end{aligned}$$

Mit anderen Worten, es ist $\mathbf{C}\boldsymbol{\omega}^k = \lambda_k\boldsymbol{\omega}^k$ für alle $k = 0, 1, \ldots, n-1$. Folglich sind genau alle $(\lambda_k, \boldsymbol{\omega}^k)$ Eigenpaare von $\mathbf{C}$, was äquivalent ist zu

$$\mathbf{C}\mathbf{T}_n = \mathbf{T}_n\mathbf{D}.$$

□

Eine zirkulante Matrix ist folglich durch die Fourier-Transformation leicht diagonalisierbar. Daher ist ein lineares Gleichungssystem $\mathbf{C}\mathbf{x} = \mathbf{b}$ mit Hilfe der schnellen Fouriertransformation gemäß

$$\mathbf{x} = \mathbf{C}^{-1}\mathbf{b} = n\mathbf{T}_n^{-\star}\mathbf{D}^{-1}\mathbf{T}_n^{-1}\mathbf{b} = \frac{1}{n}\mathbf{T}_n\mathbf{D}^{-1}\mathbf{T}_n^\star\mathbf{b}$$

lösbar mit Aufwand $\mathcal{O}(n \log n)$. Ebenso können Matrix-Vektor-Multiplikation in $\mathcal{O}(n \log n)$ Operationen durchgeführt werden:

$$\mathbf{C}\mathbf{x} = \frac{1}{n}\mathbf{T}_n\mathbf{D}\mathbf{T}_n^\star\mathbf{x}.$$

Im Fall einer Toeplitz-Matrix bleibt diese Technik anwendbar, wenn die Toeplitz-Matrix

$\mathbf{T} \in \mathbb{C}^{n\times n}$ in eine zirkulante Matrix $\mathbf{C} \in \mathbb{C}^{2n\times 2n}$ eingebettet wird:

$$\mathbf{C} = \begin{bmatrix} \mathbf{T} & \mathbf{E} \\ \mathbf{E} & \mathbf{T} \end{bmatrix} \quad \text{mit} \quad \mathbf{E} = \begin{bmatrix} 0 & t_{1-n} & t_{2-n} & \cdots & t_{-1} \\ t_{-1} & 0 & t_{1-n} & \cdots & t_{-2} \\ t_{-2} & t_{-1} & 0 & \cdots & t_{-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{1-n} & t_{2-n} & t_{3-n} & \cdots & 0 \end{bmatrix}.$$

Mit Hilfe der schnellen Fourier-Transformation lässt sich nun das Matrix-Vektor-Produkt $\mathbf{Tx}$ effizient vermittels

$$\mathbf{C} \begin{bmatrix} \mathbf{x} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{Tx} \\ \mathbf{Ex} \end{bmatrix}$$

bestimmen, wobei der Aufwand etwa doppelt so hoch wie im Fall einer zirkulanten $(n\times n)$-Matrix ist.

5. Splines

5.1 Spline-Räume

Wie wir in Kapitel 3 gesehen haben, ist die Polynominterpolation generell kein geeignetes numerisches Verfahren, wenn der Polynomgrad groß wird. Deshalb wollen wir im folgenden Polynome niederen Grads möglichst glatt miteinander verkleben.

Definition 5.1 Sei $\triangle = \{x_0, x_1, \ldots, x_n\}$ ein Gitter von $n+1$ paarweise verschiedenen Knoten mit $a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$. Ein **Spline** vom Grad m ist eine $(m-1)$-mal stetig differenzierbare Funktion s, die auf jedem Intervall $[x_{i-1}, x_i]$, $i = 1, 2, \ldots, n$, mit einem Polynom $s_i \in \Pi_m$ übereinstimmt. Den Raum der Splines vom Grad m bezüglich $\triangle$ bezeichnen wir mit $S_m(\triangle)$. Splines vom Grad 1, 2, beziehungsweise 3 werden auch **linear**, **quadratisch**, beziehungsweise **kubisch** genannt.

Sind $s_1, s_2 \in S_m(\triangle)$ zwei Splines und $\alpha, \beta \in \mathbb{R}$, dann gilt offenbar auch $\alpha s_1 + \beta s_2 \in S_m(\triangle)$. Folglich ist $S_m(\triangle)$ ein linearer Vektorraum, und es gilt

$$\Pi_m \subset S_m(\triangle).$$

Splines vom Grad ≥ 3 werden vom Auge als "glatt" empfunden. Deshalb werden kubische Splines besonders oft verwendet.

Wir betrachten zunächst den Fall linearer Splines, da hier die Situation ganz einfach ist. Speziell kann der interpolierende Spline leicht explizit angegeben werden, wie anhand der Illustration in Abbildung 5.1 sofort klar wird.

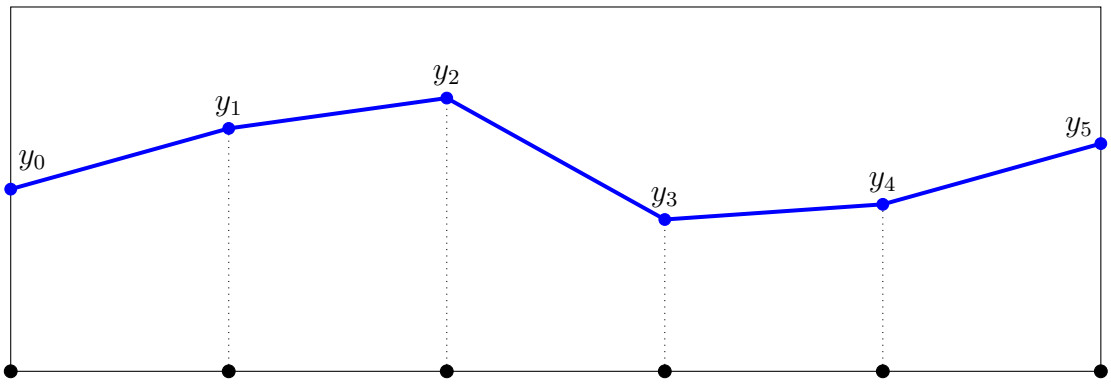


Abbildung 5.1: Linearer Spline auf dem Intervall $[0, 5]$.

Satz 5.2 Zu gegebenen Daten $y_0, y_1, \ldots, y_n$ existiert genau ein linearer Spline $s \in S_1(\triangle)$ mit

$$s(x_i) = y_i, \quad i = 0, 1, \ldots, n. \tag{5.1}$$

Der Vektorraum $S_1(\triangle)$ besitzt die Dimension $\dim S_1(\triangle) = n + 1$.

Beweis. Zu den Daten $y_0, y_1, \ldots, y_n$ konstruieren wir einen linearen Spline durch

$$s(x) := \frac{x_{i+1} - x}{x_{i+1} - x_i} y_i + \frac{x - x_i}{x_{i+1} - x_i} y_{i+1}, \quad x \in [x_i, x_{i+1}].$$

Die so beschriebene Funktion ist offensichtlich stetig und verbindet auf jedem Teilintervall $[x_i, x_{i+1}]$ die beiden Randpunkte (x_i, y_i) und (x_{i+1}, y_{i+1}) durch eine gerade Linie, das heißt, es ist $s \in S_1(\triangle)$. Speziell folgt, dass der Spline $s(x)$ eindeutig ist. Deshalb ist die lineare Abbildung

$$A : S_1(\triangle) \to \mathbb{R}^{n+1}, \qquad s \mapsto \begin{bmatrix} s(x_0) \\ s(x_1) \\ \vdots \\ s(x_n) \end{bmatrix}$$

bijektiv, womit alle Aussagen des Satzes bewiesen sind. □

Mit Hilfe der linearen Splines können wir nun zeigen, dass auch Spline-Räume höherer Ordnung Vektorräume sind.

Satz 5.3 $S_m(\triangle)$ ist ein $(n + m)$-dimensionaler Vektorraum.

Beweis. Für $m = 1$ haben wir die Behauptung im vorhergehenden Satz bereits bewiesen. Sei also $m > 1$ und sei $s_0, s_1, \ldots, s_n$ eine Basis von $S_1(\triangle)$. Weiterhin sei σ_i eine beliebige $(m-1)$-te Stammfunktion von s_i, $i = 0, 1, \ldots, n$. Dann gehören sowohl σ_i als auch die Monome $x^0, x^1, \ldots, x^{m-2}$ zu $S_m(\triangle)$. Wir zeigen, dass

$$\{\sigma_0, \sigma_1, \ldots, \sigma_n\} \cup \{x^0, x^1, \ldots, x^{m-2}\} \tag{5.2}$$

eine Basis von $S_m(\triangle)$ ist.
Ist $s \in S_m(\triangle)$, so gilt $s^{(m-1)} \in S_1(\triangle)$ und deshalb

$$s^{(m-1)}(x) = \sum_{i=0}^{n} c_i s_i(x)$$

für gewisse Koeffizienten $c_i \in \mathbb{R}$. Daraus folgt aber, dass

$$s(x) = \sum_{i=0}^{n} c_i \sigma_i(x) + \sum_{i=0}^{m-2} d_i x^i$$

gilt mit Koeffizienten $d_i \in \mathbb{R}$. Demnach lässt sich jedes $s \in S_m(\triangle)$ als Linearkombination der Vektoren aus (5.2) darstellen. Zum Nachweis der linearen Unabhängigkeit dieser

Vektoren nehmen wir an, es sei

$$\sum_{i=0}^{n} c_i \sigma_i(x) + \sum_{i=0}^{m-2} d_i x^i \equiv 0 \tag{5.3}$$

für Zahlen $c_0, c_1, \ldots, c_n, d_0, d_1, \ldots, d_{m-2} \in \mathbb{R}$. Differenzieren wir diese Gleichung $(m-1)$-mal, dann folgt

$$\sum_{i=0}^{n} c_i s_i(x) \equiv 0.$$

Da $s_0, s_1, \ldots, s_n$ eine Basis von $S_1(\triangle)$ ist, ergibt sich hieraus $c_0 = c_1 = \cdots = c_n = 0$. Aus

$$\sum_{i=0}^{m-2} d_i x^i \equiv 0$$

schließen wir auch $d_0 = d_1 = \cdots = d_{m-2} = 0$. Damit besitzt (5.3) nur die triviale Lösung und der Beweis ist vollständig erbracht. □

5.2 Kubische Splines

Lineare Splines über $\triangle$ haben genau $n+1$ Freiheitsgrade, weshalb das Interpolationsproblem (5.1) eindeutig lösbar ist. Da die Dimension von $S_m(\triangle)$ für $m > 1$ größer als $n+1$ ist, müssen wir zusätzliche Bedingungen an den interpolieren Spline stellen, um Eindeutigkeit zu erzwingen. Für ungerades m benötigen wir eine gerade Anzahl von Zusatzbedingungen, die wir symmetrisch an den Intervallenden verteilen. Obwohl die folgenden Sätze leicht auf beliebige m übertragen werden können, beschränken wir uns der Übersichtlichkeit halber auf den in der Praxis wichtigen Fall der kubischen Splines.

Problem 5.4 (kubische Spline-Interpolation) Gegeben seien ein Gitter $\triangle = \{x_0, x_1, \ldots, x_n\}$ über $[a, b]$ mit $a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$ sowie $n+1$ Werte $y_0, y_1, \ldots, y_n \in \mathbb{R}$. Gesucht ist ein kubischer Spline $s \in S_3(\triangle)$ mit

$$s(x_i) \stackrel{!}{=} y_i, \qquad i = 0, 1, \ldots, n, \tag{5.4}$$

der zusätzlich einer der folgenden Randbedingungen

$$\begin{aligned} s''(a) = s''(b) &= 0 && \text{(natürliche Randbedingungen)} \\ s'(a) = y'_0, \quad s'(b) &= y'_n && \text{(Hermite-Randbedingungen)} \\ s'(a) = s'(b), \quad s''(a) &= s''(b) && \text{(periodische Randbedingungen)} \end{aligned} \tag{5.5}$$

genügt. Dabei gelte im zweiten Fall $y'_0, y'_n \in \mathbb{R}$ sowie im dritten $s(a) = y_0 = y_n = s(b)$.

Bemerkung 5.5 Interpolierende kubische Splines haben eine interessante Optimalitätseigenschaft, die die "Glattheit" betrifft. Für eine Funktion $y : [a, b] \to \mathbb{R}$ ist

$$\kappa(x) := \frac{y''(x)}{\left(1 + y'(x)^2\right)^{3/2}}$$

die *Krümmung* der Kurve $\big(x, y(x)\big)$ für ein gegebenes Argument x. Beschreibt etwa $y(x)$ die Lage einer dünnen Holzlatte, so misst

$$E = \int_a^b |\kappa(x)|^2 \,\mathrm{d}x = \int_a^b \left| \frac{y''(x)}{\big(1 + y'(x)^2\big)^{3/2}} \right|^2 \mathrm{d}x$$

die *Biegeenergie* dieser Latte. Aufgrund des Hamiltonschen Prinzips stellt sich die Latte so ein, dass die Biegeenergie E minimal wird. Für kleine Auslenkungen $y'(x)$ gilt

$$E \approx \int_a^b |y''(x)|^2 \,\mathrm{d}x =: \|y''\|_{L^2}^2.$$

Wie wir gleich sehen werden, beschreibt daher der kubische Spline näherungsweise die Form dieser Holzlatte, falls sie an den Stellen (x_i, y_i), $i = 0, 1, \ldots, n$, fixiert wird und an den Enden

- lose und deshalb gerade ist: $s''(x) = 0$ für $x \le a$ und $x \ge b$ (natürliche Randbedingungen),
- in eine bestimmte Richtung fest eingespannt ist (Hermite-Randbedingungen), oder
- zusammengeklebt ist (periodische Randbedingungen).

△

Satz 5.6 Der kubische Spline $s \in S_3(\triangle)$ interpoliere die Punkte (x_i, y_i), $i = 0, 1, \ldots, n$, und erfülle eine der Randbedingungen aus (5.5). Die Funktion $g \in C^2([a, b])$ sei irgendeine andere interpolierende Funktion, die denselben Randbedingungen genügt. Dann gilt

$$\|g''\|_{L^2}^2 = \|s''\|_{L^2}^2 + \|g'' - s''\|_{L^2}^2, \tag{5.6}$$

insbesondere also

$$\|s''\|_{L^2} \le \|g''\|_{L^2}. \tag{5.7}$$

Beweis. Es gilt

$$\|g''\|_{L^2}^2 = \|s'' + (g'' - s'')\|_{L^2}^2 = \|s''\|_{L^2}^2 + \|g'' - s''\|_{L^2}^2 + 2\int_a^b s''(g'' - s'') \,\mathrm{d}x.$$

Wir müssen also zeigen, dass hierin der letzte Term verschwindet.
Da s und g nach Voraussetzung dieselben Randbedingungen erfüllen, gilt die Identität

$$s''(a)\big(g'(a) - s'(a)\big) = s''(b)\big(g'(b) - s'(b)\big). \tag{5.8}$$

Ferner folgt mit Hilfe partieller Integration

$$\begin{aligned}\int_a^b s''(g'' - s'') \,\mathrm{d}x &= \sum_{i=1}^n \int_{x_{i-1}}^{x_i} s''(g'' - s'') \,\mathrm{d}x \\ &= \sum_{i=1}^n \left\{ s''(g' - s')\Big|_{x_{i-1}}^{x_i} - \int_{x_{i-1}}^{x_i} s'''(g' - s') \,\mathrm{d}x \right\}.\end{aligned}$$

Da s''' im Innern von $[x_{i-1}, x_i]$ konstant ist, etwa gleich c_i, ergibt sich also

$$\int_a^b s''(g''-s'')\,\mathrm{d}x = \sum_{i=1}^n s''(g'-s')\Big|_{x_{i-1}}^{x_i} - \sum_{i=1}^n c_i \int_{x_{i-1}}^{x_i} (g'-s')\,\mathrm{d}x$$
$$= \underbrace{s''(g'-s')\Big|_a^b}_{=0 \text{ wegen (5.8)}} - \sum_{i=1}^n c_i \underbrace{(g-s)\Big|_{x_{i-1}}^{x_i}}_{=0} = 0,$$

da g und s die gleichen Daten interpolieren. □

Satz 5.7 Die durch (5.4) und (5.5) bestimmte kubische Spline-Interpolationsaufgabe ist eindeutig lösbar.

Beweis. Wie im Satz 5.3 gezeigt wurde, ist $S_3(\triangle)$ ein $(n+3)$-dimensionaler Vektorraum. Wählen wir eine Basis $\{s_0, s_1, \ldots, s_{n+2}\} \in S_3(\triangle)$ und machen den Ansatz $s(x) = \sum_{i=0}^{n+2} c_i s_i(x)$, dann erhalten wir das folgende lineare Gleichungssystem aus den Interpolationsbedingungen (5.4):

$$\sum_{i=0}^{n+2} c_i s_i(x_j) = y_j \quad \text{für } j = 0, \ldots, n.$$

Jede der Randbedingungen aus (5.5) liefert uns zwei zusätzliche Gleichungen:

$$\sum_{i=0}^{n+2} c_i s_i''(x_0) = 0, \ \sum_{i=0}^{n+2} c_i s_i''(x_n) = 0 \qquad \text{(natürliche Randbedingungen)}$$
$$\sum_{i=0}^{n+2} c_i s_i'(x_0) = y_0', \ \sum_{i=0}^{n+2} c_i s_i'(x_n) = y_n' \qquad \text{(Hermite-Randbedingungen)}$$
$$\sum_{i=0}^{n+2} c_i \big[s_i'(x_0) - s_i'(x_n)\big] = 0, \ \sum_{i=0}^{n+2} c_i \big[s_i''(x_0) - s_i''(x_n)\big] = 0 \qquad \text{(periodische Randbedingungen)}$$

Wir haben also für die $n+3$ Koeffizienten $c_0, c_1, \ldots, c_{n+2}$ immer genau $n+3$ Gleichungen. Dieses lineare Gleichungssystem ist eindeutig lösbar, wenn wir zeigen können, dass im Fall homogener Daten nur die triviale Lösung $s \equiv 0$, sprich $c_0 = c_1 = \cdots = c_{n+2} = 0$, existiert. Indem wir $g \equiv 0$ wählen, besagt Satz 5.6, dass jeder kubische Spline $s \in S_3(\triangle)$, der die Interpolationsbedingungen $(x_0, 0), \ldots, (x_n, 0)$ und eine der Randbedingungen in (5.5) erfüllt, der Abschätzung

$$0 \le \|s''\|_{L^2} \le \|g''\|_{L^2} = 0$$

genügt. Hierbei nehmen wir im Fall von Hermite-Randbedingungen zusätzlich $y_0' = y_n' = 0$ an. Aus $\|s''\|_{L^2} = 0$ folgt insbesondere $s'' = 0$ wegen der Stetigkeit von s''. Daraus ergibt sich, dass $s \in \Pi_1$ gilt. Wegen $s(a) = s(b) = 0$ schließen wir aber sofort $s \equiv 0$ aufgrund der Eindeutigkeit der Polynominterpolation. □

5.3 B-Splines

Wir führen nun eine Basis in den Spline-Räumen $S_m(\triangle)$ ein, die zur numerischen Berechnung interpolierender Splines genutzt werden kann. Dabei wollen wir uns der Einfachheit halber auf äquidistante Gitter beschränken.

Satz 5.8 Die durch

$$B_0(x) := \begin{cases} 1, & -0.5 \le x < 0.5, \\ 0, & \text{sonst}, \end{cases}$$

und

$$B_{m+1}(x) := \int_{x-1/2}^{x+1/2} B_m(t)\,\mathrm{d}t, \qquad x \in \mathbb{R}, \qquad m = 0, 1, 2, \ldots \tag{5.9}$$

rekursiv definierten Funktionen sind Splines vom Grad m auf dem Gitter

$$\triangle_m := \left\{ i - \frac{m+1}{2} : i = 0, 1, \ldots, m+1 \right\}.$$

Sie heißen *B-Splines* vom Grad m, sind nichtnegativ und erfüllen $B_m(x) = 0$ für $|x| > (m+1)/2$.

Beweis. Wir beweisen die Behauptung durch Induktion nach m. Da für $m = 0$ die Aussage klar ist, nehmen wir an, dass die Behauptung für $m \ge 0$ erfüllt ist. Dann folgt aus (5.9), dass $B_{m+1}(x) \ge 0$ ist für alle x, da $B_m(x) \ge 0$ ist für alle x, und $B_{m+1}(x) = 0$ ist für alle $|x| > (m+2)/2$, da $B_m(x) = 0$ ist für alle $|x| > (m+1)/2$. Außerdem ist

$$B'_{m+1}(x) = B_m\left(x + \frac{1}{2}\right) - B_m\left(x - \frac{1}{2}\right).$$

Aufgrund der Induktionsannahme ist daher $B'_{m+1}(x)$ überall mindestens $(m-1)$-mal stetig differenzierbar und auf allen Intervallen $[x - 1/2, x + 1/2]$ mit $x \in \triangle_m$ ein Polynom vom Grad $\le m$. Daher ist B_{m+1} ein Spline vom Grad $m + 1$. □

Wie man leicht anhand der Rekursionsformel (5.9) nachrechnet, ist der lineare B-Spline gegeben durch

$$B_1(x) = \begin{cases} 1 - |x|, & |x| \le 1, \\ 0, & |x| > 1, \end{cases}$$

der quadratische B-Spline durch

$$B_2(x) = \frac{1}{2} \begin{cases} 2 - (|x| - 0.5)^2 - (|x| + 0.5)^2, & |x| \le 0.5, \\ (|x| - 1.5)^2, & 0.5 < |x| \le 1.5, \\ 0, & |x| > 1.5, \end{cases}$$

und der kubische B-Spline durch

$$B_3(x) = \frac{1}{6} \begin{cases} (2 - |x|)^3 - 4(1 - |x|)^3, & |x| \le 1, \\ (2 - |x|)^3, & 1 < |x| \le 2, \\ 0, & |x| > 2, \end{cases}$$

vergleiche Abbildung 5.2.

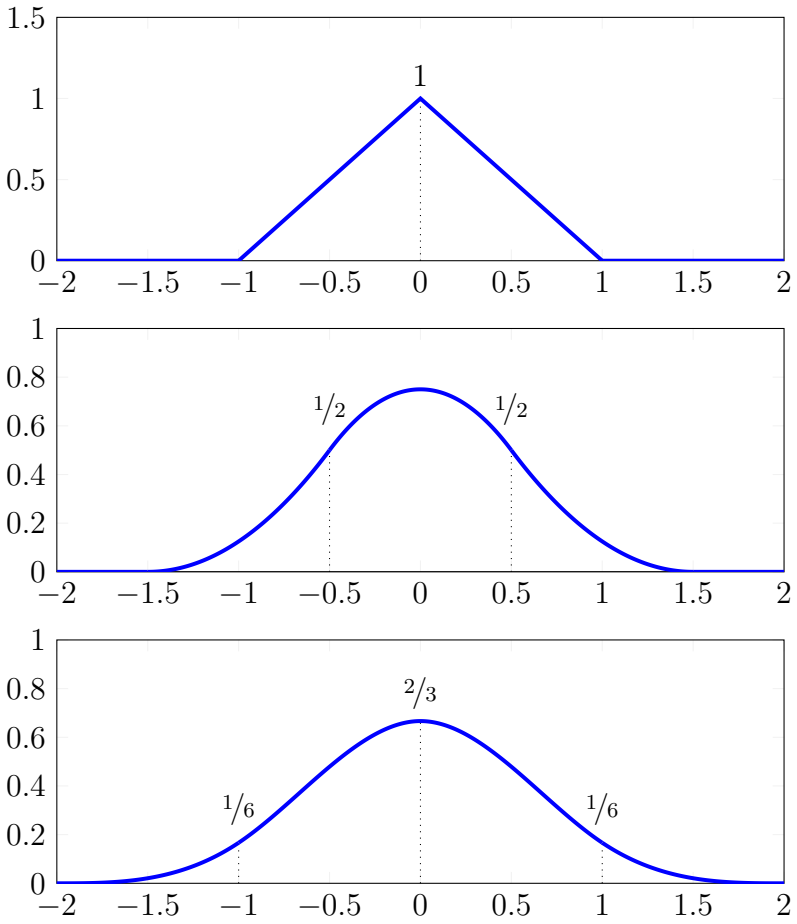


Abbildung 5.2: Linearer, quadratischer und kubischer B-Spline.

Satz 5.9 Für $m = 0, 1, 2, \ldots$ sind die B-Splines

$$B_m(\cdot - i), \qquad i = 0, 1, \ldots, m$$

linear unabhängig auf dem Intervall $I_m := [(m-1)/2, (m+1)/2]$.

Beweis. Für $m = 0$ ist die Behauptung offensichtlich. Wir nehmen also an, dass die Behauptung wahr ist für $m - 1$. Zu zeigen ist, dass

$$\sum_{i=0}^{m} c_i B_m(x - i) = 0, \qquad x \in I_m \tag{5.10}$$

nur gilt, falls $c_0 = c_1 = \cdots = c_m = 0$. Differenzieren von (5.10) liefert

$$\sum_{i=0}^{m} c_i \left\{ B_{m-1}\left(x - i + \frac{1}{2}\right) - B_{m-1}\left(x - i - \frac{1}{2}\right) \right\} = 0, \qquad x \in I_m.$$

Wegen $B_{m-1}(x) = 0$ für alle $|x| \geq m/2$, folgt

$$B_{m-1}\left(x + \frac{1}{2}\right) = B_{m-1}\left(x - m - \frac{1}{2}\right) = 0, \qquad x \in I_m.$$

Daher können wir die letzte Gleichung wie folgt umformen:

$$\sum_{i=1}^{m}(c_i - c_{i-1})B_{m-1}\left(x - i + \frac{1}{2}\right) = 0, \qquad x \in I_m.$$

Aus der Induktionsannahme folgt nun $c_i = c_{i-1}$ für alle $i = 1, 2, \ldots, m$, dies bedeutet $c_0 = c_1 = \cdots = c_m =: c$. Daher gilt nach (5.10)

$$c\sum_{i=0}^{m} B_m(x - i) = 0, \qquad x \in I_m.$$

Durch Integration dieser Gleichung über dem Interval I_m erhalten wir

$$c\int_{(m-1)/2}^{(m+1)/2}\sum_{i=0}^{m} B_m(x-i)\,\mathrm{d}x = c\sum_{i=0}^{m}\int_{(m-1)/2-i}^{(m+1)/2-i} B_m(x)\,\mathrm{d}x = c\int_{-(m+1)/2}^{(m+1)/2} B_m(x)\,\mathrm{d}x = 0.$$

Dies impliziert schließlich $c = 0$, da B_m positiv ist. □

Korollar 5.10 Sei $\triangle = \{x_0, x_1, \ldots, x_n\}$ ein äquidistantes Gitter mit Gitterweite $h > 0$, das heißt $x_i := x_0 + hi$, und sei $m = 2\ell - 1$ mit $\ell \in \mathbb{N}$. Dann bilden die B-Splines

$$B_{m,i}(x) := B_m\left(\frac{x - x_i}{h}\right), \qquad x \in [x_0, x_n]$$

für $i = 1 - \ell, 2 - \ell, \ldots, n + \ell - 1$ eine Basis von $S_m(\triangle)$.

Beweis. Nach Satz 5.8 liegen die $n + m$ Funktionen $B_{m,i}$ alle in $S_m(\triangle)$. Nach Satz 5.3 müssen wir daher lediglich zeigen, dass sie linear unabhängig sind. Dies folgt aber durch Anwendung von Satz 5.9 auf jedes Teilintervall. □

Mit Hilfe von B-Splines können interpolierende Splines effizient berechnet werden. Wir demonstrieren dies am Beipiel linearer und kubischer Splines:

Beispiel 5.11 Der lineare Spline

$$s(x) = \sum_{i=0}^{n} y_i B_1\left(\frac{x - x_i}{h}\right), \quad x \in [x_0, x_n],$$

interpoliert die Werte (x_i, y_i), $i = 0, 1, \ldots, n$. △

Beispiel 5.12 Es gilt

$$B_3(0) = \frac{2}{3}, \quad B_3(\pm 1) = \frac{1}{6}, \quad B_3'(0) = 0, \quad B_3'(\pm 1) = \mp\frac{1}{2}, \quad B_3''(0) = -2, \quad B_3''(\pm 1) = 1.$$

Daher erfüllt der kubische Spline

$$s(x) = \sum_{i=-1}^{n+1} c_i B_3\left(\frac{x - x_i}{h}\right), \quad x \in [x_0, x_n],$$

genau dann die Interpolationsbedingungen (5.4), falls je nach der ausgewählten Randbedingung (5.5) nachfolgendes Gleichungssystem erfüllt ist:

1. natürliche Randbedingungen:

$$\begin{bmatrix} 1 & -2 & 1 & & & & \\ 1/6 & 2/3 & 1/6 & & & & \\ & 1/6 & 2/3 & 1/6 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1/6 & 2/3 & 1/6 & \\ & & & 1 & -2 & 1 & \end{bmatrix} \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \\ \vdots \\ c_n \\ c_{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ y_0 \\ y_1 \\ \vdots \\ y_n \\ 0 \end{bmatrix}$$

2. Hermite-Randbedingungen:

$$\begin{bmatrix} -1/2 & 0 & 1/2 & & & & \\ 1/6 & 2/3 & 1/6 & & & & \\ & 1/6 & 2/3 & 1/6 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1/6 & 2/3 & 1/6 & \\ & & & -1/2 & 0 & 1/2 & \end{bmatrix} \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \\ \vdots \\ c_n \\ c_{n+1} \end{bmatrix} = \begin{bmatrix} hy_0' \\ y_0 \\ y_1 \\ \vdots \\ y_n \\ hy_n' \end{bmatrix}$$

3. periodischer Randbedingungen: unter der Voraussetzung $y_0 = y_n$ können wir hier aufgrund der Periodizität B_{i-1} identifizieren mit B_{i+n-1}, $i = 0, 1, 2$, dies bedeutet $c_{i+n-1} = c_{i-1}$ für $i = 0, 1, 2$. Folglich gilt

$$\begin{bmatrix} 1/6 & 2/3 & 1/6 & & & & \\ & 1/6 & 2/3 & 1/6 & & & \\ & & & \ddots & & & \\ & & & 1/6 & 2/3 & 1/6 & \\ 1/6 & & & & 1/6 & 2/3 & \\ 2/3 & 1/6 & & & & 1/6 & \end{bmatrix} \begin{bmatrix} c_{-1} \\ c_0 \\ \vdots \\ c_{n-2} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}.$$

△

5.4 Interpolationsfehler

Es sei $\triangle = \{x_0, x_1, \ldots, x_n\}$ ein Gitter mit $a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$. Mit $h_i := x_{i+1} - x_i$, $i = 0, 1, \ldots, n-1$, bezeichnet $h = \max_{i=0,1,\ldots,n-1} h_i$ die Gitterweite. Wir zeigen zunächst, dass die lineare Splineinterpolation eine Approximation zweiter Ordnung liefert. Dazu führen wir den Interpolationsprojektor

$$L_1 : C([a,b]) \to S_1(\triangle), \qquad f \mapsto s$$

ein, wobei $s \in S_1(\triangle)$ gemäß Satz 5.2 durch die Interpolationsbedingungen $s(x_i) = f(x_i)$, $i = 0, 1, \ldots, n$, eindeutig bestimmt ist.

Satz 5.13 Für $f \in C^2([a,b])$ gilt

$$\|f - L_1 f\|_{L^2} \leq \frac{h^2}{2} \|f''\|_{L^2}. \tag{5.11}$$

Beweis. Die Funktion $g := f - L_1 f$ besitzt die Nullstellen $x_0, x_1, \ldots, x_n$. Daher gilt

$$\int_{x_i}^{x_{i+1}} |g(x)|^2 \,\mathrm{d}x = \int_{x_i}^{x_{i+1}} \left| \int_{x_i}^{x} g'(t) \cdot 1 \,\mathrm{d}t \right|^2 \mathrm{d}x,$$

woraus mit Hilfe der Cauchy-Schwarzen Ungleichung folgt

$$\begin{aligned}
\int_{x_i}^{x_{i+1}} |g(x)|^2 \,\mathrm{d}x &\le \int_{x_i}^{x_{i+1}} \left(\int_{x_i}^{x} 1 \,\mathrm{d}t \right) \left(\int_{x_i}^{x} |g'(t)|^2 \,\mathrm{d}t \right) \mathrm{d}x \\
&= \int_{x_i}^{x_{i+1}} (x - x_i) \left(\int_{x_i}^{x} |g'(t)|^2 \,\mathrm{d}t \right) \mathrm{d}x \\
&\le \left(\int_{x_i}^{x_{i+1}} |g'(t)|^2 \,\mathrm{d}t \right) \int_{x_i}^{x_{i+1}} (x - x_i) \,\mathrm{d}x \\
&= \frac{h_i^2}{2} \int_{x_i}^{x_{i+1}} |g'(t)|^2 \,\mathrm{d}t.
\end{aligned}$$

Durch Summation über i erhalten wir die Abschätzung

$$\|f - L_1 f\|_{L^2} \le \frac{h}{\sqrt{2}} \|(f - L_1 f)'\|_{L^2}. \tag{5.12}$$

Man beachte, dass der Ausdruck auf der rechten Seite formal zu verstehen ist, denn $f - L_1 f$ ist nur stückweise differenzierbar. Dies bedeutet, die Funktion $(f - L_1 f)'$ ist keine Ableitung sondern setzt sich zusammen aus intervallweise berechneten Ableitungen.
Weiter liefert partielle Integration

$$\begin{aligned}
\|(f - L_1 f)'\|_{L^2}^2 &= \sum_{i=0}^{n-1} \Bigg\{ \underbrace{(f - L_1 f)'(x)(f - L_1 f)(x)\Big|_{x_i}^{x_{i+1}}}_{=0} \\
&\qquad - \int_{x_i}^{x_{i+1}} (f - L_1 f)''(x)(f - L_1 f)(x) \,\mathrm{d}x \Bigg\} \\
&= -\sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f''(x)(f - L_1 f)(x) \,\mathrm{d}x.
\end{aligned}$$

Mit Hilfe der Cauchy-Schwarzschen Ungleichung und (5.12) folgt hieraus

$$\|(f - L_1 f)'\|_{L^2}^2 \le \|f - L_1 f\|_{L^2} \|f''\|_{L^2} \le \frac{h}{\sqrt{2}} \|(f - L_1 f)'\|_{L^2} \|f''\|_{L^2}.$$

Division durch $\|(f - L_1 f)'\|_{L^2}$ liefert

$$\|(f - L_1 f)'\|_{L^2} \le \frac{h}{\sqrt{2}} \|f''\|_{L^2},$$

was in Kombination mit (5.12) die Behauptung ergibt. □

Wir führen nun den kubischen Spline-Interpolationsprojektor

$$L_3 : C([a, b]) \to S_3(\triangle), \qquad f \mapsto s$$

ein, der durch die Interpolationsbedingungen $s(x_i) = f(x_i)$, $i = 0, 1, \ldots, n$, und eine der Randbedingungen aus (5.5) eindeutig bestimmt ist.

Satz 5.14 Für $f \in C^4([a,b])$ gilt

$$\|f - L_3 f\|_{L^2} \leq \frac{h^4}{4} \|f^{(4)}\|_{L^2}.$$

Beweis. Da $L_1(f - L_3 f) = 0$ gilt, erhalten wir mit (5.11)

$$\|f - L_3 f\|_{L^2} = \|(f - L_3 f) - L_1(f - L_3 f)\|_{L^2} \leq \frac{h^2}{2} \|f'' - (L_3 f)''\|_{L^2}.$$

Wir wählen ein $s \in S_3(\triangle)$ mit $s'' = L_1(f'')$ und definieren $g := f - s$. Dabei können wir s so wählen, dass auch die Randbedingungen erfüllt sind, da beim zweimaligem Integrieren von $L_1(f'')$ zwei Konstanten frei wählbar sind.
Es gilt

$$\begin{aligned} \|f'' - (L_3 f)''\|_{L^2}^2 &= \|f'' - s'' - (L_3 f)'' + s''\|_{L^2}^2 \overset{L_3 s = s}{=} \|g'' - (L_3 g)''\|_{L^2}^2 \\ &\leq \|g'' - (L_3 g)''\|_{L^2}^2 + \|(L_3 g)''\|_{L^2}^2 \overset{(5.6)}{=} \|g''\|_{L^2}^2 \\ &= \|f'' - s''\|_{L^2}^2 = \|f'' - L_1(f'')\|_{L^2}^2 \end{aligned}$$

und eine erneute Anwendung von (5.11) auf f'' ergibt

$$\|f'' - (L_3 f)''\|_{L^2} \leq \|f'' - L_1 f''\|_{L^2} \leq \frac{h^2}{2} \|f^{(4)}\|_{L^2}.$$

Durch Kombination mit der ersten Ungleichung erhalten wir die Behauptung. □

6. Numerische Quadratur

6.1 Trapezregel

Wir wollen bestimmte Integrale

$$I[f] = \int_a^b f(x)\,\mathrm{d}x, \tag{6.1}$$

numerisch approximieren, die nicht in geschlossener Form durch Angabe einer Stammfunktion integriert werden können. Das bestimmt einfachste und oft auch schon hinreichend gute Verfahren hierfür ist die sogenannte *Trapezregel*

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{2}\big(f(a)+f(b)\big), \tag{6.2}$$

deren einfache geometrische Interpretation in Abbildung 6.1 dargestellt ist.

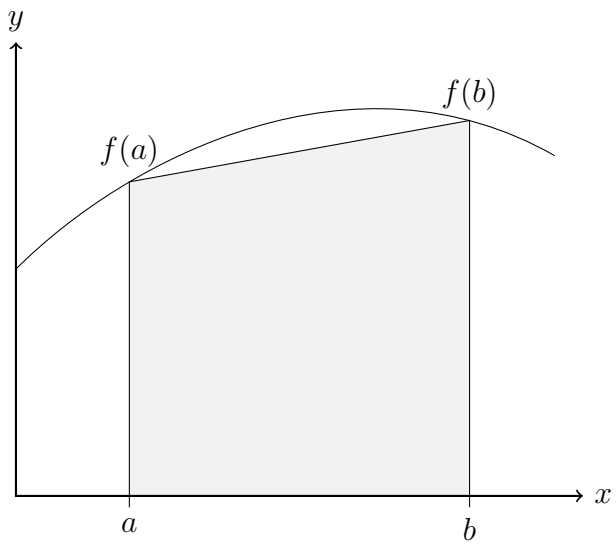


Abbildung 6.1: Geometrische Interpretation der Trapezregel.

Natürlich hat (6.2) in der Regel einen (beliebig großen) festen Fehler. Daher zerlegt man in der Praxis das Intervall $[a, b]$ in N gleichgroße Teilintervalle und wendet (6.2) auf jedes

Teilintervall an. Dieses Verfahren nennt man die *zusammengesetzte Trapezregel*:

$$\begin{aligned} &a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b, \qquad x_i = a + ih, \qquad h = \frac{b-a}{N}, \\ &T_N[f] := \sum_{i=1}^{N} \frac{x_i - x_{i-1}}{2}\big(f(x_i) + f(x_{i-1})\big) = \frac{h}{2}f(a) + h\sum_{i=1}^{N-1} f(x_i) + \frac{h}{2}f(b). \end{aligned} \tag{6.3}$$

Man macht sich leicht mit der Definition des Riemann-Integrals klar, dass $T_N[f] \to I[f]$ für $N \to \infty$, falls f über $[a,b]$ Riemann-integrierbar ist. Unter Zusatzannahmen kann sogar folgende Fehlerabschätzung bewiesen werden:

Satz 6.1 Sei $f \in C^2([a,b])$. Dann gilt mit $h = (b-a)/N$

$$|I[f] - T_N[f]| \le \frac{b-a}{12}h^2\|f''\|_{C([a,b])}.$$

Beweis. Betrachte zunächst die Näherung (6.2), also $N = 1$. Wie durch Abbildung 6.1 deutlich wird, ist

$$T_1[f] = \frac{b-a}{2}\big(f(a) + f(b)\big) = \int_a^b p(x)\,\mathrm{d}x,$$

wobei p das lineare Polynom

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

ist, also insbesondere f in den Punkten a und b interpoliert. Gemäß Satz 3.5 folgt deshalb

$$|f(x) - p(x)| = \left|\frac{f''(\xi)}{2}(x-a)(x-b)\right| \le \frac{1}{2}\|f''\|_{C([a,b])}(x-a)(b-x).$$

Folglich ist

$$|I[f] - T_1[f]| = \left|\int_a^b \big(f(x) - p(x)\big)\,\mathrm{d}x\right| \le \frac{\|f''\|_{C([a,b])}}{2}\int_a^b (x-a)(b-x)\,\mathrm{d}x.$$

Das Integral lässt sich mit Hilfe der Substitution $x := a + (b-a)t$ berechnen:

$$\int_a^b (x-a)(b-x)\,\mathrm{d}x = (b-a)^3\int_0^1 t(1-t)\,\mathrm{d}t = (b-a)^3\Big(\frac{t^2}{2} - \frac{t^3}{3}\Big)\Big|_0^1 = \frac{(b-a)^3}{6}.$$

Dies oben eingesetzt liefert für die Trapezregel auf dem Intervall $[a,b]$ die Fehlerabschätzung

$$|I[f] - T_1[f]| \le \frac{1}{12}\|f''\|_{C([a,b])}(b-a)^3.$$

Angewandt auf (6.3) ergibt sich in Anbetracht von $h = (b-a)/N$ schließlich

$$\begin{aligned} |I[f] - T_N[f]| &\le \sum_{i=1}^{N}\left|\int_{x_{i-1}}^{x_i} f(x)\,\mathrm{d}x - \frac{h}{2}\big(f(x_{i-1}) + f(x_i)\big)\right| \\ &\le \sum_{i=1}^{N}\frac{1}{12}h^3\|f''\|_{C([x_i,x_{i+1}])} \le \frac{N}{12}h^3\|f''\|_{C([a,b])} = \frac{b-a}{12}h^2\|f''\|_{C([a,b])}. \end{aligned}$$

□

Die Trapezregel ist nur ein natürlich konkretes Beispiel für eine *Quadraturformel* zur Berechnung von Integralen der Form (6.1). Für die Entwicklung weiterer Quadraturformeln wollen wir diesen Begriff daher zunächst ganz allgemein einführen.

Definition 6.2 Wir sprechen von einer **Quadraturformel**

$$Q[f] = \sum_{i=1}^{m} w_i f(x_i)$$

mit **Knoten** x_i und **Gewichten** w_i, wenn die Wahl von m, $\{x_i\}$ und $\{w_i\}$ fix ist. Unter der dazugehörigen **zusammengesetzten Quadraturformel** $Q_N[f]$ verstehen wir dann die Unterteilung von $[a, b]$ in N gleich große Teilintervalle, in denen jeweils die Quadraturformel angewandt wird.

Die Genauigkeit einer Quadraturformel hängt davon ab, wie gut Polynome integriert werden können. Im Zusammenhang mit zusammengesetzten Quadraturformeln steht dagegen die Asymptotik $N \to \infty$ im Vordergrund. Um die qualitativen Merkmale einer (zusammengesetzten) Quadraturformel beschreiben zu können, benötigen wir daher folgende Definition:

Definition 6.3 (a) Eine Quadraturformel $Q[f]$ besitzt den **Exaktheitsgrad** q, falls

$$Q[p] = I[p] \quad \text{für alle } p \in \Pi_q.$$

(b) Eine zusammengesetzte Quadraturformel konvergiert gegen $I[f]$ mit der **Ordnung** s, falls

$$|Q_N[f] - I[f]| = \mathcal{O}(N^{-s}), \quad N \to \infty.$$

Beachte: Für den Exaktheitsgrad reicht es, eine Basis von Π_q zu untersuchen, da sowohl $Q[\cdot]$ als auch $I[\cdot]$ lineare Abbildungen sind. Weiterhin konvergiert offensichtlich eine zusammengesetzte Quadraturformel mit der Ordnung $s = q + 1$, falls ihr eine Quadraturformel mit dem Exaktheitsgrad q zugrundeliegt.

Beispiel 6.4 Die Trapezregel hat Exaktheitsgrad $q = 1$. Die zusammengesetzte Trapezregel konvergiert mit Ordnung $s = 2$. △

6.2 Newton-Cotes-Formeln

Mit Hilfe der Polynominterpolation lassen sich leicht Quadraturformeln für $I[f]$ mit beliebigen Exaktheitsgrad q angeben. Seien $x_0 < x_1 < \cdots < x_m$ $m + 1$ vorgegebene Knoten in $[a, b]$ und

$$w_i := \int_a^b L_i(x)\,\mathrm{d}x \tag{6.4}$$

das Integral des i-ten zugehörigen Lagrange-Grundpolynoms. Dann gilt:

Proposition 6.5 Die Quadraturformel Q mit Knoten $\{x_i\}$ und Gewichten $\{w_i\}$ gemäß (6.4) hat den Exaktheitsgrad $q = m$.

Beweis. Sei $p \in \Pi_m$. Offensichtlich interpoliert p sich selbst. Wegen der Eindeutigkeit des Interpolationspolynoms gilt daher

$$p(x) = \sum_{i=0}^{m} p(x_i) L_i(x),$$

vergleiche Satz 3.3. Daraus folgt

$$\begin{aligned} I[p] &= \int_a^b p(x)\,\mathrm{d}x = \int_a^b \sum_{i=0}^{m} p(x_i) L_i(x)\,\mathrm{d}x \\ &= \sum_{i=0}^{m} p(x_i) \int_a^b L_i(x)\,\mathrm{d}x \overset{(6.4)}{=} \sum_{i=0}^{m} w_i p(x_i) = Q[p], \end{aligned}$$

was zu zeigen war. □

Außerdem erhalten wir aus Satz 3.5 folgende Fehlerabschätzung:

$$|I[f] - Q[f]| \le \frac{M_{m+1}}{(m+1)!} \int_a^b |w(x)|\,\mathrm{d}x. \tag{6.5}$$

Hierbei ist $M_{m+1} = \max_{a \le x \le b} |f^{(m+1)}(x)|$ und $w(x)$ das Knotenpolynom aus Definition 3.2.

Beispiel 6.6 Beschränken wir uns auf den Fall äquidistanter Knoten $a = x_0 < x_1 < \cdots < x_{m-1} < x_m = b$, dann erhalten wir die *Newton-Cotes-Formeln*. Im Fall $m = 1$ erhält man speziell die Trapezregel (6.2). Für $m = 2$ ergibt sich die *Simpson-Regel*

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right).$$

△

Beachte: Aus $1 \in \Pi_0$ folgt

$$\int_a^b 1\,\mathrm{d}x = \sum_{i=0}^{m} w_i \underbrace{f(x_i)}_{=1} = \sum_{i=0}^{m} w_i.$$

Daher gilt immer

$$\sum_{i=0}^{m} w_i = b - a$$

für die Formeln mit Exaktheitsgrad $q \ge 0$ (insbesondere also für Newton-Cotes-Formeln).

Beispiel 6.7 Da die Polynominterpolation nur bedingt eine gute Approximation an f liefert (vgl. Kapitel 3), macht es im allgemeinen nur wenig Sinn den Approximationsparameter m hochzuschrauben. Zudem treten ab $m = 7$ negative Gewichte und dadurch unter Umständen Stabilitätsverluste selbst bei positiven Integranden auf. Die Gewichte der Newton-Cotes-Formeln für $m \leq 6$ sind in nachfolgender Tabelle zusammengefasst:

m	Gewichte	$Q[f] - \int_0^{mh} f(x)\,\mathrm{d}x$	Name
1	$\frac{1}{2}$ $\frac{1}{2}$	$\frac{1}{12}h^3M_2$	Trapezregel
2	$\frac{1}{6}$ $\frac{4}{6}$ $\frac{1}{6}$	$\frac{1}{90}h^5M_4$	Simpson-Regel
3	$\frac{1}{8}$ $\frac{3}{8}$ $\frac{3}{8}$ $\frac{1}{8}$	$\frac{3}{80}h^5M_4$	3/8-Regel
4	$\frac{7}{90}$ $\frac{32}{90}$ $\frac{12}{90}$ $\frac{32}{90}$ $\frac{7}{90}$	$\frac{8}{945}h^7M_6$	Milne-Regel
5	$\frac{19}{288}$ $\frac{75}{288}$ $\frac{50}{288}$ $\frac{50}{288}$ $\frac{75}{288}$ $\frac{19}{288}$	$\frac{275}{12096}h^7M_6$	—
6	$\frac{41}{840}$ $\frac{216}{840}$ $\frac{27}{840}$ $\frac{272}{840}$ $\frac{27}{840}$ $\frac{216}{840}$ $\frac{41}{840}$	$\frac{9}{1400}h^9M_8$	Weddle-Regel

△

Um bei gegebener Quadraturformel eine genauere Approximation an ein Integral zu erhalten, unterteilt man das Intervall — wie bereits in Abschnitt 6.1 gesehen — und verwendet die zugehörige zusammengesetzte Newton-Cotes-Formel.

Im Fall $m = 2$ erhält man so die *zusammengesetzte Simpson-Regel*

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{3}\big\{f(a) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{2N-2}) + 4f(x_{2N-1}) + f(b)\big\}$$

mit $x_i = a + ih$, $i = 1, 2, \ldots, 2N-1$, und $h = (b-a)/(2N)$.

Wir benötigen das folgende Hilfsresultat:

Lemma 6.8 Sei $Q[f] = \sum_{i=0}^m w_i f(x_i)$ eine Quadraturformel für $\int_a^b f(x)\,\mathrm{d}x$ mit zu $(a+b)/2$ symmetrischen Knoten und Gewichten. Ist $Q[p] = I[p]$ für alle Polynome $p \in \Pi_{2q}$, dann besitzt $Q[\cdot]$ mindestens den Exaktheitsgrad $2q+1$.

Beweis. Betrachte die Basis

$$\left\{x^0, x^1, \ldots, x^{2q}, \left(x - \frac{a+b}{2}\right)^{2q+1}\right\}$$

von Π_{2q+1}. Nach Voraussetzung ist $Q[x^i] = I[x^i]$ für alle $i = 0, 1, \ldots, 2q$. Ferner ist

$$Q\big[\big(x - \tfrac{a+b}{2}\big)^{2q+1}\big] = 0,$$

da die Knoten und Gewichte symmetrisch zu $(a+b)/2$ liegen und $\big(x - (a+b)/2\big)^{2q+1}$ punktsymmetrisch zum Punkt $\big((a+b)/2, 0\big)$ ist. Andererseits ist auch

$$\begin{aligned} I\big[\big(x - \tfrac{a+b}{2}\big)^{2q+1}\big] &= \frac{1}{2q+2}\left(x - \frac{a+b}{2}\right)^{2q+2}\Bigg|_a^b \\ &= \frac{1}{2q+2}\left\{\left(\frac{b-a}{2}\right)^{2q+2} - \left(\frac{a-b}{2}\right)^{2q+2}\right\} \\ &= 0. \end{aligned}$$

Folglich ist $Q[p] = I[p]$ für alle $p \in \Pi_{2q+1}$. □

Offensichtlich hat die Simpson-Regel Exaktheitsgrad 3 anstelle von 2. Genauer haben wir den folgenden Satz:

Satz 6.9 Sei $f \in C^4([a,b])$. Dann gilt für die zusammengesetze Simpson-Regel $S_N[f]$ der Fehler

$$|I[f] - S_N[f]| \le \frac{b-a}{180} h^4 M_4, \quad h = \frac{b-a}{2N},$$

mit $M_4 = \max_{a \le x \le b} |f^{(4)}(x)|$.

Beweis. Wir interpolieren f in jedem der N Teilintervalle $[c,d]$ durch ein Polynom p dritten Grades mit Stützstellen c, d und $(c+d)/2$ (letztere doppelt). Da die Quadraturformel für Polynome dritten Grades exakt ist, verbleibt, das Integral über $f - p$ abzuschätzen. Dazu verwenden wir die Fehlerdarstellung (3.4) mit dem Knotenpolynom

$$w(x) = (x-c)(x-d)\Big(x - \frac{c+d}{2}\Big)^2,$$

vergleiche Bemerkung 3.8.
Durch Integration ergibt sich

$$\begin{aligned}
\int_c^d |f(x) - p(x)| \, \mathrm{d}x &\le \frac{M_4}{4!}\Big(\frac{d-c}{2}\Big)^5 \int_{-1}^1 t^2(1-t^2) \, \mathrm{d}t \\
&= \frac{M_4}{4!}\Big(\frac{d-c}{2}\Big)^5 \Big(\frac{1}{3}t^3 - \frac{1}{5}t^5\Big)\Big|_{-1}^1 \\
&= \frac{M_4}{24}\Big(\frac{d-c}{2}\Big)^5 \frac{4}{15} \\
&= \frac{d-c}{180}\Big(\frac{b-a}{2N}\Big)^4 M_4.
\end{aligned}$$

Aufsummation ergibt die gewünschte Behauptung. □

Beachte: Hier werden doppelt so viele Funktionswerte benötigt wie für die zusammengesetzte Trapezregel. Dafür erhält man allerdings auch die doppelte Ordnung, nämlich $s = 4$.

6.3 Adaptive Quadratur

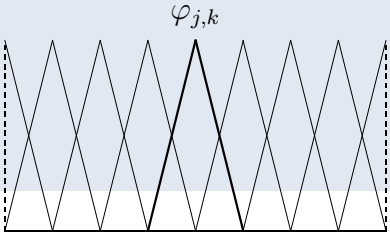
Wir wollen adaptive Quadraturstrategien anhand der Trapezsumme vorstellen. Dazu rufen wir uns den Raum der linearen Splinefunktionen in das Gedächnis zurück:

Definition 6.10 Sei

$$B_1(x) = \begin{cases} 1 - |x|, & -1 \le x \le 1 \\ 0, & \text{sonst} \end{cases}$$

die **Hutfunktion**. Für jedes $j \in \mathbb{N}_0$ bilden die Funktionen

$$\varphi_{j,k}(x) = B_1(2^j x - k)\big|_{[0,1]}, \quad k \in \Delta_j := \{0, 1, \ldots, 2^j\},$$



die **nodale Basis** im **Raum der stückweise linearen Funktionen**

$$S_1(\Delta_j) = \Big\{ f \in C([0,1]) : f\big|_{[2^{-j}k, 2^{-j}(k+1)]} \in \Pi_1 \text{ für alle } k = 0, 1, \ldots, 2^j - 1 \Big\}.$$

Eine Funktion $f \in C([0,1])$ kann mit Hilfe der nodalen Basis in $S_1(\Delta_j)$ approximiert werden gemäß

$$f(x) \approx f_j(x) := \sum_{k \in \Delta_j} \alpha_{j,k} \varphi_{j,k}(x) \quad \text{mit} \quad \alpha_{j,k} = f(2^{-j}k).$$

Folglich gilt

$$I[f] = \int_0^1 f(x)\,\mathrm{d}x \approx I[f_j] = \sum_{k \in \Delta_j} \alpha_{j,k} \underbrace{\int_0^1 \varphi_{j,k}(x)\,\mathrm{d}x}_{= \begin{cases} 2^{-j}, & \text{falls } 0 < k < 2^j \\ 2^{-(j+1)}, & \text{falls } k = 0 \text{ oder } k = 2^j \end{cases}}.$$

Offensichtlich ist $I[f_j]$ gerade die Trapezsumme

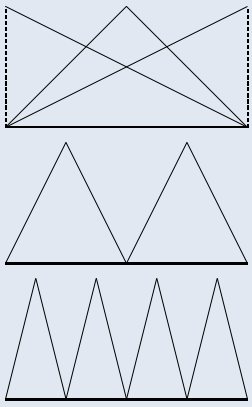
$$T_j[f] = 2^{-(j+1)} \Big\{ f(0) + 2 \sum_{k=1}^{2^j - 1} f(2^{-j}k) + f(1) \Big\}.$$

Definition 6.11 Setzen wir

$$\nabla_\ell := \begin{cases} \{0, 1\}, & \text{falls } \ell = 0, \\ \{1, 3, 5, \ldots, 2^\ell - 1\}, & \text{falls } \ell \in \mathbb{N}. \end{cases}$$

Dann ist die **hierarchische Basis** in $S_1(\Delta_j)$ definiert durch

$$\{\varphi_{\ell,k}\}_{k \in \nabla_\ell, 0 \le \ell \le j}.$$



In der hierarchischen Basis besitzt f_j die Darstellung

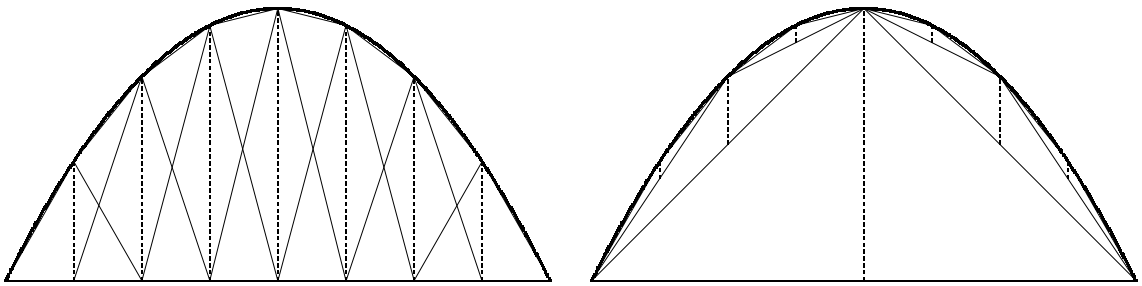
$$f_j(x) = \sum_{\ell=0}^{j} \sum_{k \in \nabla_\ell} \beta_{\ell,k} \varphi_{\ell,k}(x)$$

$$\text{mit} \quad \beta_{\ell,k} = \begin{cases} f(2^{-\ell}k), & \text{falls } \ell = 0, \\ f(2^{-\ell}k) - \frac{1}{2}\big\{ f\big(2^{-\ell}(k-1)\big) + f\big(2^{-\ell}(k+1)\big) \big\}, & \text{falls } \ell \in \mathbb{N}. \end{cases}$$

Die Funktion f_j heißt dann *hierarchischer Interpolant von* f, die Koeffizienten $\beta_{\ell,k}$ heißen *hierarchische Überschüsse*. Ist $f \in C^2([0,1])$, dann liefert eine Taylor-Entwicklung von $f(x \pm h)$ in x für $\ell > 0$ die Abschätzung

$$\beta_{\ell,k} \le 2^{-(2\ell+1)} \|f''\|_{C([2^{-\ell}(k-1), 2^{-\ell}(k+1)])} \le 2^{-(2\ell+1)} \|f''\|_{C([0,1])}. \tag{6.6}$$

Beispiel 6.12 Im Fall einer Parabel ergeben sich folgende Darstellungen auf Level $j = 3$:



nodale Basis hierarchische Basis

△

Mit Hilfe des hierarchischen Interpolanten erhalten wir die Quadraturformel

$$I[f_j] = \sum_{\ell=0}^{j} \sum_{k \in \nabla_\ell} \beta_{\ell,k} \int_0^1 \varphi_{\ell,k}(x)\,\mathrm{d}x = \frac{f(0)}{2} + \frac{f(1)}{2} + \sum_{\ell=1}^{j} \sum_{k \in \nabla_\ell} 2^{-\ell} \beta_{\ell,k}.$$

Ist die Funktion glatt, dann werden die hierarchischen Überschüsse gemäß (6.6) sehr schnell sehr klein. Sind sie hingegen groß, ist dies ein Indikator dafür, dass die Funktion an dieser Stelle nicht glatt ist. Daher ist folgender adaptive Quadraturalgorithmus naheliegend:

Algorithmus 6.13 (adaptive Quadratur)
input: Funktion $f \in C([0,1])$ und Fehlertoleranz ε
output: Approximation $Q \approx I[f]$

① Initialisierung: setze $Q := \big(f(0) + f(1)\big)/2$ und $(x,h) := (1/2, 1/2)$

② berechne den hierarchischen Überschuss

$$E(x,h) = f(x) - \frac{1}{2}\big(f(x-h) + f(x+h)\big)$$

③ falls $|E(x,h)| > \varepsilon$, dann datiere auf $Q := Q + hE(x,h)$ und gehe nach ② mit $(x,h) := (x - h/2, h/2)$ und $(x,h) := (x + h/2, h/2)$

Bemerkung 6.14 Der Algorithmus kann zu früh terminieren, insbesondere wenn die Nullstellen des Integranden f mit den Stützstellen zusammenfallen. Dies ist beispielsweise bei der Funktion $f(x) = \sin(2^\ell \pi x)$ der Fall. △

6.4 Euler-Maclaurinsche Summenformel

Definition 6.15 Die **Bernoulli-Polynome** B_n vom Grad n sind rekursiv definiert durch $B_0(x) := 1$ und

$$B_n' := B_{n-1}, \qquad \int_0^1 B_n(x)\,\mathrm{d}x = 0, \quad n = 1, 2, \ldots. \tag{6.7}$$

Die rationalen Zahlen

$$b_n = n!B_n(0), \quad n = 1, 2, \ldots$$

heißen **Bernoulli-Zahlen**.

Die ersten Bernoulli-Polynome sind gegeben durch

$$B_0(x) = 1, \quad B_1(x) = x - \frac{1}{2}, \quad B_2(x) = \frac{1}{2}x^2 - \frac{1}{2}x + \frac{1}{12}.$$

Wegen $\int_0^1 B_{n-1}\,\mathrm{d}x = B_n(1) - B_n(0)$, ist die Normalisierungsbedingung äquivalent zu

$$B_n(0) = B_n(1), \quad n = 2, 3, \ldots. \tag{6.8}$$

Lemma 6.16 Die Bernoulli-Polynome erfüllen die Symmetrieeigenschaft

$$B_n(x) = (-1)^n B_n(1 - x), \quad x \in \mathbb{R}, \quad n = 0, 1, 2, \ldots. \tag{6.9}$$

Beweis. Offensichtlich gilt (6.9) für $n = 0$. Angenommen, (6.9) gilt für ein $n \geq 0$. Integration von (6.9) liefert

$$B_{n+1}(x) = (-1)^{n+1} B_{n+1}(1 - x) + \beta_{n+1}, \quad \beta_{n+1} \in \mathbb{R}.$$

Die Normalisierungsbedingung aus (6.7) liefert $\beta_{n+1} = 0$ und somit die Behauptung im Fall $n + 1$. □

Lemma 6.17 Die Bernoulli-Polynome B_{2m+1}, $m = 1, 2, \ldots$, ungeraden Grades besitzen exakt 3 Nullstellen in $[0, 1]$, und diese sind in den Punkten $0, 1/2$ und 1. Die Bernoulli-Polynome B_{2m}, $m = 1, 2, \ldots$, geraden Grades erfüllen $B_{2m}(0) \neq 0$.

Beweis. Aus (6.8) und (6.9) folgt, dass B_{2m+1} verschwindet in den Punkten $0, 1/2$ und 1. Wir beweisen durch Induktion, dass dies die einzigen Nullstellen von B_{2m+1} in $[0, 1]$ sind. Für $m = 1$ ist die Aussage richtig, da $B_3 \in \Pi_3$ nur drei Nullstellen besitzt. Angenommen, dass die Aussage ist richtig für B_{2m+1} und dass B_{2m+3} noch eine weitere Nullstelle $\alpha \in [0, 1]$ besitzt. Wegen der Symmetrie (6.9) dürfen wir annehmen, dass $\alpha \in (0, 1/2)$ gilt. Aus dem Satz von Rolle folgt dann aber, dass $B'_{2m+3} = B_{2m+2}$ mindestens je eine Nullstelle in $(0, \alpha)$ und $(\alpha, 1/2)$ besitzt. Damit besitzt aber auf $B'_{2m+2} = B_{2m+1}$ eine Nullstelle in $(0, 1/2)$, was im Widerspruch zu unserer Annahme steht.

Aufgrund der Nullstellen von B_{2m+1} impliziert der Satz von Rolle, dass $B'_{2m+1} = B_{2m}$ eine Nullstelle in $(0, 1/2)$ besitzt. Gilt nun $B_{2m}(0) = 0$, dann besitzt $B'_{2m} = B_{2m-1}$ eine Nullstelle in $(0, 1/2)$, was dem oben schon bewiesenen widerspricht. □

Für $N \in \mathbb{N}$ sei $h = (b - a)/N$ und $x_i = a + ih$, $i = 0, 1, \ldots, N$. Die zusammengesetzte Trapezregel ist definiert durch

$$T_N[f] := \frac{h}{2}\big\{f(a) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{N-1}) + f(b)\big\}.$$

Wir beweisen nun die *Euler-Maclaurinsche Summenformel*.

Satz 6.18 Sei $f \in C^m([a,b])$ für ein $m \geq 2$. Dann gilt die Euler-Maclaurinsche Summenformel

$$\int_a^b f(x)\,\mathrm{d}x = T_N(f) - \sum_{j=1}^{\lfloor m/2 \rfloor} \frac{b_{2j} h^{2j}}{(2j)!}\big(f^{(2j-1)}(b) - f^{(2j-1)}(a)\big) + (-1)^m h^m \int_a^b \widetilde{B}_m\left(\frac{x-a}{h}\right) f^{(m)}(x)\,\mathrm{d}x, \tag{6.10}$$

wobei $\widetilde{B}_m$ das periodisierte Bernoulli-Polynom B_m bezeichnet

$$\widetilde{B}_m(x) := B_m(x - \lfloor x \rfloor).$$

Beweis. Sei $g \in C^m([0,1])$, dann ergibt $(m-1)$-malige partielle Integration zusammen mit (6.8)

$$\int_0^1 B_1(z) g'(z)\,\mathrm{d}x = \sum_{j=2}^{m} (-1)^j B_j(0)\big(g^{(j-1)}(1) - g^{(j-1)}(0)\big) - (-1)^m \int_0^1 B_m(z) g^{(m)}(z)\,\mathrm{d}z.$$

Einsetzen von

$$\int_0^1 B_1(z) g'(z)\,\mathrm{d}z = \frac{1}{2}\big(g(1) + g(0)\big) - \int_0^1 g(z)\,\mathrm{d}z$$

liefert unter Beachtung, dass alle ungeraden Bernoulli-Zahlen verschwinden, die Gleichung

$$\int_0^1 g(z)\,\mathrm{d}z = \frac{h}{2}\big(g(1) + g(0)\big) - \sum_{j=1}^{\lfloor m/2 \rfloor} \frac{b_{2j}}{(2j)!}\big(f^{(2j-1)}(b) - f^{(2j-1)}(a)\big) + (-1)^m \int_0^1 B_m(x) g^{(m)}(z)\,\mathrm{d}z.$$

Substituieren wir $x = x_k + hz$ und $g(z) = f(x_k + hz)$, dann erhalten wir

$$\int_{x_k}^{x_{k+1}} f(x)\,\mathrm{d}x = \frac{1}{2}\big(g(x_k) - g(x_{k+1})\big) - \sum_{j=1}^{\lfloor m/2 \rfloor} \frac{b_{2j} h^{2j}}{(2j)!}\big(f^{(2j-1)}(x_{k+1}) - f^{(2j-1)}(x_k)\big) + (-1)^m h^m \int_{x_k}^{x_{k+1}} B_m\left(\frac{x - x_k}{h}\right) f^{(m)}(x)\,\mathrm{d}x.$$

Aufsummieren über alle $k = 0, 1, \ldots, N-1$ liefert schließlich (6.10). □

6.5 Romberg-Verfahren

Während die Erhöhung der Knotenzahl N bei Trapez- oder der Simpson-Regel lediglich eine Verbesserung der Form N^{-p} für ein festes p ergibt, wollen wir hier eine Methode vorstellen, bei der N und p gleichzeitig erhöht werden.

Sei

$$T_N^1[f] := \frac{h}{2}\left\{ f(a) + 2\sum_{n=1}^{N-1} f(a + nh) + f(b) \right\}$$

die zusammengesetzte Trapezregel zur Schrittweite $h = (b-a)/N$. Vorausgesetzt $f \in C^4([a,b])$, dann liefert die Euler-Maclaurinsche Summenformel (6.10) die Fehlerdarstellung

$$\int_a^b f(x)\,\mathrm{d}x = T_N^1[f] + \gamma_1 h^2 + \mathcal{O}(h^4)$$

für eine Konstante γ_1, die von f abhängt, aber nicht von h. Daher folgt für die halbe Schrittweite

$$\int_a^b f(x)\,\mathrm{d}x = T_{2N}^1[f] + \gamma_1 \frac{h^2}{4} + \mathcal{O}(h^4).$$

Aus diesen beiden Gleichungen können wir nun den Term $\gamma_1 h^2$ eliminieren:

$$\int_a^b f(x)\,\mathrm{d}x = \frac{1}{3}\Big(4T_{2N}^1[f] - T_N^1[f]\Big) + \mathcal{O}(h^4).$$

Die Quadraturformel

$$T_N^2[f] := \frac{1}{3}\Big(4T_{2N}^1[f] - T_N^1[f]\Big)$$

besitzt demnach die verbesserte Fehlerordnung $\mathcal{O}(h^4)$. Wie man leicht nachrechnet, entspricht $T_N^2[f]$ gerade der zusammengesetzten Simpson-Regel $S_N[f]$.

Ist $f \in C^6([a,b])$, dann folgt aus der entsprechenden Linearkombination der Euler-Maclaurinschen Summenformeln (6.10) zu den Schrittweiten h und $h/2$, dass

$$\int_a^b f(x)\,\mathrm{d}x = T_N^2[f] + \gamma_2 h^4 + \mathcal{O}(h^6)$$

mit einer nur von f abhängigen Konstante γ_2. Geeignete Linearkombination mit

$$\int_a^b f(x)\,\mathrm{d}x = T_{2N}^2[f] + \gamma_2 \frac{h^4}{16} + \mathcal{O}(h^6)$$

liefert die Quadraturformel (es ist gerade die zusammengesetzte Milne-Regel)

$$T_N^3[f] := \frac{1}{15}\Big(16T_{2N}^2 - T_N^2[f]\Big)$$

von der Fehlerordnung $\mathcal{O}(h^6)$. Diese benötigt die Auswertungen der Trapezsummen zu den Schrittweiten $h, h/2$ und $h/4$.

Offensichtlich kann dieses Vorgehen fortgesetzt werden, was auf das *Romberg-Verfahren* führt:

Algorithmus 6.19 (Romberg-Verfahren)

input: Funktion f, Ausgangsschrittweite $h = 1/N$, Rekursionstiefe M

output: Approximation t_0^{M+1} an $I[f]$

① berechne für alle $k = 0, 1, 2, \ldots, M$ die Trapezsummen $t_k^1 := T_{2^k N}^1[f]$

② für alle $m = 1, 2, \ldots, M$ berechne

$$t_k^{m+1} := \frac{1}{4^m - 1}\big(4^m t_{k+1}^m - t_k^m\big), \quad k = 0, 1, 2, \ldots, M-m \tag{6.11}$$

Der Aufwand von Romberg-Verfahrens ist dabei

$$\mathcal{O}\Big(\sum_{k=0}^{M} 2^k N\Big) = \mathcal{O}(2^M N).$$

Der Quadraturfehler wird in folgendem Satz abgeschätzt.

Satz 6.20 Sei $f \in C^{2m}([a,b])$, dann gilt

$$\left| \int_a^b f(x)\,\mathrm{d}x - t_k^m \right| \le C_m \big\| f^{(2m)} \big\|_{C([a,b])} \left(\frac{h}{2^k}\right)^{2m}, \quad k = 0, 1, \ldots, \tag{6.12}$$

mit einer von m abhängigen Konstante C_m.

Beweis. Durch Induktion zeigen wir, dass Konstanten $\gamma_{j,i}$ existieren, so dass

$$\left| \int_a^b f(x)\,\mathrm{d}x - t_k^i - \sum_{j=i}^{m-1} \gamma_{j,i} \big(f^{(2j-1)}(b) - f^{(2j-1)}(a) \big) \left(\frac{h}{2^k}\right)^{2j} \right| \le \gamma_{m,i} \big\| f^{(2m)} \big\|_{C([a,b])} \left(\frac{h}{2^k}\right)^{2m} \tag{6.13}$$

für alle $i = 1, 2, \ldots, m$ und $m = 0, 1, \ldots$. Dabei ist die Summe gleich Null, falls $i = m$ ist. Aufgrund der Euler-Maclaurinschen Entwicklung (6.10) ist (6.13) im Fall $i = 1$ wahr, falls $\gamma_{j,1} = b_{2j}/(2j)!$ für alle $j = 1, 2, \ldots, m-1$ und

$$\gamma_{m,1} = (b-a) \| B_{2m} \|_{C([0,1])}$$

gesetzt wird.
Wir setzen

$$F_j := f^{(2j-1)}(b) - f^{(2j-1)}(a), \quad j = 1, 2, \ldots, m-1.$$

Angenommen, (6.13) gilt für ein $1 \le i < m$. Dann folgt aus (6.11), dass

$$\begin{aligned} &\frac{4^i}{4^i - 1} \left[\int_a^b f(x)\,\mathrm{d}x - t_{k+1}^i - \sum_{j=i}^{m-1} \left(\frac{h}{2^{k+1}}\right)^{2j} \gamma_{j,i} F_j \right] \\ &\qquad - \frac{1}{4^i - 1} \left[\int_a^b f(x)\,\mathrm{d}x - t_k^i - \sum_{j=i}^{m-1} \left(\frac{h}{2^k}\right)^{2j} \gamma_{j,i} F_j \right] \\ &\qquad\qquad = \int_a^b f(x)\,\mathrm{d}x - t_k^{i+1} - \sum_{j=i+1}^{m-1} \left(\frac{h}{2^k}\right)^{2j} \gamma_{j,i+1} F_j, \end{aligned}$$

wobei

$$\gamma_{j,i+1} = \frac{4^{i-j} - 1}{4^i - 1} \gamma_{j,i}, \quad j = i+1, i+2, \ldots, m-1.$$

Mit Hilfe der Induktionsannahme und

$$\gamma_{m,i+1} := \frac{4^{i-m} + 1}{4^i - 1}$$

ergibt sich

$$\left| \int_a^b f(x)\,\mathrm{d}x - t_k^{i+1} - \sum_{j=i+1}^{m-1} \left(\frac{h}{2^k}\right)^{2j} \gamma_{j,i+1} F_j \right| \le \gamma_{m,i+1} \left\| f^{(2m)} \right\|_{C([a,b])} \left(\frac{h}{2^k}\right)^{2m}.$$

□

Beispiel 6.21 Zu $f(x) = \sin^4 x$ berechnen wir $I[f] = \int_0^\pi \sin^4 x\,\mathrm{d}x$ mit dem Romberg-Verfahren. Ausgehend von

$$\begin{aligned}
T_1[f] &= \frac{\pi}{2}(\sin^4 0 + \sin^4 \pi) = 0,\\
T_2[f] &= \frac{\pi}{4}(\sin^4 0 + 2\sin^4 \frac{\pi}{2} + \sin^4 \pi) = \frac{\pi}{2},\\
T_4[f] &= \frac{\pi}{8}(\sin^4 0 + 2\sin^4 \frac{\pi}{4} + 2\sin^4 \frac{\pi}{2} + 2\sin^4 \frac{3\pi}{4} + \sin^4 \pi) = \frac{3\pi}{8},
\end{aligned}$$

erhalten wir die Näherung

$$\begin{array}{lcccl}
t_0^1 = T_1[f] = 0 & & & & \\
 & \searrow^{-\frac{1}{3}} & & & \\
t_1^1 = T_2[f] = \dfrac{\pi}{2} & \xrightarrow{\frac{4}{3}} & t_0^2 = \dfrac{2\pi}{3} & \searrow^{-\frac{1}{15}} & \\
 & \searrow^{-\frac{1}{3}} & & & \\
t_2^1 = T_4[f] = \dfrac{3\pi}{8} & \xrightarrow{\frac{4}{3}} & t_1^2 = \dfrac{\pi}{3} & \xrightarrow{\frac{16}{15}} & t_0^2 = \dfrac{14}{45}\pi
\end{array}$$

△

6.6 Quadratur periodischer Funktionen

Im Fall 2π-periodischer Funktionen $f : \mathbb{R} \to \mathbb{R}$ stimmt die Trapezregel mit der *Mittelpunktsregel* überein

$$I[f] = \int_0^{2\pi} f(x)\,\mathrm{d}x \approx Q_N[f] = \frac{2\pi}{N} \sum_{k=1}^{N} f\left(\frac{2\pi k}{N}\right).$$

Für den Quadraturfehler liefert die Euler-Maclaurinsche Summenformel (6.10) das folgende Korollar.

Korollar 6.22 Sei $f \in C^{2m+1}_{\mathrm{per}}([0, 2\pi])$ und sei $N \in \mathbb{N}$. Dann ist der Fehler der Mittelpunktsregel beschränkt durch

$$\left| I[f] - Q_N[f] \right| \le \frac{C}{N^{2m+1}} \int_0^{2\pi} \left| f^{(2m+1)}(x) \right| \mathrm{d}x$$

mit

$$C := 2 \sum_{k=1}^{\infty} \frac{1}{k^{2m+1}}.$$

Beweis. Wegen der Periodizität gilt $f^{(j)}(0) = f^{(j)}(2\pi)$, $j = 0, 1, \ldots, 2m+1$. Daher reduziert sich die Euler-Maclaurinsche Summenformel (6.10) auf

$$\bigl|I[f] - Q_N[f]\bigr| = -\left(\frac{2\pi}{N}\right)^{2m+1} \int_0^{2\pi} \widetilde{B}_{2m+1}\left(\frac{2\pi x}{N}\right) f^{(2m+1)}(x)\,\mathrm{d}x.$$

Die periodisierten Bernoulli-Polynome genügen der Fourier-Entwicklung

$$\widetilde{B}_{2m}(x) = 2(-1)^{m-1} \sum_{k=1}^{\infty} \frac{\cos(2\pi kx)}{(2\pi k)^{2m}}, \quad \widetilde{B}_{2m+1}(x) = 2(-1)^{m-1} \sum_{k=1}^{\infty} \frac{\sin(2\pi kx)}{(2\pi k)^{2m+1}}$$

für alle $m = 0, 1, 2, \ldots$. Hieraus folgt

$$\bigl|\widetilde{B}_{2m+1}(x)\bigr| \le 2 \sum_{k=1}^{\infty} \frac{1}{(2\pi k)^{2m+1}}, \quad x \in \mathbb{R},$$

und damit die Behauptung. □

Ist f eine analytische Funktion, dann kann f in jedem Punkt $x_0 \in [0, 2\pi]$ in eine Taylor-Reihe entwickelt werden

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k, \quad |x - x_0| < r(x_0)$$

mit Konvergenzradius

$$\frac{1}{r(x_0)} = \limsup_{k\to\infty} \sqrt[k]{\frac{f^{(k)}(x_0)}{k!}} > 0.$$

Folglich existiert eine Konstante $c_f > 0$, so dass

$$\bigl\|f^{(k)}\bigr\|_{C([0,2\pi])} \le c_f \frac{k!}{R^k}, \quad R := \min_{x\in[0,2\pi]} r(x) > 0.$$

Wir erhalten demnach im Falle einer *analytischen* Funktion f die Fehlerabschätzung

$$\bigl|I[f] - Q_N[f]\bigr| \le \frac{2\pi c_f\, C\,(2m+1)!}{(RN)^{2m+1}}$$

für alle $m \in \mathbb{N}$, unabhängig von N. Wählen wir nun $m \sim (RN-1)/2$, dann folgt

$$\frac{(2m+1)!}{(RN)^{2m+1}} \sim \frac{(RN)!}{(RN)^{RN}} \le \frac{(RN/2)^{RN/2}(RN)^{RN/2}}{(RN)^{RN}} \le 2^{-RN/2},$$

das heißt, die Mittelpunktsregel konvergiert *exponentiell* in N,

$$\bigl|I[f] - Q_N[f]\bigr| \le a\exp(-bN), \quad a, b > 0.$$

Exponentielle Konvergenz bedeutet, dass eine Verdoppelung der Anzahl N der Funktionsauswertungen die doppelte Anzahl richtiger Ziffern ergibt.

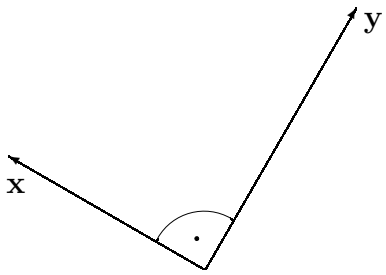


Abbildung 6.2: $\mathbf{x}$ steht senkrecht auf $\mathbf{y}$.

6.7 Orthogonalpolynome

Orthogonalität ist ein zentrales Hilfsmittel in der Mathematik, insbesondere in der Numerik. Der Begriff ist vertraut und anschaulich im $\mathbb{R}^n$: Zwei Vektoren $\mathbf{x}$ und $\mathbf{y}$ heißen zueinander orthogonal, kurz $\mathbf{x} \perp \mathbf{y}$, falls gilt

$$\mathbf{x}^\intercal \mathbf{y} = \sum_{i=1}^{n} x_i y_i = 0.$$

Geometrisch bedeutet dies, dass $\mathbf{x}$ senkrecht auf $\mathbf{y}$ fußt:

Um den Begriff der Othogonalität in beliebigen Vektorräumen einzuführen, benötigen wir ein Innenprodukt.

Definition 6.23 Eine Abbildung $\langle \cdot, \cdot \rangle : X \times X \to \mathbb{R}$ in einem Vektorraum X über $\mathbb{R}$ heißt **Innenprodukt** oder **Skalarprodukt**, falls gilt:

1. $\langle f, g \rangle = \langle g, f \rangle$ für alle $f, g \in X$ (Symmetrie)
2. $\langle \alpha f + \beta g, h \rangle = \alpha \langle f, h \rangle + \beta \langle g, h \rangle$ für alle $f, g, h \in X$ und $\alpha, \beta \in \mathbb{R}$ (Linearität)
3. $\langle f, f \rangle > 0$ für alle $f \in X \setminus \{0\}$ (Definitheit)

Bemerkung 6.24 Wegen der Symmetrie ist $\langle \cdot, \cdot \rangle$ auch im zweiten Argument linear. Man nennt eine Abbildung $\langle \cdot, \cdot \rangle$, die die ersten beiden Aussagen erfüllt, daher auch *symmetrische Bilinearform*. △

Proposition 6.25 Es gilt die Cauchy-Schwarzsche Ungleichung

$$\langle f, g \rangle^2 \leq \langle f, f \rangle \cdot \langle g, g \rangle.$$

Beweis. Im Fall $g = 0$ ist nichts zu zeigen. Sei also $g \neq 0$, dann gilt für jedes $\alpha \in \mathbb{R}$

$$0 \leq \langle f - \alpha g, f - \alpha g \rangle = \langle f, f \rangle - 2\alpha \langle f, g \rangle + \alpha^2 \langle g, g \rangle.$$

Setzt man speziell $\alpha = \langle f, g \rangle / \langle g, g \rangle$ ein, dann folgt

$$0 \leq \langle f, f \rangle - \frac{\langle f, g \rangle^2}{\langle g, g \rangle},$$

woraus die Behauptung folgt. □

Satz 6.26 Ein Innenprodukt $\langle \cdot, \cdot \rangle : X \times X \to \mathbb{R}$ induziert eine Norm in X, die wegen der dritten Aussage aus Definition 6.23 wohldefiniert ist:

$$\|f\| := \sqrt{\langle f, f \rangle}.$$

Beweis. Der einfache Beweis verbleibt dem Leser zur Übung. □

Bemerkung 6.27 Im $\mathbb{R}^n$ ist die induzierte Norm die übliche Euklid-Norm $\|\cdot\|_2$. △

Das für uns wichtigste Beispiel ist der Raum der auf dem Intervall $I = (a, b)$ quadratisch integrierbaren Funktionen ($a = -\infty$ und $b = +\infty$ sind hier ausdrücklich zugelassen)

$$L^2(I) = \left\{ \text{alle Funktionen } f : I \to \mathbb{R} : \int_I f(t)^2 \, \mathrm{d}t < \infty \right\},$$

versehen mit dem Innenprodukt

$$\langle f, g \rangle_{L^2(I)} := \int_I f(t) g(t) \, \mathrm{d}t, \qquad \|f\|_{L^2(I)} = \left(\int_I f(t)^2 \, \mathrm{d}t \right)^{1/2}.$$

Entsprechend definieren wir für eine positive Funktion w mit

$$\int_I w(x) \, \mathrm{d}x < \infty$$

den Raum $L^2_w(I)$ mit

$$\langle f, g \rangle := \int_I f(t) g(t) w(t) \, \mathrm{d}t. \tag{6.14}$$

Die Funktion w heißt *Gewichtsfunktion*.

Satz 6.28 Es sei $I = (a, b) \subset \mathbb{R}$ ein Intervall. Zu jeder Gewichtsfunktion w und zugehörigem Innenprodukt (6.14) existiert eine eindeutig bestimmte Folge von Polynomen $\{u_n\}_{n=0}^{\infty}$ mit $u_n \in \Pi_n$, für die gilt

$$\langle u_n, u_m \rangle = \delta_{m,n}, \quad u_n(x) = \gamma_n x^n + \cdots \text{ für ein } \gamma_n > 0. \tag{6.15}$$

Die Folge $\{u_n\}$ genügt zudem der Dreitermrekursion

$$a_{n+1} u_{n+1}(x) = (x - b_n) u_n(x) - a_n u_{n-1}(x), \quad n \geq 0, \tag{6.16}$$

wobei $a_n = \gamma_{n-1}/\gamma_n > 0$, $b_n = \langle x u_n, u_n \rangle$, $u_{-1} \equiv 0$ und

$$u_0 = \frac{1}{\sqrt{\langle 1, 1 \rangle}} = \frac{1}{\sqrt{\int_I w(x) \, \mathrm{d}x}}.$$

Beweis. Wir führen den Beweis induktiv. Seien $u_0, u_1, \ldots, u_n$ Orthonormalpolynome von Π_n bezüglich $\langle \cdot, \cdot \rangle$ aus (6.14), die alle im Satz genannten Eigenschaften erfüllt. Ist

$$p_{n+1}(x) = xu_n(x) - b_n u_n(x) - a_n u_{n-1}(x),$$

dann gilt

$$\langle p_{n+1}, u_n \rangle = \langle xu_n, u_n \rangle - b_n \underbrace{\langle u_n, u_n \rangle}_{=1} - a_n \underbrace{\langle u_{n-1}, u_n \rangle}_{=0} = \underbrace{\langle xu_n, u_n \rangle}_{=b_n} - b_n = 0.$$

Offensichtlich gilt zudem

$$\begin{aligned}
\langle p_{n+1}, u_{n-1} \rangle &= \langle xu_n, u_{n-1} \rangle - b_n \underbrace{\langle u_n, u_{n-1} \rangle}_{=0} - a_n \underbrace{\langle u_{n-1}, u_{n-1} \rangle}_{=1} \\
&= \langle xu_n, u_{n-1} \rangle - a_n \\
&\overset{(6.14)}{=} \langle u_n, xu_{n-1} \rangle - a_n \\
&\overset{(6.16)}{=} \langle u_n, a_n u_n + b_{n-1} u_{n-1} + a_{n-1} u_{n-2} \rangle - a_n \\
&= a_n \underbrace{\langle u_n, u_n \rangle}_{=1} + b_{n-1} \underbrace{\langle u_n, u_{n-1} \rangle}_{=0} + a_{n-1} \underbrace{\langle u_n, u_{n-2} \rangle}_{=0} - a_n \\
&= 0
\end{aligned}$$

und für jedes $m < n - 1$

$$\begin{aligned}
\langle p_{n+1}, u_m \rangle &= \langle xu_n, u_m \rangle - b_n \underbrace{\langle u_n, u_m \rangle}_{=0} - a_n \underbrace{\langle u_{n-1}, u_m \rangle}_{=0} \\
&= \langle u_n, xu_m \rangle \\
&\overset{(6.16)}{=} \langle u_n, a_{m+1} u_{m+1} + b_m u_m + a_m u_{m-1} \rangle \\
&= 0.
\end{aligned}$$

Folglich ist p_{n+1} für alle $0 \leq m \leq n$ orthogonal zu u_m, und nach Konstruktion gilt

$$p_{n+1}(x) = \gamma_n x^{n+1} + \cdots .$$

Um das normalisierte Polynom u_{n+1} zu finden, muss p_{n+1} geeignet umskaliert werden. Gesucht ist u_{n+1} derart, dass der Höchstkoeffizient γ_{n+1} positiv ist. Also gilt

$$u_{n+1}(x) = \gamma_{n+1} x^{n+1} + \cdots = \frac{\gamma_{n+1}}{\gamma_n} p_{n+1}(x),$$

beziehungsweise $p_{n+1} = a_{n+1} u_{n+1}$.
Es sei q_{n+1} ein zweites Polynom, das die Bedingung (6.15) erfüllt. Der Ansatz

$$q_{n+1} = \sum_{\ell=0}^{n+1} \lambda_\ell u_\ell$$

liefert

$$0 = \langle q_{n+1}, u_m \rangle = \sum_{\ell=0}^{n+1} \lambda_\ell \langle u_\ell, u_m \rangle = \lambda_m \quad \text{für alle } m \leq n.$$

Folglich ist $q_{n+1} = \lambda_{n+1} u_{n+1}$. Aus $\langle q_{n+1}, q_{n+1} \rangle = \lambda_{n+1}^2 \langle u_{n+1}, u_{n+1} \rangle$ ergibt sich $\lambda_{n+1}^2 = 1$, also $\lambda_{n+1} = \pm 1$. Da $\lambda_{n+1} = -1$ auf einen negativen Höchstkoeffizienten führt, folgt schließlich $\lambda_{n+1} = 1$ und $q_{n+1} = u_{n+1}$. □

Bemerkung 6.29 Die gängigsten Orthogonalpolynome sind:

- Tschebyscheff-Polynome: $w(x) = 1/\sqrt{1-x^2}$ und $I = (-1,1)$.
- Legendre-Polynome: $w(x) = 1$ und $I = (-1,1)$
- Jacobi-Polynome: $w(x) = (1-x)^\alpha(1+x)^\beta$ für $\alpha, \beta > -1$ und $I = (-1,1)$
- Hermite-Polynome: $w(x) = e^{-x^2}$ und $I = \mathbb{R}$
- Laguerre-Polynome: $w(x) = x^\alpha e^{-x}$ für $\alpha > -1$ und $I = \mathbb{R}_{>0}$

△

Beispiel 6.30 (Tschebyscheff-Polynome) Es sei $I = (-1,1)$ und $w(x) = 1/\sqrt{1-x^2}$. Die Tschebyscheff-Polynome $T_n(x) = \cos(n \arccos x)$ aus Abschnitt 3.4 erfüllen nach Definition

$$\langle T_i, T_j \rangle = \int_{-1}^{1} \cos(i \arccos x) \cos(j \arccos x) \frac{1}{\sqrt{1-x^2}} \, \mathrm{d}x.$$

Substitutieren wir

$$x = \cos\xi, \quad \mathrm{d}x = -\sin\xi \, \mathrm{d}\xi = -\sqrt{1-x^2} \, \mathrm{d}\xi,$$

so ergibt sich

$$\langle T_i, T_j \rangle = -\int_{\pi}^{0} \cos(i\xi)\cos(j\xi) \, \mathrm{d}\xi = \int_{0}^{\pi} \cos(i\xi)\cos(j\xi) \, \mathrm{d}\xi = \begin{cases} 0, & \text{falls } i \neq j, \\ \pi, & \text{falls } i = j = 0, \\ \pi/2, & \text{falls } i = j \neq 0. \end{cases}$$

Die zugehörige Dreitermrekursion haben wir bereits in (3.11) kennengelernt. △

Beispiel 6.31 (Legendre-Polynome) Wir betrachten das Intervall $I = (-1,1)$, ausgestattet mit dem Standard-Innenprodukt, das heißt, $w(x) \equiv 1$. Das *Legendre-Polynom* vom Grad n ist definiert durch

$$P_n(x) := \frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n}[(x^2-1)^n] \in \Pi_n. \tag{6.17}$$

Es gilt $\langle P_n, P_m \rangle = 0$ für $n \neq m$, denn für $m < n$ folgt durch partielle Integration

$$\begin{aligned}
2^n n!\, 2^m m! \, \langle P_n, P_m \rangle &= \int_{-1}^{1} \frac{\mathrm{d}^n}{\mathrm{d}x^n}[(x^2-1)^n] \frac{\mathrm{d}^m}{\mathrm{d}x^m}[(x^2-1)^m] \, \mathrm{d}x \\
&= \frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}}[\underbrace{(x^2-1)^n}_{n\text{-fache Nst. in } \pm 1}] \frac{\mathrm{d}^m}{\mathrm{d}x^m}[(x^2-1)^m]\Big|_{-1}^{1} \\
&\qquad - \int_{-1}^{1} \frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}}[(x^2-1)^n] \frac{\mathrm{d}^{m+1}}{\mathrm{d}x^{m+1}}[(x^2-1)^m] \, \mathrm{d}x \\
&= -\int_{-1}^{1} \frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}}[(x^2-1)^n] \frac{\mathrm{d}^{m+1}}{\mathrm{d}x^{m+1}}[(x^2-1)^m] \, \mathrm{d}x \\
&\;\;\vdots \\
&= (-1)^n \int_{-1}^{1} (x^2-1)^n \underbrace{\frac{\mathrm{d}^{m+n}}{\mathrm{d}x^{m+n}}[(x^2-1)^m]}_{\equiv 0, \text{ da } m+n > 2m = \deg[(x^2-1)^m]} \mathrm{d}x = 0.
\end{aligned}$$

Wiederum mit Hilfe partieller Integration folgt

$$\begin{aligned} 2^n n!\, 2^n n!\, \langle P_n, P_n\rangle &= (-1)^n \int_{-1}^{1} \underbrace{\frac{\mathrm{d}^{2n}}{\mathrm{d}x^{2n}}[(x^2-1)^n]}_{=(2n)!}(x^2-1)^n \,\mathrm{d}x \\ &= (-1)^n (2n)! \int_{-1}^{1} (x-1)^n (x+1)^n \,\mathrm{d}x \\ &= (n!)^2 \frac{2^{2n+1}}{2n+1}. \end{aligned}$$

Also ergibt sich $\langle P_n, P_n\rangle = 2/(2n+1)$ und daher ist

$$u_n(x) = \sqrt{\frac{2n+1}{2}} P_n(x) = \sqrt{\frac{2n+1}{2}} \frac{(2n)!}{2^n (n!)^2}\, x^n + \cdots = \gamma_n x^n + \cdots \tag{6.18}$$

das zugehörige orthonormale Polynom.
Für die Dreitermrekursion (6.16) ergibt sich

$$\begin{aligned} a_n = \frac{\gamma_{n-1}}{\gamma_n} &= \frac{2^n (n!)^2}{(2n)!} \sqrt{\frac{2}{2n+1}} \sqrt{\frac{2n-1}{2}} \frac{(2n-2)!}{2^{n-1}\big((n-1)!\big)^2} \\ &= \sqrt{\frac{2n-1}{2n+1}} \frac{2n^2}{2n(2n-1)} \\ &= \frac{n}{\sqrt{4n^2-1}}. \end{aligned}$$

Wegen (6.17) ist P_n ein gerades Polynom, falls n gerade, und ein ungerades Polynom, falls n ungerade ist. Daher muss b_n in (6.16) verschwinden. Dies bedeutet,

$$\frac{n+1}{\sqrt{4(n+1)^2-1}} u_{n+1}(x) = x u_n(x) - \frac{n}{\sqrt{4n^2-1}} u_{n-1}(x).$$

△

Zum Abschluss dieses Ausflugs zu den Orthogonalpolynomen wollen wir noch folgende für die Gauß-Quadratur wichtige Eigenschaft beweisen:

Satz 6.32 Die Nullstellen der Orthogonalpolynome $\{u_n\}$ sind alle einfach und liegen im Inneren von I.

Beweis. Wir nehmen an, die Aussage sei falsch. Hat u_n etwa eine Nullstelle z auf dem Rand von I oder in $\mathbb{R} \setminus \overline{I}$, dann ist

$$p_{n-1}(x) := \frac{u_n(x)}{x-z} \in \Pi_{n-1}$$

und daher

$$0 = \langle p_{n-1}, u_n\rangle = \int_I \frac{u_n^2(x)}{x-z} w(x)\,\mathrm{d}x.$$

Da jedoch $w(x)/(x-z)$ in I keinen Vorzeichenwechsel hat und $0 \not\equiv u_n^2(x) \geq 0$ ist, ergibt sich ein Widerspruch.
Ist hingegen $z \in I$ eine mehrfache Nullstelle von u_n oder liegt $z \notin \mathbb{R}$, dann wenden wir das entsprechende Argument auf

$$p_{n-2}(x) := \frac{u_n(x)}{(x-z)(x-\overline{z})} = \frac{u_n(x)}{|x-z|^2} \in \Pi_{n-2}$$

an. Dazu beachte man, dass gemäß Satz 6.28 das Polynom u_n reell, also mit $z \notin \mathbb{R}$ auch $\overline{z}$ eine Nullstelle von u_n ist. □

6.8 Gauß-Quadratur

Wir beantworten zunächst folgende Frage: Gegeben seien m Knoten $x_1, x_2, \ldots, x_m$ und m Gewichte $w_1, w_2, \ldots, w_m$. Wie groß ist maximal der Exaktheitsgrad der Quadraturformel

$$G_w[f] = \sum_{i=1}^{m} w_i f(x_i) \approx I_w[f] = \int_a^b f(x)w(x)\,\mathrm{d}x? \tag{6.19}$$

Proposition 6.33 Sei $w > 0$ in (a,b). Dann ist der Exaktheitsgrad der Quadraturformel $G_w[\cdot]$ aus (6.19) maximal $q = 2m-1$.

Beweis. Für

$$p(x) = \prod_{i=1}^{m}(x-x_i)^2 \in \Pi_{2m}$$

ist $G_w[p] = 0$, während gilt

$$I_w[p] = \int_a^b \underbrace{\prod_{i=1}^{m}(x-x_i)^2 w(x)}_{>0 \text{ für } x\neq a,x_1,x_2,\ldots,x_m,b}\,\mathrm{d}x > 0.$$

□

Im nachfolgenden Satz zeigen wir, dass mit Hilfe von Orthogonalpolynomen eine Quadraturformel konstruiert werden kann, die den maximalen Exaktheitsgrad annimmt.

Satz 6.34 Wählt man die Knoten $\{x_i\}_{i=1}^m$ der Quadraturformel G_w als Nullstellen des m-ten Orthonormalpolynoms u_m bezüglich des Innenprodukts

$$\langle f, g\rangle = \int_a^b f(x)g(x)w(x)\,\mathrm{d}x$$

und die Gewichte w_i gemäß

$$w_i = \int_a^b L_i(x)w(x)\,\mathrm{d}x, \tag{6.20}$$

dann besitzt G_w den Exaktheitsgrad $2m-1$.

Beweis. Die Quadraturformel G_w ist nach Proposition 6.5 exakt für Polynome vom Grad $q = m - 1$. Daher gilt

$$I_w[p] = G_w[p] \quad \text{für alle } p \in \Pi_{m-1}.$$

Wir erweitern die Monombasis $\{1, x, x^2, \ldots, x^{m-1}\}$ von Π_{m-1} durch $\{u_m, u_m x, \ldots, u_m x^{m-1}\}$, um eine Basis von Π_{2m-1} zu erhalten. Weil der Grad von $u_m x^k$ genau $m + k$ ist, ist dies in der Tat eine Basis.

Die Knoten von G_w sind gerade die Nullstellen von u_m, so dass folgt

$$G_w[u_m x^k] = 0 \quad \text{für alle } k = 0, 1, \ldots, m-1.$$

Andererseits ist aber auch

$$I_w[u_m x^k] = \int_a^b u_m(x)\, x^k\, w(x)\, \mathrm{d}x = 0 \quad \text{für alle } k = 0, 1, \ldots, m-1,$$

weil u_m senkrecht auf Π_{m-1} steht. Folglich gilt $I_w[p] = G_w[p]$ für alle $p \in \Pi_{2m-1}$. □

Beispiel 6.35 Im in der Praxis wichtigsten Fall $w \equiv 1$ sind die Orthonormalpolynome gerade die Legendre-Polynome aus Beispiel 6.31 und es gilt $a = -1$ und $b = 1$. Bis $m = 5$ lauten die Gewichte und Knoten:

m	x_i	w_i
1	0	2
2	$-\frac{1}{\sqrt{3}}$, $\frac{1}{\sqrt{3}}$	1, 1
3	$-\frac{\sqrt{15}}{5}$, 0, $\frac{\sqrt{15}}{5}$	$\frac{5}{9}$, $\frac{8}{9}$, $\frac{5}{9}$
4	$\pm\frac{1}{35}\sqrt{525 - 70\sqrt{30}}$ $\pm\frac{1}{35}\sqrt{525 + 70\sqrt{30}}$	$\frac{1}{36}(18 + \sqrt{30})$ $\frac{1}{36}(18 - \sqrt{30})$
5	0 $\pm\frac{1}{21}\sqrt{245 - 14\sqrt{70}}$ $\pm\frac{1}{21}\sqrt{245 + 14\sqrt{70}}$	$\frac{128}{225}$ $\frac{1}{900}(322 + 13\sqrt{70})$ $\frac{1}{900}(322 - 13\sqrt{70})$

△

Für den Fehler bei der Gauß-Quadratur gilt folgendes Resultat:

Satz 6.36 Sei $f \in C^{2m}([a,b])$ und G_w die m-stufige Gauß-Formel für I_w. Dann gilt

$$\left|I_w[f] - G_w[f]\right| \leq \frac{\|f^{(2m)}\|_{C([a,b])}}{(2m)!\gamma_m^2},$$

wobei γ_m wie in (6.15) den Höchstkoeffizienten des Orthonormalpolynoms $u_m(x)$ bezeichnet.

Beweis. Es sei $p \in \Pi_{2m-1}$ dasjenige Polynom, für das gilt

$$f(x_i) = p(x_i), \quad f'(x_i) = p'(x_i), \quad i = 1, 2, \ldots, m.$$

Ferner bezeichne $s(x) := \prod_{i=1}^{m}(x - x_i)^2 \in \Pi_{2m}$ das zugehörige Knotenpolynom. Wie im Beweis von Satz 3.5 betrachten wir die Funktion

$$h(t) := f(t) - p(t) - \frac{s(t)}{s(x)}\big(f(x) - p(x)\big), \quad x \in [a, b].$$

Da sie m doppelte Nullstellen in den Punkten $x_1, x_2, \ldots, x_m$, sowie eine einfache Nullstelle in $t = x$ besitzt, folgt, dass h' $2m$ Nullstellen $\tau_1^{(1)} < \tau_2^{(1)} < \cdots < \tau_{2m}^{(1)}$ in $[a, b]$ besitzt, h'' $2m - 1$ Nullstellen $a < \tau_1^{(2)} < \tau_2^{(2)} < \cdots < \tau_{2m-1}^{(2)} < b$ besitzt, und so weiter. Schließlich besitzt $h^{(2m)}$ eine Nullstelle $\xi \in (a, b)$. Da $p^{(2m)} \equiv 0$ gilt, folgt

$$0 = h^{(2m)}(\xi) := f^{(2m)}(\xi) - \frac{(2m)!}{s(x)}\big(f(x) - p(x)\big),$$

beziehungsweise

$$f(x) - p(x) = \frac{f^{(2m)}(\xi)}{(2m)!}\prod_{i=1}^{m}(x - x_i)^2.$$

Da die x_i gerade die Nullstellen des $(m + 1)$-ten Orthonormalpolynoms sind, gilt

$$f(x) - p(x) = \frac{f^{(2m)}(\xi)}{(2m)!\gamma_m^2}u_m^2(x).$$

Es folgt

$$\begin{aligned}\big|I_w[f] - G_w[f]\big| &= \big|I_w[f] - G_w[p]\big| = \big|I_w[f - p]\big| \\ &\le \frac{\|f^{(2m)}\|_{C([a,b])}}{(2m)!\gamma_m^2}\underbrace{\int_a^b u_m^2(x)w(x)\,\mathrm{d}x}_{=1} = \frac{\|f^{(2m)}\|_{C([a,b])}}{(2m)!\gamma_m^2}.\end{aligned}$$

□

Bemerkung 6.37 Gemäß (6.18) ist für die Gauß-Legendre-Formel ($w \equiv 1$)

$$\gamma_m^2 = \frac{2m + 1}{2} \cdot \frac{(2m)!^2}{2^{2m}m!^4} \approx \frac{4^m}{\pi},$$

wobei sich die letzte Abschätzung Dank der *Stirlingschen Formel*

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \underbrace{e^{\lambda(n)}}_{\approx 1} \quad \text{mit} \quad \frac{1}{12n + 1} < \lambda(n) < \frac{1}{12n}$$

ergibt. △

7. Lineare Ausgleichsprobleme

7.1 Normalengleichungen

Im folgenden sei $\mathbf{A} \in \mathbb{R}^{m\times n}$ und $\mathbf{b} \in \mathbb{R}^m$. Gesucht ist ein Vektor $\mathbf{x} \in \mathbb{R}^n$ mit

$$\mathbf{A}\mathbf{x} \approx \mathbf{b}.$$

Da wir m Gleichungen für n Unbekannte haben, ist das lineare Gleichungssystem im allgemeinen nicht — oder nicht eindeutig — lösbar. Ist $m > n$, dann nennen wir das lineare Gleichungssystem *überbestimmt*, ist $m < n$, dann nennen wir es *unterbestimmt*.
In den Anwendungen treten häufig überbestimmte Probleme auf, weil es darum geht, Modellparameter an Messdaten anzupassen.

Beispiel 7.1 Unter Einfluss der Schwerkraft fliegen geworfene Körper auf Parabeln. Hat der Körper die Anfangsgeschwindigkeit $\mathbf{v} = (v_x, v_y)$ zum Zeitpunkt $t = 0$ am Punkt $\mathbf{0}$ und fliegt er anschließend nur unter Einfluss der Schwerkraft, so ist er zum Zeitpunkt $t > 0$ am Ort

$$x = v_x t, \qquad y = v_y t - \frac{1}{2} g t^2,$$

wobei g die Erdbeschleunigung ist. Die Anfangsgeschwindigkeit v_y und die Erdbeschleunigung g seien unbekannt und sollen aus Messungen bestimmt werden. Hierzu wurde die Höhe über Grund des Körpers zu folgenden Zeiten gemessen:

i	1	2	3	4	5	6	7
t_i [s]	0.1	0.4	0.5	0.9	1.0	1.2	2.0
y_i [m]	0.96	3.26	3.82	5.11	5.2	5.05	0.58

Es ergeben sich damit sieben Gleichungen für die zwei unbekannten Parameter v_y und g:

$$y_i = t_i v_y - \frac{1}{2} t_i^2 g, \qquad i = 1, 2, \ldots, 7.$$

Führt man die Matrix $\mathbf{A} \in \mathbb{R}^{7\times 2}$ mit

$$a_{i,1} = t_i, \qquad a_{i,2} = -\frac{1}{2} t_i^2, \qquad i = 1, 2, \ldots, 7,$$

ein, so ergibt sich

$$\mathbf{A} \begin{bmatrix} v_y \\ g \end{bmatrix} = \begin{bmatrix} 0.1 & -0.005 \\ 0.4 & -0.08 \\ 0.5 & -0.125 \\ 0.9 & -0.405 \\ 1.0 & -0.5 \\ 1.2 & -0.72 \\ 2.0 & -2.0 \end{bmatrix} \cdot \begin{bmatrix} v_y \\ g \end{bmatrix} = \begin{bmatrix} 0.96 \\ 3.26 \\ 3.82 \\ 5.11 \\ 5.2 \\ 5.05 \\ 0.5 \end{bmatrix} = \mathbf{y}. \tag{7.1}$$

△

In Beispiel 7.1 erhalten wir also ein überbestimmtes Gleichungssystem mit $m = 7$ und $n = 2$, das keine klassische Lösung besitzt. Die Ausgleichsrechnung liefert nun eine Methode, sinnvolle Lösungen von überbestimmten Gleichungssystemen zu definieren.

Da wir für $m > n$ die m Gleichungen im allgemeinen nicht alle exakt erfüllen können, suchen wir nun nach Vektoren $\mathbf{x} \in \mathbb{R}^n$, für die das *Residuum*

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x} \tag{7.2}$$

möglichst klein ist.

Definition 7.2 Für eine Matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ und ein $\mathbf{b} \in \mathbb{R}^m$ heißt das Problem

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \to \min \tag{7.3}$$

ein lineares **Ausgleichsproblem**. Eine Lösung $\mathbf{x} \in \mathbb{R}^n$ des Ausgleichsproblems heißt **Ausgleichslösung** oder **kleinste-Quadrate-Lösung**.

Bemerkung 7.3 Der Lösungsbegriff in (7.3) ist eine Verallgemeinerung der klassischen Lösung. Ist nämlich $m = n$ und ist $\mathbf{x} \in \mathbb{R}^n$ eine klassische Lösung, das heißt, gilt $\mathbf{A}\mathbf{x} = \mathbf{b}$, dann ist offensichtlich $\mathbf{x}$ ebenfalls eine Lösung von (7.3). △

Satz 7.4 Die Lösungen von (7.3) sind genau die Lösungen der **Gaußschen Normalengleichungen**

$$\mathbf{A}^\intercal\mathbf{A}\mathbf{x} = \mathbf{A}^\intercal\mathbf{b}, \tag{7.4}$$

insbesondere existiert eine Lösung $\mathbf{x}$. Ist $\mathbf{z}$ eine weitere Lösung, so gilt $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{z}$. Das Residuum (7.2) ist eindeutig bestimmt und genügt der Gleichung $\mathbf{A}^\intercal\mathbf{r} = \mathbf{0}$.

Beweis. Das Bild der Matrix $\mathbf{A}$ ist der lineare Teilraum

$$\operatorname{img}(\mathbf{A}) = \{\mathbf{A}\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\} \subset \mathbb{R}^m,$$

der von den Spalten von $\mathbf{A}$ aufgespannt wird. Wegen

$$\begin{aligned}\operatorname{img}(\mathbf{A})^\perp &= \{\mathbf{r} \in \mathbb{R}^m : \mathbf{r}^\intercal\mathbf{z} = 0 \text{ für alle } \mathbf{z} \in \operatorname{img}(\mathbf{A})\}\\ &= \{\mathbf{r} \in \mathbb{R}^m : \mathbf{r}^\intercal\mathbf{A} = \mathbf{0}\}\\ &= \operatorname{kern}(\mathbf{A}^\intercal),\end{aligned}$$

folgt

$$\mathbb{R}^m = \operatorname{img}(\mathbf{A}) \overset{\perp}{\oplus} \operatorname{kern}(\mathbf{A}^\intercal),$$

das heißt, der Vektor $\mathbf{b} \in \mathbb{R}^m$ lässt sich eindeutig schreiben als

$$\mathbf{b} = \mathbf{y} + \mathbf{r}, \quad \mathbf{y} \in \operatorname{img}(\mathbf{A}), \quad \mathbf{r} \in \operatorname{kern}(\mathbf{A}^\intercal).$$

Folglich gibt es mindestens ein $\mathbf{x} \in \mathbb{R}^n$ mit $\mathbf{Ax} = \mathbf{y}$ und es gilt

$$\mathbf{A}^\intercal \mathbf{b} = \mathbf{A}^\intercal \mathbf{y} + \underbrace{\mathbf{A}^\intercal \mathbf{r}}_{=\mathbf{0}} = \mathbf{A}^\intercal \mathbf{Ax},$$

das heißt, $\mathbf{x}$ löst die Normalengleichungen (7.4).
Um zu zeigen, dass $\mathbf{x}$ auch Ausgleichslösung ist, setzen wir für beliebiges $\mathbf{z} \in \mathbb{R}^n$

$$\mathbf{y} := \mathbf{Az} - \mathbf{Ax}, \quad \mathbf{r} := \mathbf{b} - \mathbf{Ax}.$$

Nun gilt

$$\|\mathbf{b} - \mathbf{Az}\|_2^2 = \|(\mathbf{r} + \mathbf{Ax}) - (\mathbf{y} + \mathbf{Ax})\|_2^2 = \|\mathbf{r} - \mathbf{y}\|_2^2 = \|\mathbf{r}\|_2^2 - 2\mathbf{r}^\intercal \mathbf{y} + \|\mathbf{y}\|_2^2.$$

Wegen

$$\mathbf{A}^\intercal \mathbf{r} = \mathbf{A}^\intercal(\mathbf{b} - \mathbf{Ax}) = \mathbf{A}^\intercal \mathbf{b} - \mathbf{A}^\intercal \mathbf{Ax} = \mathbf{0}$$

folgt weiter $\mathbf{r}^\intercal \mathbf{y} = \mathbf{r}^\intercal \mathbf{A}(\mathbf{z} - \mathbf{x}) = 0$ und wir schließen

$$\|\mathbf{b} - \mathbf{Az}\|_2^2 = \|\mathbf{r}\|_2^2 + \|\mathbf{y}\|_2^2 \geq \|\mathbf{r}\|_2^2 = \|\mathbf{b} - \mathbf{Ax}\|_2^2.$$

Der Vektor $\mathbf{x}$ minimiert demnach (7.3). Gleichheit gilt in dieser Abschätzung nur dann für ein $\mathbf{z} \in \mathbb{R}^n$, wenn

$$0 = \|\mathbf{y}\|_2^2 = \|\mathbf{A}(\mathbf{z} - \mathbf{x})\|_2^2.$$

Dies ist genau dann der Fall, wenn

$$\mathbf{z} - \mathbf{x} \in \operatorname{kern}(\mathbf{A}) \subset \operatorname{kern}(\mathbf{A}^\intercal \mathbf{A}),$$

Es ist also

$$\mathbf{0} = \mathbf{A}^\intercal \mathbf{A}(\mathbf{z} - \mathbf{x}) = \mathbf{A}^\intercal \mathbf{Az} - \mathbf{A}^\intercal \mathbf{Ax} = \mathbf{A}^\intercal \mathbf{Az} - \mathbf{A}^\intercal \mathbf{b}.$$

das heißt, auch $\mathbf{z}$ löst die Nomalengleichungen (7.4). □

Bemerkung 7.5 Aus $\mathbf{A}^\intercal \mathbf{r} = \mathbf{0}$ folgt, dass das Residuum senkrecht auf den Spalten von $\mathbf{A}$ steht. Das Residuum $\mathbf{r}$ ist folglich eine Normale zum von den Spalten der Matrix $\mathbf{A}$ aufgespannten Raum. Daher erklärt sich die Bezeichnung Normalengleichungen. △

Satz 7.6 Die Matrix $\mathbf{A}^\intercal \mathbf{A} \in \mathbb{R}^{n \times n}$ ist symmetrisch und positiv semidefinit. Darüber hinaus ist $\mathbf{A}^\intercal \mathbf{A}$ genau dann positiv definit, wenn der Kern von $\mathbf{A}$ trivial ist, das heißt, wenn $\operatorname{kern}(\mathbf{A}) = \{\mathbf{0}\}$. Dies ist genau dann der Fall, wenn die Spalten von $\mathbf{A}$ linear unabhängig sind.

Beweis. Offensichtlich ist $\mathbf{A}^\intercal \mathbf{A}$ symmetrisch und wegen

$$\mathbf{x}^\intercal \mathbf{A}^\intercal \mathbf{Ax} = \|\mathbf{Ax}\|_2^2 \geq 0 \text{ für alle } \mathbf{x} \in \mathbb{R}^n$$

auch positiv semidefinit. Ist $\operatorname{kern} \mathbf{A} = \{\mathbf{0}\}$, so gilt Gleichheit ("=") nur im Fall $\mathbf{x} = \mathbf{0}$, das heißt, $\mathbf{A}^\intercal \mathbf{A}$ ist positiv definit. □

Beispiel 7.7 (Fortsetzung von Beipiel 7.1) Für das überbestimmte System (7.1) ist die gesuchte Normalengleichung

$$\mathbf{A}^\intercal\mathbf{A} = \begin{bmatrix} 0.1 & 0.4 & 0.5 & 0.9 & 1.0 & 1.2 & 2.0 \\ -0.005 & -0.08 & -0.125 & -0.405 & -0.5 & -0.72 & -2.0 \end{bmatrix} \begin{bmatrix} 0.1 & -0.005 \\ 0.4 & -0.08 \\ 0.5 & -0.125 \\ 0.9 & -0.405 \\ 1.0 & -0.5 \\ 1.2 & -0.72 \\ 2.0 & -2.0 \end{bmatrix}$$

$$= \begin{bmatrix} 7.6700 & -5.8235 \\ -5.8235 & 4.954475 \end{bmatrix}$$

und

$$\mathbf{A}^\intercal\mathbf{y} = \begin{bmatrix} 0.1 & 0.4 & 0.5 & 0.9 & 1.0 & 1.2 & 2.0 \\ -0.005 & -0.08 & -0.125 & -0.405 & -0.5 & -0.72 & -2.0 \end{bmatrix} \begin{bmatrix} 0.96 \\ 3.26 \\ 3.82 \\ 5.11 \\ 5.2 \\ 5.05 \\ 0.58 \end{bmatrix}$$

$$= \begin{bmatrix} 20.3290 \\ -10.20865 \end{bmatrix}.$$

Die gesuchte Ausgleichslösung (v_y, g) erfüllt also

$$\begin{bmatrix} 7.6700 & -5.8235 \\ -5.8235 & 4.954475 \end{bmatrix} \begin{bmatrix} v_y \\ g \end{bmatrix} = \begin{bmatrix} 20.3290 \\ -10.20865 \end{bmatrix}.$$

Die Matrix $\mathbf{A}^\intercal\mathbf{A}$ ist in der Tat symmetrisch und positiv definit, und die damit eindeutige Lösung (v_y, g) ist mit drei Stellen Genauigkeit

$$\begin{bmatrix} v_y \\ g \end{bmatrix} = \begin{bmatrix} 10.1 \\ 9.81 \end{bmatrix}.$$

△

Prinzipiell könnte man das lineare Ausgleichsproblem mit den Gaußschen Normalengleichungen auch numerisch behandeln, etwa mittels einer Cholesky-Zerlegung der Matrix $\mathbf{A}^\intercal\mathbf{A}$ in den Normalengleichungen. Dies ist jedoch ein typisches Beispiel für einen numerischen Algorithmus, der deutlich weniger stabil ist als das eigentlich zu lösende Problem. Man sieht das am einfachsten in dem Fall, dass $\mathbf{A} \in \mathbb{R}^{n\times n}$ invertierbar ist. Dann ist die Kondition des linearen Gleichungssystems $\mathbf{A}\mathbf{x} = \mathbf{b}$

$$\operatorname{cond}_2 \mathbf{A} = \|\mathbf{A}\|_2 \cdot \|\mathbf{A}^{-1}\|_2 = \sqrt{\frac{\lambda_{\max}(\mathbf{A}^\intercal\mathbf{A})}{\lambda_{\min}(\mathbf{A}^\intercal\mathbf{A})}},$$

aber die Kondition der Normalengleichung ist

$$\operatorname{cond}_2(\mathbf{A}^\intercal\mathbf{A}) = \|\mathbf{A}^\intercal\mathbf{A}\|_2 \cdot \|(\mathbf{A}^\intercal\mathbf{A})^{-1}\|_2 = \frac{\lambda_{\max}(\mathbf{A}^\intercal\mathbf{A})}{\lambda_{\min}(\mathbf{A}^\intercal\mathbf{A})} = (\operatorname{cond}_2 \mathbf{A})^2.$$

Wir wollen daher in den nachfolgenden Abschnitten einen Algorithmus herleiten, der die Normalengleichungen vermeidet.

7.2 QR-Zerlegung

Im folgenden sei $\mathbf{A} \in \mathbb{R}^{m\times n}$, $m \geq n$, eine gegebene Matrix mit $\operatorname{rang} \mathbf{A} = n$. Die Grundidee der QR-Zerlegung ist eine Faktorisierung $\mathbf{A} = \mathbf{QR}$ in eine rechte obere Dreiecksmatrix $\mathbf{R} \in \mathbb{R}^{m\times n}$ und eine unitäre Matrix $\mathbf{Q} \in \mathbb{R}^{m\times m}$.

Definition 7.8 Eine Matrix $\mathbf{Q} \in \mathbb{R}^{n\times n}$ heißt **orthogonal**, falls

$$\mathbf{Q}^\intercal \mathbf{Q} = \mathbf{I},$$

das heißt, falls die Spalten von $\mathbf{Q}$ eine Orthonormalbasis bilden.

Eigenschaften orthogonaler Matrizen:

1. Wegen
$$\|\mathbf{Qx}\|_2^2 = (\mathbf{Qx})^\intercal \mathbf{Qx} = \mathbf{x}^\intercal \underbrace{\mathbf{Q}^\intercal \mathbf{Q}}_{=\mathbf{I}} \mathbf{x} = \mathbf{x}^\intercal \mathbf{x} = \|\mathbf{x}\|_2^2$$
gilt
$$\|\mathbf{Qx}\|_2 = \|\mathbf{x}\|_2 \quad \forall \mathbf{x} \in \mathbb{R}^n.$$
2. Es gilt
$$\operatorname{cond}_2 \mathbf{Q} = 1,$$
da
$$\|\mathbf{Q}\|_2 = \max_{\|\mathbf{x}\|_2=1} \|\mathbf{Qx}\|_2 = 1$$
und
$$\|\mathbf{Q}^{-1}\|_2 = \|\mathbf{Q}^\intercal\|_2 = \|\mathbf{Q}\|_2 = 1.$$
3. Mit $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{n\times n}$ orthogonal ist auch $\mathbf{PQ}$ orthogonal, da
$$(\mathbf{PQ})^\intercal \mathbf{PQ} = \mathbf{Q}^\intercal \underbrace{\mathbf{P}^\intercal \mathbf{P}}_{=\mathbf{I}} \mathbf{Q} = \mathbf{Q}^\intercal \mathbf{Q} = \mathbf{I}.$$

Definition 7.9 Sei $\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$. Die Matrix

$$\mathbf{P} = \mathbf{I} - \frac{2}{\|\mathbf{v}\|_2^2} \mathbf{v}\mathbf{v}^\intercal \in \mathbb{R}^{n\times n}$$

heißt **Householder-Transformation**.

Lemma 7.10 $\mathbf{P}$ ist eine symmetrische orthogonale Matrix mit

$$\mathbf{Pv} = -\mathbf{v}$$

und für alle $\mathbf{w} \in \mathbb{R}^n$ mit $\mathbf{w} \perp \mathbf{v}$ gilt

$$\mathbf{Pw} = \mathbf{w}.$$

Beweis. Aus der Definition von $\mathbf{P}$ folgt unmittelbar, dass $\mathbf{P}$ symmetrisch ist. Weiter gilt

$$\begin{aligned}\mathbf{P}^\mathsf{T}\mathbf{P} = \mathbf{P}^2 &= \left(\mathbf{I} - \frac{2}{\|\mathbf{v}\|_2^2}\mathbf{v}\mathbf{v}^\mathsf{T}\right)\left(\mathbf{I} - \frac{2}{\|\mathbf{v}\|_2^2}\mathbf{v}\mathbf{v}^\mathsf{T}\right)\\ &= \mathbf{I} - \frac{4}{\|\mathbf{v}\|_2^2}\mathbf{v}\mathbf{v}^\mathsf{T} + \frac{4}{\|\mathbf{v}\|_2^4}\mathbf{v}\underbrace{\mathbf{v}^\mathsf{T}\mathbf{v}}_{=\|\mathbf{v}\|_2^2}\mathbf{v}^\mathsf{T}\\ &= \mathbf{I} - \frac{4}{\|\mathbf{v}\|_2^2}\mathbf{v}\mathbf{v}^\mathsf{T} + \frac{4}{\|\mathbf{v}\|_\mathbf{2}^\mathbf{2}}\mathbf{v}\mathbf{v}^\mathsf{T} = \mathbf{I}.\end{aligned}$$

Außerdem ergibt sich für den Vektor $\mathbf{v}$ aus der Definition von $\mathbf{P}$

$$\mathbf{P}\mathbf{v} = \mathbf{I}\mathbf{v} - \frac{2}{\|\mathbf{v}\|_2^2}\mathbf{v}\underbrace{\mathbf{v}^\mathsf{T}\mathbf{v}}_{=\|\mathbf{v}\|_2^2} = \mathbf{v} - 2\mathbf{v} = -\mathbf{v}$$

und für beliebiges $\mathbf{w} \perp \mathbf{v}$

$$\mathbf{P}\mathbf{w} = \mathbf{I}\mathbf{w} - \frac{2}{\|\mathbf{v}\|_2^2}\mathbf{v}\underbrace{\mathbf{v}^\mathsf{T}\mathbf{w}}_{=0} = \mathbf{w}.$$

□

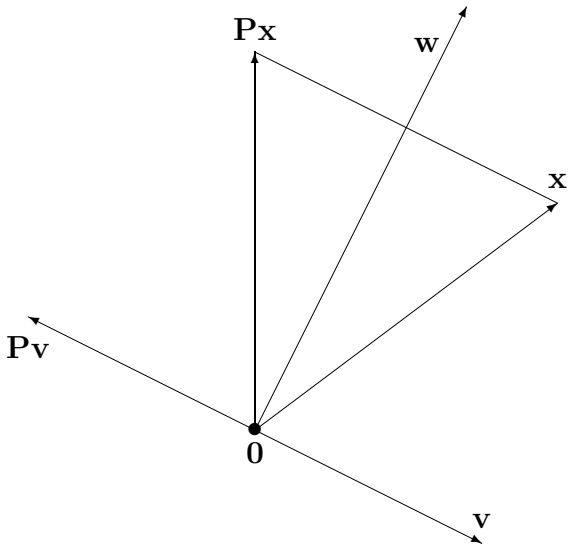


Abbildung 7.1: Householder-Transformationen sind Spiegelungen!

Eine QR-Zerlegung kann erzeugt werden, indem man schrittweise die Matrix $\mathbf{A}$ durch Multiplikation mit geeigneten Householder-Transformationen $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_n$ auf rechte obere Dreicksgestalt bringt. Das nächste Lemma erlaubt uns, solche Householder-Transformationen zu konstruieren, indem es für jedes $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ eine Householder-Transformation $\mathbf{Q}$ angibt, so dass

$$\mathbf{Q}\mathbf{x} = \sigma\mathbf{e}_1 \quad \text{mit} \quad \sigma \in \mathbb{R} \setminus \{0\}.$$

Lemma 7.11 Gegeben sei $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ mit $\mathbf{x} \notin \operatorname{span}\{\mathbf{e}_1\}$. Für

$$\mathbf{v} = \mathbf{x} + \sigma \mathbf{e}_1 \quad \text{mit} \quad \sigma = \pm \begin{cases} \frac{x_1}{|x_1|}\|\mathbf{x}\|_2, & \text{falls } x_1 \neq 0, \\ \|\mathbf{x}\|_2, & \text{falls } x_1 = 0, \end{cases} \tag{7.5}$$

gilt

$$\left(\mathbf{I} - \frac{2}{\|\mathbf{v}\|_2^2}\mathbf{v}\mathbf{v}^\mathsf{T}\right)\mathbf{x} = -\sigma \mathbf{e}_1.$$

Beweis. Wegen $\mathbf{x} \notin \operatorname{span}\{\mathbf{e}_1\}$ ist $\mathbf{v} \neq \mathbf{0}$. Weiter gilt

$$\|\mathbf{x} + \sigma \mathbf{e}_1\|_2^2 = \|\mathbf{x}\|_2^2 + 2\sigma \mathbf{x}^\mathsf{T}\mathbf{e}_1 + \sigma^2 = 2(\mathbf{x} + \sigma \mathbf{e}_1)^\mathsf{T}\mathbf{x}.$$

Daraus erhält man

$$2\mathbf{v}^\mathsf{T}\mathbf{x} = 2(\mathbf{x} + \sigma \mathbf{e}_1)^\mathsf{T}\mathbf{x} = \|\mathbf{x} + \sigma \mathbf{e}_1\|_2^2 = \|\mathbf{v}\|_2^2,$$

was zusammen mit (7.5) die Darstellung

$$\frac{2}{\|\mathbf{v}\|_2^2}\mathbf{v}(\mathbf{v}^\mathsf{T}\mathbf{x}) = \mathbf{x} + \sigma \mathbf{e}_1$$

liefert. Dies impliziert die Behauptung. □

Bemerkung 7.12 Damit im Fall $x_1 \neq 0$ bei der Berechnung von $\mathbf{v}$ keine Auslöschung auftritt, wählen wir σ mit dem oberen Vorzeichen, das heißt

$$\mathbf{v} = \mathbf{x} + \frac{x_1}{|x_1|}\|\mathbf{x}\|_2 \mathbf{e}_1, \qquad \|\mathbf{v}\|_2^2 = 2\|\mathbf{x}\|_2^2 + 2|x_1|\|\mathbf{x}\|_2. \tag{7.6}$$

△

Satz 7.13 Sei $\mathbf{A} \in \mathbb{R}^{m \times n}$ mit $\operatorname{rang}(\mathbf{A}) = n$ (also $m \geq n$). Dann existiert eine unitäre Matrix $\mathbf{Q} \in \mathbb{R}^{m \times m}$ und eine obere Dreiecksmatrix $\mathbf{R} \in \mathbb{R}^{m \times n}$ mit

$$\mathbf{A} = \mathbf{Q} \cdot \mathbf{R} = \mathbf{Q} \cdot \underbrace{\begin{bmatrix} \star & \cdots & \star \\ & \ddots & \vdots \\ & & \star \\ \hline & \mathbf{0} & \end{bmatrix}}_{n} \begin{matrix} \rceil \\ n \\ \rfloor \\ \rceil \\ m-n \\ \rfloor \end{matrix}.$$

Beweis. Wir bestimmen die gesuchte Zerlegung, indem wir in jedem Schritt eine Householder-Transformation an $\mathbf{A}$ heranmultiplizieren, um sukzessive die Spalten von 1 bis n von $\mathbf{R}$ zu erhalten:

$$\mathbf{Q}_{n-1}\mathbf{Q}_{n-2}\cdots\mathbf{Q}_1\mathbf{A} = \mathbf{R}. \tag{7.7}$$

Wegen der Symmetrie der $\mathbf{Q}_i$, $1 \leq i < n$, ist $\mathbf{Q}$ dann gegeben durch

$$\mathbf{Q} = \mathbf{Q}_1 \mathbf{Q}_2 \cdots \mathbf{Q}_{n-1}.$$

Im ersten Schritt setzen wir $\mathbf{A}_1 := \mathbf{A}$ und $\mathbf{x} = \mathbf{a}_1$ (erste Spalte von $\mathbf{A}_1$) und bestimmen die Householder-Transformation $\mathbf{Q}_1 \in \mathbb{R}^{m \times m}$ gemäß (7.6). Es folgt

$$\mathbf{Q}_1 \mathbf{a}_1 = r_{1,1} \mathbf{e}_1 \quad \text{mit} \quad |r_{1,1}| = \|\mathbf{a}_1\|_2 \neq 0,$$

beziehungsweise

$$\mathbf{Q}_1 \mathbf{A}_1 = \left[\begin{array}{c|c} r_{1,1} & \mathbf{r}_1 \\ \hline \mathbf{0} & \mathbf{A}_2 \end{array}\right], \qquad \mathbf{A}_2 \in \mathbb{R}^{(m-1)\times(n-1)}, \qquad \mathbf{r}_1^\mathsf{T} \in \mathbb{R}^{n-1}.$$

Im nächsten Schritt setzen wir $\mathbf{x} = \mathbf{a}_2 \in \mathbb{R}^{m-1}$ (erste Spalte von $\mathbf{A}_2$) und wählen wiederum die Householder-Matrix $\widetilde{\mathbf{Q}}_2 \in \mathbb{R}^{(m-1)\times(m-1)}$ gemäß (7.6). Wir erhalten

$$\widetilde{\mathbf{Q}}_2 \mathbf{A}_2 = \left[\begin{array}{c|c} r_{2,2} & \mathbf{r}_2 \\ \hline \mathbf{0} & \mathbf{A}_3 \end{array}\right], \qquad |r_{2,2}| = \|\mathbf{a}_2\|_2 \neq 0, \qquad \mathbf{A}_3 \in \mathbb{R}^{(m-2)\times(n-2)}, \qquad \mathbf{r}_2^\mathsf{T} \in \mathbb{R}^{n-2},$$

beziehungsweise

$$\underbrace{\left[\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & \widetilde{\mathbf{Q}}_2 \end{array}\right]}_{=:\mathbf{Q}_2} \mathbf{Q}_1 \mathbf{A} = \left[\begin{array}{c|c} r_{1,1} & \mathbf{r}_1 \\ \hline \mathbf{0} & \widetilde{\mathbf{Q}}_2 \mathbf{A}_2 \end{array}\right] = \left[\begin{array}{c|cc} r_{1,1} & \multicolumn{2}{c}{\mathbf{r}_1} \\ \hline & r_{2,2} & \mathbf{r}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3 \end{array}\right].$$

Die erste Zeile $\mathbf{r}_1$ verändert sich nicht mehr. Die Matrix $\mathbf{Q}_2$ kann ebenfalls als $(m \times m)$-Householder-Transformation aufgefasst werden mit $\mathbf{v} = \left[\begin{smallmatrix} 0 \\ \tilde{\mathbf{v}} \end{smallmatrix}\right]$.

Auf diese Weise erhalten wir sukzessive die gewünschte Zerlegung (7.7). Man beachte, dass $|r_{i,i}| = \|\mathbf{a}_i\|$ $(1 \leq i \leq n)$ immer von Null verschieden ist, da ansonsten $\mathbf{A}_i$ und damit auch $\mathbf{A}$ einen Rangdefekt hätte. □

Bemerkungen 7.14

1. Bei der Implementierung ist darauf zu achten, dass Householder-Transformationen $\mathbf{P}$ *niemals* explizit gebildet werden, denn sonst kostet die Berechnung $\mathbf{P} \cdot \mathbf{A}$ $m^2 n$ Multiplikationen. Besser ist

$$\mathbf{P}\mathbf{A} = \mathbf{A} - \frac{2}{\|\mathbf{v}\|_2^2} \mathbf{v} \underbrace{\mathbf{v}^\mathsf{T} \mathbf{A}}_{=\mathbf{w}^\mathsf{T}} = \mathbf{A} - \frac{2}{\|\mathbf{v}\|_2^2} \mathbf{v}\mathbf{w}^\mathsf{T}, \quad \mathbf{w} = \mathbf{A}^\mathsf{T}\mathbf{v}$$

   mit $\mathcal{O}(mn)$ Multiplikationen. Wenn man $\mathbf{P}$ später verwenden will, speichert man den Vektor $\mathbf{v}$ ab.

2. Die während der QR-Zerlegung anfallenden Vektoren $\mathbf{v}_i = [0, \ldots, 0, 1, v_{i,i+1}, \ldots, v_{i,m}]^\mathsf{T}$ lassen sich analog zur LR-Zerlegung wieder in der freiwerdenden linken unteren Dreiecksmatrix von $\mathbf{A}$ speichern. Die Matrix $\mathbf{Q}$ ist dann wie folgt gegeben

$$\mathbf{Q} = \prod_{i=1}^{n} \left(\mathbf{I} - \frac{2}{\|\mathbf{v}_i\|_2^2} \mathbf{v}_i \mathbf{v}_i^\mathsf{T} \right).$$

△

Algorithmus 7.15
Initialisierung: sei $\mathbf{A}_1 := \mathbf{A}$ und $\mathbf{a}_1$ die erste Spalte
Für $i = 1, 2, \ldots, n$:

① setze $\mathbf{x} = \mathbf{a}_i$ und bestimme gemäß (7.6)

$$\mathbf{v} = \mathbf{x} + \frac{x_1}{|x_1|}\|\mathbf{x}\|_2 \mathbf{e}_1,$$

② setze

$$\beta = \frac{2}{\|\mathbf{v}\|_2^2} = \frac{1}{\|\mathbf{x}\|_2^2 + |x_1|\|\mathbf{x}\|_2},$$

vgl. (7.6).

③ berechne $\mathbf{w} = \beta \mathbf{A}_i^\intercal \mathbf{v}$

④ ersetze $\mathbf{A}_i$ durch $\mathbf{A}_i - \mathbf{v}\mathbf{w}^\intercal$

⑤ $\mathbf{A}_{i+1}$ bezeichne die rechte untere $(m-i+1) \times (n-i+1)$-Teilmatrix von $\mathbf{A}_i$, und $\mathbf{a}_{i+1}$ deren erste Spalte

Aufwand: Wir bilanzieren den Aufwand im i-ten Schritt:

$\mathbf{v}$:	$m-i+3$ Multiplikationen
β:	2 Multiplikationen
$\mathbf{w}$:	$(m-i+2)(n-i+1)$ Multiplikationen
$\mathbf{A}_i$:	$(m-i)(n-i+1)$ Multiplikationen
	$\approx 2(m-i+1)(n-i+1)$ Multiplikationen

Für den Gesamtaufwand ergibt sich daher

$$\begin{aligned} 2\sum_{i=1}^{n}(m-i+1)(n-i+1) &\overset{j:=n-i}{=} 2\sum_{j=1}^{n} j(m-n+j) \\ &= 2\sum_{j=1}^{n} j^2 + 2(m-n)\sum_{j=1}^{n} j \\ &= \frac{2}{3}n^3 + (m-n)n^2 + \mathcal{O}(mn) \\ &= mn^2 - \frac{1}{3}n^3 + \mathcal{O}(mn), \end{aligned}$$

dies bedeutet dass der Aufwand etwa doppelt so hoch ist wie bei der LR-Zerlegung.

Die QR-Zerlegung kann wie die LR-Zerlegung zur Lösung eines nichtsingulären linearen Gleichungssystems $\mathbf{A}\mathbf{x} = \mathbf{b}$ (also $m = n$) verwendet werden. Dies geschieht in folgender Weise: Zerlege $\mathbf{A} = \mathbf{Q}\mathbf{R}$, und löse $\mathbf{Q}\mathbf{R}\mathbf{x} = \mathbf{b}$ durch Rückwärtssubstitution

$$\mathbf{R}\mathbf{x} = \underbrace{\mathbf{Q}^\intercal \mathbf{b}}_{\mathcal{O}(n^2) \text{ Operationen}}.$$

Bemerkung 7.16 Die QR-Zerlegung gehört zu den "stabilsten" Algorithmen in der numerischen linearen Algebra. Der Grund dafür ist, dass Orthogonaltransformationen keine Fehlerverstärkung bringen, da $\operatorname{cond}_2 \mathbf{Q} = 1$. Die abschließende Rückwärtssubstitution hat die gleiche Kondition wie das Ausgangsproblem, da wegen $\|\mathbf{Qx}\|_2 = \|\mathbf{x}\|_2$ folgt

$$\begin{aligned}
\|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2 &= \Big(\max_{\|\mathbf{x}\|_2=1} \|\mathbf{Ax}\|_2\Big)\Big(\max_{\|\mathbf{x}\|_2} \|\mathbf{A}^{-1}\mathbf{x}\|_2\Big) \\
&= \Big(\max_{\|\mathbf{x}\|_2=1} \|\mathbf{QRx}\|_2\Big)\Big(\max_{\|\mathbf{x}\|_2=1} \|\mathbf{R}^{-1}\mathbf{Q}^\intercal\mathbf{x}\|_2\Big) \\
&= \Big(\max_{\|\mathbf{x}\|_2=1} \|\mathbf{Rx}\|_2\Big)\Big(\max_{\|\mathbf{y}\|_2=1} \|\mathbf{R}^{-1}\mathbf{y}\|_2\Big) \\
&= \|\mathbf{R}\|_2\|\mathbf{R}^{-1}\|_2,
\end{aligned}$$

das heißt

$$\operatorname{cond}_2 \mathbf{R} = \operatorname{cond}_2 \mathbf{A}.$$

△

Beispiel 7.17 Gesucht ist die QR-Zerlegung von

$$\mathbf{A} = \mathbf{A}_1 = \begin{bmatrix} 1 & -11/2 \\ -2 & 0 \\ 2 & -1 \end{bmatrix}.$$

Die erste Spalte von $\mathbf{A}$ ist

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix}, \qquad \|\mathbf{a}_1\|_2 = \sqrt{1+4+4} = 3.$$

Also ist

$$\mathbf{v} = \mathbf{a}_1 + \operatorname{sign}(a_{1,1})\|\mathbf{a}_1\|_2\mathbf{e}_1 = \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix} + 3\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ -2 \\ 2 \end{bmatrix}.$$

Somit folgt

$$\beta = \frac{1}{\|\mathbf{a}_1\|_2^2 + |a_{1,1}|\|\mathbf{a}_1\|_2} = \frac{1}{12}$$

woraus sich

$$\mathbf{w} = \beta\mathbf{A}_1^\intercal\mathbf{v} = \frac{1}{12}\begin{bmatrix} 1 & -2 & 2 \\ -11/2 & 0 & -1 \end{bmatrix}\begin{bmatrix} 4 \\ -2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

ergibt. Dies bedeutet

$$\mathbf{Q}_1\mathbf{A}_1 = \mathbf{A}_1 - \mathbf{vw}^\intercal = \begin{bmatrix} 1 & -11/2 \\ -2 & 0 \\ 2 & -1 \end{bmatrix} - \begin{bmatrix} 4 & -8 \\ -2 & 4 \\ 2 & -4 \end{bmatrix} = \begin{bmatrix} -3 & 5/2 \\ 0 & -4 \\ 0 & 3 \end{bmatrix}.$$

Die erste Spalte stimmt dabei mit $-\sigma\mathbf{e}_1$ überein, so war die Householder-Transformation schließlich konstruiert. Sie ist übrigens gegeben durch

$$\mathbf{Q}_1 = \mathbf{I} - \beta\mathbf{vv}^\intercal = \frac{1}{3}\begin{bmatrix} -1 & 2 & -2 \\ 2 & 2 & 1 \\ -2 & 1 & 2 \end{bmatrix}.$$

Nun ist

$$\mathbf{A}_2 = \mathbf{a}_2 = \begin{bmatrix} -4 \\ 3 \end{bmatrix}, \qquad \|\mathbf{a}_2\|_2 = \sqrt{16+9} = 5.$$

Mit

$$\mathbf{v} = \mathbf{a}_2 + \operatorname{sign}(a_{2,1})\|\mathbf{a}_2\|_2 \mathbf{e}_1 \begin{bmatrix} -4 \\ 3 \end{bmatrix} - 5 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -9 \\ 3 \end{bmatrix}$$

und

$$\beta = \frac{1}{\|\mathbf{a}_2\|_2^2 + |a_{2,1}|\|\mathbf{a}_2\|_2} = \frac{1}{45}$$

folgt

$$\mathbf{w} = \beta \mathbf{A}_2^\mathsf{T} \mathbf{v} = \frac{1}{45} \begin{bmatrix} -4 & 3 \end{bmatrix} \begin{bmatrix} -9 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \end{bmatrix}.$$

Damit ergibt sich

$$\widetilde{\mathbf{Q}}_2 \mathbf{A}_2 = \mathbf{A}_2 - \mathbf{v}\mathbf{w}^\mathsf{T} = \begin{bmatrix} -4 \\ 3 \end{bmatrix} - \begin{bmatrix} -9 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}.$$

Dabei hat die Matrix $\mathbf{Q}_2$ die Form

$$\mathbf{Q}_2 = \left[\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I} - \beta \mathbf{v}\mathbf{v}^\mathsf{T} \end{array}\right] = \frac{1}{5} \begin{bmatrix} 5 & 0 & 0 \\ 0 & -4 & -3 \\ 0 & -3 & 4 \end{bmatrix}.$$

Für $\mathbf{Q}$ erhalten wir schließlich

$$\mathbf{Q} = \mathbf{Q}_1 \mathbf{Q}_2 = \frac{1}{15} \begin{bmatrix} -1 & 2 & -2 \\ 2 & 2 & -1 \\ -2 & 1 & 2 \end{bmatrix} \begin{bmatrix} 5 & 0 & 0 \\ 0 & -4 & -3 \\ 0 & -3 & 4 \end{bmatrix} = \frac{1}{15} \begin{bmatrix} -5 & -2 & -14 \\ 10 & -11 & -2 \\ -10 & -10 & 5 \end{bmatrix},$$

während $\mathbf{R}$ gegeben ist durch

$$\mathbf{R} = \begin{bmatrix} -3 & 5/2 \\ 0 & 5 \\ 0 & 0 \end{bmatrix}.$$

△

Bemerkung 7.18 Algorithmus 7.15 bricht zusammen, wenn $\operatorname{rang}(\mathbf{A}) = p < n$. In diesem Fall muss man Spalten von $\mathbf{A}$ permutieren (ähnlich zur Pivotsuche) und erhält eine Faktorisierung der Art

$$\mathbf{Q}^\mathsf{T} \mathbf{A} \mathbf{P} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

mit einer Permutationsmatrix $\mathbf{P} \in \mathbb{R}^{n \times n}$, einer oberen Dreiecksmatrix $\mathbf{R}_1 \in \mathbb{R}^{p \times p}$, und einer eventuell vollbesetzten Matrix $\mathbf{R}_2 \in \mathbb{R}^{p \times (n-p)}$. △

7.3 Methode der Orthogonalisierung

Wir wollen nun die Methode der Orthogonalisierung herleiten, mit deren Hilfe das lineare Ausgleichsproblem ohne Verschlechterung der Kondition lösbar ist. Dazu betrachten wir

zunächst den Fall, dass die Matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ mit $m \geq n = \operatorname{rang}(\mathbf{A})$ Rechtsdreiecksstruktur hat, das heißt

$$\mathbf{A} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

mit einer rechten oberen Dreiecksmatrix $\mathbf{R} \in \mathbb{R}^{n\times n}$. Der Vektor $\mathbf{b} \in \mathbb{R}^m$ sei analog zerlegt:

$$\mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}, \qquad \mathbf{c} \in \mathbb{R}^n, \qquad \mathbf{d} \in \mathbb{R}^{m-n}.$$

Damit haben wir

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 = \left\| \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} - \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{x} \right\|_2^2 = \left\| \begin{bmatrix} \mathbf{c} - \mathbf{R}\mathbf{x} \\ \mathbf{d} - \mathbf{0}\mathbf{x} \end{bmatrix} \right\|_2^2 = \|\mathbf{c} - \mathbf{R}\mathbf{x}\|_2^2 + \|\mathbf{d}\|_2^2.$$

Da die rechte obere Dreiecksmatrix $\mathbf{R}$ invertierbar ist, ist die Lösung des Minimierungsproblems (7.3) offensichtlich gegeben durch $\mathbf{x} = \mathbf{R}^{-1}\mathbf{c} \in \mathbb{R}^n$. Die Größe des Residuums $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$ können wir auch sofort ablesen, nämlich $\|\mathbf{r}\|_2 = \|\mathbf{d}\|_2$.

Satz 7.19 Sei $m \geq n$, $\mathbf{A} \in \mathbb{R}^{m\times n}$ eine Matrix mit linear unabhängigen Spalten und der QR-Zerlegung

$$\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}, \qquad \mathbf{Q} \in \mathbb{R}^{m\times m}, \qquad \mathbf{R} \in \mathbb{R}^{n\times n}.$$

Für beliebiges $\mathbf{b} \in \mathbb{R}^m$ sei

$$\mathbf{Q}^\intercal \mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}, \qquad \mathbf{c} \in \mathbb{R}^n, \qquad \mathbf{d} \in \mathbb{R}^{m-n}.$$

Dann ist die Lösung $\mathbf{x} \in \mathbb{R}^n$ des Ausgleichsproblems (7.3) eindeutig bestimmt durch

$$\mathbf{R}\mathbf{x} = \mathbf{c}.$$

Die Norm des Residuums $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$ ist gegeben durch $\|\mathbf{r}\|_2 = \|\mathbf{d}\|_2$.

Beweis. Wegen $\|\mathbf{Q}\mathbf{z}\|_2 = \|\mathbf{z}\|_2$ für alle $\mathbf{z} \in \mathbb{R}^m$ folgt

$$\begin{aligned} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 = \|\mathbf{Q}(\mathbf{Q}^\intercal\mathbf{b} - \mathbf{Q}^\intercal\mathbf{A}\mathbf{x})\|_2^2 = \|\mathbf{Q}^\intercal\mathbf{b} - \mathbf{Q}^\intercal\mathbf{A}\mathbf{x}\|_2^2 &= \left\| \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} - \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{x} \right\|_2^2 \\ = \|\mathbf{c} - \mathbf{R}\mathbf{x}\|_2^2 + \|\mathbf{d}\|_2^2. \end{aligned}$$

Da $\mathbf{A}$ vollen Rang besitzt, ist $\mathbf{R}$ nicht singulär. Damit wird dieser Ausdruck minimal für die eindeutig bestimmte Lösung $\mathbf{x} := \mathbf{R}^{-1}\mathbf{c} \in \mathbb{R}^n$. □

Beispiel 7.20 Wir führen das Vorgehen wieder mit den Zahlen aus Beispiel 7.1 vor. Die

QR-Zerlegung der Matrix $\mathbf{A}$ aus (7.1) lautet $\mathbf{A} = \mathbf{QR}$ mit

$$\mathbf{Q} = \begin{bmatrix} -0.0361 & -0.0972 & -0.1684 & -0.3121 & -0.3493 & -0.4252 & -0.7488 \\ -0.1444 & -0.3064 & -0.3553 & -0.3943 & -0.3700 & -0.2804 & 0.6229 \\ -0.1805 & -0.3488 & 0.8855 & -0.1444 & -0.1433 & -0.1309 & 0.0562 \\ -0.3250 & -0.3813 & -0.1429 & 0.8069 & -0.1967 & -0.1934 & -0.0375 \\ -0.3611 & -0.3551 & -0.1412 & -0.1959 & 0.7984 & -0.2031 & -0.0802 \\ -0.4333 & -0.2618 & -0.1272 & -0.1905 & -0.2010 & 0.7844 & -0.1887 \\ -0.7222 & 0.6595 & 0.0693 & -0.0202 & -0.0628 & -0.1720 & 0.0690 \end{bmatrix}$$

und

$$\left[\begin{array}{c} \mathbf{R} \\ \hline \mathbf{0} \end{array}\right] = \left[\begin{array}{cc} -2.7695 & 2.1027 \\ 0 & -0.73 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array}\right].$$

Wir berechnen

$$\mathbf{Q}^\intercal \mathbf{y} = \mathbf{Q}^\intercal \begin{bmatrix} 0.96 \\ 3.26 \\ 3.82 \\ 5.11 \\ 5.2 \\ 5.05 \\ 0.58 \end{bmatrix} = \begin{bmatrix} -7.3404 \\ -7.1590 \\ -0.0037 \\ -0.0063 \\ 0.0058 \\ -0.0055 \\ 0.0046 \end{bmatrix}$$

Die gewünschte Partitionierung von $\mathbf{Q}^\intercal \mathbf{y}$ ist damit

$$\mathbf{Q}^\intercal \mathbf{y} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}, \qquad \mathbf{c} = \begin{bmatrix} -7.3404 \\ -7.1590 \end{bmatrix}, \qquad \mathbf{d} = \begin{bmatrix} -0.0037 \\ -0.0063 \\ 0.0058 \\ -0.0055 \\ 0.0046 \end{bmatrix}.$$

Somit erhalten wir als das zu lösende Gleichungssystem

$$\mathbf{Rx} = \begin{bmatrix} -2.7695 & 2.1027 \\ 0 & -0.73 \end{bmatrix} \begin{bmatrix} v_y \\ g \end{bmatrix} = \begin{bmatrix} -7.3404 \\ -7.1590 \end{bmatrix} = \mathbf{c}.$$

Aufgelöst ergibt sich also

$$v_y = 10.1, \qquad g = 9.81.$$

Die Norm des Residuums ist übrigens

$$\|\mathbf{r}\|_2 = \|\mathbf{d}\|_2 = 0.118.$$

△

8. Iterative Lösungsverfahren

8.1 Fixpunktiterationen

In der Praxis tritt oft das Problem auf, eine nichtlineare Gleichung oder gar ein System von nichtlinearen Gleichungen lösen zu müssen. Während wir für lineare Gleichungssysteme Verfahren in Kapitel 2 kennengelernt haben, mit denen man in endlich vielen Schritten eine Lösung erhält, ist dies im allgemeinen bei nichtlinearen Gleichungssystemen nicht möglich. Es werden deshalb fast immer *iterative* Verfahren angewendet, bei denen eine Folge von Approximationen konstruiert wird, die gegen die gesuchte Lösung konvergiert.

Definition 8.1 Eine Abbildung Φ heißt **Selbstabbildung** von $D \subset \mathbb{R}^n$, falls $\Phi : D \to D$. Gilt zusätzlich

$$\|\Phi(\mathbf{y}) - \Phi(\mathbf{z})\| \leq L\|\mathbf{y} - \mathbf{z}\| \qquad \forall \mathbf{y}, \mathbf{z} \in D$$

für ein $L < 1$, so heißt Φ **kontrahierend**.

Definition 8.2 Sei $K \subset \mathbb{R}^n$ eine abgeschlossene Menge und Φ eine Selbstabbildung von K. Ein Punkt $\mathbf{x} \in K$ heißt ein **Fixpunkt** von Φ, falls dieser der **Fixpunktgleichung**

$$\mathbf{x} = \Phi(\mathbf{x})$$

genügt.

Nichtlineare Gleichungen werden zumeist als Fixpunktgleichung umgeformt, weil zu ihrer Lösung folgendes iteratives Verfahren naheliegt: für einen geeigneten Startwert $\mathbf{x}_0$ definiert man die Folge $\{\mathbf{x}_i\}_{i\in\mathbb{N}_0}$ durch

$$\mathbf{x}_{i+1} = \Phi(\mathbf{x}_i), \qquad i = 0, 1, 2, \ldots. \tag{8.1}$$

Eine solche Iteration heißt *Fixpunktiteration*.

Satz 8.3 (Banachscher Fixpunktsatz) Sei $K \subset \mathbb{R}^n$ eine abgeschlossene Menge. Ferner sei Φ eine kontrahierende Selbstabbildung von K. Dann existiert genau ein Fixpunkt $\mathbf{x} \in K$ und für jeden Startwert $\mathbf{x}_0 \in K$ konvergiert die durch die Iterationsvorschrift (8.1) definierte Folge $\{\mathbf{x}_i\}_{i\in\mathbb{N}_0}$ gegen diesen Fixpunkt. Ferner gelten die folgenden Fehlerabschätzungen:

(i)	$\|\mathbf{x}-\mathbf{x}_i\| \le L\|\mathbf{x}-\mathbf{x}_{i-1}\|$	"Monotonie"
(ii)	$\|\mathbf{x}-\mathbf{x}_i\| \le \frac{L^i}{1-L}\|\mathbf{x}_1-\mathbf{x}_0\|$	"a-piori-Schranke"
(iii)	$\|\mathbf{x}-\mathbf{x}_i\| \le \frac{L}{1-L}\|\mathbf{x}_i-\mathbf{x}_{i-1}\|$	"a-posteriori-Schranke"

Beweis. Sei $\mathbf{x}_0 \in K$ beliebig. Wir zeigen zunächst, dass die Folge $\{\mathbf{x}_i\}_{i\in\mathbb{N}_0}$ eine Cauchy-Folge ist. Da $\Phi(\mathbf{z}) \in K$ für beliebige $\mathbf{z} \in K$ ist, gilt $\mathbf{x}_i \in K$ für alle $i \in \mathbb{N}_0$. Für beliebiges $i \in \mathbb{N}$ folgt die Abschätzung

$$\begin{aligned}\|\mathbf{x}_{i+1}-\mathbf{x}_i\| &= \|\Phi(\mathbf{x}_i)-\Phi(\mathbf{x}_{i-1})\| \le L\,\|\mathbf{x}_i-\mathbf{x}_{i-1}\| \\ &= L\,\|\Phi(\mathbf{x}_{i-1})-\Phi(\mathbf{x}_{i-2})\| \le L^2\|\mathbf{x}_{i-1}-\mathbf{x}_{i-2}\| \\ &= L^2\|\Phi(\mathbf{x}_{i-2})-\Phi(\mathbf{x}_{i-3})\| \le L^3\|\mathbf{x}_{i-2}-\mathbf{x}_{i-3}\| \\ &= \ldots \le L^i\|\mathbf{x}_1-\mathbf{x}_0\|.\end{aligned}$$

Eingesetzt in die Dreiecksungleichung

$$\|\mathbf{x}_j-\mathbf{x}_i\| \le \sum_{k=i}^{j-1}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|$$

liefert dies

$$\begin{aligned}\|\mathbf{x}_j-\mathbf{x}_i\| &\le \sum_{k=i}^{j-1} L^k\|\mathbf{x}_1-\mathbf{x}_0\| = \|\mathbf{x}_1-\mathbf{x}_0\|\sum_{k=i}^{j-1}L^k = \|\mathbf{x}_1-\mathbf{x}_0\|L^i\sum_{k=0}^{j-i-1}L^k \\ &= \|\mathbf{x}_1-\mathbf{x}_0\|L^i\frac{1-L^{j-i}}{1-L} \le \|\mathbf{x}_1-\mathbf{x}_0\|\frac{L^i}{1-L} \overset{i\to\infty}{\longrightarrow} 0.\end{aligned}$$

Dies bedeutet, zu jedem $\varepsilon > 0$ existiert ein $N \in \mathbb{N}$, so dass für alle $N < i < j$

$$\|\mathbf{x}_j-\mathbf{x}_i\| \le \varepsilon.$$

Demnach ist $\{\mathbf{x}_i\}_{i\in\mathbb{N}_0}$ eine Cauchy-Folge. Da $K \subset \mathbb{R}^n$ vollständig ist, existiert das Grenzelement

$$\mathbf{x} = \lim_{i\to\infty}\mathbf{x}_i \in K.$$

Ferner ist Φ nach Voraussetzung Lipschitz-stetig, also insbesondere stetig auf K, woraus

$$\mathbf{x} = \lim_{i\to\infty}\mathbf{x}_{i+1} = \lim_{i\to\infty}\Phi(\mathbf{x}_i) = \Phi\Big(\lim_{i\to\infty}\mathbf{x}_i\Big) = \Phi(\mathbf{x})$$

folgt, das heißt, $\mathbf{x} \in K$ ist Fixpunkt von Φ.
Sei $\boldsymbol{\xi} \in K$ ein weiterer Fixpunkt von Φ, dann gilt

$$0 \le \|\boldsymbol{\xi}-\mathbf{x}\| = \|\Phi(\boldsymbol{\xi})-\Phi(\mathbf{x})\| \le L\|\boldsymbol{\xi}-\mathbf{x}\|,$$

und daher $\|\boldsymbol{\xi}-\mathbf{x}\| = 0$, das heißt, der Fixpunkt ist eindeutig.

Die Monotonie folgt gemäß

$$0 \leq \|\mathbf{x} - \mathbf{x}_i\| = \|\Phi(\mathbf{x}) - \Phi(\mathbf{x}_{i-1})\| \leq L\|\mathbf{x} - \mathbf{x}_{i-1}\|.$$

Um die Fehlerabschätzungen zu zeigen, wenden wir noch die Dreiecksungleichung an

$$\|\mathbf{x} - \mathbf{x}_i\| \leq L\|\mathbf{x} - \mathbf{x}_i + \mathbf{x}_i - \mathbf{x}_{i-1}\| \leq L\|\mathbf{x} - \mathbf{x}_i\| + L\|\mathbf{x}_i - \mathbf{x}_{i-1}\|.$$

Hieraus ergibt sich dann

$$\|\mathbf{x} - \mathbf{x}_i\| \leq \frac{L}{1-L}\|\mathbf{x}_i - \mathbf{x}_{i-1}\| \leq \frac{L^2}{1-L}\|\mathbf{x}_{i-1} - \mathbf{x}_{i-2}\| \leq \ldots \leq \frac{L^i}{1-L}\|\mathbf{x}_1 - \mathbf{x}_0\|.$$

□

Beispiel 8.4 Jede Lösung des nichtlinearen Gleichungssystems

$$\begin{aligned} x &= 0.7 \sin x + 0.2 \cos y \\ y &= 0.7 \cos x - 0.2 \sin y \end{aligned}$$

ist ein Fixpunkt der Abbildung $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$ mit

$$\Phi(x, y) := \begin{bmatrix} 0.7 \sin x + 0.2 \cos y \\ 0.7 \cos x - 0.2 \sin y \end{bmatrix}.$$

Für die Jacobi-Matrix Φ' von Φ,

$$\Phi'(x, y) = \begin{bmatrix} 0.7 \cos x & -0.2 \sin y \\ -0.7 \sin x & -0.2 \cos y \end{bmatrix},$$

folgt

$$\begin{aligned} L &:= \|\Phi'(x, y)\|_F \\ &= \sqrt{0.49 \cos^2 x + 0.04 \sin^2 y + 0.49 \sin^2 x + 0.04 \cos^2 y} \\ &= \sqrt{0.53} \approx 0.728. \end{aligned}$$

Also ist $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$ kontrahierend, und besitzt nach dem Banachschen Fixpunktsatz in $\mathbb{R}^2$ genau einen Fixpunkt $(\xi, \eta)^T$. Die Folge

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} := \Phi(x_i, y_i)$$

konvergiert für jeden Startvektor $(x, y)^T \in \mathbb{R}^2$ gegen $(\xi, \eta)^T$.
Wir wählen $x_0 = y_0 = 0$ und verlangen

$$\left\| \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} \xi \\ \eta \end{bmatrix} \right\|_2 \leq 10^{-4}.$$

Unter Verwendung der a-priori-Fehlerabschätzung

$$\left\| \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} \xi \\ \eta \end{bmatrix} \right\|_2 \leq \frac{L^i}{1-L} \left\| \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \right\|_2$$

ergibt sich $i \geq 33$ als hinreichende Bedingung. Die folgende Tabelle enthält neben x_i und y_i die Schranken

$$e_i := \frac{L}{1-L} \left\| \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} x_{i-1} \\ y_{i-1} \end{bmatrix} \right\|_2$$

aus der a-posteriori-Fehlerabschätzung.

i	x_i	y_i	e_i
1	0.200000	0.700000	1.9
2	0.292037	0.557203	0.45
3	0.371280	0.564599	0.21
4	0.422927	0.545289	0.14
5	0.458297	0.534591	$0.99 \cdot 10^{-1}$
⋮			
10	0.518525	0.511306	$0.13 \cdot 10^{-1}$
⋮			
20	0.526420	0.507964	$0.16 \cdot 10^{-3}$
21	0.526456	0.507948	$0.11 \cdot 10^{-3}$
22	0.526480	0.507938	$0.69 \cdot 10^{-4}$
⋮			
33	0.526522	0.507920	$0.57 \cdot 10^{-6}$

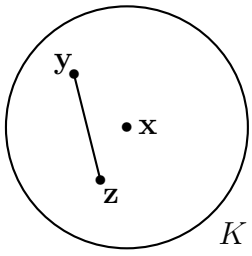
Offensichtlich wird die Fehlerschranke $\varepsilon = 10^{-4}$ bereits nach $i = 22$ Iterationen unterschritten. △

Der Banachsche Fixpunktsatz garantiert die Existenz eines Fixpunktes und die Konvergenz der Fixpunktiteration. In der Praxis ist die Existenz eines Fixpunktes meist bekannt (beispielsweise durch graphische Betrachtungen) und sogar seine ungefähre Lage. In dieser Situation besteht die Relevanz des Banachschen Fixpunktsatzes darin, ein relativ einfach zu überprüfendes Kriterium für die Konvergenz der Fixpunktiteration zu liefern:

Korollar 8.5 Sei $D \subset \mathbb{R}^n$ eine offene Menge und $\Phi : D \to \mathbb{R}^n$ stetig differenzierbar, und $\mathbf{x} \in D$ ein Fixpunkt von Φ. Ferner sei $\|\cdot\|_V$ eine Norm in $\mathbb{R}^n$ und $\|\cdot\|_M$ eine verträgliche Matrixnorm, für die $\|\Phi'(\mathbf{x})\|_M < 1$. Dann gibt es ein $\varepsilon > 0$ derart, dass für jedes $\mathbf{x}_0$ mit $\|\mathbf{x} - \mathbf{x}_0\|_V \leq \varepsilon$ die Folge $\{\mathbf{x}_i\}_{i\in\mathbb{N}_0}$, definiert durch die Fixpunktiteration $\mathbf{x}_{i+1} = \Phi(\mathbf{x}_i)$, gegen $\mathbf{x}$ konvergiert.

Beweis. Sei $\delta := 1 - \|\Phi'(\mathbf{x})\|_M > 0$. Wegen der Stetigkeit von Φ' lässt sich dann ein $\varepsilon > 0$ finden, so dass $\|\Phi'(\mathbf{y})\|_M \leq 1 - \delta/2$ für alle $\mathbf{y} \in K := \{\mathbf{z} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{z}\|_V \leq \varepsilon\} \subset D$. Für alle $\mathbf{y}, \mathbf{z} \in K$ folgt nach dem Fundamentalsatz der Integralrechnung.

$$\Phi(\mathbf{y}) - \Phi(\mathbf{z}) = \int_0^1 \Phi'\big(t\mathbf{y} + (1-t)\mathbf{z}\big)(\mathbf{y} - \mathbf{z})\,\mathrm{d}t.$$



Da mit $\mathbf{y}$ und $\mathbf{z}$ auch deren Verbindungsstrecke

$$[\mathbf{y}, \mathbf{z}] := \{t\mathbf{y} + (1-t)\mathbf{z} : t \in [0,1]\}$$

in K liegt, folgt

$$\|\Phi(\mathbf{y}) - \Phi(\mathbf{z})\|_V \leq \int_0^1 \underbrace{\big\|\Phi'\big(t\mathbf{y} + (1-t)\mathbf{z}\big)\big\|_M}_{\leq\left(1-\frac{\delta}{2}\right)} \|\mathbf{y} - \mathbf{z}\|_V \,\mathrm{d}t \leq \left(1 - \frac{\delta}{2}\right)\|\mathbf{y} - \mathbf{z}\|_V.$$

Dies bedeutet, Φ ist eine Kontraktion auf der abgeschlossenen Menge K. Die Behauptung des Korollars folgt nun aus dem Banachschen Fixpunktsatz, wenn wir zeigen können, dass

$\Phi(\mathbf{z}) \in K$ für alle $\mathbf{z} \in K$. Sei hierzu $\mathbf{z} \in K$ beliebig. Dann gilt wegen $\mathbf{x} = \Phi(\mathbf{x})$ und dem soeben bewiesenen

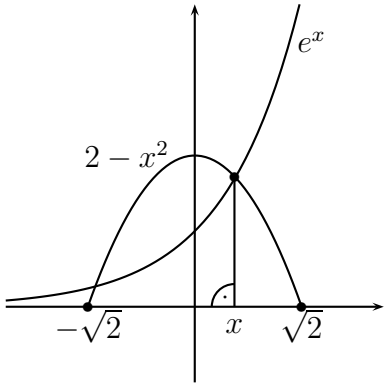
$$\|\mathbf{x} - \Phi(\mathbf{z})\|_V = \|\Phi(\mathbf{x}) - \Phi(\mathbf{z})\|_V \leq \left(1 - \frac{\delta}{2}\right)\|\mathbf{x} - \mathbf{z}\|_V \leq \left(1 - \frac{\delta}{2}\right)\varepsilon \leq \varepsilon,$$

was zu zeigen war. □

Beispiel 8.6 Gesucht werde eine Lösung der nichtlinearen Gleichung

$$2 - x^2 - e^x = 0.$$

Durch graphische Überlegungen sieht man, dass es genau eine positive Lösung $x \approx 0.5$ gibt:



Für $x > 0$ kann die Gleichung in verschiedener Weise in eine Fixpunktgleichung umgeformt werden:

$$x = \sqrt{2 - e^x} =: \Phi_1(x), \qquad x = \ln(2 - x^2) =: \Phi_2(x),$$

Die Fixpunktiterationen, die auf diesen beiden Iterationsfunktionen basieren, verhalten sich aber unterschiedlich, wenn man mit $x_0 = 0.5$ startet, wie nachfolgende Tabelle zeigt:

i	$x_{i+1} = \Phi_1(x_i)$	$x_{i+1} = \Phi_2(x_i)$
0	0.592687716508341	0.559615787935423
1	0.437214425050104	0.522851128605001
2	0.672020792350124	0.546169619063046
3	0.204473907097276	0.531627015197373
4	0.879272743474883	0.540795632739194
5	Abbruch ($2 - e^{0.87} < 0$)	0.535053787215218
6		0.538664955236433
7		0.536399837485597
8		0.537823020842571
9		0.536929765486145

Die Fixpunktiteration $x_{i+1} = \Phi_1(x_i)$ konvergiert nicht (sie bricht sogar ab), während die Iteration $x_{i+1} = \Phi_2(x_i)$ gegen den korrekten Wert $x = 0.5372744491738\ldots$ konvergiert. Korollar 8.5 erklärt das unterschiedliche Verhalten: Am Fixpunkt gilt

$$\Phi_1'(x) \approx -1.59, \qquad \Phi_2'(x) \approx -0.62.$$

Wir erwarten deshalb Konvergenz der auf Φ_2 basierenden Iteration für hinreichend nahe am Fixpunkt gelegene Startwerte. Für die auf Φ_1 basierende Iteration ist Korollar 8.5 nicht anwendbar. Man kann sogar zeigen, dass für stetig differenzierbare Iterationsfunktionen Φ die Bedingung $|\Phi'(x)| > 1$ zu Divergenz der Fixpunktiteration führt. △

Um einen Vergleich von verschiedenen Fixpunktverfahren zu ermöglichen, will man die Konvergenzgeschwindigkeit messen können. Dazu führen wir den Begriff der Konvergenzordnung ein.

Definition 8.7 Sei $\{\varepsilon_k\}_{k\in\mathbb{N}_0}$ eine Folge positiver reeller Zahlen mit $\varepsilon_k \to 0$ für $k \to \infty$. Wir sagen, dass die Konvergenz (mindestens) die Ordnung $p \geq 1$ hat, wenn ein $C > 0$ existiert, so dass

$$\varepsilon_{k+1} \leq C\varepsilon_k^p.$$

Ist $p = 1$, so fordert man zusätzlich, dass $C < 1$!

Beispiel 8.8 Um die Konvergenzordnung iterativer Verfahren zu studieren, muss man nur die Folge $\{\varepsilon_k\}_{k\in\mathbb{N}_0}$ der Fehler der Iterierten betrachten und obige Definition anwenden.

- Der Fall $p = 1$ ist von besonderer Bedeutung und uns bereits im Zusammenhang mit dem Banachschen Fixpunktsatz 8.3 begegnet. In diesem Fall spricht man von *linearer Konvergenz*.
- Der Fall $p = 2$ wird uns im Abschnitt 8.3 begegenen, im Zusammenhang mit dem *Newton-Verfahren*. Hier spricht man von *quadratischer Konvergenz*.

△

Bei nichtlinearen Problemen ist es wichtig, zwischen lokaler und globaler Konvergenz zu unterscheiden:

Definition 8.9 Ein Iterationsverfahren mit Iterierten $\mathbf{x}_i \in \mathbb{R}^n$ heißt **lokal konvergent** gegen $\mathbf{x} \in \mathbb{R}^n$, falls eine Umgebung $U \subset \mathbb{R}^n$ um $\mathbf{x} \in U$ existiert, so dass

$$\mathbf{x}_i \overset{i\to\infty}{\longrightarrow} \mathbf{x} \quad \text{für alle } \mathbf{x}_0 \in U.$$

Man spricht von **globaler Konvergenz**, falls $U = \mathbb{R}^n$.

8.2 Iterationsverfahren für lineare Gleichungsysteme

Wenn die Matrizen sehr groß sind, verbieten sich direkte Löser zur Lösung eines linearen Gleichungssystems $\mathbf{Ax} = \mathbf{b}$ wegen ihres $\mathcal{O}(n^3)$-Aufwands. Zudem sind die großen, in der Praxis auftretenden Systeme ($n \gtrsim 10^5$) meist dünn besetzt, das heißt, nur wenige ($\lesssim 10$) Einträge in jeder Zeile sind ungleich Null. Während die Matrix eines solchen Problems noch gut in den Speicher passen mag, trifft dies für die L- und R-Faktoren in der Gauß-Elimination in der Regel nicht mehr zu ("fill-in"). In solchen Fällen behilft man sich gerne mit Iterationsverfahren.

Am einfachsten ist hierbei vermutlich das *Gesamtschritt-* oder *Jacobi-Verfahren*:

Algorithmus 8.10 (Gesamtschrittverfahren)

input: Matrix $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{n\times n}$, rechte Seite $\mathbf{b} = [b_i] \in \mathbb{R}^n$ und Startnäherung $\mathbf{x}^{(0)} = [x_i^{(0)}] \in \mathbb{R}^n$

output: Folge von Iterierten $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}}$ mit $\mathbf{x}^{(k)} = [x_i^{(k)}] \in \mathbb{R}^n$

① Initialisierung: $k = 0$

② für $i = 1, 2, \ldots, n$ setze

$$x_i^{(k+1)} := \frac{1}{a_{i,i}}\left(b_i - \sum_{j\neq i} a_{i,j}x_j^{(k)}\right)$$

③ erhöhe $k := k + 1$ und gehe nach ②

Vorausgesetzt werden muss offensichtlich, dass $a_{i,i} \neq 0$ ist für alle $i = 1, 2, \ldots, n$. Die Frage nach der Konvergenz ist dadurch jedoch nicht beantwortet. Sicher ist nur, dass bei vorliegender Konvergenz die Iterierten $\mathbf{x}^{(k)}$ gegen eine Lösung von $\mathbf{Ax} = \mathbf{b}$ konvergieren.

Das Gesamtschrittverfahren lässt sich auch in Matrixnotation formulieren. Dazu zerlegen wir (aus historischen Gründen mit "−")

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{R}$$

in eine Diagonal- und in *strikte* linke untere und rechte obere Dreiecksmatrizen. Dann ist

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}\big(\mathbf{b} + (\mathbf{L} + \mathbf{R})\mathbf{x}^{(k)}\big). \tag{8.2}$$

Beim *Einzelschritt-* oder *Gauß-Seidel-Verfahren* verwendet man in ② bereits alle berechneten Komponenten von $\mathbf{x}^{(k+1)}$:

Algorithmus 8.11 (Einzelschrittverfahren)

input: Matrix $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{n\times n}$, rechte Seite $\mathbf{b} = [b_i] \in \mathbb{R}^n$ und Startnäherung $\mathbf{x}^{(0)} = [x_i^{(0)}] \in \mathbb{R}^n$

output: Folge von Iterierten $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}}$ mit $\mathbf{x}^{(k)} = [x_i^{(k)}] \in \mathbb{R}^n$

① Initialisierung: $k = 0$

② für $i = 1, 2, \ldots, n$ setze

$$x_i^{(k+1)} := \frac{1}{a_{i,i}}\left(b_i - \sum_{j<i} a_{i,j}x_j^{(k+1)} - \sum_{j>i} a_{i,j}x_j^{(k)}\right)$$

③ erhöhe $k := k + 1$ und gehe nach ②

Entsprechend zu (8.2) erhält man die Matrixformulierung, indem man alle Komponenten von $\mathbf{x}^{(k+1)}$ in ② auf die linke Seite bringt. Dann folgt

$$a_{i,i}x_i^{(k+1)} + \sum_{j<i} a_{i,j}x_j^{(k+1)} = b_i - \sum_{j>i} a_{i,j}x_j^{(k)}, \qquad i = 1, 2, \ldots, n,$$

das heißt, $\mathbf{x}^{(k+1)}$ ergibt sich durch Auflösen des Dreiecksystems

$$(\mathbf{D}-\mathbf{L})\mathbf{x}^{(k+1)} = \mathbf{b} + \mathbf{R}\mathbf{x}^{(k)},$$

also

$$\mathbf{x}^{(k+1)} = (\mathbf{D}-\mathbf{L})^{-1}\big(\mathbf{b} + \mathbf{R}\mathbf{x}^{(k)}\big). \tag{8.3}$$

Verschiedene, mehr oder weniger praktikable, Konvergenzkriterien für das Gesamtschrittverfahren (8.2) und das Einzelschrittverfahren (8.3) sind in der Literatur bekannt. Wir wollen uns auf nachfolgendes, einfach nachzuweisendes Kriterium beschränken.

Definition 8.12 Eine Matrix $\mathbf{A}$ heißt **strikt diagonaldominant**, falls

$$|a_{i,i}| > \sum_{j\neq i} |a_{i,j}| \quad \text{für alle} \quad i = 1, 2, \ldots, n.$$

Satz 8.13 Ist $\mathbf{A}$ strikt diagonaldominant, dann konvergieren Gesamt- und Einzelschrittverfahren für jeden Startvektor $\mathbf{x}^{(0)} \in \mathbb{R}^n$ gegen die eindeutige Lösung von $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Beweis. Da sowohl Gesamt-, als auch Einzelschrittverfahren, Fixpunktiterationen $\mathbf{x}^{(k+1)} = \Phi(\mathbf{x}^{(k)})$ sind, wobei die Abbildung Φ sogar affin ist, können wir den Banachschen Fixpunktsatz 8.3 anwenden. Da wir $K = \mathbb{R}^n$ wählen können, müssen wir lediglich noch zeigen, dass Φ eine Kontraktion ist.

Zunächst betrachten wir das Gesamtschrittverfahren. Zu zeigen ist, dass ein $L < 1$ existiert mit

$$\|\mathbf{D}^{-1}(\mathbf{L}+\mathbf{R})\mathbf{w}\| \leq L\|\mathbf{w}\| \tag{8.4}$$

für alle $\mathbf{w} \in \mathbb{R}^n$. Aus der strikten Diagonaldominanz von $\mathbf{A}$ folgt

$$\|\mathbf{D}^{-1}(\mathbf{L}+\mathbf{R})\|_\infty = \max_{1\leq i\leq n} \sum_{\substack{j=1\\ j\neq i}}^{n} \frac{|a_{i,j}|}{|a_{i,i}|} < 1,$$

das ist (8.4) mit $\|\cdot\| = \|\cdot\|_\infty$ und $L = \|\mathbf{D}^{-1}(\mathbf{L}+\mathbf{R})\|_\infty$.

Für das Einzelschrittverfahren ist der Beweis komplizierter. Wieder verwenden wir ∞-Norm und müssen entsprechend zu (8.4) nachweisen, dass

$$\max_{\|\mathbf{x}\|_\infty = 1} \|(\mathbf{D}-\mathbf{L})^{-1}\mathbf{R}\mathbf{x}\|_\infty = \|(\mathbf{D}-\mathbf{L})^{-1}\mathbf{R}\|_\infty < 1. \tag{8.5}$$

Sei also $\|\mathbf{x}\|_\infty = 1$ und $L < 1$ wie zuvor definiert. Die einzelnen Komponenten y_i von $\mathbf{y} = (\mathbf{D}-\mathbf{L})^{-1}\mathbf{R}\mathbf{x}$ ergeben sich gemäß Algorithmus 8.11 aus

$$y_i := \frac{1}{a_{i,i}}\Big(-\sum_{j<i} a_{i,j}y_j - \sum_{j>i} a_{i,j}x_j \Big). \tag{8.6}$$

Wir zeigen induktiv, dass $|y_i| \le L < 1$ für alle $i = 1, 2, \ldots, n$ gilt. Hierzu schätzen wir in (8.6) $|y_i|$ mit der Dreiecksungleichung und der Induktionsannahme ab (der Fall $i = 1$ ist klar):

$$\begin{aligned} |y_i| &\le \frac{1}{|a_{i,i}|}\left(\sum_{j<i} |a_{i,j}||y_j| + \sum_{j>i} |a_{i,j}||x_j|\right) \\ &\le \frac{1}{|a_{i,i}|}\left(\sum_{j<i} |a_{i,j}|L + \sum_{j>i} |a_{i,j}|\|\mathbf{x}\|_\infty\right) \\ &\le \frac{1}{|a_{i,i}|}\left(\sum_{j<i} |a_{i,j}| + \sum_{j>i} |a_{i,j}|\right) \le L. \end{aligned}$$

Hieraus folgt $\|\mathbf{y}\|_\infty \le L$ und somit (8.5). □

8.3 Newton-Verfahren

Das *Newton-Verfahren* und seine Varianten sind wohl die wichtigsten Verfahren zum Lösen von nichtlinearen Gleichungen.

Die Funktion $f : \mathbb{R}^n \to \mathbb{R}^n$ sei differenzierbar. Wir wollen die Nullstelle $\mathbf{x} \in \mathbb{R}^n$ der nichtlinearen Gleichung

$$f(\mathbf{x}) = \mathbf{0}$$

finden. Ist $\mathbf{x}_0 \in \mathbb{R}^n$ ein Nährerungswert an diese Lösung, dann approximieren wir

$$f(\mathbf{x}) = \mathbf{0} \approx f(\mathbf{x}_0) + f'(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0).$$

Falls $\big(f'(\mathbf{x}_0)\big)^{-1} \in \mathbb{R}^{n\times n}$ existiert, so folgt

$$\mathbf{x} \approx \mathbf{x}_0 - \big(f'(\mathbf{x}_0)\big)^{-1} f(\mathbf{x}_0).$$

Setzen wir

$$\mathbf{x}_1 := \mathbf{x}_0 - \big(f'(\mathbf{x}_0)\big)^{-1} f(\mathbf{x}_0),$$

so ist $\mathbf{x}_1$ möglicherweise eine bessere Näherung an $\mathbf{x}$. Dies ist in einer geeigneten Umgebung von $\mathbf{x}$ wahrscheinlich der Fall. Daher ist es naheliegend, das folgende Iterationsverfahren

$$\mathbf{x}_{i+1} := \mathbf{x}_i - \big(f'(\mathbf{x}_i)\big)^{-1} f(\mathbf{x}_i), \qquad i = 0, 1, 2, \ldots \tag{8.7}$$

zum Auffinden einer Nullstelle zu verwenden. Dies ist das bekannte Newton-Verfahren. Es ist von der Form (8.1), das heißt eine Fixpunktiteration. Wann und wie schnell dieses Verfahren konvergiert, beantwortet der nächste Satz.

Satz 8.14 Sei $D \subset \mathbb{R}^n$ offen und konvex, $\|\cdot\|_V$ eine Norm in $\mathbb{R}^n$ und $\|\cdot\|_M$ eine verträgliche Matrixnorm. Ferner sei $f : D \to \mathbb{R}^n$ eine stetig differenzierbare Funktion mit invertierbarer Jacobi-Matrix $f'(\mathbf{z})$ mit

$$\big\|\big(f'(\mathbf{z})\big)^{-1}\big\|_M \le \alpha$$

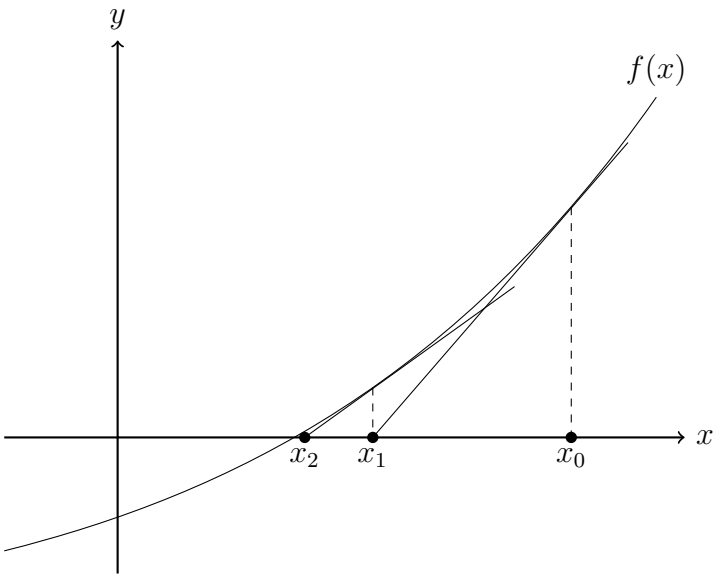


Abbildung 8.1: Geometrische Interpretation des Newton-Verfahrens.

für alle $\mathbf{z} \in D$. Zusätzlich sei $f'(\mathbf{z})$ auf D Lipschitz-stetig mit der Konstanten β,

$$\|f'(\mathbf{y}) - f'(\mathbf{z})\|_M \leq \beta \|\mathbf{y} - \mathbf{z}\|_V, \qquad \mathbf{y}, \mathbf{z} \in D.$$

Der Punkt $\mathbf{x} \in D$ sei eine Nullstelle, das heißt $f(\mathbf{x}) = \mathbf{0}$, und $\mathbf{x}_0$ eine Startnäherung mit

$$\mathbf{x}_0 \in K := \{\mathbf{z} \in \mathbb{R}^n : \|\mathbf{z} - \mathbf{x}\|_V \leq \gamma\},$$

wobei γ hinreichend klein sei, so dass $K \subset D$ und

$$\gamma \leq \frac{2}{\alpha\beta}.$$

Dann bleibt die durch das Newton-Verfahren (8.7) definierte Folge $\{\mathbf{x}_i\}_{i \in \mathbb{N}_0}$ innerhalb der Kugel K und konvergiert quadratisch gegen $\mathbf{x}$, das heißt

$$\|\mathbf{x}_{i+1} - \mathbf{x}\|_V \leq \frac{\alpha\beta}{2} \|\mathbf{x}_i - \mathbf{x}\|_V^2, \qquad i = 0, 1, 2, \ldots.$$

Beweis. Wegen (8.7) und $f(\mathbf{x}) = \mathbf{0}$ hat man für $\mathbf{x}_i \in D$

$$\begin{aligned}
\mathbf{x}_{i+1} - \mathbf{x} &= \mathbf{x}_i - \big(f'(\mathbf{x}_i)\big)^{-1} f(\mathbf{x}_i) - \mathbf{x} \\
&= \mathbf{x}_i - \mathbf{x} - \big(f'(\mathbf{x}_i)\big)^{-1} \big[f(\mathbf{x}_i) - f(\mathbf{x})\big] \\
&= \big(f'(\mathbf{x}_i)\big)^{-1} \big[f(\mathbf{x}) - f(\mathbf{x}_i) - f'(\mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)\big].
\end{aligned}$$

Daher gilt

$$\begin{aligned}\|\mathbf{x}_{i+1}-\mathbf{x}\|_V &\le \underbrace{\left\|\big(f'(\mathbf{x}_i)\big)^{-1}\right\|_M}_{\le\alpha}\big\|f(\mathbf{x})-f(\mathbf{x}_i)-f'(\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)\big\|_V\\ &\le \alpha\big\|f(\mathbf{x})-f(\mathbf{x}_i)-f'(\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)\big\|_V. \end{aligned}\tag{8.8}$$

Letzter Term ist nun abzuschätzen. Dazu setzen wir für $\mathbf{y},\mathbf{z}\in D$

$$g(t)=f\big((1-t)\mathbf{y}+t\mathbf{z}\big)$$

und bemerken, dass $g(0)=f(\mathbf{y})$ und $g(1)=f(\mathbf{z})$ gilt. Nach Voraussetzung ist g differenzierbar und nach der Kettenregel folgt

$$g'(t)=f'\big((1-t)\mathbf{y}+t\mathbf{z}\big)(\mathbf{z}-\mathbf{y}).$$

Also ist wegen der Lipschitz-Stetigkeit

$$\begin{aligned}\|g'(t)-g'(0)\|_V &= \big\|\big[f'\big((1-t)\mathbf{y}+t\mathbf{z}\big)-f'(\mathbf{y})\big](\mathbf{z}-\mathbf{y})\big\|_V\\ &\le \underbrace{\|f'\big((1-t)\mathbf{y}+t\mathbf{z}\big)-f'(\mathbf{y})\|_M}_{\le\beta t\|\mathbf{z}-\mathbf{y}\|_V}\|\mathbf{z}-\mathbf{y}\|_V\\ &\le \beta t\|\mathbf{z}-\mathbf{y}\|_V^2.\end{aligned}$$

Mit

$$f(\mathbf{z})-f(\mathbf{y})-f'(\mathbf{y})(\mathbf{z}-\mathbf{y})=g(1)-g(0)-g'(0)=\int_0^1 g'(t)-g'(0)\,\mathrm{d}t$$

folgt hieraus

$$\|f(\mathbf{z})-f(\mathbf{y})-f'(\mathbf{y})(\mathbf{z}-\mathbf{y})\|_V\le\beta\|\mathbf{z}-\mathbf{y}\|_V^2\int_0^1 t\,\mathrm{d}t=\frac{\beta}{2}\|\mathbf{z}-\mathbf{y}\|_V^2.$$

Wegen (8.8) erhalten wir daher die quadratische Konvergenzrate

$$\|\mathbf{x}_{i+1}-\mathbf{x}\|_V\le\frac{\alpha\beta}{2}\|\mathbf{x}_i-\mathbf{x}\|_V^2.\tag{8.9}$$

Es verbleibt zu zeigen, dass für alle i die Ungleichung $\|\mathbf{x}-\mathbf{x}_i\|_V\le\gamma$ gilt. Da dies nach Voraussetzung für $i=0$ gilt, bietet sich vollständige Induktion an. Der Induktionschritt ergibt sich aus (8.9) gemäß

$$\|\mathbf{x}_{i+1}-\mathbf{x}\|_V\le\underbrace{\frac{\alpha\beta}{2}\|\mathbf{x}_i-\mathbf{x}\|_V}_{\le 1}\underbrace{\|\mathbf{x}_i-\mathbf{x}\|_V}_{\le\gamma}\le\gamma.$$

□

Beachte: Die Konvergenz des Newton-Verfahrens ist im allgemeinen nur lokal!

Bemerkung 8.15 Im Newton-Verfahren wird die inverse Jacobi-Matrix nicht berechnet, sondern die Iterationsvorschrift (8.7) wie folgt modifiziert:

1. Löse das lineare Gleichungssystem $f'(\mathbf{x}_i)\triangle\mathbf{x}_i=-f(\mathbf{x}_i)$ und

2. setze $\mathbf{x}_{i+1} = \mathbf{x}_i + \triangle\mathbf{x}_i$.

Zur Lösung des linearen Gleichungssystems kann man beispielsweise die LR-Zerlegung verwenden. △

Beispiel 8.16 In Beispiel 8.4 haben wir die Lösung $(\xi, \eta)^T$ des nichtlinearen Gleichungssystems

$$\begin{aligned} x &= 0.7 \sin x + 0.2 \cos y \\ y &= 0.7 \cos x - 0.2 \sin y \end{aligned}$$

mit dem Banachschen Fixpunktsatz berechnet. Zum Vergleich soll $(\xi, \eta)^T$ auch mit dem Newton-Verfahren approximiert werden. Dazu wird das Fixpunktproblem als Nullstellenaufgabe einer Funktion f formuliert, nämlich

$$f(x, y) := \begin{bmatrix} x - 0.7 \sin x - 0.2 \cos y \\ y - 0.7 \cos x + 0.2 \sin y \end{bmatrix} \stackrel{!}{=} \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Das Newtonsche Iterationsverfahren lautet dann

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} := \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \big(f'(x_i, y_i)\big)^{-1} f(x_i, y_i) \in \mathbb{R}^2.$$

Hierbei ist

$$f'(x, y) = \begin{bmatrix} 1 - 0.7 \cos x & 0.2 \sin y \\ 0.7 \sin x & 1 + 0.2 \cos y \end{bmatrix}$$

mit Determinante

$$\det f'(x, y) = 1 - 0.7 \cos x + 0.2 \cos y - 0.14 \cos x \cos y - 0.14 \sin x \sin y.$$

Folglich gilt

$$\big(f'(x, y)\big)^{-1} = \frac{1}{\det f'(x, y)} \begin{bmatrix} 1 + 0.2 \cos y & -0.2 \sin y \\ -0.7 \sin x & 1 - 0.7 \cos x \end{bmatrix}.$$

Aus der untenstehenden Tabelle erkennt man, dass x_i und y_i ab $i = 4$ auf sechs Stellen genau sind (Startwerte $x_0 = y_0 = 0$). Es gilt sogar $|x_i - \xi| < 5 \cdot 10^{-13}$ für $i \geq 5$ und $|y_i - \eta| < 5 \cdot 10^{-13}$ für $i \geq 6$.

	Banachscher Fixpunktsatz		Newton-Verfahren	
i	x_i	y_i	x_i	y_i
1	0.200000	0.700000	0.666667	0.583333
2	0.292037	0.557203	0.536240	0.508849
3	0.371280	0.564599	0.526562	0.507932
4	0.422927	0.545289	0.526523	0.507920
⋮			⋮	⋮
22	0.526480	0.507938	⋮	⋮
⋮			⋮	⋮
33	0.526522	0.507920	0.526523	0.507920

Exakte Lösung ist $\xi = 0.526522621917$ und $\eta = 0.507919719037$. △

8.4 Verfahren der konjugierten Gradienten

Das Verfahren der konjugierten Gradienten von Hestenes und Stiefel (1952), welches auch als cg-Verfahren (von engl.: conjugate gradient method) bekannt ist, ist wohl das effektivste Verfahren zur Lösung großer linearer Gleichungssysteme $\mathbf{Ax} = \mathbf{b}$ mit hermitescher und positiv definiter Matrix $\mathbf{A}$.

Definition 8.17 Ist $\mathbf{A} \in \mathbb{R}^{n\times n}$ hermitesch und positiv definit, dann definiert

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{A}} = \mathbf{x}^T \mathbf{A} \mathbf{y}, \qquad \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$$

ein Skalarprodukt. Die induzierte Norm

$$\|\mathbf{x}\|_{\mathbf{A}} := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbf{A}}} = \sqrt{\mathbf{x}^T \mathbf{A} \mathbf{x}}$$

wird **Energienorm** bezüglich $\mathbf{A}$ genannt. Zwei Vektoren $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ heißen **konjugiert** bezüglich $\mathbf{A}$, falls

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{A}} = \mathbf{x}^T \mathbf{A} \mathbf{y} = 0.$$

Lemma 8.18 Seien $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1}$ bezüglich $\mathbf{A}$ konjugierte Richtungen. Für jedes $\mathbf{x}_0 \in \mathbb{R}^n$ liefert die durch

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$$

mit

$$\alpha_i = \frac{\mathbf{d}_i^T \mathbf{r}_i}{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i}, \qquad \mathbf{r}_i = \mathbf{b}_i - \mathbf{A}\mathbf{x}_i$$

für $i \geq 0$ erzeugte Folge nach (höchstens) n Schritten die Lösung $\mathbf{x}_n = \mathbf{A}^{-1}\mathbf{b}$.

Beweis. Mit dem Ansatz

$$\mathbf{x} - \mathbf{x}_0 = \sum_{j=0}^{n-1} \alpha_j \mathbf{d}_j$$

erhalten wir wegen den Orthogonalitätsrelationen

$$\mathbf{d}_i^T \mathbf{A}(\mathbf{x} - \mathbf{x}_0) = \mathbf{d}_i^T \mathbf{A} \left(\sum_{j=0}^{n-1} \alpha_j \mathbf{d}_j \right) = \sum_{j=0}^{n-1} \alpha_j \underbrace{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_j}_{=0 \text{ falls } i \neq j} = \alpha_i \mathbf{d}_i^T \mathbf{A} \mathbf{d}_i$$

die Beziehung

$$\alpha_i = \frac{\mathbf{d}_i^T \mathbf{A}(\mathbf{x} - \mathbf{x}_0)}{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i}.$$

Weil $\mathbf{d}_i$ zu den anderen Richtungen konjugiert ist, gilt

$$\mathbf{d}_i^T \mathbf{A}(\mathbf{x}_i - \mathbf{x}_0) = \mathbf{d}_i^T \mathbf{A} \left(\sum_{j=0}^{i-1} \alpha_j \mathbf{d}_j \right) = \sum_{j=0}^{i-1} \alpha_j \underbrace{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_j}_{=0} = 0.$$

Deshalb ist

$$\alpha_i = \frac{\mathbf{d}_i^T(\overbrace{\mathbf{A}\mathbf{x}}^{=\mathbf{b}} - \mathbf{A}\mathbf{x}_i + \mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_0)}{\mathbf{d}_i^T\mathbf{A}\mathbf{d}_i} = \frac{\mathbf{d}_i^T(\mathbf{b} - \mathbf{A}\mathbf{x}_i)}{\mathbf{d}_i^T\mathbf{A}\mathbf{d}_i} + \underbrace{\frac{\mathbf{d}_i^T\mathbf{A}(\mathbf{x}_i - \mathbf{x}_0)}{\mathbf{d}_i^T\mathbf{A}\mathbf{d}_i}}_{=0} = \frac{\mathbf{d}_i^T\mathbf{r}_i}{\mathbf{d}_i^T\mathbf{A}\mathbf{d}_i}.$$

□

Bemerkung 8.19 Der Vektor $\mathbf{r}_i = \mathbf{b} - \mathbf{A}\mathbf{x}_i$ wird *Residuum* genannt. Seine Norm $\|\mathbf{r}_i\|_2$ ist ein Maß für den Fehler im i-ten Schritt. Gilt $\|\mathbf{r}_i\|_2 = 0$, so stimmt $\mathbf{x}_i$ mit der Lösung $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ überein. △

Beim Verfahren der konjugierten Gradienten werden die Richtungen $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1}$ nicht von vornherein gewählt, sondern aus dem jeweils aktuellen Residuum $\mathbf{r}_i$ durch Addition einer Korrektur ermittelt. Das Verfahren der konjugierten Gradienten lautet:

Algorithmus 8.20 (cg-Verfahren)

input: Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, rechte Seite $\mathbf{b} \in \mathbb{R}^n$ und Startnäherung $\mathbf{x}_0 \in \mathbb{R}^n$

output: Folge von Iterierten $\{\mathbf{x}_k\}_{k>0}$

① Initialisierung: setze $\mathbf{d}_0 = \mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$ und $i := 0$

② berechne

$$\alpha_i := \frac{\mathbf{d}_i^T\mathbf{r}_i}{\mathbf{d}_i^T\mathbf{A}\mathbf{d}_i} \tag{8.10}$$

$$\mathbf{x}_{i+1} := \mathbf{x}_i + \alpha_i\mathbf{d}_i \tag{8.11}$$

$$\mathbf{r}_{i+1} := \mathbf{r}_i - \alpha_i\mathbf{A}\mathbf{d}_i \tag{8.12}$$

$$\beta_i := \frac{\mathbf{d}_i^T\mathbf{A}\mathbf{r}_{i+1}}{\mathbf{d}_i^T\mathbf{A}\mathbf{d}_i} \tag{8.13}$$

$$\mathbf{d}_{i+1} := \mathbf{r}_{i+1} - \beta_i\mathbf{d}_i \tag{8.14}$$

③ falls $\|\mathbf{r}_{i+1}\|_2 \neq 0$ erhöhe $i := i + 1$ und gehe nach ②

Satz 8.21 Solange $\mathbf{r}_i \neq \mathbf{0}$ ist, gelten die folgenden Aussagen:

1. Es ist $\mathbf{d}_j^T\mathbf{r}_i = 0$ für alle $j < i$.
2. Es gilt $\mathbf{r}_j^T\mathbf{r}_i = 0$ für alle $j < i$.
3. Die Vektoren $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_i$ sind paarweise konjugiert.

Beweis. Wir bemerken zunächst, dass gilt

$$\mathbf{d}_i^T\mathbf{r}_{i+1} \overset{(8.12)}{=} \mathbf{d}_i^T(\mathbf{r}_i - \alpha_i\mathbf{A}\mathbf{d}_i) = \mathbf{d}_i^T\mathbf{r}_i - \underbrace{\alpha_i}_{\overset{(8.10)}{=}\frac{\mathbf{d}_i^T\mathbf{r}_i}{\mathbf{d}_i^T\mathbf{A}\mathbf{d}_i}} \mathbf{d}_i^T\mathbf{A}\mathbf{d}_i = 0. \tag{8.15}$$

Wir wollen nun den Beweis vermittels Induktion führen.

Im Falle $i = 1$ folgen die ersten beiden Aussagen direkt aus (8.15) während sich die dritte wegen

$$\mathbf{d}_0^T \mathbf{A}\mathbf{d}_1 \overset{(8.14)}{=} \mathbf{d}_0^T \mathbf{A}\mathbf{r}_1 - \underbrace{\beta_i}_{\overset{(8.13)}{=} \frac{\mathbf{d}_0^T \mathbf{A}\mathbf{r}_1}{\mathbf{d}_0^T \mathbf{A}\mathbf{d}_0}} \mathbf{d}_0^T \mathbf{A}\mathbf{d}_0 = 0$$

ergibt.

Für den Induktionsschritt $i \mapsto i+1$ nehmen wir an, dass alle drei Aussagen für i gelten. Nun ergibt sich die erste Aussage für $i+1$ und $j = i$ wieder aus (8.15) während sie für $j < i$ mit Hilfe der Induktionsannahme aus

$$\mathbf{d}_j^T \mathbf{r}_{i+1} \overset{(8.12)}{=} \mathbf{d}_j^T (\mathbf{r}_i - \alpha_i \mathbf{A}\mathbf{d}_i) = \underbrace{\mathbf{d}_j^T \mathbf{r}_i}_{=0} - \alpha_i \underbrace{\mathbf{d}_j^T \mathbf{A}\mathbf{d}_i}_{=0} = 0$$

folgt. Wegen

$$\mathbf{r}_j \overset{(8.14)}{=} \mathbf{d}_j + \beta_{j-1}\mathbf{d}_{j-1}, \qquad 1 \leq j \leq i$$

ergibt sich die zweite Aussage direkt aus der ersten.

Weiter sind $\mathbf{d}_i$ und $\mathbf{d}_{i+1}$ konjugiert, da

$$\mathbf{d}_i^T \mathbf{A}\mathbf{d}_{i+1} \overset{(8.14)}{=} \mathbf{d}_i^T \mathbf{A}\mathbf{r}_{i+1} - \underbrace{\beta_i}_{\overset{(8.13)}{=} \frac{\mathbf{d}_i^T \mathbf{A}\mathbf{r}_{i+1}}{\mathbf{d}_i^T \mathbf{A}\mathbf{d}_i}} \mathbf{d}_i^T \mathbf{A}\mathbf{d}_i = 0.$$

Für $\mathbf{d}_j$ mit $j < i$ folgt aufgrund der Induktionsannahme

$$\mathbf{d}_j^T \mathbf{A}\mathbf{d}_{i+1} \overset{(8.14)}{=} \mathbf{d}_j^T \mathbf{A}(\mathbf{r}_{i+1} - \beta_i \mathbf{d}_i) = \mathbf{d}_j^T \mathbf{A}\mathbf{r}_{i+1} - \beta_i \underbrace{\mathbf{d}_j^T \mathbf{A}\mathbf{d}_i}_{=0} = \mathbf{d}_j^T \mathbf{A}\mathbf{r}_{i+1},$$

und damit wegen der schon bewiesenen zweiten Aussage

$$\alpha_j \mathbf{d}_j^T \mathbf{A}\mathbf{d}_{i+1} = \alpha_j \mathbf{d}_j^T \mathbf{A}\mathbf{r}_{i+1} \overset{(8.12)}{=} (\mathbf{r}_{j+1} - \mathbf{r}_j)^T \mathbf{r}_{i+1} = \mathbf{r}_{j+1}^T \mathbf{r}_{i+1} - \mathbf{r}_j^T \mathbf{r}_{i+1} = 0.$$

Es bleibt nur noch zu zeigen, dass α_j nicht Null werden kann. Angenommen $\alpha_j = 0$, dann folgt

$$\mathbf{d}_j^T \mathbf{r}_j = 0,$$

und wegen (8.14) ergibt sich

$$0 = (\mathbf{r}_j - \beta_{j-1}\mathbf{d}_{j-1})^T \mathbf{r}_j = \mathbf{r}_j^T \mathbf{r}_j - \beta_{j-1} \underbrace{\mathbf{d}_{j-1}^T \mathbf{r}_j}_{=0} = \|\mathbf{r}_j\|_2^2.$$

Dies steht jedoch im Widerspruch zur Voraussetzung $\mathbf{r}_j \neq 0$. □

Bemerkung 8.22

1. Äquivalent zu (8.10) und (8.13), aber effizienter und numerisch stabiler, hat sich die Wahl

$$\alpha_i = \frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{d}_i^T \mathbf{A}\mathbf{d}_i}, \qquad \beta_i = -\frac{\mathbf{r}_{i+1}^T \mathbf{r}_{i+1}}{\mathbf{r}_i^T \mathbf{r}_i}$$

erwiesen.

2. Das Verfahren der konjugierten Gradienten wird generell als Iterationsverfahren verwendet, das heißt, man bricht die Iteration ab, falls $\|\mathbf{r}_i\|_2$ klein ist. Pro Iterationsschritt wird nur eine Matrix-Vektor-Multiplikation benötigt. Die Konvergenz des Verfahrens hängt dabei stark von der Kondition der Matrix ab. Genauer, es gilt die Fehlerabschätzung

$$\|\mathbf{x} - \mathbf{x}_i\|_{\mathbf{A}} \leq 2\left(\frac{\sqrt{\text{cond}_2\,\mathbf{A}} - 1}{\sqrt{\text{cond}_2\,\mathbf{A}} + 1}\right)^i \|\mathbf{x} - \mathbf{x}_0\|_{\mathbf{A}},$$

   siehe z.B. J. Stoer und R. Bulirsch *Numerische Mathematik II*.

3. Es bezeichne $\mathcal{K}_i$ den sogenannten *Krylov-Raum*

$$\mathcal{K}_i := \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots \mathbf{A}^{i-1}\mathbf{r}_0\} = \text{span}\{\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{i-1}\}.$$

   Man kann zeigen, dass die Iterierte $\mathbf{x}_i$ des i-ten Schritts die Funktion

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}$$

   unter allen $\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_i$ minimiert.

△

Index

3/8-Regel, 74
LR-Zerlegung, 22
    mit Rang-1-Updates, 25
QR-Zerlegung, 30, 96
`eps`, 8
Überschuss
    hierarchischer, 76

absolute Konditionszahl, 9
Abstand, 13
Algorithmus
    LR-Zerlegung, 25
    QR-Zerlegung, 100
    adaptive Quadratur, 77
    Auf-dem-Platz-Algorithmus, 23
    cg-Verfahren, 118
    Cholesky-Zerlegung, 34
    Einzelschrittverfahren, 111
    Gesamtschrittverfahren, 111
    Horner-Schema, 43
    Rückwärtssubstitution, 23
    schnelle Fourier-Synthese, 55
    Vorwärtssubstitution, 23
Auf-dem-Platz-Algorithmus, 23
Ausgleichslösung, 93
Ausgleichsproblem, 93
Auslöschung, 10

B-Splines, 64
Banachscher Fixpunktsatz, 105
Basis
    hierarchische, 76
    nodale, 76
Bernoulli-Polynom, 77
Bernoulli-Zahlen, 78
Biegeenergie, 62
Bilinearform
    symmetrische, 84

cg-Verfahren, 118
Cholesky-Zerlegung, 32
Cholesky-Zerlegung mit Rang-1-Updates, 34

Distanz, 13
dividierte Differenzen, 43

Einzelschrittverfahren, 111
Energienorm, 117
Euler-Maclaurinsche Summenformel, 78
Exaktheitsgrad, 72
Exponent, 7

Faltung, 56
FFT, 53
Fixpunkt, 105
    -gleichung, 105
    -iteration, 105
Fourier
    -Synthese, 53
    -Transformation, 53

Gauß-Quadratur, 89
Gauß-Seidel-Verfahren, 111
Gesamtpivotsuche, 20
Gesamtschrittverfahren, 111
Gewicht, 72
Gewichtsfunktion, 85
Gleichungssystem
    überbestimmt, 92
    unterbestimmt, 92

hierarchische Basis, 76
Horner-Schema, 42, 43
Householder-Transformation, 96
Hutfunktion, 75

Innenprodukt, 84
Interpolant
    hierarchischer, 76
Interpolation

Hermite-, 39
kubische Spline-, 61
Lagrange-, 36
lineare Spline-, 60
polynomiale, 36
trigonometrische, 49
Interpolationspolynom
Lagrange-Darstellung, 37
Newton-Darstellung, 42

Jacobi-Verfahren, 111

kleinste-Quadrate-Lösung, 93
Knoten, 36, 72
-polynom, 37
Tschebyscheff-, 46
Kondition
absolute, 9
einer Matrix, 19
relative, 9
Konditionszahl
absolute, 9
relative, 9
Kontraktion, 105
Konvergenz
-ordnung, 110
exponentielle, 83
globale, 110
lineare, 110
lokale, 110
quadratische, 110
Krümmung, 62
Krylov-Raum, 120

Lagrange
-Interpolation, 36
-Polynome, 37
Legendre-Polynom, 87

Mantisse, 7
Maschinengenauigkeit, 8
Maschinenzahl, 7
Matrix
orthogonale, 96
Permutations-, 26
positiv definite, 31
strikt diagonaldominate, 112
Toeplitz-, 56
Vandermonde-, 37
zirkulante, 56
Matrixnorm, 14
Frobenius-Norm, 14
induzierte, 16
Spaltensummennorm, 14
Spektralnorm, 17
submultiplikative, 15
verträgliche, 15
Zeilensummennorm, 14
Milne-Regel, 74
Mittelpunktsregel, 82

Neville-Schema, 41
Newton-
Polynom, 42
Schema, 43
Newton-Cotes-Formeln, 73
Newton-Verfahren, 113
Newtonsche
Interpolationsformel, 44
nodale Basis, 76
Norm, 13
induzierte, 16
Normalengleichungen, 93

Ordnung
der Quadratur, 72
Overflow, 8

Permutationsmatrix, 26
Pivotelement, 20, 21
Pivotisierung
partielle, 20
totale, 20
Polynom
Bernoulli-, 77
Interpolations-, 37, 42
Knoten-, 37
Legendre-, 87
Newton-, 42
trigonometrisches, 49
Tschebyscheff-, 45
Polynome
Lagrange-, 37
Problem
gut konditioniertes, 9
schlecht konditioniertes, 9

Quadratur
3/8-Regel, 74
-ordnung, 72

adaptive, 77
Gauß-Formel, 89
Milne-Regel, 74
Mittelpunktsregel, 82
Simpson-Regel, 73
Trapezregel, 70
Weddle-Regel, 74
Quadraturformel, 72
zusammengesetzte, 72

Rückwärts
-stabilität, 10
-substitution, 23
Randbedingungen
Hermite-, 61
natürliche, 61
periodische, 61
Regel
3/8-, 74
Milne-, 74
Simpson-, 73
Trapez-, 70
Weddle-, 74
relative Konditionszahl, 9
Residuum, 93, 118
Romberg-Verfahren, 80
Rundung, 8

Schema
Horner-, 42
Neville-, 41
Newton-, 43
schlecht konditioniertes Problem, 9
schnelle Fourier-Synthese, 55
Schur-Komplement, 25
Selbstabbildung, 105
kontrahierende, 105
Simpson-Regel, 73
zusammengesetzte, 74
Skalarprodukt, 84
Spaltenpivotsuche, 20
Spline, 59
kubischer, 59
linearer, 59
quadratischer, 59
Stützstelle, 36
Stabilität
Rückwärts-, 10
Vorwärts-, 10
Stirlingsche Formel, 91

Transformation
diskrete Fourier-, 53
Householder, 96
schnelle Fourier-, 53
Trapezregel, 70
zusammengesetzte, 71
Tschebyscheff
-Knoten, 46
Tschebyscheff-Polynom, 45

Underflow, 8

Vandermonde-Matrix, 37
Vektoren
konjugierte, 117
Vektornorm, 14
Betragssummennorm, 14
Euklid-Norm, 14
Maximumsnormorm, 14
Verfahren
der konjugierten Gradienten, 117, 118
Einzelschritt-, 111
Gesamtschritt-, 111
Newton-, 113
Romberg-, 80
Vorwärts
-stabilität, 10
-substitution, 23

Weddle-Regel, 74
Wurzelsatz von Vieta, 12