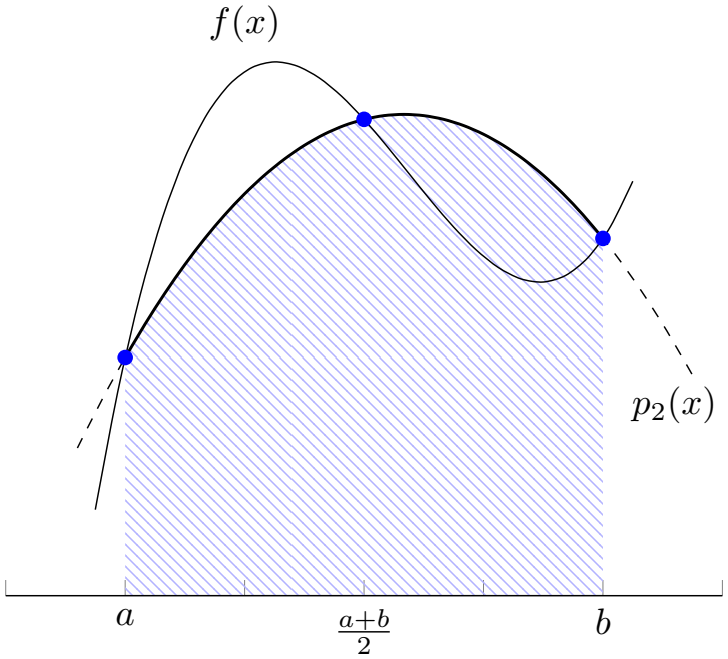
Vorlesungsskript

Einführung in die Numerik

Robert Scheichl & Nils Friess

r.scheichl@uni-heidelberg.de
Institut für Angewandte Mathematik und
Interdisziplinäres Zentrum für wissenschaftliches Rechnen (IWR)
Universität Heidelberg



Letzte Änderung: 24. Februar 2021

Inhaltsverzeichnis

1 Einführung

Im folgenden Kapitel motivieren wir das mathematische Teilgebiet der Numerik anhand einiger Beispiele und erläutern die Ziele der Vorlesung.

1.1 Warum Numerische Mathematik?

Forschung in der Technik und den Naturwissenschaften basiert auf der wissenschaftlichen Methode, bestehend aus:

- **Experiment:** Beobachte und messe was passiert;
- **Theorie:** Versuche die Beobachtung mit Hilfe von *Modellen* zu erklären.

Theorie und Experiment werden sukzessive verfeinert und verglichen, bis eine akzeptable Übereinstimmung vorliegt. Die Modelle liegen oft in Form von mathematischen Gleichungen vor (z. B. gewöhnliche oder partielle Differentialgleichungen). Typischerweise können diese Modellgleichungen nicht geschlossen gelöst werden (d. h. mit Papier und Bleistift oder Computeralgebrasystemen wie *Mathematica*).

Deshalb bedarf es **numerischer Simulation**, d.h. mathematische Modelle numerisch lösen. Untrennbar damit verbunden sind die Gebiete der **Optimierung** und der **Unsicherheitsbestimmung (Uncertainty Quantification)**. Das ermöglicht

- Ersetzen undurchführbarer Experimente (z. B. Atommüllendlager),
- Einsparen teurer Experimente (z. B. Windkanaltests),
- schneller durchführbare Parameterstudien,
- (automatische) Optimierung von Prozessen,
- Identifikation von Modellparametern aus Messungen,
- Abschätzung von Unsicherheiten.

Das führt zu vielfältigen Einsatzmöglichkeiten in Naturwissenschaft, Technik und Industrie: Strömungsberechnungen (Wetter, Klima, Blutkreislauf), Festigkeit von Brücken oder Flugzeugtragflächen und viele weitere.

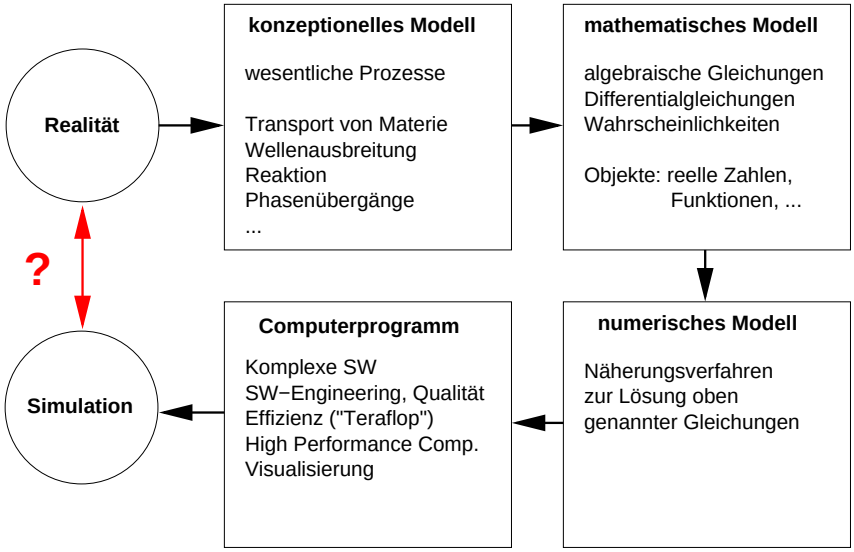


Abbildung 1.1: Von der Realität zur Simulation mittels wissenschaftlichem Rechnen.

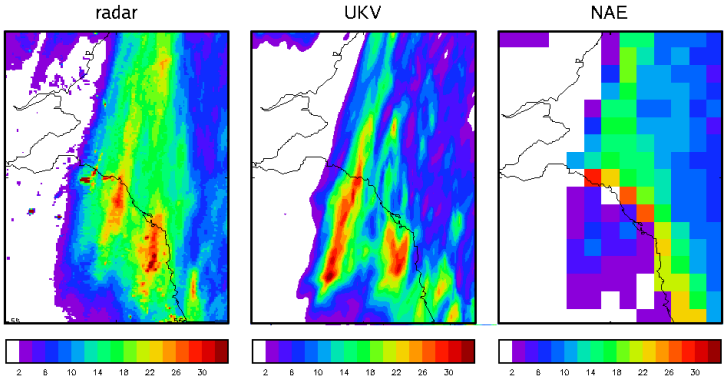


Abbildung 1.2: Ein Radar Bild sowie zwei Wettervorhersagen des UK Met Office: UKV = Model für Großbritannien (Auflösung = 1.5km); NAE = Nord Atlantk Model (Auflösung = 12km).

Grundlage für all diese Anwendungen sind numerische Algorithmen!

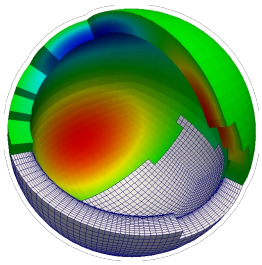
Eine erfolgreiche Simulation erfordert dabei die interdisziplinäre Zusammenarbeit von Physikern oder Ingenieuren mit Mathematikern und Informatikern, die bspw. in den Bereichen Software-Engineering oder High Performance (Parallel) Computing zentral beteiligt sind (vgl. auch Abb. 1.1).

Beispiel (Wetter- und Klimavorhersage)**.** Das 12km Nordatlantik Modell des UK Met Office kann den Zeitpunkt und die genaue Position einer Sturmfront nicht vorhersagen. Das 1.5km UK Modell schon (vgl. Abb 1.2).

Das UK Met Office hat den Plan ca. 2025 ihr globales Modell auf eine Gitterauflösung mit ca. 1km umzustellen. Betrachten wir, was das bedeutet:

Die zeitliche Veränderung der Zustandsgrößen

$$\vec{\phi} := (u, v, w, p, \theta, \rho),$$



d.h. Windgeschwindigkeiten, Druck, Temperatur und Dichte, lässt sich durch die Euler-Gleichungen beschreiben:

$$\frac{\mathrm{D}\vec{\phi}}{\mathrm{D}t}(\vec{x}, t) = N\left[\vec{\phi}(\vec{x}, t)\right] + f_{\phi}^{\text{subgrid}}(\vec{x}, t)\,,$$

wobei $N(\cdot)$ ein nichtlinearer (örtlicher) Differentialoperator ist und $f_{\phi}^{\text{subgrid}}$ Prozesse unter der Gitterauflösung beinhaltet, die auch nichtlinear von ϕ abhängen.

Man kann leicht ausrechnen, dass zum Vernetzen der Erdatmosphäre mit einem Gitter von 1km horizontaler Gitterweite und 200 vertikalen Schichten ca. 10^{11} Gitterzellen benötigt werden. Um die Euler-Gleichungen auf dem 1km-Gitter numerisch stabil zu lösen, kann man höchstens Zeitschritte von 30 Sekunden verwenden, d. h. ca. 15,000 Zeitschritte für eine Simulation von fünf Tagen. In jedem dieser Zeitschritte muss mehrmals ein *lineares Gleichungssystem* der Form $Ax = b$ mit **10^{11} Unbekannten** gelöst werden. Jede Nacht steht jedoch nur eine Stunde zur Lösung dieser partiellen Differentialgleichung zur Verfügung (neben anderen Aufgaben, wie Datenassimilierung, lokalen Detailsimulationen, ...), d. h. zur Lösung jedes linearen Gleichungssystems mit 10^{11} Unbekannten stehen ungefähr 0.05s zur Verfügung!

Dazu benötigt man:

- einen Höchstleistungsrechner (Top 500),
- mathematisch optimal skalierende Algorithmen, wie z. B. Multigrid,
- Chip-optimierten parallelen Code,

Siehe, z. B., [E.H. Müller, R. Scheichl, E. Vainikko „Petascale solvers for anisotropic PDEs in atmospheric modelling on GPU clusters“, *Parallel Computing* **50**, 2015] (ein Physiker, ein Mathematiker und ein Informatiker!), wo eine Leistung von **0.78 PetaFLOP** auf **16384 GPUs** von Titan[1] (Nr. 2 der Welt in 2015) erreicht wurden, um ein lineares Gleichungssystem mit 10^{11} Unbekannten in $\sim$0.2 Sekunden zu lösen.

1.2 Lernziele

In dieser Vorlesung soll/sollen

[1] Siehe `https://www.top500.org/system/177975`.

- anhand von Beispielen vermittelt werden, warum numerische Mathematik und wissenschaftliches Rechnen **unentbehrlich** für das Lösen mathematischer, wissenschaftlicher und technischer Probleme sind;
- einige **grundlegende numerische Methoden** beschrieben und entwickelt werden, die den Studenten die Fähigkeit und das „Handwerkszeug“ geben, eine **breite Vielfalt von Problemen rechnerisch zu lösen**;
- die vorgestellten und entwickelten Algorithmen bezüglich ihrer Genauigkeit, Robustheit und Effizienz **mathematisch analysiert** werden.

Des Weiteren befassen wir uns mit verschiedenen Fehlerquellen, die Unterschiede zwischen Experiment und Simulation erklären, wie zum Beispiel:

1. **Modellfehler:** Ein relevanter Prozess wurde nicht oder ungenau modelliert (Temperatur als konstant angenommen, Luftwiderstand vernachlässigt, ...).
2. **Datenfehler:** Messungen von Anfangsbedingungen, Randbedingungen, Werte für Parameter sind fehlerbehaftet.
3. **Approximationsfehler:** Abbruch von Reihen oder Iterationsverfahren, Approximation von Funktionen (z. B. stückweise Polynome).
4. **Rundungsfehler:** Reelle Zahlen werden im Rechner genähert dargestellt.

In dieser Vorlesung konzentrieren wir uns auf die Untersuchung von Punkten 3 und 4, d.h. Rundungsfehler und Abschneidefehler (nicht Modell- und Datenfehler), und untersuchen außerdem Stabilität und Effizienz von numerischen Methoden.

Wir wollen zu Beginn ein paar der Begriffe anhand eines Beispiels kennenlernen:

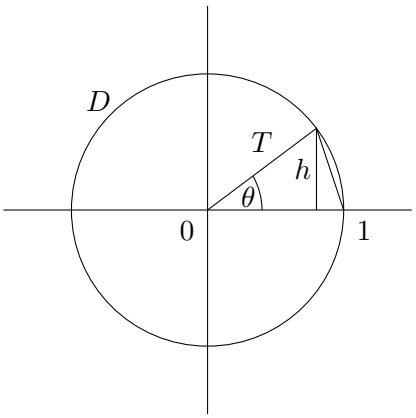
Beispiel 1.1 (Numerische Approximation von π)**.** Wir wollen π so genau wie möglich und nur mittels der Operationen $+, -, \times, \div, \sqrt{}$ (typisch für Computer) berechnen.

Dazu unterteilen wir die Einheitskreisscheibe D zunächst in $2N$ gleiche Segmente, die wir mit T bezeichnen. Dann ist

$$\theta = \frac{2\pi}{2N} = \frac{\pi}{N}$$

und

$$h = \sin\left(\frac{\pi}{N}\right),$$



und wir können somit π approximieren durch

$$\pi = |D| = 2N \cdot |T| \;\approx\; 2N \cdot \frac{1}{2}\sin\left(\frac{\pi}{N}\right) = N \sin\left(\frac{\pi}{N}\right). \tag{1.1}$$

Aber wie berechnet man $\sin(\frac{\pi}{N})$ ohne π zu benutzen? Hierzu betrachten wir die folgenden trigonometrischen Identitäten:

$$\cos(2\theta) = 1 - 2\sin^2\theta \tag{T1}$$

$$\cos(2\theta) = \sqrt{1 - \sin^2(2\theta)} \tag{T2}$$

Für $\theta \leq \frac{\pi}{4}$ ist dann die Gleichung (T1) − (T2) äquivalent zu

$$\sin\theta = \sqrt{\frac{1 - \sqrt{1 - \sin^2(2\theta)}}{2}}\,. \tag{1.2}$$

Dadurch kann man folgende *Rekursion* definieren:

$$\sin\left(\frac{\pi}{2}\right) = 1\,, \tag{1.3}$$

$$\sin\left(\frac{\pi}{2n}\right) = \sqrt{\frac{1 - \sqrt{1 - \sin^2(\pi/n)}}{2}}\,. \tag{1.4}$$

Durch rekursives Anwenden von (1.4) erhalten wir somit

$$\sin\left(\frac{\pi}{N}\right) \text{ für } N = 2^2, 2^3, 2^4, \ldots$$

Formel (1.4) ist ein typisches Beispiel einer **Iteration** – einfache arithmetische Operationen werden wiederholt angewendet, um eine Näherungslösung eines „schwierigeren“ Problems zu finden.

Obige Iteration ist sehr leicht zu programmieren (z. B. in Matlab):

```
K = input('Waehlen Sie die Anzahl der Iterationen K: ');
s = 1;     % s = sin(pi/2)
N = 2;     % Initialisieren von N

for i = 2:K % i = 2, 3, ..., K
    s = sqrt((1-sqrt(1-s^2))/2);
    N = 2*N;
end

p = N*s;

fprintf('K =%3.0d, N = %4.0d: Naeherung fuer pi =  %12.10g',
   K,N,p)
```

Das Programm erzeugt folgende Ausgabe:

K	N	$A_N := N \sin \frac{\pi}{N}$	$\Delta_N := A_N - A_{N/2}$	$\frac{\Delta_{N/2}}{\Delta_N}$
2	4	2.828427		
3	8	3.061467	2.33×10^{-1}	
4	16	3.121445	6.00×10^{-2}	3.89
5	32	3.136548	1.51×10^{-2}	3.97
6	64	3.140331	3.78×10^{-3}	3.99
7	128	3.141277	9.46×10^{-4}	4.00
8	256	3.141514	2.37×10^{-4}	4.00

A_N scheint gegen π zu konvergieren, wenn $N \to \infty$. Um zu schätzen *wie schnell*, machen wir den Ansatz

$$A_N = \pi - CN^{-\alpha} \tag{1.5}$$

und schätzen die Konstanten $C, \alpha > 0$. Wenn (1.5) gilt, dann folgt

$$\Delta_N = A_N - A_{N/2} = C\left(N^{-\alpha} - \left(\frac{N}{2}\right)^{-\alpha}\right) = C(1 - 2^{\alpha})N^{-\alpha}$$

und wir erhalten

$$\frac{\Delta_{N/2}}{\Delta_N} = 2^{\alpha}.$$

Aus der Tabelle kann man ersehen, dass $\alpha \approx 2$.

Definition 1.2 (Landausche Symbole)**.**

(a) Für zwei Funktionen $g(t), h(t)$ der Variablen $t \in \mathbb{R}_+$ schreibt man

$$g(t) = \mathcal{O}(h(t)), \quad \text{für } t \to 0$$

falls $t_0 > 0, C \geq 0$ existieren, sodass für alle $t \in (0, t_0]$ gilt, dass

$$|g(t)| \leq C|h(t)|$$

(Anschauliche Bedeutung: „$g(t)$ geht wie $h(t)$ gegen 0“).

(b) Außerdem schreibt man

$$g(t) = o(h(t)), \quad \text{für } t \to 0\,,$$

falls

$$\lim_{t \to 0} \left| \frac{g(t)}{h(t)} \right| = 0$$

(Anschauliche Bedeutung: „$g(t)$ geht schneller als $h(t)$ gegen 0“).

Unsere Experimente lassen die Vermutung zu, dass

$$|A_N - \pi| \approx \mathcal{O}(N^{-2}) .$$

Können wir das mathematisch rigoros beweisen? Um diese Frage zu beantworten, benötigen wir zunächst einen der wichtigsten Sätze der Numerik.

Satz 1.3 (Satz von Taylor)**.** *Sei $f : (a, b) \to \mathbb{R}$ $(m+1)$-mal stetig differenzierbar und seien $x, x_0 \in (a, b)$ mit $x_0 \neq x$. Dann gilt*

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 + \ldots \\ \ldots + \frac{f^{(m)}(x_0)}{m!}(x - x_0)^m + R_m(x)$$

mit Restglied

$$R_m(x) = \frac{f^{(m+1)}(\xi)}{(m+1)!}(x - x_0)^{m+1}$$

für ein ξ strikt zwischen x und x_0, d. h. $\xi \in (x, x_0)$ oder $\xi \in (x_0, x)$.

Beweis. *Analysis I*, *Mathe für Informatiker II*, bzw. [For16, §22, Satz 1 und 2]. □

Wir können nun diesen Satz anwenden, um die Konvergenz unserer Methode zur Berechnung von π zu zeigen:

Satz 1.4. *Es existiert ein $C > 0$ konstant, sodass für alle $N \geq 2$ gilt:*

$$|A_N - \pi| \leq \frac{C}{N^2} .$$

Beweis. Man wende Taylors Satz auf $f(x) = \sin(x)$ an mit $(a, b) = (-\frac{\pi}{2}, \frac{\pi}{2})$, $x \in (0, \frac{\pi}{2})$, $x_0 = 0$ und $m = 2$. Dann existiert ein $\xi \in (0, x)$, sodass

$$\sin(x) = x - \frac{x^3}{3!}\cos(\xi).$$

Also, für $N \geq 2$,

$$\sin\left(\frac{\pi}{N}\right) = \frac{\pi}{N} - \frac{\pi^3}{6N^3}\cos(\xi)$$

woraus durch Umstellen folgt

$$\left|N \sin\left(\frac{\pi}{N}\right) - \pi\right| = \frac{\pi^3}{6N^2}|\cos(\xi)| \leq \frac{\pi^3}{6}N^{-2} \qquad (\text{d. h. } C = \tfrac{\pi^3}{6} \text{ reicht}).$$

□

Wir wollen in diesem Kurs sowohl **Verlässlichkeit** als auch **Effizienz** untersuchen, deshalb betrachten wir auch die **Komplexität** des Algorithmus:

- Sei $N = 2^{K+1}$ (wobei K die Anzahl der Iterationen ist).
- In (1.4) werden in jeder Iteration zwei Subtraktionen, eine Multiplikation, eine Division und zwei $\sqrt{}$ benötigt, also insgesamt sechs Operationen (wenn alle gleich gezählt werden).
- Da (1.4) K-mal ausgewertet wird, sind das $6K$ Operationen.
- In jeder Iteration gibt es einen Update von N ($N = 2N$) und am Schluss werden s und N multipliziert, d.h gesamt

$$7K + 1 \text{ Operationen.}$$

Die Komplexität wächst also *linear* mit K wenn $K \to \infty$ oder (äquivalent) *logarithmisch* mit N, d. h. proportional zu $\log_2 N$, wenn $N \to \infty$.

Der dritte Punkt, den wir in diesem Kurs untersuchen wollen ist die **Robustheit** (oder **Stabilität**) von Algorithmen.

Frage: Ist der durch (1.4) beschriebene Algorithmus robust? Untersuchen Sie, was passiert, wenn man $K > 10$ wählt? Woran kann das liegen?

Danksagung

Das vorliegende Vorlesungsskript basiert fast zur Gänze auf dem Vorlesungsskript von Prof. Rolf Ranacher [Ran17], welches selbst eng dem Buch von Stoer und Bullirsch [SB] folgt. Kapitel 3.7 zu Haar-Wavelets ist aus dem Vorlesungsskript von Prof. Peter Bastian [Bas20] adaptiert.

Die erste Version des Skripts entstand im Wintersemester 2018/19 und wurde im Laufe des Wintersemesters 2020/21 zu dem vorliegenden Skript überarbeitet. Der erste Autor dankt dem zweiten Autor für die sorgfältig ausgearbeitete Latex-Version inklusive der klaren Illustrationen.

2 Zwischenspiele zu wichtigen Kernkonzepten der Numerik

In diesem Kapitel führen wir zuerst zwei Begriffe ein, die wesentlich in der Beschreibung numerischer Aufgaben und numerischer Algorithmen sind. Danach befassen wir uns mit der Darstellung von Zahlen auf einem Computer und mit der Analyse von Rundungsfehlern. Um nicht gleich mit abstrakten Konzepten zu starten und den Fluss der Vorlesung nicht zu sehr zu stören, werden die folgenden Kapitel **an passenden Stellen in der Vorlesung als sogenannte „Zwischenspiele"** eingebaut.

2.1 Konditionierung

Unter einer **numerischen Aufgabe** versteht man die Berechnung endlich vieler Größen $y_i,\ i = 1, \ldots, n$ aus einer Reihe anderer Größen $x_j,\ j = 1, \ldots, m$, mittels einer funktionalen Vorschrift

$$y_i = f_i(x_1, \ldots, x_m), \quad i = 1, \ldots, n,$$

oder kompakt

$$y = f(x)\,.$$

Das Lösen eines linearen Gleichungssystems $Ax = b$ führt bspw. zur numerischen Aufgabe

$$x = f(A, b) := A^{-1}b\,.$$

Definition 2.1. Bei Verwendung fehlerhafter Eingangsdaten $x_j + \Delta x_j$ (z. B. aufgrund von Rundungs- oder Messfehlern) ergeben sich fehlerhafte Resultate $y_i + \Delta y_i$. Wir bezeichnen mit

$$|\Delta x_j| \qquad \text{den absoluten Datenfehler ,}$$

$$\left|\frac{\Delta x_j}{x_j}\right| \qquad \text{den relativen Datenfehler ;}$$

$\lvert\Delta y_j\rvert$	den absoluten Fehler (im Resultat) ;
$\left\lvert\dfrac{\Delta y_j}{y_j}\right\rvert$	den relativen Fehler (im Resultat) .

Bemerkung. Große absolute Fehler können, *relativ gesehen*, klein sein und umgekehrt; ±100km beim Messen der Distanz Erde–Mond kann als (relativ) klein angesehen werden, während der selbe Fehler auf die Distanz Heidelberg–Paris relativ groß ist.

Um den Einfluss relativ kleiner Datenfehler $|\Delta x_j| \ll |x_j|$ (z. B. Rundungsfehler) auf das Resultat einer numerischen Aufgabe zu bestimmen, betreiben wir nun eine sogenannte **differentielle Fehleranalyse**. [1]

Nehmen wir an, dass jedes der f_i zweimal stetig partiell differenzierbar ist bezüglich x_j für $j = 1, \ldots, m$, und dass $\|\Delta x\| \ll |y_i|$ für alle $i = 1, \ldots, n$ (d. h. die Datenfehler sind kleiner als die Resultate, absolut betrachtet). Daraus folgt für $i = 1, \ldots, m$ mit dem Satz von Taylor im Mehrdimensionalen (vgl. Satz 1.3 für $m = 1$), dass

$$\Delta y_i = f_i(x + \Delta x) - f_i(x) = \sum_{j=1}^{m} \frac{\partial f_i}{\partial x_j}(x)\Delta x_j + R_i^f(x; \Delta x)\,.\text{‘} \tag{2.1}$$

Für das Restglied gilt

$$R_i^f(x; \Delta x) = \mathcal{O}(\|\Delta x\|^2) \tag{2.2}$$

und deshalb

$$\begin{aligned} \frac{\Delta y_i}{y_i} &= \sum_{j=1}^{m} \frac{\partial f_i}{\partial x_j}(x)\frac{\Delta x_j}{y_i} + \mathcal{O}(\|\Delta x\|^2) \\ &= \sum_{j=1}^{m} \underbrace{\left(\frac{\partial f_i}{\partial x_j}(x)\frac{x_j}{f_i(x)}\right)}_{=:\ k_{ij}(x)} \frac{\Delta x_j}{x_j} + \mathcal{O}(\|\Delta x\|^2) \end{aligned} \tag{2.3}$$

Definition 2.2 (Konditionierung)**.** Die Größen $k_{ij}(x)$ in (2.3) heißen **relative Konditionszahlen** der Funktion f im Punkt x (sie sind ein Maß, wie relativ kleine Datenfehler sich im Resultat auswirken). Man nennt die Aufgabe $y = f(x)$ **schlecht konditioniert**, wenn eines der $|k_{ij}(x)| \gg 1$; andernfalls heißt die Aufgabe **gut konditioniert**.

Im Fall $|k_{ij}(x)| < 1$ spricht man von **Fehlerdämpfung**, im Fall $|k_{ij}(x)| > 1$ spricht man von **Fehlerverstärkung**.

[1] Das Symbol $\ll$ steht für „viel kleiner als“.

Beispiel 2.3. Wir betrachten die Konditionierung arithmetischer Grundoperationen:

(a) Die Addition

$$y = f(x) := x_1 + x_2, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2, \quad x_1, x_2 \neq 0$$

mit

$$k_1(x) = \frac{\partial f}{\partial x_1}(x)\frac{x_1}{f(x)} = 1\frac{x_1}{x_1 + x_2} = \frac{1}{1 + x_2/x_1},$$

und

$$k_2(x) = \frac{\partial f}{\partial x_2}(x)\frac{x_2}{f(x)} = 1\frac{x_2}{x_1 + x_2} = \frac{1}{1 + x_1/x_2}$$

ist **schlecht konditioniert** für $x_1 \approx -x_2$. Bei Addition ähnlich großer Zahlen mit unterschiedlichen Vorzeichen kann bei Fortpflanzung von kleinen Datenfehlern sogar **Auslöschung** wesentlicher (dezimaler) Stellen auftreten.

(b) Die Multiplikation $y = f(x) = x_1 \cdot x_2$, $x \in \mathbb{R}^2$, mit

$$k_1(x) = \frac{\partial f}{\partial x_1}(x)\frac{x_1}{f(x)} = x_2\frac{x_1}{x_1 x_2} = 1$$

und (analog)

$$k_2(x) = 1$$

ist generell *gut konditioniert.*

(c) Auch die Division ist generell gut konditioniert. MMS

Beispiel 2.4. Betrachten wir das lineare Gleichungssystem

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.8642 \\ 0.1440 \end{pmatrix}$$

mit (eindeutiger) Lösung $(x \quad y)^\intercal = (2 \quad -2)^\intercal$. Eine Störung der rechten Seite zu $(0.86419999 \quad 0.14400001)^\intercal$, d. h. ein relativer Datenfehler von $\mathcal{O}(10^{-8})$ erzeugt die **Näherungslösung**

$$(x + \Delta x \quad y + \Delta y)^\intercal = (0.9911 \quad -0.4870)^\intercal,$$

d. h. ein relativer Fehler im Resultat von $\mathcal{O}(1)$. MMS

2.2 Stabilität

Leider kann ein schlecht gewähltes Verfahren (**Algorithmus**) auch bei einer gut konditionierten Aufgabe zu einer großen Fehlerverstärkung führen. Man spricht in diesem Fall von **Instabilität** des Algorithmus.

Für eine gegebene numerische Aufgabe $y = f(x)$ mit $f : \mathbb{R}^m \to \mathbb{R}^n$ versteht man unter einem **numerischen Verfahren** (Algorithmus) zur (näherungsweisen) Berechnung von y aus x eine endliche Folge von **elementaren Abbildungen** (Operationen) $\varphi^{(k)}$, die durch sukzessive Anwendung einen Näherungswert $\tilde{y}$ zu y liefern:

$$x \longrightarrow \varphi^{(1)}(x) \longrightarrow \varphi^{(2)}(\varphi^{(1)}(x)) \longrightarrow \ldots \longrightarrow \varphi^{(k)}(\cdots\varphi^{(2)}(\varphi^{(1)}(x))\cdots) =: \tilde{y}\,.$$

Im einfachsten Fall sind die $\varphi^{(k)}$ arithmetische Grundoperationen.

Definition 2.5 (Stabilität)**.** Bei der Durchführung eines Algorithmus mit gestörten Daten treten in jedem Schritt Fehler auf, die sich bis zum Ende der Rechnung akkumulieren können. Der Algorithmus wird **stabil** genannt, wenn die im Laufe der Ausführung akkumulierten Fehler den durch die Konditionierung bedingten unvermeidbaren Problemfehler **nicht signifikant** übersteigen.

Aufgrund der (endlichen) Darstellung von reellen Zahlen am Rechner müssen Daten und Zwischenresultate gerundet werden und sind generell **immer** gestört (siehe Kapitel 2.4). Deshalb reicht es im Allgemeinen schon, dass eine der elementaren Abbildung $\varphi^{(k)}$ schlecht konditioniert ist, dass ein Algorithmus instabil ist.

Beispiel 2.6 (Lösen quadratischer Gleichungen)**.** Gegeben seien $p, q \in \mathbb{R}$ mit $q \neq 0$. Wir betrachten die quadratische Gleichung

$$y^2 - py + q = 0\,.$$

Für die Lösungen („Wurzeln“) y_1, y_2 gilt

$$y_1 = f_1(p,q) = \frac{p}{2} + \sqrt{\frac{p^2}{4} - q} \qquad \text{und} \qquad y_2 = f_2(p,q) = \frac{p}{2} - \sqrt{\frac{p^2}{4} - q}$$

sowie

$$y_1 + y_2 = p \quad \text{und} \quad y_1 y_2 = q\,.$$

Daraus folgt

$$\frac{\partial f_1}{\partial p} + \frac{\partial f_2}{\partial p} = 1 \quad \text{und} \quad \frac{\partial f_1}{\partial p} y_2 + \frac{\partial f_2}{\partial p} y_1 = 0$$

und somit

$$\frac{\partial f_1}{\partial p} = \frac{y_1}{y_1 - y_2} \quad \text{und} \quad \frac{\partial f_2}{\partial p} = \frac{y_2}{y_2 - y_1}\,.$$

Damit erhalten wir

$$k_{11}(p,q) = \frac{\partial f_1}{\partial p}\frac{p}{y_1} = \frac{y_1}{y_1 - y_2}\frac{y_1 + y_2}{y_1} = \frac{1 + \frac{y_2}{y_1}}{1 - \frac{y_2}{y_1}} \;=\; -k_{21}(p,q)\,.$$

MMS

Auf ähnliche Weise folgt

$$k_{12}(p,q) = \frac{1}{1 - \frac{y_1}{y_2}} \quad \text{and} \quad k_{22}(p,q) = \frac{1}{1 - \frac{y_2}{y_1}}\,.$$

Wir sehen also, dass die Berechnung von y_1, y_2 für $y_1 \approx y_2$, d. h. für dicht nebeneinanderliegende Wurzeln, schlecht konditioniert ist.

Betrachten wir nun die gut konditionierte Aufgabe

$$y^2 - py + q = 0 \quad \text{mit } p < 0 \;\text{ und }\; 0 \neq q \ll \frac{p^2}{4}\,,$$

d. h. $|y_1/y_2| \gg 1$. Der Algorithmus zur Berechnung von y_1, y_2 könnte wie folgt aussehen: Setze

$$u = \frac{p^2}{4}, \quad v = u - q, \quad w = \sqrt{v}\,.$$

Diese Operationen führen alle zu keiner wesentlichen Fehlerverstärkung:

$$\begin{aligned}
\frac{\Delta u}{u} &= \left(\frac{2p}{4}\frac{p}{u}\right)\frac{\Delta p}{p} + \mathcal{O}(|\Delta p|^2) = 2\frac{\Delta p}{p} + \mathcal{O}(|\Delta p|^2) \\
\frac{\Delta v}{v} &= \underbrace{\left(1\frac{u}{v}\right)}_{\approx 1}\frac{\Delta u}{u} + \underbrace{\left((-1)\frac{q}{v}\right)}_{\ll 1}\frac{\Delta q}{q} + \mathcal{O}(|\Delta u|^2 + |\Delta q|^2) \\
\frac{\Delta w}{w} &= \left(\frac{1}{2}v^{-1/2}\frac{v}{w}\right)\frac{\Delta v}{v} + \mathcal{O}(|\Delta v|^2) = \frac{1}{2}\frac{\Delta v}{v} + \mathcal{O}(|\Delta v|^2)
\end{aligned}$$

Um Auslöschung zu vermeiden, berechnen wir zuerst

$$\tilde{y}_2 = \frac{p}{2} - w$$

und erhalten für $\Delta y_2 = \tilde{y}_2 - y_2$

$$\begin{aligned}
\frac{\Delta y_2}{y_2} &= \left(\frac{1}{2}\frac{p}{y_2}\right)\frac{\Delta p}{p} + \left((-1)\frac{w}{y_2}\right)\frac{\Delta w}{w} + \mathcal{O}\left(|\Delta p|^2 + |\Delta w|^2\right) \\
&= \underbrace{\left(\frac{p}{p - 2w}\right)}_{\approx \frac{1}{2}}\frac{\Delta p}{p} + \underbrace{\left(\frac{2w}{p - 2w}\right)}_{\approx -\frac{1}{2}}\frac{\Delta w}{w} + \mathcal{O}\left(|\Delta p|^2 + |\Delta w|^2\right)\,,
\end{aligned}$$

da $w \approx -\frac{p}{2}$.

Um die zweite Wurzel zu berechnen, betrachten wir

$$\begin{array}{ll} \text{Variante A:} & \text{Variante B:} \\ \tilde{y}_1 = \dfrac{p}{2} + w & \tilde{y}_1 = \dfrac{q}{\tilde{y}_2} \end{array}$$

Wegen $w \approx -\frac{p}{2}$ kommt es bei Variante A zwangsläufig zu Auslöschung:

$$\frac{\Delta y_1}{y_1} = \underbrace{\left(\frac{p}{p+2w}\right)}_{\gg 1} \frac{\Delta p}{p} + \underbrace{\left(\frac{2w}{p+2w}\right)}_{\gg 1} \frac{\Delta w}{w} + \mathcal{O}\left(|\Delta p|^2 + |\Delta w|^2\right).$$

Der Algorithmus ist also für $q \ll p^2/4$ **sehr** instabil.

Bei Variante B dagegen gilt

$$\frac{\Delta y_1}{y} = \underbrace{\left(\frac{1}{\tilde{y}_2}\frac{q\tilde{y}_2}{q}\right)}_{=1} \frac{\Delta q}{q} + \underbrace{\left(\frac{q}{\tilde{y}_2^2}\frac{\tilde{y}_2^2}{q}\right)}_{=1} \frac{\Delta y_2}{y_2} + \mathcal{O}\left(|\Delta q|^2 + |\Delta y_2|^2\right),$$

d. h. der Algorithmus ist stabil.

Bemerkung 2.7. Bei der Lösung quadratischer Gleichungen sollten als **nicht** beide Wurzeln aus der klassischen Lösungsformel berechnet werden!

Beispiel 2.8. Sei bspw. $p = -4$, $q = 0.01$ und wir runden auf 4 Stellen. Mit den Werten

$$\begin{aligned} \tilde{u} &= 4 \qquad \text{(exakt)} \\ \tilde{v} &= 3.99 \quad \text{(exakt)} \\ \tilde{w} &= 1.997(4984\ldots) \\ \tilde{y}_2 &= -3.997 \end{aligned}$$

erhalten wir für das exakte Ergebnis $y_1 = -0.0025015645\ldots$ die Näherungslösungen

$$\begin{aligned} &\text{Variante A:} \quad \tilde{y}_1 = -0.003 \\ &\text{Variante B:} \quad \tilde{y}_1 = -0.002502(018764\ldots) \end{aligned}$$

Der relative Fehler ist also **20%** in A und **0.0174%** in B!

2.3 Zahlendarstellung

Bei der Ausführung numerischer Algorithmen auf einem Computer treten durch die notwendigerweise endliche Darstellung von Zahlen zwangsläufig Fehler auf. Wir

betrachten nun wie Zahlen am Computer dargestellt werden und kommen dann nochmal zurück auf Konditionierung und Stabilität (vgl. Kapitel 2.1-2.2).

Zur Approximation reeller (und komplexer) Zahlen und elementaren arithmetischen Operationen werden sog. **Maschinenzahlen** und **Maschinenoperationen** verwenden, die am Computer realisierbar sind.

Definition 2.9. (a) Eine **(normalisierte) Fließkommazahl** zur **Basis** $b \in \mathbb{N},\ b \geq 2$, ist eine Zahl, dargestellt in der Form

$$x = \pm\, m \times b^{\pm e} \tag{2.4}$$

mit der **Mantisse**

$$m = m_1 b^{-1} + m_2 b^{-2} + \ldots + m_r b^{-r} + \ldots \ \in \mathbb{R}$$

und dem **Exponenten**

$$e = e_0 b^0 + e_1 b^1 + \ldots + e_{s-1} b^{s-1} + \ldots \ \in \mathbb{N}\,,$$

wobei $m_i, e_i \in \{0, \ldots, b-1\}$. Jedes $x \in \mathbb{R}$ kann so dargestellt werden.

(b) Mit den Bedingungen $m_k = 0$ für alle $k > r$ und $e_k = 0$ für alle $k \geq s$ erhält man einen Unterraum der reellen Zahlen mit **endlicher Fließkommadarstellung**, das numerische **Fließkommagitter**

$$\mathbb{F} := \mathbb{F}(b, r, s) \subset \mathbb{R}\,. \tag{2.5}$$

(c) Jedes $x \neq 0$ ist eindeutig bestimmt durch die Zusatzbedingung $m_1 \neq 0$.

(d) Für $x = 0$ setzen wir $m = 0$ und $e = 0$.

Am Computer stehen für jede Zahl aufgrund von endlichem Speicher nur endlich viele Stellen zur Verfügung. In $\mathbb{F}(b, r, s)$, müssen für jede Zahl

- r Ziffern + 1 Vorzeichen für die Mantisse
- s Ziffern + 1 Vorzeichen für den Exponenten

gespeichert werden, d. h. gesamt $r + s + 2$ Werte. Alternativ könnte man mit **Festkommazahlen** arbeiten, d. h.

$$x = \pm \sum_{i=-k}^{l} m_i b^i\,,$$

was für wissenschaftliche Anwendungen aber aufgrund der sehr unterschiedlichen Größen vieler Zahlen ineffizient und unbrauchbar ist:

- Ruhemasse eines Elektrons $= 0.9109384 \times 10^{-30}$ kg
- Lichtgeschwindigkeit $= 0.2997925 \times 10^{9}$ $^{\mathrm{m}}/_{\mathrm{s}}$
- Avogadrokonstante $= 0.6021415 \times 10^{24}{}^{1}/_{\mathrm{mol}}$

Um alle drei Zahlen mit ähnlicher Genauigkeit wie in der üblichen Fließkommadarstellung (z. B. "double precision" mit Basis $b = 2$, siehe unten), müsste man über 200 Werte speichern.

Bemerkung 2.10. Da $\mathbb{F}(b, r, s)$ endlich ist, gibt es eine größte/kleinste darstellbare Zahl. Mit $m_i = (b-1) = e_i$ (und einem positiven Vorzeichen im Exponenten) erhält man

$$\begin{aligned} x^{+}_{\max} &= (b-1)(b^{-1} + \ldots + b^{-r}) b^{(b-1)(b^0+\ldots+b^{s-1})} \\ &= (b-1) b^{-1} \frac{b1-r-1}{b^{-1}-1} b^{(b-1)\frac{b^s-1}{b-1}} \\ &= (1-b^{-r}) b^{b^s-1}\,, \end{aligned} \tag{2.6}$$

sowie

$$x^{-}_{\min} = -(1-b^{-r}) b^{b^s-1}\,. \tag{2.7}$$

Auch nahe 0 gibt es eine kleinste positive bzw. größte negative, darstellbare Zahl. Setze dazu $m_1 = 1$, $m_i = 0$ für $i > 1$ und $e_i = b-1$ (und einem negativen Vorzeichen im Exponenten), folgt

$$x^{+}_{\min} = b^{-1} b^{-(b-1)(b^0+\ldots+b^{s-1})} = b^{-b^s} \tag{2.8}$$

und

$$-b^{-b^s}\,. \tag{2.9}$$

Beispiel 2.11. $\mathbb{F}(10, 3, 1)$ besteht aus $x = 0$ sowie Zahlen der Form

$$x = \pm(m_1 \times 0.1 + m_2 \times 0.01 + m_r \times 0.001) \times 10^{\pm e_0}$$

mit $m_1 \in \{1, \ldots, 9\}$ und $m_2, m_3, e_0 \in \{0, \ldots, 9\}$, z. B.

$$0.999 \times 10^{4} \qquad \text{oder} \qquad 0.124 \times 10^{-1}\,,$$

aber *nicht*

$$0.140 \times 10^{-10} \qquad \text{oder} \qquad 0.7934 \times 10^{7}\,.$$

Beispiel 2.12. Als Basis wird auf fast allen modernen Rechnern $b = 2$ gewählt (ab und zu $b = 16$). Beim sog. **IEEE double precision** Format (Standard in `MATLAB`, Datentyp `double` in C++) werden 8 Bytes = 64 Bits zum Speichern einer Fließkommazahl verwendet. Statt des Exponenten $\pm e \in \mathbb{Z}$ wird die sog. **Charakteristik** $c \in \mathbb{N}_0$ verwendet:

$$x = \pm m \times 2^{c+e_{\min}+1} \tag{2.10}$$

mit $c = c_0 2^0 + c_1 2^1 + \ldots + c_s 2^s$. Dadurch entfällt die Speicherung des Vorzeichens im Exponenten.

Durch den extra Faktor 2 in (2.10) ist diese Darstellung nicht ganz äquivalent zu $\mathbb{F}(2, r, s)$, sondern leicht verschoben in den Bereich der positiven Exponenten. Da $b = 2$ ist $m_1 = 1$ und muss deshalb nicht gespeichert werden. Die zur Verfügung stehenden 64 Bit werden dann folgendermaßen verwendet:

$$\begin{aligned} &1 \text{ Bit für das Vorzeichen;} \\ &52 \text{ Bit für } m_2, \ldots, m_r \in \{0, 1\}; \\ &11 \text{ Bit für die Charakteristik.} \end{aligned}$$

Damit ergibt sich $r = 53$ und $s = 10$. Es folgt

$$x = \pm m \times 2^{c-1022}\,. \tag{2.11}$$

Dabei werden die Werte $c = 0$ und $c = 2047$ ausgenommen für spezielle Werte, sodass $c \in \{1, \ldots, 2046\}$ und

$$\begin{aligned} x^+_{\max} &= (2^{-1} + \ldots + 2^{-53}) \times 2^{1024} \approx 2^{1024} \approx 1.8 \times 10^{308} \\ x^-_{\min} &= 2^{-1} \times 2^{-1021} = 2^{-1022} \approx 2.2 \times 10^{-308} \end{aligned}$$

und ähnliche Minima/ Maxima im negativen Bereich. Wir erhalten außerdem für

$$\begin{aligned} c = 0,\ m = 1 \quad &\equiv \text{Null} \\ c = 2047,\ m = \pm 1 \quad &\equiv \pm\infty \\ c = 2047,\ m \neq 1 \quad &\equiv \text{NaN} \end{aligned}$$

NaN steht hierbei für **N**ot **a** **N**umber und wird bspw. verwendet, um das Ergebnis bei Division durch Null zu speichern.

2.4 Rundungsfehleranalyse

Ausgangsdaten $x \in \mathbb{R}$ einer numerischen Aufgabe und Zwischenergebnisse müssen durch Maschinenzahlen dargestellt werden. Dazu definieren wir eine **Rundungsoperation** $\mathrm{rd} : \mathbb{R} \to \mathbb{F}$ mit der natürlichen Forderung

$$|x - \mathrm{rd}(x)| = \min_{y \in \mathbb{F}} |x - y| \qquad \forall x \in D \tag{2.12}$$

mit

$$D \coloneqq [x^-_{\min}, x^-_{\max}] \cup \{0\} \cup [x^+_{\min}, x^+_{\max}]$$

(üblicherweise setzt man $\mathrm{rd}(x) = 0$ für $x \in ([x^-_{\max}, x^+_{\min}])$, $\mathrm{rd}(x) = +\infty$ für $x > x^+_{\max}$ und $\mathrm{rd}(x) = -\infty$ für $x < x^-_{\min}$).

Bei der **natürlichen Rundung**, z. B. im IEEE Format ist

$$\mathrm{rd}(x) = \mathrm{sgn}(x) \times \begin{cases} (m_1 b^{-1} + \ldots + m_r b^{-r}) \times b^e & \text{, für } m_{r+1} < b/2 \\ (m_1 b^{-1} + \ldots + (m_r + 1) b^{-r}) \times b^e & \text{, für } m_{r+1} \geq b/2\,. \end{cases} \tag{2.13}$$

Der **absolute Rundungsfehler** in (2.13) erfüllt

$$|x - \mathrm{rd}(x)| \leq \frac{1}{2} b^{-r} b^e\,, \tag{2.14}$$

hängt also vom Exponenten e ab. Dagegen ist der sog. **relative Rundungsfehler** nur von b und r abhängig und ist wegen

$$\left| \frac{x - \mathrm{rd}(x)}{x} \right| \leq \frac{\frac{1}{2} b^{-r} b^e}{|m| b^e} \leq \frac{1}{2} b^{-r+1} =: \texttt{eps} \tag{2.15}$$

für alle $x \in D$, $x \neq 0$, durch die sog. **Maschinengenauigkeit** `eps` beschränkt.

Für jedes $x \in D$ gilt offenbar

$$\mathrm{rd}(x) = x(1 + \epsilon) \quad \text{mit } |\epsilon| \leq \texttt{eps}\,. \tag{2.16}$$

Beim IEEE double Format ist der maximale relative Rundungsfehler kleiner gleich

$$\texttt{eps}_{\text{double}} = \frac{1}{2} 2^{-52} \approx 10^{-16}\,.$$

Die arithmetischen Grundoperationen $+,\ -,\ \times,\ /$ werden durch **Maschinenoperationen** $\oplus,\ \ominus,\ \otimes,\ \oslash$ ersetzt, wobei das Resultat einfach gerundet wird, also bspw. für $+$

$$x \oplus y = rd(x + y) = (x + y)(1 + \epsilon) \qquad \text{mit } |\epsilon| \leq \texttt{eps}\,. \tag{2.17}$$

Achtung! Die Maschinenoperationen genügen dem Assoziativ- und Distributivgesetz nur näherungsweise. Die Reihenfolge der Operationen ist also wichtig; wir erhalten für je nach Reihenfolge also unter Umständen Stabilitätsprobleme.

Beispiel 2.13 (Vorwärtsrundungsfehleranalyse)**.** In diesem Beispiel analysieren wir die folgende Aufgabe

$$y = x_1^2 - x_2^2 \quad \text{bzw.} \quad y = (x_1 + x_2)(x_1 - x_2)\,.$$

Algorithmus A: Der erste Ansatz führt auf

$$\begin{aligned}\tilde{y}_A &= (x_1 \otimes x_1) \ominus (x_2 \otimes x_2)\\ &= \Big(x_1^2(1+\epsilon_1) - x_2^2(1+\epsilon_2)\Big)\,(1+\epsilon_3)\\ &= \underbrace{x_1^2 - x_2^2}_{=y} + x_1^2\epsilon_1 - x_2^2\epsilon_2 + \underbrace{(x_1^2 - x_2^2)}_{=y}\epsilon_3 + \mathcal{O}(\texttt{eps}^2)\,.\end{aligned}$$

Wir erhalten für den relativen Rundungsfehler

$$\begin{aligned}\left|\frac{y-\tilde{y}_A}{y}\right| &\le \frac{x_1^2 + x_2^2 + |x_1^2 - x_2^2|}{|x_1^2 - x_2^2|}\texttt{eps} + \mathcal{O}(\texttt{eps}^2)\\ &= \left(1 + \frac{x_1^2 + x_2^2}{|x_1^2 - x_2^2|}\right)\texttt{eps} + \mathcal{O}(\texttt{eps}^2)\,.\end{aligned}$$

Für $x_1^2 \approx x_2^2$ kommt es daher zu starker **Rundungsfehlerverstärkung**.

Algorithmus B: Für den zweiten Ansatz ergibt sich

$$\begin{aligned}\tilde{y}_B &= (x_1 \oplus x_2) \otimes (x_1 \ominus x_2)\\ &= ((x_1 + x_2)(1+\epsilon_1))\,((x_1 - x_2)(1+\epsilon_2))\,(1+\epsilon_3)\\ &= y(1+\epsilon_1)(1+\epsilon_2)(1+\epsilon_3)\,,\end{aligned}$$

woraus folgt

$$\begin{aligned}\left|\frac{y-\tilde{y}_B}{y}\right| &\le |\epsilon_1 + \epsilon_2 + \epsilon_3| + \mathcal{O}(\texttt{eps}^2)\\ &\le 3\texttt{eps} + \mathcal{O}(\texttt{eps}^2)\,.\end{aligned}$$

Algorithmus B ist offenbar im Allgemeinen stabiler als Algorithmus A.

Rundungsfehlerfortpflanzung kann aber noch viel katastrophaler sein, wie folgendes Beispiel zeigt.

Beispiel 2.14. Für Integrale der Form

$$I_k = \int_0^1 x^k \exp(x) \mathrm{d}x$$

kann man mit Hilfe von partieller Integration leicht die folgende Rekursionsformel herleiten:

$$I_0 = e - 1 \quad \text{und} \quad I_k = e - k \cdot I_{k-1}\,, \quad \text{für } k \geq 1.$$

Scheint ein sehr einfacher Algorithmus zu sein, aber wie funktioniert er in der Praxis?

Die ersten 26 Werte für I_k, mit IEEE-double precision Genauigkeit berechnet (also mit eps $\approx 10^{-16}$), ergeben:

k	berechnetes I_k	Fehler $\lvert\Delta I_k\rvert$
0	1.718281828459050	$2.6 \cdot 10^{-15}$
1	1	(Null)
2	0.718281828459045	$1.5 \cdot 10^{-15}$
3	0.563436343081910	$5.5 \cdot 10^{-16}$
4	0.464536456131406	$1.0 \cdot 10^{-15}$
5	0.395599547802016	$6.0 \cdot 10^{-15}$
6	0.344684541646949	$3.8 \cdot 10^{-14}$
7	0.305490036930402	$2.7 \cdot 10^{-13}$
8	0.274361533015832	$2.1 \cdot 10^{-12}$
9	0.249028031316559	$1.9 \cdot 10^{-11}$
10	0.228001515293454	$1.9 \cdot 10^{-10}$
11	0.210265160231056	$2.1 \cdot 10^{-9}$
12	0.195099905686377	$2.5 \cdot 10^{-8}$
13	0.181983054536145	$3.3 \cdot 10^{-7}$
14	0.170519064953013	$4.6 \cdot 10^{-6}$
15	0.160495854163853	$7.0 \cdot 10^{-5}$
16	0.150348161837404	$1.1 \cdot 10^{-3}$
17	0.162363077223183	$1.9 \cdot 10^{-2}$
18	-0.204253561558257	$3.4 \cdot 10^{-1}$
19	6.59909949806592	$6.7 \cdot 10^{0}$
20	-129.263708132859	$1.3 \cdot 10^{1}$
21	2717.25615261851	$2.7 \cdot 10^{3}$
22	-59776.9170757787	$6.0 \cdot 10^{4}$
23	1374871.81102474	$1.4 \cdot 10^{6}$
24	-32996920.7463119	$3.3 \cdot 10^{7}$
25	824923021.376079	$8.2 \cdot 10^{8}$

Die Rekursionsformel $I_k = e - k \cdot I_{k-1}$ führt zu einer Fehlerverstärkung um den Faktor k im k-ten Schritt **!!**

3 Interpolation und Approximation

In diesem Kapitel versuchen wir die folgenden Fragen zu beantworten:

(i) Eine Funktion $f(x)$ sei nur in einer diskreten Menge von Argumenten $x_0, \ldots, x_n$ bekannt. Wie kann $f(x)$ mittels dieser Information rekonstruiert werden (z. B. zur graphischen Darstellung oder zur Auswertung an Zwischenstellen)?

(ii) Wie kann eine analytisch gegeben Funktion $f(x)$ am Computer dargestellt werden (und berechenbar werden für beliebiges x, z. B. trigonometrische Funktionen)?

In beiden Fällen stellt sich das Problem folgendermaßen dar: Wir wollen ein System mit unendlich vielen Freiheitsgraden, $y = f(x)$, durch einen endlichen Datensatz simulieren.

Man bedient sich dazu gewisser Klassen $\mathcal{P}$ von „einfachen“ Funktionen, z. B.:

$$\begin{aligned}
&\text{Polynome:} && p(x) = a_0 + a_1 x + \ldots + a_n x^n \\
&\text{rationale Polynome:} && r(x) = \frac{a_0 + a_1 x + \ldots + a_n x^n}{b_0 + b_1 x + \ldots + b_n x^n} \\
&\text{trigonometrische Polynome:} && t(x) = \tfrac{1}{2} a_0 + \sum_{k=1}^{n} \left(a_k \cos(kx) + b_k \sin(kx)\right) \\
&\text{Exponentialsummen:} && e(x) = \sum_{k=1}^{n} a_k \exp(b_k x)
\end{aligned}$$

Definition 3.1. (a) Man spricht von **Interpolation** einer Funktion f durch ein Element $g \in \mathcal{P}$, wenn für alle $i = 0, \ldots, n$ gilt:

$$g(x_i) = y_i := f(x_i)\,.$$

(b) Man spricht von **Best-Approximation** einer Funktion f durch ein Element $g \in \mathcal{P}$ bezüglich einer Norm $\|\cdot\|$, wenn

$$\|f - g\| = \min_{h \in \mathcal{P}} \|f - h\|.$$

Typische Normen, die zur Best-Approximation verwendet werden, sind

$$\|f\|_\infty := \max_{x \in [a,b]} |f(x)| \quad \text{oder} \quad \|f\|_2 := \left(\int_a^b |f(x)|^2 \, \mathrm{d}x \right)^{1/2}.$$

Beispiel.

$f(t) =$ Live Musik

$f(t_i) =$ elektronisch gesamplete Live Musik auf einer CD zu diskreten Zeiten t_i

$g(t) =$ Lautsprecheroutput, der die Daten zu allen Zeiten t interpoliert

3.1 Lineare Interpolation

Hier $n = 1$. Gegeben seien $x_0 \neq x_1$ und $f(x_0)$ und $f(x_1)$. Die lineare Interpolierende ist

$$p_1(x) = f(x_0) + \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) (x - x_0). \tag{3.1}$$

Beispiel 3.2. Sei $f(x) = \sqrt{x}, x_0 = \frac{1}{4}, x_1 = 1$. Damit erhalten wir

$$p_1(x) = \sqrt{\frac{1}{4}} + \left(\frac{1 - \sqrt{\frac{1}{4}}}{1 - \frac{1}{4}} \right) \left(x - \frac{1}{4} \right) = \frac{2}{3}x + \frac{1}{3}$$

Der maximale Fehler ist ca. 0.0417 (vgl. Abb. 3.1). Für $x_0 = 0$ und $x_1 = \frac{3}{4}$ ist der maximale Fehler ca. 0.217. Wie können wir diesen Unterschied erklären? Zum Beweis benötigen wir die folgenden wichtigen Sätze aus der Analysis:

Satz 3.3 (Mittelwertsatz)**.** *Sei f stetig auf $[a, b]$ und differenzierbar auf (a, b). Dann existiert ein $\xi \in (a, b)$, sodass*

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

Beweis. Anwendung von Satz 1.3 (Satz von Taylor) mit $m = 0$. □

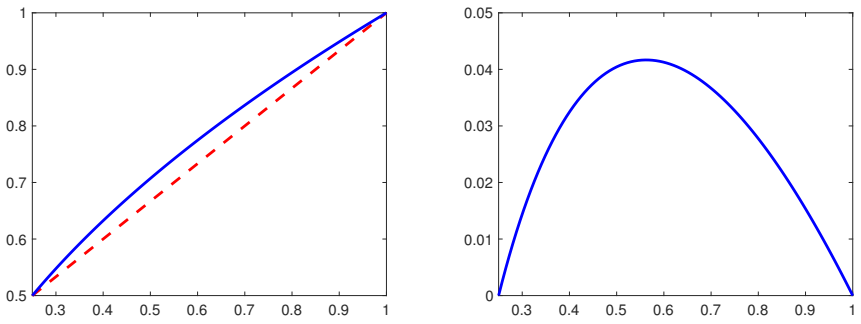


Abbildung 3.1: Der Graph von $f(x) = \sqrt{x}$ (blau) und der Interpolierenden p_1 (rot) für die Stützstellen $x_0 = \frac{1}{4}, x_1 = 1$ (links), sowie der Graph des Interpolationsfehler $f - p_1$ (rechts).

Korollar 3.4 (Satz von Rolle)**.** *Sei f stetig auf $[a, b]$ und differenzierbar auf (a, b) mit $f(a) = f(b)$. Dann existiert ein $\xi \in (a, b)$, sodass*

$$f'(\xi) = 0\,.$$

Um den linearen Interpolationsfehler zu analysieren, setzen wir

$$e(x) := f(x) - p_1(x)\,.$$

Offenbar gilt $e(x_0) = 0 = e(x_1)$.

Satz 3.5. *Sei $f : [x_0, x_1] \to \mathbb{R}$ stetig auf dem Intervall $[x_0, x_1]$ und zwei mal stetig differenzierbar auf (x_0, x_1). Dann existiert für alle $x \in (x_0, x_1)$ ein $\xi \in (x_0, x_1)$, sodass*

$$e(x) = \frac{f''(\xi)}{2} \underbrace{(x - x_0)(x - x_1)}_{=:\, w_2(x)}\,.$$

Beweis. Setze $w_2(x) = (x-x_0)(x-x_1)$. Für festes $x \in (x_0, x_1)$ gilt dann $w_2(x) \neq 0$. Mit

$$g(t) := e(t) - \frac{e(x)}{w_2(x)} w_2(t), \quad t \in [x_0, x_1], \tag{3.2}$$

erhält man

$$g(x_0) = e(x_0) - \frac{e(x)}{w_2(x)} w_x(x_0) = 0\,.$$

Analog ist $g(x_1) = 0$ und $g(x) = 0$. Da f (und deshalb auch g) die Annahmen von Korollar 3.4 auf $[x_0, x]$ und auf $[x, x_1]$ erfüllt, existieren $\eta_1 \in (x_0, x), \eta_2 \in (x, x_1)$, sodass

$$g'(\eta_1) = 0 = g'(\eta_2)\,.$$

Auch g' erfüllt die Voraussetzungen von Korollar 3.4 auf $[\eta_1, \eta_2]$, sodass ein $\xi \in (\eta_1, \eta_2)$ existiert mit

$$g''(\xi) = 0\,. \tag{3.3}$$

Da p_1 linear ist, folgt $p_1''(t) = 0$ und wir erhalten mit (3.2)

$$g''(t) \;=\; \frac{\mathrm{d}^2}{\mathrm{d}t^2}\left(f(t) - p_1(t) - \frac{e(x)}{w_2(x)}(t-x_0)(t-x_1)\right) \;=\; f''(t) - 0 - 2\frac{e(x)}{w_2(x)}$$

woraus wir mit (3.3)

$$0 = f''(\xi) - 2\frac{e(x)}{w_2(x)}$$

erhalten. Durch Umstellen folgt die Behauptung. □

Bemerkung. Der Wert von ξ in Satz 3.5 hängt von x ab, d.h. $\xi = \xi(x)$, und ist normalerweise nicht bekannt. Es ist wichtig, dass f zwei stetige Ableitungen hat!

Korollar 3.6. *Unter den Voraussetzungen von Satz 3.5 gilt*

$$\|e\|_{\infty,[x_0,x_1]} \leq \frac{(x_1-x_0)^2}{8}\|f''\|_{\infty,[x_0,x_1]}\,.$$

Beweis. Aus Satz 3.5 folgt, dass für alle $x \in (x_0, x_1)$ ein $\xi \in (x_0, x_1)$ existiert, sodass

$$|e(x)| = \frac{|f''(\xi)|}{2}|w_2(x)|\,. \tag{3.4}$$

Es gilt $|w_2(x)| = (x-x_0)(x_1-x)$ und eine einfache Kurvendiskussion ergibt

MMS

$$\max_{x\in[x_0,x_1]} |w_2(x)| = \frac{(x_1-x_0)^2}{4}\,. \tag{3.5}$$

Zusammen mit (3.4) folgt

$$|e(x)| \leq \frac{(x_1-x_0)^2}{8}\|f''\|_{\infty,[x_0,x_1]}.$$

Die Ungleichung gilt trivialerweise auch für $x = x_0$ und $x = x_1$. □

3.2 Polynominterpolation

Wir bezeichnen im Folgenden mit

$$\mathcal{P}_n := \{p(x) = a_0 + a_1x + \ldots + a_nx^n \mid a_i \in \mathbb{R},\ i = 0,\ldots,n\}$$

den Vektorraum der Polynome vom Grad kleiner oder gleich n.

Definition 3.7. Die **Lagrange-Interpolationsaufgabe** besteht darin, zu $n+1$ paarweise verschiedene Stützstellen („Knoten“) $x_0, \dots, x_n \in \mathbb{R}$ und Knotenwerten $y_0, \dots, y_n \in \mathbb{R}$ ein Polynom $p \in \mathcal{P}_n$ zu bestimmen, sodass

$$p(x_i) = y_i \quad \text{für } i = 0, \dots, n\,. \tag{3.6}$$

Satz 3.8. *Die Lagrange-Interpolationsaufgabe ist eindeutig lösbar.*

Beweis. Die Bedingung (3.6) entspricht $n+1$ Gleichungen in den $n+1$ Unbekannten $a_0, \dots, a_n$, ist also insb. ein endlichdimensionales Gleichungssystem. Daher sind Lösbarkeit und Eindeutigkeit äquivalent und es genügt die Eindeutigkeit zu zeigen.

Seien $p_1, p_2 \in \mathcal{P}_n$ zwei Lösungen. Wir definieren nun

$$q := p_1 - p_2 \in \mathcal{P}_n \quad \text{sodass} \quad q(x_i) = 0 \text{ für alle } i = 0, \dots, n.$$

Wir benötigen nun zunächst einen weiteren Satz:

Satz (Restpolynomsatz)**.** *Sei $p \in \mathcal{P}_n$ und c eine Nullstelle von p mit $p(c) = 0$. Dann existiert $g \in \mathcal{P}_{n-1}$, sodass*

$$p(x) = (x - c)g(x).$$

Beweis. Siehe bspw. [Fis14, S. 64ff]. ♢

Durch n-maliges Anwenden dieses Satzes erhalten wir

$$q = (x - x_0)(x - x_1) \dots (x - x_{n-1})g \quad \text{mit } g \in \mathcal{P}_0,$$

aber

$$q(x_n) = 0 \;\text{ und }\; x_n \neq x_i, \;\text{ für }\; i = 0, \dots, n-1.$$

Also ist $g \equiv 0$ und somit auch $q \equiv 0$, woraus folgt, dass $p_1 \equiv p_2$. □

Bemerkung. Satz 3.8 kann auch durch n-maliges Anwenden von Korollar 3.4 (Satz von Rolle) bewiesen werden. MMS

Zur Konstruktion verwendet man die **Lagrange-Basispolynome**

$$L_i^{(n)}(x) := \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \in \mathcal{P}_n, \quad i = 0, \dots, n \tag{3.7}$$

(z. B. $L_2^{(3)}(x) = \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}$).

Bemerkung. Die Familie von Polynomen $\left(L_i^{(n)}\right)_{i=0}^{n}$ ist eine Basis von $\mathcal{P}_n$.

MMS

Offenbar gilt

$$L_i^{(n)}(x_k) = \left\{ \begin{array}{ll} 1, & \text{falls } i = k \\ 0, & \text{falls } i \neq k \end{array} \right\} =: \delta_{i,k} \quad (\text{„Kronecker Symbol“}) \tag{3.8}$$

Definition 3.9. Es folgt klarerweise aus (3.8), dass das Polynom

$$p := \sum_{i=0}^{n} y_i L_i^{(n)} \in \mathcal{P}_n$$

die Bedingung (3.6) erfüllt. Es wird **Lagrange-Darstellung** des Interpolationspolynoms für die Stützpunkte $(x_0, y_0), \ldots, (x_n, y_n)$ genannt.

Die Lagrange-Darstellung hat den Nachteil, dass die Basis $(L_i^{(n)})$ von n abhängt und sich ändert, wenn ein neuer Stützpunkt (x_{n+1}, y_{n+1}) dazukommt.

Besser eignet sich die **Newton-Darstellung** mittels der **Newton-Basispolynome**:

$$N_0(x) := 1 \quad \text{und} \quad N_i(x) := (x - x_0) \ldots (x - x_{i-1}), \quad i = 1, \ldots, n \tag{3.9}$$

Das Interpolationspolynom lässt sich aus diesen durch sukzessives Auswerten des Ansatzes

$$p(x) = \sum_{i=0}^{n} a_i N_i(x) \tag{3.10}$$

an den Stützstellen $x_0, \ldots, x_n$ systematisch aufbauen; es führt auf das gestaffelte (Dreiecks-)Gleichungssystem

$$\left.\begin{array}{l} y_0 = p(x_0) = a_0\,, \\ y_1 = p(x_1) = a_0 + a_1(x_1 - x_0)\,, \\ \quad \vdots \\ y_n = p(x_n) = a_0 + a_1(x_1 - x_0) + \ldots + a_n(x_n - x_0) \ldots (x_n - x_{n-1})\,, \end{array}\right\} \tag{3.11}$$

woraus sich die Koeffizienten a_i rekursiv berechnen lassen.

Eine weitere Stützstelle (x_{n+1}, y_{n+1}) kann nun leicht hinzugefügt werden. Eine numerisch stabilere Berechnung liefert folgender Satz.

Satz 3.10. *Die Newton-Darstellung des Interpolationspolynoms für* $(x_0, y_0), \ldots, (x_n, y_n)$ *lässt sich schreiben als*

$$p(x) = \sum_{i=0}^{n} y[x_0, \ldots, x_i] N_i(x), \tag{3.12}$$

wobei $y[x_0, \ldots, x_i]$ *die zu den Punkten* $(x_0, y_0), \ldots, (x_i, y_i)$ *gehörenden* ***dividierten Differenzen*** *bezeichnen, welche rekursiv definiert sind durch*

$$\begin{aligned} y[x_i] &= y_i, && i = 0, \ldots, n \\ y[x_i, \ldots, x_j] &= \frac{y[x_{i+1}, \ldots, x_j] - y[x_i, \ldots, x_{j-1}]}{x_j - x_i}, && 0 \le i < j \le n. \end{aligned}$$

Beweis. Gegeben $0 \le i < j \le n$ und $k := j - i \le n$, sei $p_{i,j} \in \mathcal{P}_k$ das Polynom, das die Punkte $(x_i, y_i), \ldots, (x_j, y_j)$ interpoliert. Dann ist $p = p_{0,n}$ das gesuchte Polynom. Wir beweisen per Induktion bezüglich k, dass

$$p_{i,j}(x) = y[x_i] + y[x_i, x_{i+1}](x - x_i) + \ldots + y[x_i, \ldots, x_j](x - x_i) \ldots (x - x_{j-1}) \tag{3.13}$$

(woraus (3.12) mit $i = 0$ und $j = n$ folgt).

Induktionsanfang: Da $p_{i,i}(x) \equiv y_i = y[x_i]$ gilt die Aussage für $k = 0$.

Induktionsannahme: Gleichung (3.13) gelte für alle $i < j$ mit $j - i = k - 1$.

Induktionsschritt: Dann folgt für $i < j$ mit $j - i = k$, dass

$$p_{i,j}(x) = \underbrace{p_{i,j-1}(x)}_{\in \mathcal{P}_{k-1}} + a'_k (x - x_i) \ldots (x - x_{j-1}) \,. \tag{3.14}$$

Zu zeigen ist nun, dass $a'_k = y[x_i, \ldots, x_j]$. Weil $p_{i,j-1} \in \mathcal{P}_{k-1}$, ist der führende Koeffizient von $p_{i,j}$ (d. h. der Koeffizient zu x^k) gleich a'_k. Man kann leicht verifizieren, dass die folgende Formel gilt:

MMS

$$p_{i,j}(x) = \frac{(x - x_i) p_{i+1,j}(x) - (x - x_j) p_{i,j-1}(x)}{x_j - x_i} \,. \tag{3.15}$$

Aus der Induktionsannahme folgt weiterhin, dass die führenden Koeffizienten von $p_{i+1,j}$ und $p_{i,j-1}$ respektive $y[x_{i+1}, \ldots, x_j]$ und $y[x_i, \ldots, x_{j-1}]$ sind. Daraus folgt mit (3.14) und (3.15), dass

$$a'_k = \frac{y[x_{i+1}, \ldots, x_j] - y[x_i, \ldots, x_{j-1}]}{x_j - x_i} = y[x_i, \ldots, x_j] \,,$$

was zu zeigen war. □

Bemerkung. Aus der Eindeutigkeit des Interpolationspolynoms und der Unabhängigkeit von a_n in (3.10) von der Nummerierung der Stützstellen folgt

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

für jede beliebige Permutation $(i_0, \ldots, i_n)$ von $(0, \ldots, n)$.

Die im Beweis von Satz 3.10 verwendete Beziehung (3.13) der Polynome $p_{i,j}$ führt auf das **Neville-Schema**

(x_0, y_0)	$p_{0,0}(x)$	$\searrow$								
			$p_{0,1}(x)$	$\searrow$						
(x_1, y_1)	$p_{1,1}(x)$	$\nearrow$ $\searrow$			$p_{0,2}(x)$					
			$p_{1,2}(x)$	$\nearrow$		$\ddots$				
(x_2, y_2)	$p_{2,2}(x)$	$\nearrow$					$p_{0,n-1}(x)$	$\searrow$		
$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$		$p_{0,n}(x)$	
(x_{n-2}, y_{n-2})	$p_{n-2,n-2}(x)$	$\searrow$					$p_{1,n}(x)$	$\nearrow$		
			$p_{n-2,n-1}(x)$	$\searrow$		$\iddots$				
(x_{n-1}, y_{n-1})	$p_{n-1,n-1}(x)$	$\nearrow$ $\searrow$			$p_{n-2,n}(x)$					
			$p_{n-1,n}(x)$	$\nearrow$						
(x_n, y_n)	$p_{n,n}(x)$	$\nearrow$								

und auf die **Neville-Darstellung**.

Auch hier ist die Hinzunahme einer weiteren Stützstelle einfach.

Zudem bietet das Neville-Schema eine sehr effiziente und numerisch stabile Methode zur Auswertung von $p(\xi)$ für $\xi \neq x_i$, ohne vorheriges Bestimmen der Koeffizienten in (3.10) oder (3.12). Man setzt $x = \xi$ im obigen Schema und verwendet die in Satz 3.10 definierte Rekursion (3.15).

3.2.1 Auswertung von Polynomen

Das Auswerten von Polynomen $p \in \mathcal{P}_n$, die in der Form

$$p(x) = a_0 + a_1 x + \ldots + a_n x^n$$

gegeben sind, an einem Punkt $\xi \neq x_i$, lässt sich effizient und numerisch stabil über das **Horner-Schema** realisieren:

MMS

$$\begin{aligned} b_n &= a_n, \\ b_k &= a_k + b_{k+1}\,\xi\,, \qquad \text{für } k = n-1, \ldots, 0, \end{aligned}$$

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

Für Polynome gegeben in Newton-Darstellung

$$p(x) = a_0 + a_1(x - x_0) + \ldots + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

verwendet man das **verallgemeinerte Horner-Schema**:

$$\begin{aligned} b_n &= a_n, \\ b_k &= a_k + b_{k+1}\,(\xi - x_k)\,, \qquad \text{für } k = n-1, \ldots, 0, \\ p(\xi) &= b_0. \end{aligned}$$

Beispiel 3.11. Gegeben die Knotenpaare $(x_0, y_0) = (0, 1), (x_1, y_1) = (1, 4), (x_2, y_2) = (2, 3)$, finde die Interpolierende $p \in \mathcal{P}_2$.

(a) Direkte Berechnung von p in der **Monombasis** $\{1, x, x^2\}$ führt auf das Gleichungssystem

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \\ 3 \end{pmatrix}.$$

Gauß-Elimination führt auf:

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 4 \\ 1 & 2 & 4 & 3 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 3 \\ 0 & 2 & 4 & 2 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 3 \\ 0 & 0 & 2 & -4 \end{array}\right)$$

mit Lösung $(a_0, a_1, a_2)^\intercal = (1, 5, -2)^\intercal$, und deshalb

$$p(x) = 1 + 5x - 2x^2\,.$$

(b) In der Lagrange-Darstellung erhalten wir:

$$\begin{aligned} p(x) &= 1 \cdot \underbrace{\frac{(x-1)(x-2)}{(0-1)(0-2)}}_{L_0^{(2)}(x)} + 4 \cdot \underbrace{\frac{(x-0)(x-2)}{(1-0)(1-2)}}_{L_1^{(2)}(x)} + 3 \cdot \underbrace{\frac{(x-0)(x-1)}{(2-0)(2-1)}}_{L_2^{(2)}(x)} \\ &= \frac{1}{2}(x-1)(x-2) - 4x(x-2) + \frac{3}{2}x(x-1) \quad \text{(identisch!)} \end{aligned}$$

(c) Für die Newton-Basis ergibt sich das Tableu

$$
\begin{array}{lll}
a_0 = y[x_0] = 1 & \searrow & \\
 & a_1 = y[x_0, x_1] = \frac{4-1}{1-0} = 3 & \searrow \\
y[x_1] = 4 & \nearrow\!\searrow & a_2 = y[x_0, x_1, x_2] = \frac{-1-3}{2-0} = -2 \\
 & y[x_1, x_2] = \frac{3-4}{2-1} = -1 & \nearrow \\
y[x_2] = 3 & \nearrow &
\end{array}
$$

und daher

$$p(x) = \underbrace{1}_{N_0(x)} + 3 \cdot \underbrace{x}_{N_1(x)} - 2 \cdot \underbrace{x(x-1)}_{N_2(x)} \quad \text{(identisch!)}$$

3.2.2 Interpolation von Funktionen und Interpolationsfehler

Wir betrachten nun den Fall, dass die Knotenwerte y_i durch eine Funktion f auf einem Intervall $[a, b]$ gegeben sind, welches die Stützpunkte x_i enthält:

$$y_i = f(x_i), \quad x_i \in [a, b], \quad i = 0, \ldots, n \tag{3.16}$$

Frage: Wie gut approximiert das zugehörige Interpolationspolynom $p \in \mathcal{P}_n$ die Funktion f auf $[a, b]$?

Um diese Frage zu beantworten, führen wir zunächst die folgende Notation ein:

- $I(x_0, \ldots, x_n)$ ist das kleinste Intervall, das alle in Klammern eingeschlossenen Punkte enthält.
- $\mathcal{C}[a, b]$ ist der Vektorraum der über $[a, b]$ stetigen Funktionen.
- $\mathcal{C}^k[a, b]$ ist der Vektorraum der über $[a, b]$ $k-$mal stetig differenzierbaren Funktionen.

Satz 3.12 (Interpolationsfehler)**.** *Sei $f \in \mathcal{C}^{n+1}[a, b]$. Für alle $x \in [a, b]$ existiert ein $\xi_x \in I(x_0, \ldots, x_n, x)$, sodass*

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{j=0}^{n} (x - x_j) \tag{3.17}$$

(Vgl. Satz 3.5 für die lineare Interpolation).

Beweis. Für $x \in \{x_0, \ldots, x_n\}$ folgt das Resultat trivialerweise.

Sei also $x \in [a,b] \setminus \{x_0, \ldots, x_n\}$. Wir setzen

$$w_{n+1}(t) := \prod_{j=0}^{n} (t - x_j) \quad \text{und} \quad e(x) := f(x) - p(x)\,.$$

Wie im Beweis von Satz 3.5 überzeugt man sich leicht, dass die Funktion

$$g(t) = e(t) - \frac{e(x)}{w_{n+1}(x)} w_{n+1}(t)$$

$(n+2)$ Nullstellen an den Punkten $x_0, \ldots, x_n, x$ hat. Durch wiederholtes Anwenden von Korollar 3.4 (Satz von Rolle) folgt, dass ein $\xi_x \in I(x_0, \ldots, x_n, x)$ existiert, sodass

MMS

$$g^{(n+1)}(\xi_x) = 0,$$

das heißt

$$0 = f^{(n+1)}(\xi_x) - \underbrace{p^{(n+1)}(\xi_x)}_{=0} - \frac{e(x)}{w_{n+1}(x)} \underbrace{w_{n+1}^{(n+1)}(\xi_x)}_{=(n+1)!}\,.$$

Umstellen ergibt

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

□

Korollar 3.13. *Unter den Voraussetzungen von Satz 3.12 mit $a = x_0 < x_1 < \ldots < x_n = b$ gilt*

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

Beweis. Folgt direkt aus Satz 3.12 und der Beobachtung, dass

$$|w_{n+1}(x)| = |(x - x_0) \ldots (x - x_n)| \leq |x_n - x_0|^{n+1}$$

□

Die Abschätzung im Beweis von Korollar 3.13 ist **nicht scharf**; z. B. gilt für äquidistante Stützstellen $x_j = jh$ und $h = \frac{1}{n}$ auf $[0,1]$:

$$\|f - p\|_{\infty,[0,1]} \;\leq\; \frac{\left\|f^{(n+1)}\right\|_{\infty,[0,1]}}{2^n (n+1)!}\,.$$

Das heißt, wenn $f^{(n+1)}$ für $n \to \infty$ beschränkt bleibt, reduziert sich der Interpolationsfehler **mindestens** mit $\mathcal{O}\left(\frac{2^{-n}}{(n+1)!}\right)$, was sich für $n = 2, 4, 8, \ldots$ wie in der folgenden Tabelle aufgelistet verhält:

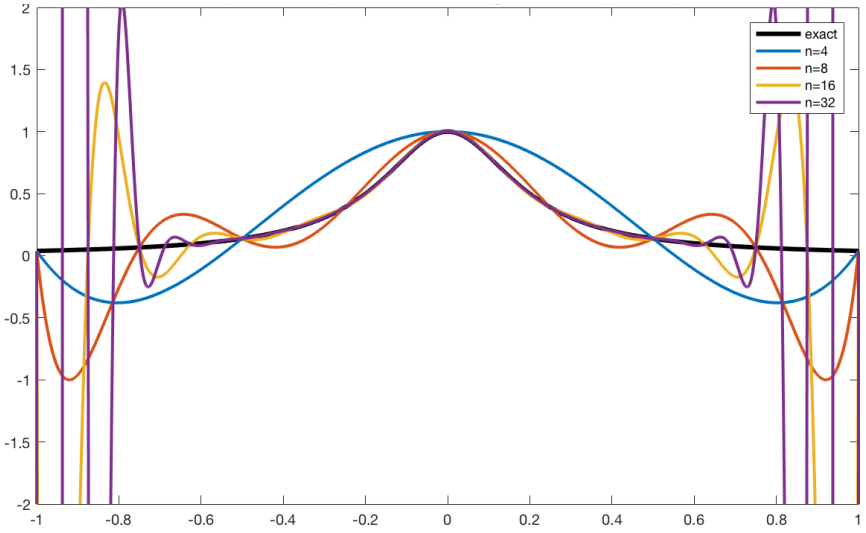


Abbildung 3.2: Graph der Runge-Funktion $f(x) = (1 + 25x^2)^{-1}$ mit äquidistanten Stützstellen für $n = 4, 8, 16, 32$.

n	2	4	8	16
$\frac{2^{-n}}{(n+1)!}$	4.2×10^{-2}	5.2×10^{-4}	1.1×10^{-8}	4.3×10^{-20}

Meistens haben die Ableitungen von f jedoch ein zu starkes Wachstum für $n \to \infty$, z. B. die „Runge“-Funktion

$$f(x) = \frac{1}{1 + 25x^2}$$

für die gilt

$$\left| f^{(n)}(x) \right| \approx 2^n n! \, \mathcal{O}\left(|5x|^{-(n+2)} \right)$$

(vgl. Abb. 3.2). Der relative Fehler

$$\max_{x \in [-1,1]} \left\| \frac{f(x) - p(x)}{f(x)} \right\|_\infty$$

beträgt 351.5 für $n = 16$ und 12802 für $n = 32$.

Bemerkung 3.14. Der **Approximationssatz von Weierstraß** besagt, dass jede Funktion $f \in \mathcal{C}[a.b]$ beliebig gut gleichmäßig auf $[a, b]$ durch Polynome approximiert werden kann. Das heißt aber **nicht**, dass das durch Interpolation erreicht werden kann. Geeignet gewählte nicht-äquidistante Stützstellen helfen — passende Stützstellen zu finden ist aber schwer.

Ein weiterer Defekt der Lagrange-Interpolation ist die große Fehlerempfindlichkeit: Sei $p(y; x)$ das Interpolationspolynom zu $(x_0, y_0), \ldots, (x_n, y_n)$ mit **festen** Stützstellen

$x_0, \ldots, x_n$. Dann gilt

$$\begin{aligned} p_{y+\Delta y}(x) - p_y(x) &= \sum_{i=0}^{n} (y_i + \Delta y_i) L_i^{(n)}(x) - \sum_{i=0}^{n} y_i L_i^{(n)}(x) \\ &= \sum_{i=0}^{n} \Delta y_i L_i^{(n)}(x) \end{aligned}$$

und deshalb

$$\frac{p_{y+\Delta y}(x) - p_y(x)}{p_y(x)} = \sum_{i=0}^{n} \underbrace{\frac{y_i L_i^{(n)}(x)}{p_y(x)}}_{k_i(x)} \frac{\Delta y_i}{y_i}.$$

Für große n kann $\max_{x\in[a,b]} |L_i^{(n)}(x)|$ sehr groß werden, d. h. auch wenn $p_y(x)$ beschränkt ist gilt $\max_{x\in[a,b]} |k_i(x)| \gg 1$ und Fehler in y_i werden stark verstärkt!

Man sagt: Die Lagrange-Interpolationsaufgabe ist **schlecht konditioniert** und nennt $k_i(x)$ die **(relativen) Konditionszahlen** (siehe Appendix 2.1).

3.3 Richardson Extrapolation (zum Limes)

Eine Anwendung der Polynominterpolation ist die Berechnung von

$$a(0) = \lim_{x\to+0} a(x) \qquad \text{(betrachten nur } x > 0)$$

aus den Werten $(x_i, a(x_i)), i = 0, \ldots, n$ (z. B. wenn $a(0)$ nicht direkt berechenbar ist) mit Hilfe des zugehörigen Interpolationspolynoms.

Beispiel 3.15. Wir wollen

$$a(0) = \lim_{x\to+0} \frac{\cos x - 1}{\sin x}$$

berechnen. Dazu interpolieren wir $a(x) = \frac{\cos x - 1}{\sin x}$ an einigen Stützstellen nahe 0:

$$\begin{array}{lll} x_0 = \frac{1}{8}: & a(x_0) = -6.258151 \cdot 10^{-2} = p_{0,0}(0) & \searrow \\ & & p_{0,1}(0) = 6.115 \cdot 10^{-5} \quad \searrow \\ x_1 = \frac{1}{16}: & a(x_1) = -3.126018 \cdot 10^{-2} = p_{1,1}(0) & \nearrow \\ & & p_{1,2}(0) = 7.64 \cdot 10^{-6} \quad \nearrow \\ x_2 = \frac{1}{32}: & a(x_2) = -1.562627 \cdot 10^{-2} = p_{2,2}(0) & \nearrow \end{array} \quad \ldots$$

$$\ldots \Rightarrow a(0) \approx p_2(0) = p_{0,2}(0) = -1.02 \cdot 10^{-5} \quad \text{(“Neville Schema”).}$$

(Die exakte Lösung, z.B. mittels der Regel von l’Hôpital berechnet, ist $a(0) = 0$.)

Falls $a \in \mathcal{C}^{n+1}[0,h]$, für ein $h > 0$, erhält man aus Satz 1.3 (Taylor), dass

$$a(x) = a(0) + \sum_{j=1}^{n} a_j x^j + a^{(n+1)}(\xi_x)\, x^{n+1}, \quad \text{für } x \le h, \tag{3.18}$$

für gewisse Koeffizienten $a_1, \ldots, a_n \in \mathbb{R}$ und für ein $\xi_x \in (0,h)$.

Satz 3.16 (Extrapolationsfehler)**.** *Es seien $n \in \mathbb{N}$, $h > 0$ und $a \in \mathcal{C}^{n+1}[0,h]$ gegeben. Sei außerdem $(x_k)_{k=0,1,2,\ldots}$ eine monoton fallende Folge positiver Zahlen, mit*

$$x_0 \le h \quad \text{und} \quad \frac{x_{k+1}}{x_k} \le \rho < 1 \quad \text{für } k \ge 0. \tag{3.19}$$

Dann gilt für das Interpolationspolynom $p \in \mathcal{P}_n$ durch die Punkte

$$(x_0, a(x_0)), \ldots, (x_n, a(x_n)),$$

dass

$$|a(0) - p(0)| \;\le\; \|a^{(n+1)}\|_{\infty,[0,h]}\, h^{n+1}.$$

Beweis. In Lagrange-Darstellung gilt

$$p(x) = \sum_{i=0}^{n} a(x_i) L_i^{(n)}(x), \quad \text{mit } L_i^{(n)}(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

und weil $a \in \mathcal{C}^{n+1}[0,h]$ folgt mit (3.18), dass

$$\begin{aligned} p(0) &= \sum_{i=0}^{n} \left(a(0) + \sum_{j=1}^{n} a_j x_i^j + a^{(n+1)}(\xi_x) x_i^{n+1} \right) L_i^{(n)}(0) \qquad (3.20)\\ &= a(0) \left(\sum_{i=0}^{n} L_i^{(n)}(0) \right) + \sum_{j=1}^{n} a_j \left(\sum_{i=0}^{n} x_i^j L_i^{(n)}(0) \right) + a^{(n+1)}(\xi_x) \left(\sum_{i=0}^{n} x_i^{n+1} L_i^{(n)}(0) \right). \end{aligned}$$

Die Terme in den Klammern können über die Fehlerformel in Satz 3.12 für die Funktionen $f_r(x) := x^r$, $r = 0, \ldots, n+1$, berechnet werden, d.h., für $x \in [0,h]$ gilt

$$f_r(x) - \sum_{i=0}^{n} x_i^r L_i^{(n)}(x) = \frac{f_r^{(n+1)}(\eta_r)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

für ein $\eta_r \in (0,h)$. Indem man $x = 0$ wählt, erhält man

MMS

$$\sum_{i=0}^{n} x_i^r L_i^{(n)}(0) = \begin{cases} 1, & \text{für } r = 0,\\ 0, & \text{für } r = 1, \ldots, n,\\ (-1)^n \prod_{i=0}^{n} x_i, & \text{für } r = n+1. \end{cases}$$

Substituieren wir in (3.20), so folgt

$$p(0) = a(0) + a^{(n+1)}(\xi_x)\,(-1)^n \prod_{i=0}^{n} x_i$$

und deshalb zusammen mit (3.19)

$$|p(0) - a(0)| = \left|a^{(n+1)}(\xi_x)\right| \prod_{i=0}^{n} x_i \leq \|a^{n+1}\|_{\infty,[0,h]}\, h^{n+1},$$

weil für alle $n \in \mathbb{N}$ gilt

$$\prod_{i=0}^{n} x_i \leq \prod_{i=0}^{n} \left(x_0 \rho^i\right) \leq h^{n+1} \rho^{(\sum_{i=0}^{n} i)} = h^{n+1} \rho^{\frac{n(n+1)}{2}} \leq h^{n+1}\,.$$

□

Wie im Beispiel 3.15, fúhren wir den Extrapolationsprozess mit Hilfe des Neville-Algorithmus durch. Das führt zu folgendem Extrapolationstableau. Der Konvention folgend setzen wir $a_{i,k} \equiv p_{i-k,i}$:

$$\begin{array}{c|cccccccc}
x_0 & a_{0,0} = a(x_0) & \searrow & & & & & & \\
x_1 & a_{1,0} = a(x_1) & \to & a_{1,1} & & & & & \\
 & & \searrow & & \searrow & & & & \\
\vdots & \vdots & & a_{2,1} & & \ddots & & & \\
\vdots & \vdots & & \vdots & & & \ddots & & \\
x_n & a_{n,0} = a(x_n) & \to & a_{n,1} & \to & \dots & \dots & a_{n,n}
\end{array}$$

wobei

$$a_{i,k} \;=\; \frac{x_{i-k}\, a_{i,k-1} - x_i\, a_{i-1,k-1}}{x_{i-k} - x_i}\,, \quad k = 1, \dots, i, \quad i = 1, \dots, n\,. \tag{3.21}$$

Nach Satz 3.16 gilt dann

$$|a(0) - a_{n,n}| \;=\; \mathcal{O}(h^{n+1}).$$

Falls a glatt genug ist, kann der Fehler durch Dazunehmen von weiteren Punkten exponentiell reduziert werden. Alternativ kann auch Reduktion von h der Fehler (algebraisch) reduziert werden. Für sehr kleine h gibt es aber Stabilitätsprobleme (siehe das folgende Beispiel).

3.3.1 Numerische Differentiation

Eine wichtige Anwendung der Richardson Extrapolation ist in der **numerischen Differentiation**:

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

Aus dem Satz von Taylor folgt für $f \in \mathcal{C}^2$, dass ein $\xi_x \in (x, x+h)$ existiert, sodass

$$\frac{f(x+h) - f(x)}{h} - f'(x) = \frac{f''(\xi_x)}{2} h\,,$$

d.h., der Fehler in der Approximation durch den Differenzenquotienten ist $\mathcal{O}(h)$.

Aber Vorsicht! Für h sehr klein, gilt $f(x+h) \approx f(x)$, was zu **Auslöschung** führt!

Für glatte Funktionen $f \in C^{n+1}$, $n > 1$, können wir aber, numerisch mehr stabil, eine viel bessere Näherung über Extrapolation berechnen.

Beispiel 3.17. Sei $f(x) = \sin(x)$. Berechne $f'(0)$ numerisch per Extrapolation. Wir definieren

$$a(h) \;=\; \frac{f(h) - f(0)}{h} \;=\; \frac{\sin(h)}{h}\,.$$

Auswertung von $a(h)$ in $h_0 = 1/8$, $h_1 = 1/16$, $h_2 = 1/32$, d.h.,

$$a(h_0) = 0.9973979\ldots, \quad a(h_1) = 0.9993491\ldots, \quad a(h_2) = 0.9998372\ldots,$$

ergibt

$$\begin{aligned} f'(0) &\approx p_2(0) \\ &= a(h_0)\frac{(0-\frac{1}{16})(0-\frac{1}{32})}{(\frac{1}{8}-\frac{1}{16})(\frac{1}{8}-\frac{1}{32})} + a(h_1)\frac{(0-\frac{1}{8})(0-\frac{1}{32})}{(\frac{1}{16}-\frac{1}{8})(\frac{1}{16}-\frac{1}{32})} + a(h_2)\frac{(0-\frac{1}{8})(0-\frac{1}{16})}{(\frac{1}{32}-\frac{1}{8})(\frac{1}{32}-\frac{1}{16})} \\ &= \frac{1}{3}a(h_0) - 2a(h_1) + \frac{8}{3}a(h_2) = 1.0000004\ldots \end{aligned}$$

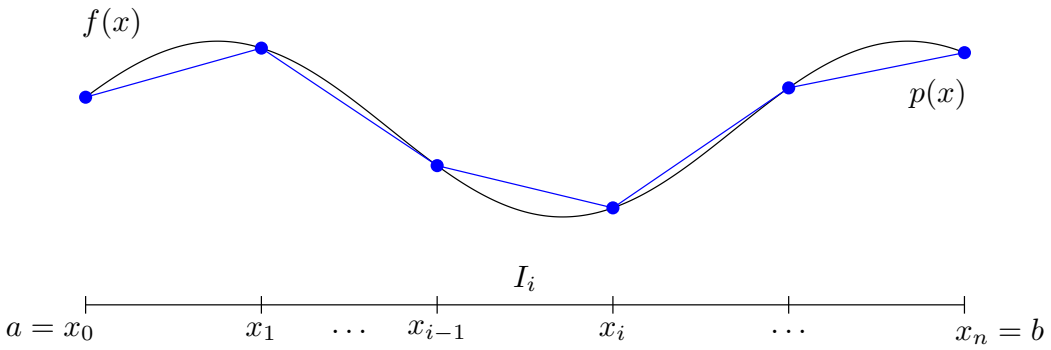
Die exakte Lösung ist $f'(0) = \cos(0) = 1$, sodass der Fehler ungefähr $4 \cdot 10^{-7}$ beträgt.

3.4 Stückweise Polynominterpolation und Splines

Die Lagrange-Interpolation eignet sich nicht besonders gut zur **stabilen** Approximation (nicht glatter) Funktionen, da die Basispolynome zwischen den Stützstellen immer größere Werte annehmen. Besser ist es, **stückweise polynomial** bezüglich einer Zerlegung

$$a = x_0 < x_1 < \ldots < x_n = b$$

zu interpolieren, z. B. stückweise linear:



Die Länge des Teilintervalls $I_i = [x_{i-1}, x_i]$ wird mit $h_i = x_i - x_{i-1}$ bezeichnet. Die Feinheit der Unterteilung wird mit $h = \max_{i=1}^{n} h_i$ beschrieben. Auf dieser Intervallunterteilung betrachten wir die Vektorräume der stückweise polynomialen Funktionen

$$S_n^{(k,r)}[a,b] := \left\{ p \in \mathcal{C}^r[a,b] : \; p|_{I_i} \in \mathcal{P}_k\left(I_i\right), \, i = 1, \ldots, n \right\} \tag{3.22}$$

mit $k, r \in \{0, 1, 2, \ldots\}$, d. h. wir verlangen, dass p an den Intervallendpunkten $r-$mal stetig differenzierbar ist.

3.4.1 Lokale Approximation ($r = 0$)

Wenn wir nur Stetigkeit verlangen, d.h. $p \in \mathcal{C}^0[a,b]$, können die Interpolationspolynome einfach lokal in jedem Teilintervall berechnet werden.

Beispiel 3.18 (Stückweise lineare Interpolation, d.h. $k = 1$ und $r = 0$)**.** Gegeben sei $f \in \mathcal{C}^2[a,b]$ in den Stützstellen $a = x_0 < x_1 < \ldots < x_n = b$. Sei

$$p \in S_n^{(1,0)}[a,b] = \left\{ p \in \mathcal{C}[a,b] : \; p|_{I_i} \text{ linear}, \, i = 1, \ldots, n \right\}$$

der stetige, stückweise lineare Polygonzug mit

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

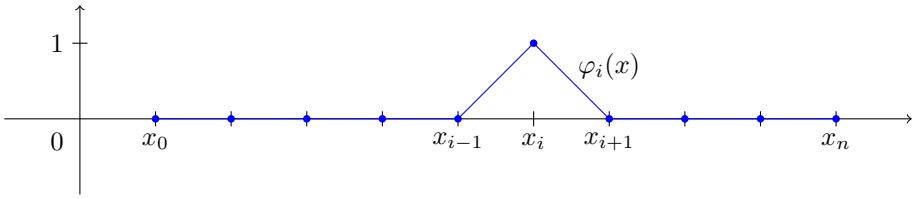
Dann folgt aus Korollar 3.6, angewendet in jedem der Intervalle $I_i, \; i = 1, \ldots, n$, dass

MMS

$$\|f - p\|_{\infty,[a,b]} \; \leq \; \frac{h^2}{8} \, \|f''\|_{\infty,[a,b]} \, .$$

Für die Konstruktion von p kann man die sog. **lokale Knotenbasis** von $S_n^{(1,0)}[a,b]$, bestehend aus den **Hütchenfunktionen** $\varphi_i \in S_n^{(1,0)}[a,b], \; i = 0, \ldots, n$, verwenden, die eindeutig bestimmt sind durch

$$\varphi_i(x_j) = \begin{cases} 1 & \text{für } i = j \\ 0 & \text{für } i \neq j \end{cases}$$



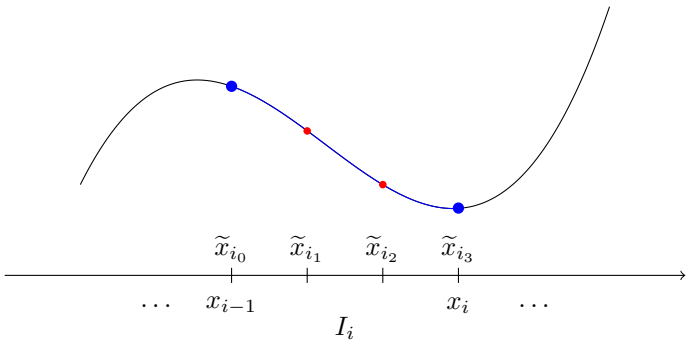
Es folgt

$$p(x) = \sum_{i=0}^{n} f(x_i)\varphi_i(x). \tag{3.23}$$

Beispiel 3.19. Ähnlich funktioniert auch eine stückweise kubische Interpolation $(k = 3, r = 0)$. Hierbei muss die Funktion f in jedem Intervall I_i an 4 Punkten

$$\widetilde{x}_{i_0} = x_{i-1},\ \widetilde{x}_{i_1},\ \widetilde{x}_{i_2},\ \widetilde{x}_{i_3} = x_i \in I_i$$

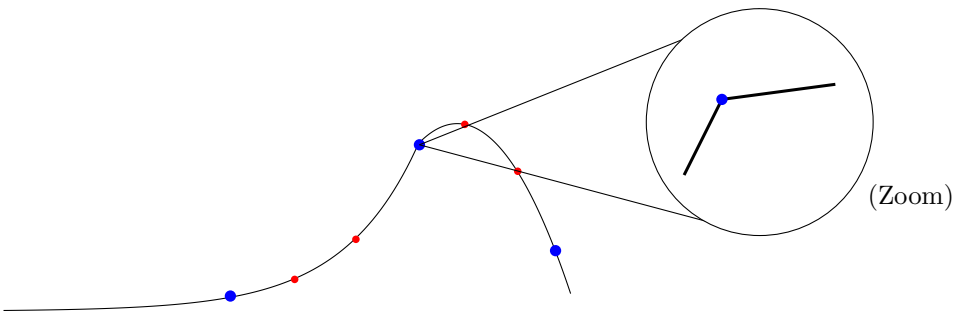
ausgewertet werden:



Über Korollar 3.13 erhält man die verbesserte Fehlerabschätzung für die Interpolierende $p \in S^{(3,0)}[a, b]$:

$$\|f - p\|_{\infty,[a,b]} \ \leq\ \frac{h^4}{4!} \left\|f^{(4)}\right\|_{\infty,[a,b]}.$$

Aber an den Intervallendpunkten ist p jedoch nur stetig:

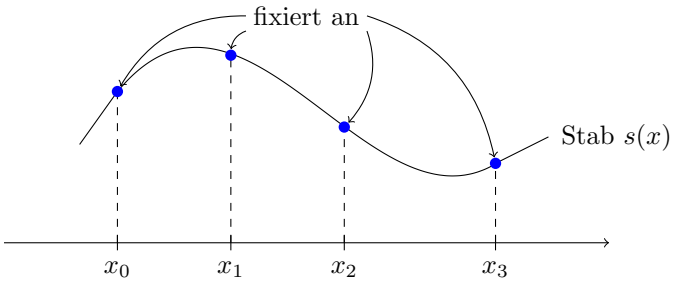


Beide Interpolationsmethoden können verallgemeinert werden auf stückweise glatte $f \in \mathcal{C}[a,b]$ mit $f|_{I_i} \in \mathcal{C}^k(I_i)$, $i = 1, \ldots, n$, mit den selben Fehlerabschätzungen.[1]

3.4.2 Splineinterpolation ($r = 2$)

Diese Art der Interpolation ist bspw. wichtig für die glatte Darstellung von Flächen in der Computergraphik. Der Einfachheit halber betrachten wir nur den Fall $p \in S_n^{(3,2)}$, die am häufigsten verwendeten **kubischen Spline-Funktionen**.

Der Begriff **Spline** stammt aus dem Englischen und bedeutet soviel wie **"biegsames Kurvenlineal"** oder **"Biegestab"**:



Ein Grundgesetz der Mechanik (das Prinzip der minimalen potentiellen Energie) bedeutet, das die Gesamtkrümmung eines Biegestabes der an vier Punkten fixiert ist minimiert wird, d. h.

$$\int_{x_0}^{x_n} |s''(x)|^2 \, \mathrm{d}x \quad \text{ist minimal} \tag{3.24}$$

über alle hinreichend glatten, interpolierenden Funktionen.

In den Endpunkten x_0 und x_n benötigen wir Randbedingungen. Wir betrachten nur den Fall

$$s''(x_0) = s''(x_n) = 0, \tag{3.25}$$

d. h. $s(x)$ ist außerhalb des Intervalls $[x_0, x_n]$ nicht weiter fixiert, also linear.

Definition 3.20. Eine Funktion $s_n : [a,b] \to \mathbb{R}$ heißt **natürlicher kubischer Spline** bzgl. der Zerlegung $a = x_0 < x_1 < \ldots < x_n = b$, wenn

(i) $s_n \in \mathcal{C}^2[a,b]$

(ii) $s_n|_{I_i} \in \mathcal{P}_3(I_i), \quad i = 1, \ldots, n$

(iii) $s_n''(a) = s_n''(b) = 0$

[1] Die Norm in den Fehlerabschätzungen muss dann ersetzt werden durch die (sogenannte) L^∞-norm, d.h. $\|f\|_{\infty,[a,b]} := \operatorname{ess\,sup}_{x\in[a,b]} |f(x)|$.

Wir betrachten nun den interpolierenden kubischen Spline zu den vorgegeben Knotenwerten

$$s_n(x_i) = y_i, \quad i = 0, \dots, n. \tag{3.26}$$

Satz 3.21 (Spline-Interpolation)**.** *Der interpolierende (natürliche) kubische Spline existiert und ist eindeutig.*

Beweis. Es sei $s_n|_{I_i} = p_i \in \mathcal{P}_3(I_i)$. Die jeweils 4 Koeffizienten $a_0^{(i)}, \dots, a_3^{(i)}$ dieser n Polynome ergeben $4n$ freie Parameter.

Die folgenden $4n$ Bedingungen müssen an den Punkten x_i erfüllt sein:

$$\begin{array}{lll}
i = 0: & p_1(x_0) = y_0,\ p_1''(x_0) = 0 & 2 \\
i = 1, \dots, n-1: & p_i(x_i) = y_i = p_{i+1}(x_i) & 2(n-1) \\
i = 1, \dots, n-1: & p_i'(x_i) = p_{i+1}'(x_i),\ p_i''(x_i) = p_{i+1}''(x_i) & 2(n-1) \\
i = n: & p_n(x_n) = y_n,\ p_n''(x_n) = 0 & 2
\end{array}$$

Deshalb reicht es zu zeigen, dass die Lösung eindeutig ist. Sei

$$N := \left\{ w \in \mathcal{C}^2[a,b] : w(x_i) = 0,\ i = 0, \dots, n \right\}. \tag{3.27}$$

Für ein beliebiges $w \in N$ gilt nach zweimaliger partieller Integration, dass

$$\begin{aligned}
\int_a^b s_n''(x) w''(x)\,\mathrm{d}x &= \sum_{i=1}^n \int_{x_{i-1}}^{x_i} s_n''(x) w''(x)\,\mathrm{d}x \\
&= \sum_{i=1}^n \left((s_n'' w')\big|_{x_{i-1}}^{x_i} - \left(s_n^{(3)} w \right)\Big|_{x_{i-1}}^{x_i} + \int_{x_{i-1}}^{x_i} s_n^{(4)}(x) w(x)\,\mathrm{d}x \right) \\
&= \sum_{i=1}^n \Big((s_n'' w')\big|_{x_{i-1}}^{x_i} - 3! a_3^{(i)} \big(\underbrace{w(x_i)}_{=0} - \underbrace{w(x_{i-1})}_{=0} \big) + 0 \Big) \\
&= s_n''(x_n) w'(x_n) - s_n''(x_0) w'(x_0) = 0
\end{aligned}$$

Seien nun $s_n^{(1)}, s_n^{(2)} \in N$ zwei interpolierende Splines. Dann ist $s \equiv s_n^{(1)} - s_n^{(2)} \in N$ und

$$\int_a^b \left| s''(x) \right|^2 \mathrm{d}x = \int_a^b \left(s_n^{(1)} \right)'' (x) s''(x)\,\mathrm{d}x - \int_a^b \left(s_n^{(2)} \right)'' (x) s''(x)\,\mathrm{d}x = 0,$$

d. h. s ist linear. Wegen $s(a) = s(b) = 0$, folgt somit: $s \equiv 0$. □

Satz 3.22. *Für alle $f \in \mathcal{C}^2[a,b]$ mit $f(x_i) = y_i,\ i = 0,\dots,n$, gilt*

$$\int_a^b |s_n''(x)|^2 \,\mathrm{d}x \ \le\ \int_a^b |f''(x)|^2 \,\mathrm{d}x\,. \tag{3.28}$$

Beweis. Sei $f \in \mathcal{C}^2[a,b]$ mit $f(x_i) = y_i$ beliebig. Dann existiert ein $w \in N$ (definiert in (3.27)), sodass $f = s_n + w$ und

$$\int_a^b s_n''(x) w''(x)\,\mathrm{d}x = 0,$$

woraus folgt, dass

$$\begin{aligned}\int_a^b |f''(x)|^2 \,\mathrm{d}x &= \int_a^b |s_n''(x) + w''(x)|^2 \,\mathrm{d}x \\ &= \int_a^b |s_n''(x)|^2 \,\mathrm{d}x + 2\underbrace{\int_a^b s_n''(x) w''(x)\,\mathrm{d}x}_{=0} + \underbrace{\int_a^b |w_n''(x)|^2 \,\mathrm{d}x}_{\ge 0} \\ &\ge \int_a^b |s_n''(x)|^2 \,\mathrm{d}x\,.\end{aligned}$$

□

Zur expliziten Berechnung von s_n schreiben wir für $i = 1,\dots,n$ die Polynome $p_i = s_n|_{I_i} \in \mathcal{P}_3$ in der Form

$$p_i(x) = a_0^{(i)} + a_1^{(i)}(x - x_i) + a_2^{(i)}(x - x_i)^2 + a_3^{(i)}(x - x_i)^3.$$

Aus den Interpolationsbedingungen $p_i(x_i) = y_i$ und $p_i(x_{i-1}) = y_{i-1}$ folgt für $i = 1,\dots,n$

(1) $a_0^{(i)} = y_i$

(2) $-a_1^{(i)} h_i + a_2^{(i)} h_i^2 - a_3^{(i)} h_i^3 = y_{i-1} - y_i$

Aus der Stetigkeit der ersten und zweiten Ableitung folgt für $i = 2,\dots,n$

(3) $a_1^{(i-1)} = a_1^{(i)} - 2a_2^{(i)} h_i + 3a_3^{(i)} h_i^2$

(4) $2a_2^{(i-1)} = 2a_2^{(i)} - 6a_3^{(i)} h_i$

Die Randbedingungen $p_1''(x_0) = 0 = p_n''(x_n)$ ergeben

(5) $2a_2^{(1)} - 6a_3^{(1)} h_1 = 0 \quad$ und $\quad 2a_2^{(n)} = 0,$

also $4n$ Gleichungen für die $4n$ Unbekannten $a_0^{(i)},\dots,a_3^{(i)},\ i = 1,\dots,n$.

Zur Vereinfachung führen wir $a_2^{(0)} := 0$ ein. Aus (4) und (5) folgt dann

(6) $a_3^{(i)} = \dfrac{a_2^{(i)} - a_2^{(i-1)}}{3h_i}, \quad i = 1, \ldots, n.$

Aus (2) und (6) folgt MMS

(7) $a_1^{(i)} = \frac{y_i - y_{i-1}}{h_i} + \frac{h_i}{3}\left(2a_2^{(i)} + a_2^{(i-1)}\right), \quad i = 1, \ldots, n.$

Schließlich folgt aus (3), (6) und (7) MMS

$$h_i a_2^{(i-1)} + 2\left(h_i + h_{i+1}\right) a_2^{(i)} + h_{i+1} a_2^{(i+1)} = 3\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right), \quad i = 1, \ldots, n-1,$$

d. h. zu lösen ist das lineare $(n-1) \times (n-1)$-Gleichungssystem

$$Aa_2 = b \tag{3.29}$$

mit

$$A = \begin{pmatrix} 2(h_1+h_2) & h_2 & & & & \\ h_2 & 2(h_2+h_3) & h_3 & & & \\ & h_3 & 2(h_3+h_4) & h_4 & & \\ & & \ddots & \ddots & \ddots & \\ & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ & & & & h_{n-1} & 2(h_{n-1}+h_n) \end{pmatrix}$$

und

$$b_i = 3\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right), \quad i = 1, \ldots, n-1$$

zur Bestimmung des $(n-1)$-Vektors $a_2 = (a_2^{(1)}, \ldots, a_2^{(n-1)})^\mathsf{T}$, d.h. die Koeffizienten der quadratischen Terme. Aufgrund der zweiten Randbedingung in (5) ist $a_2^{(n)} = 0$.

Die Matrix A in (3.29) ist symmetrisch und **strikt diagonaldominant**, d. h.

$$\sum_{j=1, j\neq i}^{n-1} |a_{ij}| < |a_{ii}|, \quad i = 1, \ldots, n-1.$$

Es folgt daraus (mit Hilfe eines Satzes aus der linearen Algebra), dass (3.29) eindeutig lösbar ist. Zur Lösung kann eine spezielle Variante der Gauß-Elimination verwendet werden — für Details zu diesen Punkten siehe Kapitel 5 und 6.

Einsetzen der berechneten Werte $a_2^{(i)}$, $i = 1, \ldots, n$, in (6) und (7) liefert zusammen mit (1), alle restlichen Koeffizienten.

Satz 3.23 (Approximationsfehler)**.** *Sei $f \in \mathcal{C}^4[a,b]$. Erfüllt der interpolierende kubische Spline die (inhomogenen) natürlichen Randbedingungen $s_n''(a) = f''(a)$ sowie $s_n''(b) = f''(b)$, so gilt*

$$\|f - s_n\|_{\infty,[a,b]} \;\leq\; \frac{h^4}{2} \left\|f^{(4)}\right\|_{\infty,[a,b]} .$$

Beweis. Siehe z.B. [WS72]. □

Neben den guten Approximationseigenschaften, weisen Splines auch eine wesentlich bessere numerische Stabilität als Lagrange-Polynome auf gegenüber kleinen Störung in den Daten.

3.5 Trigonometrische Interpolation

In vielen Anwendungsbereichen treten **periodische Funktionen** mit der Eigenschaft

$$f(x+\omega) = f(x), \quad \text{für alle} \;\; x \in \mathbb{R},$$

auf. Man bezeichnet $\omega > 0$ als **Periode**. Hier bietet sich die Interpolation mit ω-periodischen **trigonometrischen „Polynomen“** an

$$t_n(x) = \frac{1}{2}a_0 + \sum_{k=1}^{m} \left(a_k \cos\left(\frac{2\pi k x}{\omega}\right) + b_k \sin\left(\frac{2\pi k x}{\omega}\right)\right) \tag{3.30}$$

mit $n := 2m$. Ohne Beschränkung der Allgemeinheit setzen wir im Folgenden $\omega = 2\pi$; das Interpolationsintervall ist dann $[0, 2\pi]$. Als Stützstellen wählen wir

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

Beachte, dass aufgrund der Periodizität der Wert in $x_{n+1} = 2\pi$ implizit gleich dem Wert in $x_0 = 0$ sein muss und deshalb nicht als Stützstelle gewählt ist – sonst wäre das Problem nicht gut gestellt.

Es ist vorteilhaft das Interpolationsproblem mit t_n in (3.30) zuerst über den komplexen Zahlen $\mathbb{C}$ zu betrachten. Mit $i = \sqrt{-1}$ gilt

$$\cos(x) = \frac{e^{ix} + e^{-ix}}{2} \quad \text{und} \quad \sin(x) = \frac{e^{ix} - e^{-ix}}{2i} . \tag{3.31}$$

Satz 3.24 (Trigonometrische Interpolation)**.** *Gegeben seien $y_0, \dots, y_n \in \mathbb{C}$. Dann existiert genau eine (2π-periodische) Funktion der Form*

$$t_n^*(x) = \sum_{k=0}^{n} c_k e^{ikx}, \quad x \in [0, 2\pi], \tag{3.32}$$

sodass

$$t_n^*(x_j) = y_j, \quad j = 0, \dots, n.$$

Die Koeffizienten sind bestimmt durch

$$c_k = \frac{1}{n+1} \sum_{j=0}^{n} y_j e^{-ijx_k}, \quad k = 0, \dots, n. \tag{3.33}$$

Beweis. Siehe [Ran17, Satz 2.11]. (Man verwendet die Substitution $w = e^{ix}$ und dann eine Verallgemeinerung von Satz 3.8 von $\mathbb{R}$ auf $\mathbb{C}$.) □

Mit Hilfe von Satz 3.24 kann nun die ursprünglich gestellte trigonometrische Interpolationsaufgabe über $\mathbb{R}$ gelöst werden. Sei $n = 2m \in \mathbb{N}$ (betrachten hier der Einfachheit halber nur gerade n).

Satz 3.25 (Diskrete Fourier-Analyse)**.** *Gegeben seien $y_0, \dots, y_n \in \mathbb{R}$. Dann existiert genau ein trigonometrisches Polynom der Form* (3.30) *mit*

$$t_n(x_j) = y_j, \quad j = 0, \dots, n.$$

Die Koeffizienten sind bestimmt durch

$$a_k = \frac{2}{n+1} \sum_{j=0}^{n} y_j \cos(jx_k) \quad \textit{und} \quad b_k = \frac{2}{n+1} \sum_{j=0}^{n} y_j \sin(jx_k).$$

Beweis. Sei $t^*(x)$ das Interpolationspolynom aus Satz 3.24.

Wir beweisen das Resultat in vier Schritten.

(i) Aus der 2π-Periodizität von e^{-ix} folgt

$$e^{-ijx_{n+1-k}} = e^{-ij(n+1-k)\frac{2\pi}{n+1}} = e^{-ij2\pi + ijx_k} = e^{ijx_k}, \tag{3.34}$$

und somit

$$c_{n+1-k} = \frac{1}{n+1} \sum_{j=0}^{n} y_j e^{-ijx_{n+1-k}} = \frac{1}{n+1} \sum_{j=0}^{n} y_j e^{ijx_k} =: c_{-k}\,, \tag{3.35}$$

für $k = 1, \dots, m$. Wir setzen $a_0 := 2c_0$ und für $k = 1, \dots, m$

$$a_k := c_k + c_{-k} \quad \text{und} \quad b_k := i(c_k - c_{-k}).$$

(ii) Als nächstes zeigen wir $t_n(x_j) = y_j$ für $j = 0, \dots, n$. Es gilt

$$\begin{aligned} t_n(x_j) &= c_0 + \sum_{k=1}^{m} \{(c_k + c_{-k}) \cos(kx_j) + i(c_k - c_{-k}) \sin(kx_j)\} \\ &= c_0 + \sum_{k=1}^{m} c_k \left(\cos(kx_j) + i \sin(kx_j)\right) + c_{-k} \left(\cos(kx_j) - i \sin(kx_j)\right) . \end{aligned}$$

Daraus folgt wegen $e^{iz} = \cos(z) + i \sin(z)$, dass

$$t_n(x_j) = c_0 + \sum_{k=1}^{m} c_k e^{ikx_j} + \sum_{k=1}^{m} c_{-k} e^{-ikx_j} .$$

Aus (3.34) und (3.35) folgt

$$\begin{aligned} t_n(x_j) &= c_0 + \sum_{k=1}^{m} c_k e^{ikx_j} + \sum_{k=1}^{m} c_{n+1-k} e^{i(n+1-k)x_j} \\ &= c_0 + \sum_{k=1}^{n} c_k e^{ikx_j} = t_n^*(x_j) = y_j , \end{aligned}$$

wobei wir in der zweiten Summe $k' = n + 1 - k$ substituiert haben. Also erfüllt t_n die Interpolationsbedingungen.

(iii) Um die Koeffizienten a_k, b_k zu bestimmen, verwenden wir (3.31), sodass

$$a_k = c_k + c_{-k} = \frac{1}{n+1} \sum_{j=0}^{n} y_j \left(e^{-ijx_k} + e^{ijx_k}\right) = \frac{2}{n+1} \sum_{j=0}^{n} y_j \cos(jx_k)$$

und

$$b_k = i(c_k + c_{-k}) = \frac{1}{n+1} \sum_{j=0}^{n} y_j i \left(e^{-ijx_k} - e^{ijx_k}\right) = \frac{2}{n+1} \sum_{j=0}^{n} y_j \sin(jx_k) .$$

Insbesondere sind a_k und b_k reell.

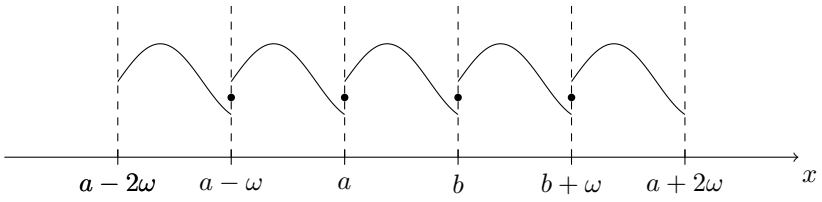
(iv) Die $n + 1$ Interpolationsbedingungen bilden ein lineares Gleichungssystem für die $n+1$ Koeffizienten a_k, b_k. Wir haben soeben gezeigt, dass dieses Gleichungssystem für beliebige $y_0, \dots, y_n$ lösbar ist, woraus die Eindeutigkeit wieder direkt folgt (im Endlichdimensionalen).

□

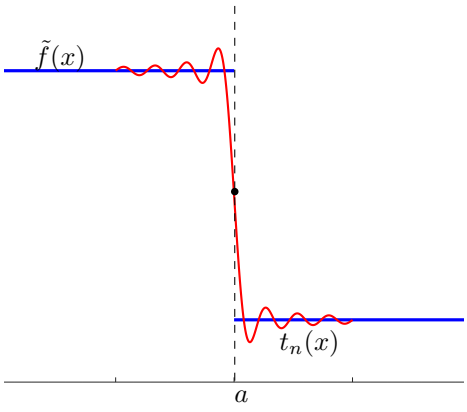
Man kann aber auch beliebige, nicht notwendigerweise periodische Funktionen f auf einem Intervall $[a, b]$ mit trigonometrischen Polynomen interpolieren. Sei $\omega = b - a$. Wir setzen f zunächst zu folgender ω-periodischen Funktion fort:

$$\tilde{f}(x) := \begin{cases} f(x), & x \in (a, b)\,, \\ \frac{1}{2}\left(f(a) + f(b)\right), & x = a\,, \\ \omega\text{-periodisch auf } \mathbb{R} \text{ fortgesetzt.} \end{cases}$$

Es ergibt sich das folgende Bild:



Leider tritt bei der trigonometrischen Interpolation von $\tilde{f}$ im Falle einer Unstetigkeit bei $x = a$ das sog. **Gibbs Phänomen** auf: es kommt zu starken Oszillationen, deren Amplituden auch mit steigendem m nicht abnehmen:

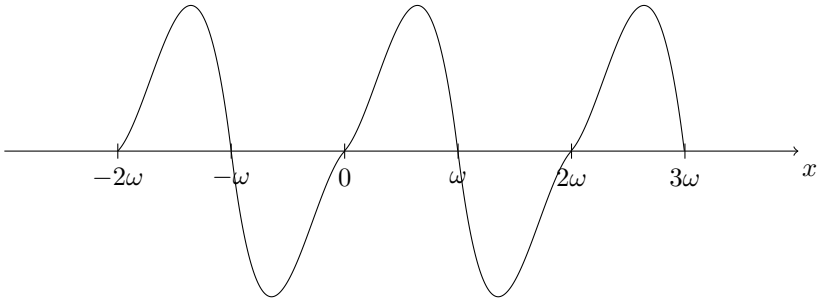


Um Unstetigkeiten zu vermeiden und bessere Approximationseigenschaften zu erzielen, kann f auch 2ω-periodisch fortgesetzt werden:

Der Einfachheit halber betrachten wir nur den Fall $a = 0, f(a) = f(b) = 0$. Eine **ungerade 2π-periodische** Fortsetzung von f ist dann gegeben durch

$$\tilde{f}(x) := \begin{cases} f(x), & x \in (0, \omega) \\ -f(2\omega - x), & x \in (\omega, 2\omega) \\ 2\omega\text{-periodisch auf } \mathbb{R} \text{ fortgesetzt} \end{cases}$$

Offenbar ist dann $\tilde{f}$ in $x = \omega$ stetig differenzierbar.



Man kann zeigen, dass in diesem Fall $c_k = -c_{-k}$, d. h. $a_k = 0$ für $k = 0, 1, \ldots, n$ und es bleiben nur die Sinusterme, d. h. eine reine Sinusreihe. Man kann f auch **gerade 2π-periodisch** fortsetzen. Dann erhält man eine reine Cosinusreihe. Für Details siehe [Ran17, S. 52ff.].

MMS

3.5.1 Fast Fourier Transformation (FFT)

Wir betrachten nun eine Methode zur effizienteren Berechnung des trigonometrischen Interpolationspolynoms. Die in Satz 3.25 behandelte Aufgabe wird **diskrete Fourier Analyse** genannt. Den $n + 1$ Werten $y_j = f(x_j)$ einer Funktion $f : [0, 2\pi] \to \mathbb{R}$ werden (**eindeutig**) die $n + 1$ Koeffizienten

$$a_0, a_k, b_k, \quad k = 1, \ldots, m = \frac{n}{2}$$

des trigonometrischen Interpolationspolynoms

$$t_n(x) = \frac{1}{2}a_0 + \sum_{k=1}^{m} (a_k \cos(kx) + b_k \sin(kx))$$

zugeordnet. Die Abbildung

$$\{y_j \ : \ j = 0, \ldots, n\} \to \{a_0, a_k, b_k \ : \ k = 1, \ldots, m\}$$

heißt **diskrete Fourier-Transformation**. Sie ist offenbar umkehrbar (**Inverse Fourier Transformation**).

Interpretiert man $\cos(kx)$ und $\sin(kx)$ als Grundschwingungen eines 2π periodischen Prozesses $y = f(x)$, so ist die (diskrete) Fourier-Analyse eine Frequenzanalyse zur Bestimmung der jeweiligen Anteile dieser Grundschwingungen am gesamten Prozess. Aus dem Beweis von Satz 3.25 folgt

$$a_k = c_k + c_{-k}, \qquad b_k = i(c_k - c_{-k})$$

mit

$$c_k = \frac{1}{n+1}\sum_{j=0}^{n} y_j e^{i\frac{jk2\pi}{n+1}} = \sum_{j=0}^{n} \overline{y_j} w^{jk}, \quad k = 0, \ldots, n, \tag{3.36}$$

wobei

$$\overline{y_j} := \frac{1}{n+1} y_j \quad \text{und} \quad w := e^{-i\frac{2\pi}{n+1}}$$

(beachte $w^{n+1} = 1$). Zur Auswertung auf Basis dieser Formeln (mit Hilfe des Horner-Schemas) werden die folgenden (komplexen) Operationen benötigt:

$y_j \to \overline{y_j}$	$n+1$	Divisionen	(once)
$w^{k-1} \to w^k$	1	Multiplikation	
c_k	n	Multiplikationen	(for each k)
	n	Additionen	
a_k, b_k	2	Additionen	
Insgesamt:	$(n+1)(2n+4)$	$=$	$2(n+1)(n+2)$ Operationen

Cooley & Tukey entwickelten 1965 einen bahnbrechenden Algorithmus, der diese Aufgabe in $2(n+1)\log_2(n+1)$ Operationen löst: die **schnelle Fourier Transformation** oder **Fast Fourier Transform (FFT)**.

Wir betrachten nur die Idee für den Fall $n+1 = 2^p$ für $p \in \mathbb{N}$. Für den genauen Beweis der Komplexitätsaussage, siehe [Ran17, Satz 2.13].

Wir spalten die Summe in c_k in gerade und ungerade Terme:

$$\begin{aligned} c_k &= \left(\overline{y_0}\omega^{0k} + \overline{y_1}\omega^{1k}\right) + \left(\overline{y_2}\omega^{2k} + \overline{y_3}\omega^{3k}\right) + \ldots + \left(\overline{y_{n-1}}\omega^{(n-1)k} + \overline{y_n}\omega^{nk}\right) \\ &= \left(\overline{y_0}(\omega^2)^{0k} + \overline{y_1}(\omega^2)^{0k}\omega^k\right) + \left(\overline{y_2}(\omega^2)^{1k} + \overline{y_3}(\omega^2)^{1k}\omega^k\right) + \ldots \\ &\quad \ldots + \left(\overline{y_{n-1}}(\omega^2)^{\frac{n-1}{2}k} + \overline{y_n}(\omega^2)^{\frac{n-1}{2}k}\omega^k\right) \\ &= \sum_{j=0}^{\frac{n-1}{2}} \overline{y_{2j}}(\omega^2)^{jk} + \omega^k \sum_{j=0}^{\frac{n-1}{2}} \overline{y_{2j+1}}(\omega^2)^{jk}\,. \end{aligned} \tag{3.37}$$

Aber: $\frac{n-1}{2} = \frac{n+1}{2} - 1 = 2^{p-1} - 1$. Dann gilt wegen $\omega^{n+1} = \omega^{2^p} = 1$, dass

MMS

$$(\omega^2)^{jk} = (\omega^2)^{jk_1} \quad \text{mit } k_1 = k \mod 2^{p-1}$$

(der ganzzahlige Rest bei Division von k durch 2^{p-1}). Folglich ist

$$c_k = c^g_{k_1} + \omega^k c^u_{k_1}, \quad \text{für } k = 0, \ldots, n, \tag{3.38}$$

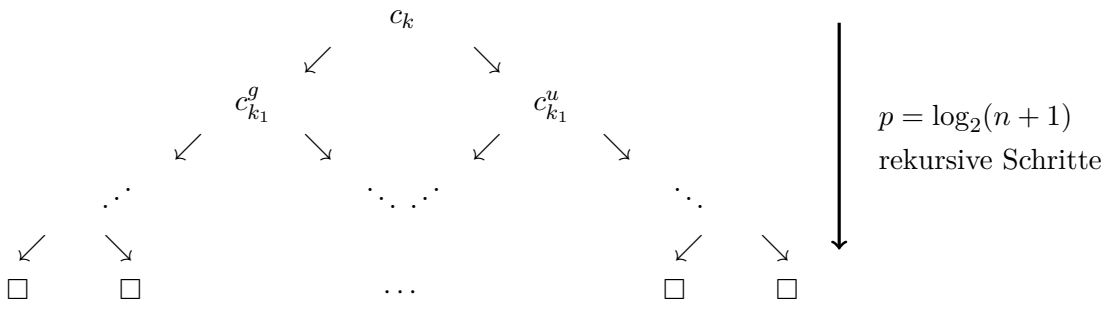
mit

$$c^g_{k_1} = \sum_{j=0}^{\frac{n-1}{2}} \overline{y_{2j}}\widetilde{\omega}^{jk_1} \quad \text{und} \quad c^u_{k_1} = \sum_{j=0}^{\frac{n-1}{2}} \overline{y_{2j+1}}\widetilde{\omega}^{jk_1}, \quad \text{für} \;\; k_1 = 0, \ldots, \frac{n-1}{2}$$

und

$$\widetilde{\omega} := \omega^2 = e^{-i\frac{2\pi}{\left(\frac{n+1}{2}\right)}}.$$

Das heißt, zur Berechnung der $n+1$ Größen c_k mit jeweils $n+1$ Termen genügt es die $2\frac{n+1}{2}$ Größen $c^g_{k_1}$ und $c^u_{k_1}$ mit jeweils $\frac{n+1}{2}$ Termen zu berechnen. Jede der Summen $c^g_{k_1}$ und $c^u_{k_1}$ ist wieder eine Fourier-Reihe der Form (3.36), jedoch nur mit der Hälfte der Terme. Wir können als wieder, wie in (3.37), in gerade und ungerade Terme zerlegen. Gesamt erhält man



wobei jede der „Summen“ in den Quadraten nur mehr aus einem Term besteht und deshalb durch eine Multiplikation berechnet werden kann.

Rekursiv lässt sich daraus c_k in $2\log_2(n+1)$ Operationen mit Hilfe von (3.38) berechnen. Insgesamt werden also $2(n+1)\log_2(n+1)$ Operationen zur Berechnung aller c_k benötigt.

Beispiel 3.26. Sei $p = 7$, d. h. $n + 1 = 2^7 = 128$. Dann ist

$$2(n+1)(n+2) = 2 \times 128 \times 129 = 33024$$
$$\text{und} \quad 2(n+1)\log_2(n+1) = 2 \times 128 \times 7 = 1792,$$

was nur 5.4% der ursprünglichen Anzahl an Operationen entspricht.

Bei $p = 10$ reduziert sich die Anzahl an Operationen bereits auf 1%!

3.6 Gauß-Approximation

Zum Ende des Kapitels gehen wir nun weg von der Interpolation und betrachten die Best-Approximation von Funktionen. Der Einfachheit halber betrachten wir vorerst

nur Funktionen $f \in \mathcal{C}[-1,1]$. Die Best-Approximation geschieht dabei immer bzgl. einer Norm $\|\cdot\|$ und einem Teilraum $S \subset \mathcal{C}[-1,1]$.

3.6.1 Best-Approximationspolynom

Wir betrachten $S = \mathcal{P}_n$, also den endlichdimensionalen Teilraum der Polynome vom Grad kleiner oder gleich n auf $[-1,1]$, und nehmen als Norm das sog. **quadratische Mittel** (oder die L_2**-Norm**):

$$\|f\| := \left(\int_{-1}^{1} |f(x)|^2 \, \mathrm{d}x \right)^{1/2}$$

(vgl. die bekannte euklidische Vektornorm im $\mathbb{R}^n$). Das Gegenstück zum euklidischen Skalarprodukt ist in diesem Fall das L_2**-Skalarprodukt**:

$$\langle f, g \rangle \;:=\; \int_{-1}^{1} f(x) g(x) \, \mathrm{d}x. \tag{3.39}$$

Wie in Appendix A kurz erläutert ist $\langle \cdot, \cdot \rangle$ linear in beiden Argumenten, symmetrisch und es gilt

$$\langle f, f \rangle \geq 0 \qquad \text{mit} \qquad \langle f, f \rangle = 0 \;\Rightarrow f \equiv 0\,.$$

Für die L_2-Norm gilt

$$\begin{aligned} \|f\| &= 0 \Leftrightarrow f \equiv 0 \\ \|\lambda f\| &= |\lambda| \|f\| \\ \|f+g\| &\leq \|f\| + \|g\| \end{aligned}$$

für $f, g \in \mathcal{C}[-1,1],\ \lambda \in \mathbb{R}$ beliebig. Außerdem gilt auch wieder die **Cauchy-Schwarz-Ungleichung**

$$|\langle f, g \rangle| \;\leq\; \|f\| \, \|g\|\,.$$

Satz 3.27 (Gauß-Approximation in $\mathcal{C}[-1,1]$ bzgl. $\|\cdot\|$)**.** *Zu jedem $f \in \mathcal{C}[-1,1]$ existiert ein eindeutig bestimmtes* **Best-Approximationspolynom** *$g \in \mathcal{P}_n$ vom Grad kleiner oder gleich n, sodass*

$$\|f - g\| = \min_{\varphi \in \mathcal{P}_n} \|f - \varphi\|\,. \tag{3.40}$$

Beweis. Spezialfall von Satz 3.31. Für den Beweis siehe dort. □

Ein wichtiges Zwischenresultat im Beweis von Satz 3.27 ist, dass

$$\langle f - g, \varphi \rangle = 0 \quad \text{für alle} \;\; \varphi \in \mathcal{P}_n, \tag{3.41}$$

d.h. der Fehler in der Best-Approximation ist orthogonal zu $\mathcal{P}_n$ in $\mathcal{C}[-1,1]$ bzgl. $\langle\cdot,\cdot\rangle$.

Das heißt für eine Basis $\{\psi_0,\dots,\psi_n\}$ von $\mathcal{P}_n$, und $g := \sum_{k=0}^n \alpha_k\psi_k$ gilt:

$$0 = \left\langle f - \sum_{k=0}^n \alpha_k\psi_k, \psi_i \right\rangle = \langle f,\psi_i\rangle - \sum_{k=0}^n \alpha_k\langle\psi_k,\psi_i\rangle, \quad i = 0,1,\dots,n. \tag{3.42}$$

Dieses Gleichungssystem hat eine besonders einfache Lösung, wenn $\{\psi_0,\dots,\psi_n\}$ ein **Orthonormalsystem** (ONS) ist, d. h.

$$\langle\psi_k,\psi_i\rangle = \delta_{ki} = \begin{cases} 1 & \text{für } k = i, \\ 0 & \text{für } k \neq i. \end{cases}$$

Dann folgt offenbar aus (3.42), dass $\alpha_k = \langle f,\psi_k\rangle$ und damit

$$g = \sum_{k=0}^n \langle f,\psi_k\rangle\psi_k\,. \tag{3.43}$$

Ausserdem gilt für alle $f \in \mathcal{P}_n$ die **Parseval-Identität**

MMS

$$\|f\|^2 = \sum_{k=0}^n \langle f,\psi_k\rangle^2. \tag{3.44}$$

Zur Konstruktion eines Orthonormalsystems aus einer Basis eines endlichdimensionalen Vektorraums kann das **Gram-Schmidt-Verfahren** verwendet werden (vgl. Appendix A und [Ran17, Hilfssatz 2.1]).

Beispiel 3.28. Die Orthonormalisierung der Monombasis $\{1,x,x^2,\dots,x^n\}$ von $\mathcal{P}_n$ mit dem Gram-Schmidt-Verfahren führt auf die sog. **Legendre-Polynome** (siehe [Ran17, Satz 2.15]), welche rekursiv definiert sind durch

$$\begin{aligned} p_0 \equiv 1, \quad p_1(x) = x, \\ p_{k+1}(x) = x\,p_k(x) - \frac{k^2}{4k^2-1}p_{k-1}(x), \quad \text{für } k = 1,2,\dots,n-1 \end{aligned} \tag{3.45}$$

sowie

$$\|p_k\| = \frac{(k!)^2}{(2k)!}\sqrt{\frac{2^{2k+1}}{2k+1}}, \quad p_k(1) = \frac{2^k(k!)^2}{(2k)!}\,. \tag{3.46}$$

Der üblichen Konvention folgend normieren wir diese Polynom bei $x = 1$, um die klassischen **(Gauß-) Legendre Polynome** zu erhalten:

$$L_k(x) := \frac{(2k)!}{2^k(k!)^2}\,p_k(x), \quad \text{für } k = 0,1,\dots,n, \tag{3.47}$$

die die Bedingung $L_k(1) = 1$ erfüllen. Das Best-Approximationspolynom bzgl. $\|\cdot\|$ in $\mathcal{P}_n$ ist dann gegeben durch

$$g = \sum_{k=0}^{n} \frac{\langle f, L_k \rangle}{\|L_k\|^2} L_k = \sum_{k=0}^{n} \left(k + \frac{1}{2}\right) \langle f, L_k \rangle L_k. \tag{3.48}$$

3.6.2 Allgemeine Gauß Approximation

Definition 3.29. Ein Vektorraum von Funktionen über $\mathbb{R}$ auf dem ein Skalarprodukt $\langle \cdot, \cdot \rangle$ definiert ist, heißt **Skalarproduktraum**. Da durch

$$\|f\| = \sqrt{\langle f, f \rangle}$$

stets eine Norm definiert wird, ist jeder Skalarproduktraum normiert. Wenn er zusätzlich noch vollständig ist wird ein Skalarproduktraum zum **Hilbertraum**.

Beispiel 3.30.

(i) $\mathcal{C}[-1, 1]$ mit dem L^2-Skalarprodukt in (3.39) ist ein Skalarproduktraum.

(ii) Der Raum $L^2(-1, 1)$ der quadratintegrierbaren Funktionen auf $[-1, 1]$, d.h. alle Funktionen f für die gilt $\|f\| = \left(\int_{-1}^{1} f(x)\,\mathrm{d}x\right)^{1/2} < \infty$, ist ein Hilbertraum. Für beliebige $a < b \in \mathbb{R}$ ist der Raum $L^2(a, b)$ analog definiert.[2]

Im folgenden sei H ein Skalarproduktraum, mit Norm $\|\cdot\|$ induziert durch das Skalarprodukt, und $S \subset H$ ein endlich-dimensionaler Teilraum von H. Wir betrachten die **allgemeine Gauß-Approximationsaufgabe**:

Zu beliebigem $f \in H$ finde $g \in S$, sodass

$$\|f - g\| \to \min. \tag{3.49}$$

Satz 3.31 (Allgemeine Gauß-Approximation)**.** *Sei $S \subset H$ ein endlich-dimensionaler Teilraum eines Skalarproduktraumes H und sei $f \in H$ beliebig. Es gibt genau eine Funktion $g \in S \subset H$, die $\|f - g\|$ minimiert, und g erfüllt die Bedingung*

$$\langle g, \varphi \rangle = \langle f, \varphi \rangle \quad \forall\, \varphi \in S. \tag{3.50}$$

Beweis. Sei $\|f - g\|$ minimal. Für jedes $\varphi \in S$, definieren wir die Funktion

$$t \in \mathbb{R} \;\to\; F_\varphi(t) \coloneqq \|(f - g) + t\varphi\|^2,$$

[2]Dabei ist das Integral $\int_a^b f\,\mathrm{d}x$ als **Lebesgue-Integral** zu verstehen. Für Details siehe die **Funktionalanalysis** Vorlesung.

die (nach Vorraussetzung) ihr Minimum bei $t = 0$ animmt. Daraus folgt $F'_\varphi(0) = 0$. Ableiten nach t ergibt

$$\begin{aligned} F'_\varphi(t) &= \frac{\mathrm{d}}{\mathrm{d}t}\Big\langle (f-g) + t\varphi, (f-g) + t\varphi \Big\rangle \\ &= \frac{\mathrm{d}}{\mathrm{d}t}\left[\|f-g\|^2 + 2t\langle f-g, \varphi\rangle + t^2\|\varphi\|^2\right] = 2\langle f-g, \varphi\rangle + 2t\|\varphi\|^2 \end{aligned}$$

und deshalb folgt $0 = F'_\varphi(0) = 2\langle f-g, \varphi\rangle$, d.h. der Fehler ist (geometrisch gesehen) orthogonal zu allen $\varphi \in S$.

Sei nun umgekehrt $\langle f-g, \varphi\rangle = 0$ für alle $\varphi \in S$. Dann gilt für jedes beliebige $g' = g - \varphi \in S$, dass

$$\begin{aligned} \|f-g'\|^2 &= \langle f-g+\varphi, f-g+\varphi\rangle \\ &= \|f-g\|^2 + 2\underbrace{\langle f-g, \varphi\rangle}_{=0 \text{ nach Vor.}} + \|\varphi\|^2 \geq \|f-g\|^2 \end{aligned}$$

und somit ist g minimal. Damit ist die folgende Äquivalenz gezeigt:

$$\|f-g\| \to \min \iff \langle f-g, \varphi\rangle = 0 \iff \langle g, \varphi\rangle = \langle f, \varphi\rangle \quad \forall\, \varphi \in S.$$

Angenommen es gäbe zwei Minima g_1 und g_2 mit $\varphi = g_2 - g_1 \neq 0$. Dann folgt aus obiger Äquivalenz, dass

$$\|f-g_1\|^2 = \|f-g_2+\varphi\|^2 = \|f-g_2\|^2 + \|\varphi\|^2 > \|f-g_2\|^2,$$

und damit ein Widerspruch zur Annahme, dass g_1 minimal ist. Das Minimum ist also eindeutig.

Da $\dim S = N < \infty$, folgt die Existenz wieder direkt aus der Eindeutigkeit und den N Bedingungen, die wir aus (3.50) erhalten, z.B. durch Einzetzen einer Basis. □

MMS

3.7 Adaptive Best-Approximation mit Haar-Wavelets

Wir knüpfen an die allgemeine Gauß-Approximation von Funktionen $f \in H$ und möchten diese nun **möglichst effizient** gestalten im Sinne von:

- garantierte Fehlerkontrolle

$$\|f-g\| \leq \mathrm{TOL} \cdot \|f\|, \tag{3.51}$$

  wobei $\mathrm{TOL} > 0$ eine vorgegebene Fehlerschranke ist,

- und einen Approximationsraum S von möglichst kleiner Dimension $\dim S$.

Diese Frage lässt sich wieder sehr einfach mittels eines Orthonormalsystems lösen.

Sei $(S_N)_{N\in\mathbb{N}} \subset H$ eine Folge von Approximationsräumen, mit $\dim S_N = N$ und Orthonormalsystemen (ONS) $\{\psi_1, \dots, \psi_N\}$ für $N \in \mathbb{N}$. Dann gilt $S_N \subset S_{N+1}$.

Für die Best-Approximation g_N von f in S_N gilt

$$\begin{aligned}
\|f - g_N\|^2 &= \|f\|^2 - 2\langle f, g_N\rangle + \|g_N\|^2 \\
&\overset{\text{Basis einsetzen}}{=} \|f\|^2 - 2\left\langle f, \sum_{i=1}^{N}\langle f,\psi_i\rangle\psi_i\right\rangle + \left\langle \sum_{i=1}^{N}\langle f,\psi_i\rangle\psi_i, \sum_{i=1}^{N}\langle f,\psi_i\rangle\psi_i\right\rangle \\
&= \|f\|^2 - 2\sum_{i=1}^{N}\langle f,\psi_i\rangle\langle f,\psi_i\rangle + \sum_{i=1}^{N}\sum_{j=1}^{N}\langle f,\psi_i\rangle\langle f,\psi_j\rangle \underbrace{\langle \psi_i,\psi_j\rangle}_{=\delta_{ij}} \\
&= \|f\|^2 - \sum_{i=1}^{N}\langle f,\psi_i\rangle^2.
\end{aligned}$$

Mit dieser expliziten Darstellung des Fehlers und der Bedingung (3.51) erhält man dann

$$\begin{aligned}
&\|f\|^2 - \sum_{i=1}^{N}\langle f,\psi_i\rangle^2 = \|f - g_N\|^2 \le \underbrace{\mathrm{TOL}^2 \cdot \|f\|^2}_{\text{relative Toleranz}} \\
\iff \quad &\sum_{i=1}^{N}\langle f,\psi_i\rangle^2 \ge \left(1 - \mathrm{TOL}^2\right)\|f\|^2. \qquad (3.52)
\end{aligned}$$

Aus dieser Fehlerdarstellung folgt auch unmittelbar, dass

MMS

$$\|f - g_{N+1}\| \le \|f - g_N\|, \quad \text{für alle } N \in \mathbb{N},$$

d.h. der Fehler kann durch Hinzufügen von Basisfunktionen **nicht** zunehmen.

3.7.1 Waveletfunktionen

Häufig variieren zu approximierende Funktionen nur lokal stark. So variiert die Funktion

$$f(x) = \frac{1}{10x^2 + \epsilon} \quad \text{mit} \quad |\epsilon| \ll 1$$

sehr stark in der Nähe von 0. Ansonsten ist die Funktion deutlich glatter und sollte gut mit wenig Aufwand approximiert werden können. In diesem Kapitel lernen wir eine Methode kennen die dies ermöglicht.

Um diese Lokalität besser quantifizieren zu können, zunächst eine Definition.

Definition 3.32. Der Support (oder Träger) einer Funktion ist die kleinste abgeschlossene Teilmenge der Definitionsmenge D einer Funktion $f : D \to \mathbb{R}$, in der alle Punkte liegen, an denen die Funktion Werte ungleich Null annimmt:

$$\operatorname{supp}(f) = \overline{\{x \in D : f(x) \neq 0\}}.$$

Für die Legendre-Polynome $p_n \in \mathcal{P}_n$ auf dem Intervall $D = [-1, 1]$ gilt zum Beispiel:

$$\operatorname{supp}(p_n) = [-1, 1] = D \text{ für alle } n \in \mathbb{N} \cup \{0\}.$$

Man sagt, diese Basisfunktionen haben einen **globalen** Träger. Das Verhalten einer Funktion f, die nur **lokal** an bestimmten Stellen sehr stark variiert, kann dann mit globalen Basisfunktionen nicht gut abgebildet werden. Besser wäre es eine Basis mit **lokalen Trägern** zu benutzen.

Wir untersuchen insbesondere L^2-orthogonale Basisfunktionen mit lokalem Träger. Sei nun $S_N \subset L^2[a, b]$ mit $S_N = \operatorname{span} \Psi_N$ und $\Psi_N := \{\psi_i : 1 \leq i \leq N\}$, dann wünschen wir uns folgende Eigenschaften von den Basisfunktionen ψ_i:

1. **Orthonormalität**: $\langle \psi_i, \psi_j \rangle = \int_a^b \psi_i(x)\psi_j(x)\,\mathrm{d}x = \delta_{ij}$.
2. **Hierarchische Basis**: $\Psi_N \subset \Psi_{N+1}$ (dies erlaubt adaptive Verfeinerung).
3. **Lokalität**: Der Träger $\operatorname{supp}(\psi_i)$ schrumpft für $i \to \infty$.

Funktionen mit diesen Eigenschaften nennt man **Waveletfunktionen**.

3.7.2 Haar-Wavelets

Wir betrachten hier das einfachste Wavelet, das sogennante **Haar-Wavelet** (benannt nach dem Mathematiker *Alfréd Haar*, 1909).

Sei $\chi(x)$ die **charakteristische Funktion** (bzw. **Indikatorfunktion**) des Intervalls $(0, 1]$ (Abbildung 3.3, links):

$$\chi(x) = \begin{cases} 1 & 0 < x \leq 1, \\ 0 & \text{sonst.} \end{cases}.$$

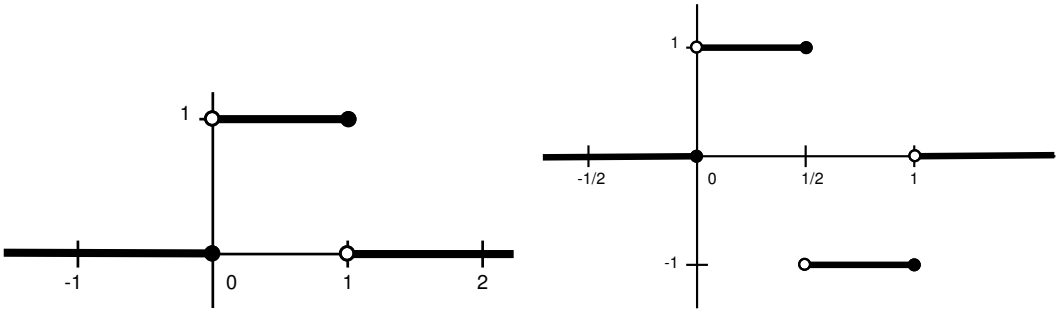


Abbildung 3.3: Die Indikatorfunktion $\chi(x)$ (links) und das daraus abgeleitete Haar-Wavelet der Stufe 0.

Definition 3.33 (Haar-Waveletbasis)**.** Wir definieren zunächst das sogenannte **Mother Wavelet**, das in Abbildung 3.3 (rechts) abgebildet ist:

$$\psi(x) = \chi(2x) - \chi(2x-1) = \begin{cases} 1 & 0 < x \leq 1/2 \\ -1 & 1/2 < x \leq 1 \\ 0 & \text{sonst} \end{cases}, \tag{3.53}$$

Die **hierarchische Haar-Waveletbasis** wird rekursiv definiert. Dabei werden die Basisfunktionen $\psi_i^k(x)$ mit einem zweidimensionalen Index (i,k) nummeriert, wobei $k \geq 0$ die Stufe und $i \geq 0$ die Nummer innerhalb einer Stufe k bezeichnet.

Auf den Stufe 0 und 1 gibt es je eine Basisfunktion und auf der Stufe $k > 1$ gibt es 2^{k-1} Basisfunktionen. Die **Haar-Waveletbasisfunktionen** sind definiert als

$$\psi_0^0(x) = \chi(x) \qquad \text{für } k = 0 \tag{3.54}$$

$$\psi_0^1(x) = \psi(x) \qquad \text{für } k = 1 \tag{3.55}$$

$$\psi_i^k(x) = 2^{\frac{k-1}{2}} \cdot \psi\left(2^{k-1}x - i\right) \qquad \text{für } k > 1, \quad 0 \leq i < 2^{k-1} \tag{3.56}$$

Der Vorfaktor ergibt sich aus der Normierung.

Schließlich besteht die Haar-Waveletbasis der Stufe $\ell \in \mathbb{N} \cup \{0\}$ aus der Vereinigungsmenge aller Wavelet-Funktionen bis zur Stufe ℓ:

$$\Psi^\ell := \left\{\psi_0^0\right\} \cup \bigcup_{k=1}^{\ell} \bigcup_{i=0}^{2^{k-1}-1} \left\{\psi_i^k\right\}. \tag{3.57}$$

Die Basisfunktionen bis zur Stufe $\ell = 3$ sind in Abbildung 3.4 aufgetragen.

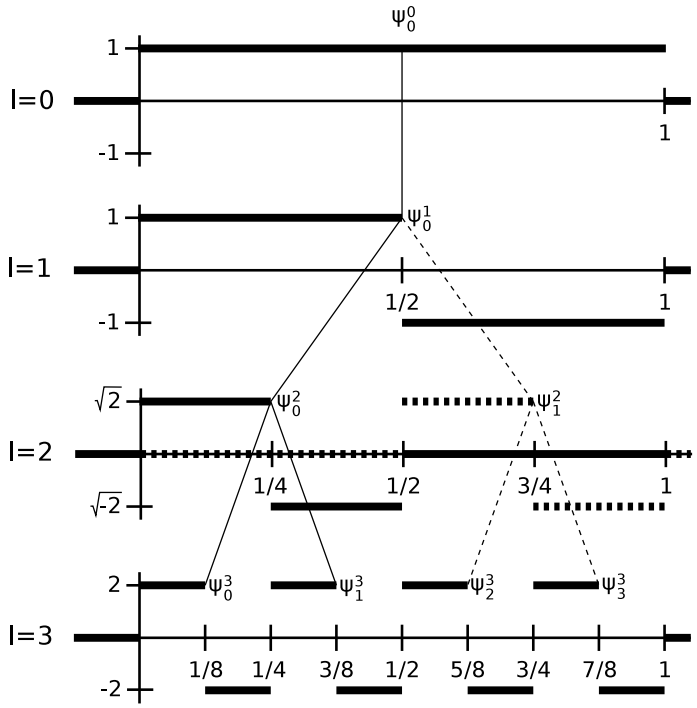


Abbildung 3.4: Baumstruktur von Haar-Wavelets bis zur Stufe 3.

Lemma 3.34 (Eigenschaften der Haar-Wavelets)**.** *Die Funktionen in Ψ^ℓ bilden eine Orthonormalbasis von $\operatorname{span}\Psi^\ell$ und haben die folgende Eigenschaften:*

(i) Die Anzahl der Wavelets bis zur Stufe k ist $|\Psi^k| = 2^k$.

(ii) $\operatorname{supp}\psi_0^0 = \operatorname{supp}\psi_0^1 = [0,1]$ *und* $\operatorname{supp}\psi_i^k = \left[\frac{i}{2^{k-1}}, \frac{i+1}{2^{k-1}}\right]$

(d.h. um für $k > 0$ das Wavelet ψ_i^k zu erhalten, schrumpft man den Träger des Mother Wavelets um den Faktor 2^{k-1}, verschiebt ihn um $\frac{i}{2^{k-1}}$ nach rechts, und skaliert seinen Wert um den Faktor $2^{\frac{k-1}{2}}$)

(iii) Für $k > 0$ gilt

$$\operatorname{supp}(\psi_i^k) = \operatorname{supp}(\psi_{2i}^{k+1}) \cup \operatorname{supp}(\psi_{2i+1}^{k+1})$$

(d.h. die Wavelets bilden eine Baumstruktur bzgl. der Inklusion ihrer Träger, siehe Abbildung 3.4)

(iv) Wavelets auf Stufe k sind unstetig an den Stellen $x = \frac{i}{2^k}$ für $0 \le i \le 2^k$. Insbesondere ist ψ_i^k unstetig in $x = \frac{2i}{2^k}, \frac{2i+1}{2^k}$ und $\frac{2i+2}{2^k}$. Zwischen diesen Punkten ist ψ_i^k konstant.

(v) Die Wavelets sind zueinander orthonormal, d.h. $\langle \psi_i^k, \psi_j^\ell \rangle = \delta_{ij} \cdot \delta_{k\ell}$.

(vi) Die Basis ist hierarchisch, d.h. $\Psi^k \subset \Psi^{k+1}$.

Beweis.

(i) Nach Definition gibt es auf den Stufen $k = 0, 1$ jeweils ein Haar-Wavelet und 2^{k-1} Wavelets auf den Stufen $k > 1$. Somit ist $\dim \Psi^0 = 1$, $\dim \Psi^1 = 2 = 2^1$ und $\dim \Psi^k = \dim \Psi^{k-1} + 2^{k-1} = 2^{k-1} + 2^{k-1} = 2^k$ für $k > 1$.

(ii) Nach Definition gilt $\operatorname{supp} \psi_0^0 = \operatorname{supp} \psi_0^1 = [0, 1]$. Nun betrachte die rekursive Definitionsgleichung (3.56): aus $2^{k-1}x - i = 2^{k-1}\left(x - \frac{i}{2^{k-1}}\right)$ folgt, dass das Mother-Wavelet ψ um $\frac{i}{2^{k-1}}$ nach rechts verschoben wird und durch die Skalierung im Argument der Trägers um den Faktor 2^{k-1} kontrahiert wird. Damit ist ψ_i^k auf einem Intervall der Länge $2^{-(k-1)}$ ungleich Null, welches bei $\frac{i}{2^{k-1}}$ beginnt. Somit ist $\operatorname{supp} \psi_i^k = [\frac{i}{2^{k-1}}, \frac{i+1}{2^{k-1}}]$.

(iii) Wir setzen die Träger aus ii) ein:

$$\begin{aligned}\operatorname{supp}(\psi_{2i}^{k+1}) \cup \operatorname{supp}(\psi_{2i+1}^{k+1}) &= \left[\frac{2i}{2^k}, \frac{2i+1}{2^k}\right] \cup \left[\frac{2i+1}{2^k}, \frac{2i+2}{2^k}\right] \\ &= \left[\frac{2i}{2^k}, \frac{2i+2}{2^k}\right] = \left[\frac{i}{2^{k-1}}, \frac{i+1}{2^{k-1}}\right] \quad = \operatorname{supp}(\psi_i^k).\end{aligned}$$

(iv) Da ψ_i^k ein skaliertes und verschobenes Mother-Wavelet ist, ist die Funktion auf der ersten Hälfte des Trägers 1 und auf der zweiten Hälfte -1, d.h. die Funktion hat gerade die drei angegebenen Unstetigkeitsstellen.

(v) Zur Orthogonalität. Zunächst sei $k = \ell > 1$. Dann ist entweder $i = j$ oder die Träger sind disjunkt und somit $\langle \psi_i^k, \psi_j^\ell \rangle = 0$. Gesamt folgt $\langle \psi_i^k, \psi_j^\ell \rangle = \delta_{ij} \cdot \delta_{k\ell}$.

Wegen der Symmetrie genügt es nun $k > \ell$ anzunehmen. Die Funktionen auf Stufe ℓ sind unstetig an Punkten, die ein Vielfaches von $2^{-\ell}$ sind, also an den Endpunkten von Trägern von Stufe-k Funktionen, aber nie im Inneren deren Träger. Somit gilt

$$\langle \psi_i^k, \psi_j^l \rangle = \int_{\frac{i}{2^{k-1}}}^{\frac{i+1}{2^{k-1}}} \psi_i^k(x)\psi_j^l(x)\,dx = c \int_{\frac{i}{2^{k-1}}}^{\frac{i+1}{2^{k-1}}} \psi_i^k(x)\,dx = 0,$$

Zuletzt rechnen wir die Normierung für $i = j$ und $k = \ell$ nach:

$$\begin{aligned}k = 0 : \langle \psi_0^0, \psi_0^0 \rangle = \langle \chi, \chi \rangle &= \int_0^1 1\,dx = 1, \\ k > 0 : \langle \psi_i^k, \psi_i^k \rangle &= \int_{2^{-(k-1)} \cdot i}^{2^{-(k-1)} \cdot (i+1)} \left(2^{\frac{k-1}{2}} \underbrace{\psi(2^{k-1}x - i)}_{=\pm 1}\right)^2 dx \\ &= \int_{2^{-(k-1)} \cdot i}^{2^{-(k-1)} \cdot (i+1)} 2^{k-1}\,dx = \frac{1}{2^{k-1}} \cdot 2^{k-1} = 1.\end{aligned}$$

(vi) Folgt unmittelbar aus der Definition. □

Wie in (3.43), ist die Best-Approximation einer Funktion $f \in L^2(0,1)$ in der Haar-Waveletbasis der Stufe ℓ bzgl. der L^2-Norm:

$$g = \bar{f} + \sum_{k=1}^{\ell} \sum_{i=0}^{2^{k-1}-1} \langle f, \psi_i^k \rangle \psi_i^k. \tag{3.58}$$

wobei $\bar{f} := \langle f, \psi_0^0 \rangle = \int_0^1 f(x)\,\mathrm{d}x$.

Aber wie sehen diese Funktionen nun aus? Wir charakterisieren jetzt die mittels Haar-Wavelets darstellbaren Funktionen.

Definition 3.35. Für $\ell \in \mathbb{N}$ sei $S^\ell \subset L^2(0,1)$ der Raum aller **stückweise konstanten Funktionen** bzgl. dem Gitter $x_0^\ell, \ldots, x_{2^\ell}^\ell$ mit $x_j^\ell := \frac{j}{2^\ell}$. Die 2^ℓ Indikatorfunktionen

$$\chi_j^\ell(x) := \begin{cases} 1 & x_j^\ell < x \le x_{j+1}^\ell, \\ 0 & \text{sonst}, \end{cases}$$

der Intervalle $(x_j, x_{j+1}]$, für $0 \le j < 2^\ell$, stellen eine Basis von S^ℓ dar.

Lemma 3.36. *Die Haar-Waveletbasis der Stufe ℓ ist eine ONS für S^ℓ, d.h.*

$$S^\ell = \mathit{span}\, \Psi^\ell.$$

Beweis. Wir verwenden die Eigenschaften von Ψ^ℓ aus Lemma 3.34. Aus (iv) folgt $\Psi^\ell \subset S^\ell$ und aus (i) folgt $\dim(S^\ell) = 2^\ell = |\Psi^\ell|$. Da Ψ^ℓ ein ONS für $\operatorname{span} \Psi^\ell$ ist und die beiden Räume die selbe Dimension haben folgt $S^\ell = \operatorname{span} \Psi^\ell$. □

Ein der wichtigsten Eigenschaften von Haar-Wavelets (und von allen Arten von Wavelets) ist, dass die Umwandlung zwischen der Darstellung einer Funktion $v \in S^\ell$ in der Waveletbasis Ψ^ℓ und in der kanonischen Basis $\{\chi_j^\ell : 0 \le j < 2^\ell\}$ effizient durchführbar ist, wie das folgende Lemma zeigt (vgl. die Fast Fourier Transformation).

Lemma 3.37 (Basistransformation)**.** *Für $\ell \ge 0$ kann jedes beliebige Element von S^ℓ in der Haar-Waveletbasis und in der kanonischen Basis dargestellt werden, d.h. es existieren Koeffizienten $(\alpha_j^\ell)_{j=0}^{2^\ell-1}$ und β_0^0, $(\beta_i^k)_{i=0}^{2^{k-1}-1}$, für $k = 1, \ldots, \ell$, sodass*

$$\sum_{j=0}^{2^\ell-1} \alpha_j^\ell \chi_j^\ell \;=\; \beta_0^0 \psi_0^0 + \sum_{k=1}^{\ell} \sum_{i=0}^{2^{k-1}-1} \beta_i^k \psi_i^k\,.$$

Die folgenden Rekursionsformeln gelten für die Transformationen:

(a) Haar-Waveletbasis → Standardbasis: Man setzt $\alpha_0^0 = \beta_0^0$ und berechnet rekursiv

$$\alpha_j^k = \alpha_{\lfloor j/2 \rfloor}^{k-1} + (-1)^j \, 2^{\frac{k-1}{2}} \beta_{\lfloor j/2 \rfloor}^k \,, \quad 0 \le j < 2^k \,, \tag{3.59}$$

wobei die Stufen $k = 1, \dots, \ell$ von unten nach oben durchlaufen werden.

(b) Standardbasis → Haar-Waveletbasis: Man berechnet rekursiv

$$\beta_i^k = 2^{-\frac{\ell+1}{2}} \left(\alpha_{2i}^k - \alpha_{2i+1}^k \right), \quad \alpha_i^{k-1} = \frac{1}{2} \left(\alpha_{2i}^k + \alpha_{2i+1}^k \right), \quad 0 \le i < 2^{k-1}, \tag{3.60}$$

und $\beta_0^0 = \alpha_0^0$, wobei die Stufen $k = \ell, \dots, 1$ von oben nach unten durchlaufen werden.

Der Rechenaufwand ist für beide Richtungen $\mathcal{O}(2^k)$, also linear in der Anzahl der Basisfunktionen.

Beweis. (a) Wir berechnen die Koeffizienten rekursiv startend mit $\ell = 1$:

$\boxed{\ell = 1:}$ Aus Definition des Mother-Wavelets folgt $\psi_0^0 = \chi_0^1 + \chi_1^1$, $\psi_0^1 = \chi_0^1 - \chi_1^1$ und

$$\beta_0^0 \psi_0^0 + \beta_0^1 \psi_0^1 = \beta_0^0 (\chi_0^0 + \chi_1^1) + \beta_0^1 (\chi_0^1 - \chi_1^1) = (\beta_0^0 + \beta_0^1) \chi_0^1 + (\beta_0^0 - \beta_0^1) \chi_1^1,$$

also

$$\alpha_0^1 = \beta_0^0 + \beta_0^1, \qquad \alpha_1^1 = \beta_0^0 - \beta_0^1.$$

$\boxed{\ell - 1 \Rightarrow \ell:}$ Wir nehmen an, auf Stufe $\ell - 1$ gilt

$$\beta_0^0 \psi_0^0 + \sum_{k=1}^{\ell-1} \sum_{i=0}^{2^{k-1}-1} \beta_i^k \psi_i^k \;=\; \sum_{j=0}^{2^{\ell-1}-1} \alpha_j^{\ell-1} \chi_j^{\ell-1},$$

wobei die Koeffizienten $\alpha_j^{\ell-1}$ bereits mittels (3.59) aus den Koeffizienten β_i^k mit $k < \ell$ berechnet wurden. Dann folgt aus Lemma 3.34(ii) auf Stufe ℓ dass

$$\begin{aligned}
\beta_0^0 \psi_0^0 + \sum_{k=1}^{\ell} \sum_{i=0}^{2^{k-1}-1} \beta_i^k \psi_i^k &= \sum_{i=0}^{2^{\ell-1}-1} \alpha_i^{\ell-1} \underbrace{\chi_i^{\ell-1}}_{=\chi_{2i}^\ell + \chi_{2i+1}^\ell} + \sum_{i=0}^{2^{\ell-1}-1} \beta_i^\ell \underbrace{\psi_i^\ell}_{=2^{\frac{\ell-1}{2}}(\chi_{2i}^\ell - \chi_{2i+1}^\ell)} \\
&= \sum_{i=0}^{2^{\ell-1}-1} \left[\left(\alpha_i^{\ell-1} + 2^{\frac{\ell-1}{2}} \beta_i^\ell \right) \chi_{2i}^\ell + \left(\alpha_i^{\ell-1} - 2^{\frac{\ell-1}{2}} \beta_i^\ell \right) \chi_{2i+1}^\ell \right] \\
&= \sum_{j=0}^{2^\ell-1} \left(\alpha_{\lfloor j/2 \rfloor}^{\ell-1} + (-1)^j \, 2^{\frac{\ell-1}{2}} \beta_{\lfloor j/2 \rfloor}^\ell \right) \chi_j^k,
\end{aligned}$$

und damit die Rekursionsformel (3.59) auf Stufe ℓ.

(b) Nun die Richtung von der Standardbasis in die Waveletbasis. Auch hier gehen wir rekursiv über die Stufen vor.

Zur Herleitung der Rekursion beobachten wir für $k > 1$ und $0 \le i < 2^{k-1}$:

$$\begin{aligned} \chi_i^{k-1} &= \chi_{2i}^k + \chi_{2i+1}^k \\ \psi_i^k &= 2^{\frac{\ell-1}{2}} \left(\chi_{2i}^k - \chi_{2i+1}^k\right) \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \chi_{2i}^k &= \tfrac{1}{2}\chi_i^{k-1} + 2^{-\frac{\ell+1}{2}}\psi_i^k \\ \chi_{2i+1}^k &= \tfrac{1}{2}\chi_i^{k-1} - 2^{-\frac{\ell+1}{2}}\psi_i^k \end{aligned} .$$

Damit erhalten wir für beliebiges $1 \le k \le \ell$:

$$\begin{aligned} \sum_{j=0}^{2^k-1} \alpha_j^k \chi_j^k &= \sum_{i=0}^{2^{k-1}-1} \left(\alpha_{2i}^k \chi_{2i}^k + \alpha_{2i+1}^k \chi_{2i+1}^k\right) \\ &= \sum_{i=0}^{2^{k-1}-1} \left(\alpha_{2i}^k \left(\frac{1}{2}\chi_i^{k-1} + 2^{-\frac{\ell+1}{2}}\psi_i^k\right) + \alpha_{2i+1}^k \left(\frac{1}{2}\chi_i^{k-1} - 2^{-\frac{\ell+1}{2}}\psi_i^k\right)\right) \\ &= \sum_{i=0}^{2^{k-1}-1} \frac{1}{2}\left(\alpha_{2i}^k + \alpha_{2i+1}^k\right)\chi_i^{k-1} + \sum_{i=0}^{2^{k-1}-1} 2^{-\frac{\ell+1}{2}}\left(\alpha_{2i}^k - \alpha_{2i+1}^k\right)\psi_i^k . \end{aligned}$$

$\boxed{k \to k-1:}$ Daraus erhalten wir $1 \le k \le \ell$ die Rekursion (3.60):

$$\begin{aligned} \beta_i^k &= 2^{-\frac{\ell+1}{2}}\left(\alpha_{2i}^k - \alpha_{2i+1}^k\right), && 0 \le i < 2^{k-1}, \\ \alpha_i^{k-1} &= \frac{1}{2}\left(\alpha_{2i}^k + \alpha_{2i+1}^k\right), && 0 \le i < 2^{k-1} . \end{aligned}$$

$\boxed{k = 0:}$ Da $\Psi_0^0 = \chi = \chi_0^0$, gilt auch $\beta_0^0 = \alpha_0^0$.

Die Komplexitätsaussage folgt direkt, da für jedes i (bzw. j) nur eine konstante Anzahl von (≤ 4) arithmetische Operationen durchgeführt werden müssen. □

3.7.3 Datenkompression mit Haar-Wavelets

Die Anwendung von Gleichung (3.52) zu Beginn des Kapitels erlaubt nun eine Datenkompression mit vorgegebener Toleranz bei der Speicherung einer Funktion mit Hilfe der Haar-Wavelets.

Definition 3.38. Durch die Indexmenge $I_N \subseteq \left\{(k,i) : k \ge 0,\ 0 \le i < 2^{k-1}\right\}$ sei eine beliebige Menge von N Haar-Waveletbasisfunktionen ausgewählt. I_N heisst **nach unten abgeschlossen**, wenn für $k > 0$ aus $(k,i) \in I_N$ auch folgt, dass $(k-1, \lfloor i/2 \rfloor) \in I_N$ ist, d.h. alle "Vorfahren" eines Wavelets sind auch im Indexset. Die höchste auftretende Stufe ℓ in I_N bezeichnen wir als **Tiefe** der Indexmenge und der Waveletbasis.

Lemma 3.39. *Sei I_N eine nach unten abgeschlossene Indexmenge der Tiefe ℓ. Dann existiert eine Untermenge von $N+1$ Gitterpunkten $\tilde{x}_j^\ell \in \{x_0^\ell, \dots, x_{2^\ell}^\ell\}$, sodass $\{\psi_i^k : (k,i) \in I_N\}$ ein ONS für den Raum $\widetilde{S}^N$ der stückweise konstanten Funktionen bzgl. dem Gitter $\tilde{x}_0^\ell < \tilde{x}_1^\ell < \ldots < \tilde{x}_N^\ell$ darstellt.*

Beweis. Folgt aus der Bedingung an I_N und aus Lemma 3.34. □ MMS

Zum Abschluss geben wir in Algorithmus 3.1 nun einen sogenannten **Greedy-Algorithmus** an, um den kleinsten Unterraum $\widetilde{S}^N \subset L^2(0,1)$ zur Approximation einer Funktion $f \in L^2(0,1)$ aus Haar-Wavelets mit vorgegebener Toleranz zu finden.

Algorithmus 3.1 Greedy-Algorithmus zur Best-N-Term Approximation

Gegeben: $f \in L^2(0,1)$, Toleranz TOL.
Gesucht: $\widetilde{S}^N \subset L^2(0,1)$ mit N möglichst klein, sodass $\|f - \tilde{g}^N\| \leq \text{TOL}\,\|f\|$.

Initialisiere die Indexmenge I, z.B. $I = \{(0,0)\}$.
Berechne $s = \sum_{(k,i)\in I} \langle f, \psi_i^k \rangle^2$
while $s < (1 - \text{TOL}^2)\langle f, f\rangle$ **do**
  Wähle Kind $(k+1, j)$ von $(k,i) \in I$ mit $\langle f, \psi_j^{k+1}\rangle^2$ maximal.
  $I = I \cup \{(k+1, j)\}$ ▷ Füge Basisfunktion mit maximalem Fehler hinzu
  $s = s + \langle f, \psi_j^{k+1}\rangle^2$
end while

Wichtig. Das ist ein Beispiel eines **nichtlinearen Approximationsverfahrens** dar, d.h. der Approximationsraum $\widetilde{S}_N$ hängt von der zu approximierenden Funktion ab und wird für jedes f anders aussehen. Der Algorithmus passt den Raum an die Funktion an.

4 Numerische Integration (Quadratur)

In diesem Kapitel behandeln wir die numerische Berechnung bestimmter Integrale von Funktionen einer Variablen $f \in \mathcal{C}[a,b]$:

$$I(f) := \int_a^b f(x)\,\mathrm{d}x$$

durch sog. **Quadraturformeln** der Form

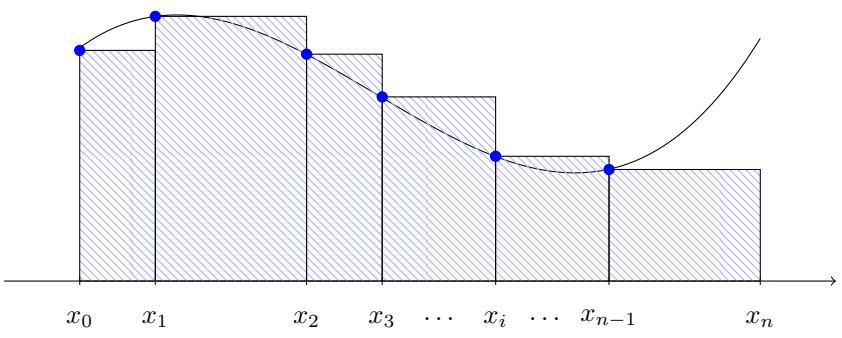
$$Q(f) := \sum_{i=0}^{n} w_i f(x_i)$$

mit **Stützstellen** $a \le x_0 < x_1 < \cdots < x_n \le b$ und **Gewichten** $w_i \in \mathbb{R}$, für $i = 0, \ldots, n$. Solche Quadraturformeln sind wichtig,

- weil es für viele Funktionen, deren Integral wichtig ist (z. B. $f(x) = \exp(-x^2)$) keine geschlossenen Formeln zur Berechnung des Integrals gibt;
- weil f oft nur an endlich vielen Punkten bekannt ist (z. B. Messreihen, oder wenn die Auswertung teuer ist);
- zur Berechnung statistischer Größen, wie Erwartungswert, Varianz etc.

Beispiel 4.1 („Summierte Rechtecksregel“)**.** Wir machen hier den Ansatz

$$I(f) \approx Q(f) := \sum_{i=0}^{n-1} (x_{i+1} - x_i) f(x_i) :$$



4.1 Interpolatorische Quadraturformeln

Eine naheliegende Konstruktionsmöglichkeit besteht darin, f durch die Lagrange-Interpolation an den Stützstellen $x_0, \ldots, x_n$ zu approximieren und dann das Interpolationspolynom exakt zu integrieren, d. h. wir setzen

$$p_n(x) = \sum_{i=0}^{n} f(x_i) L_i^{(n)}(x)$$

und dann

$$Q_n(f) \;:=\; \int_a^b p_n(x)\,\mathrm{d}x \;=\; \sum_{i=0}^{n} \underbrace{\left(\int_a^b L_i^n(x_i)\,\mathrm{d}x \right)}_{=:w_i} f(x_i)\,. \tag{4.1}$$

(Beachte, dass die Gewichte w_i nur von a, b und den Stützstellen $\{x_i\}$ abhängen.)

Aus der Interpolationsfehlerdarstellung der Lagrange-Interpolation in Satz 3.12 folgt eine allgemeine Darstellung des Quadraturfehlers in (4.1).

Satz 4.2. *Für den Quadraturfehler* $E_n^{[a,b]}(f) := I(f) - Q_n(f)$ *gilt*

$$E_n^{[a,b]}(f) \;=\; \int_a^b \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{j=0}^{n} (x - x_j)\,\mathrm{d}x$$

für ein $\xi_x \in [a, b]$.

Beweis. Folgt direkt aus Satz 3.12. □

Aus dieser Fehlerdarstellung folgt, dass $E_n^{[a,b]}(p) = 0$ für alle $p \in \mathcal{P}_n$. Es stellt sich aber heraus, dass bei geeigneter Wahl der Stützstellen auch Polynome höheren Grades mit einer $(n+1)$-Punkt-Formel integriert werden können.

Definition 4.3. Eine Quadraturformel $Q_n(f)$ mit $n+1$ Stützstellen hat **Genauigkeitsgrad** $d \in \mathbb{N}$, wenn

$$E_n^{[a,b]}(x^r) = 0 \quad \text{für} \quad 0 \le r \le d$$

und

$$E_n^{[a,b]}(x^{d+1}) \neq 0.$$

Interpolatorische Quadraturformeln der Form (4.1) mit $n+1$ Stützstellen sind deshalb *mindestens* vom Genauigkeitsgrad n.

4.2 Newton-Cotes-Formeln

Ein wichtiger Spezialfall interpolatorischer Quadraturformeln sind die auf äquidistanten Stützstellen basierenden **Newton-Cotes-Formeln**.

Wir betrachten nur den wichtigeren Fall der **abgeschlossenen** Newton-Cotes-Formeln mit $x_0 = a$ und $x_n = b$ (**offene** Newton-Cotes-Formeln verwenden nur innere Punkte, d. h. $x_0 > a$ und $x_n < b$).

Es sei also im Folgenden

$$x_i = a + ih \quad \text{für} \quad i = 0, \ldots, n \quad \text{mit} \quad h = \frac{b-a}{n}.$$

Zur Berechnung der Gewichte w_i verwenden wir die Substitution $x = a + th$, $t \in [0, n]$, sodass

$$L_i^{(n)}(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{a + th - a - jh}{a + ih - a - jh} = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{t - j}{i - j}$$

und somit (mit $\frac{\mathrm{d}x}{\mathrm{d}t} = h$)

$$w_i = \int_a^b L_i^{(n)}(x)\,\mathrm{d}x = \int_0^n \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{t-j}{i-j} h \;\mathrm{d}t \quad \text{für } i = 0, \ldots, n. \tag{4.2}$$

Diese Gewichte werden ein für allemal berechnet und tabelliert. Wie man leicht in (4.2) sieht ist $\frac{1}{b-a} w_i$ unabhängig von a und b und hängt nur von n ab.

Beispiel 4.4. (a) **Zwei Punkte:** $n = 1$ und $x_0 = a$, $x_1 = b$. Dann gilt

$$w_0 = h \int_0^1 \frac{t-1}{-1}\,\mathrm{d}t = (b-a)\left[-\frac{(t-1)^2}{2}\right]_0^1 = \frac{b-a}{2}$$
$$w_1 = h \int_0^1 \frac{t}{1}\,\mathrm{d}t = (b-a)\left[-\frac{t^2}{2}\right]_0^1 = \frac{b-a}{2}$$

ergibt die bekannte **„Trapezregel“** (siehe Abbildung 4.1a)

$$Q_1(f) = \frac{b-a}{2}\left(f(a) + f(b)\right). \tag{4.3}$$

(b) **Drei Punkte:** $n = 2$ und $x_0 = a$, $x_1 = \frac{a-b}{2}$ und $x_2 = b$.. Dann gilt

$$w_0 = h \int_0^2 \frac{(t-1)(t-2)}{(-1)(-2)}\,\mathrm{d}t = \frac{b-a}{4} \int_0^2 t^2 - 3t + 2\,\mathrm{d}t = \frac{b-a}{6}$$
$$w_1 = h \int_0^2 \frac{t(t-2)}{1(1-2)}\,\mathrm{d}t = -\frac{b-a}{2} \int_0^2 t^2 - 2t\,\mathrm{d}t = \frac{4(b-a)}{6}$$

$$w_2 = h \int_0^2 \frac{t(t-1)}{2(2-1)}\,\mathrm{d}t = \frac{b-a}{4}\int_0^2 t^2 - t\,\mathrm{d}t = \frac{b-a}{6}$$

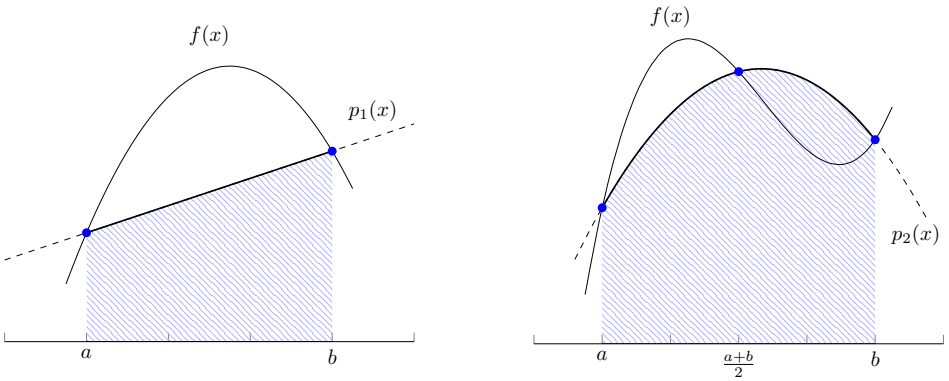
erhalten wir die sog. **„Simpson-Regel“** (siehe Abbildung 4.1b)

$$Q_2(f) = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right). \tag{4.4}$$

(c) **Vier Punkte:** Für $n = 3$ und $x_0 = a$, $x_1 = \frac{2a+b}{3}$, $x_2 = \frac{a+2b}{3}$, $x_3 = b$ kann man ähnlich wie oben die sog. **„3/8-Regel“** herleiten

MMS

$$Q_3(f) = \frac{b-a}{8}\left(f(a) + 3f\left(\frac{2a+b}{3}\right) + 3f\left(\frac{a+2b}{3}\right) + f(b)\right).$$



(a) Trapezregel $Q_1(f)$ (b) Simpsonregel $Q_2(f)$

Abbildung 4.1: Abgeschlossene Newton-Cotes-Formeln für zwei bzw. drei Punkte.

Beispiel 4.5. Es sei $f(x) = \sqrt{x}$. Wir wollen $\int_{\frac{1}{4}}^{1} f(x)\,\mathrm{d}x$ berechnen. Es ist

$$Q_1(f) = \frac{1-\frac{1}{4}}{2}\left(\sqrt{\frac{1}{4}} - 1\right) = \frac{3}{8}\cdot\frac{3}{2} = \frac{9}{16} = 0.5625$$
$$Q_2(f) = \frac{1-\frac{1}{4}}{6}\left(\sqrt{\frac{1}{4}} + 4\sqrt{\frac{5}{8}} + 1\right) \approx 0.582785$$

und

$$I(f) = \int_{\frac{1}{4}}^{1} \sqrt{x}\,\mathrm{d}x = \frac{2}{3}\left[x^{3/2}\right]_{\frac{1}{4}}^{1} = \frac{7}{12} = 0.58\overline{3}.$$

Deshalb ist der relative Fehler jeweils

$$\left|\frac{E_1^{\left[\frac{1}{4},1\right]}(f)}{I(f)}\right| = \left|\frac{\frac{7}{12} - \frac{9}{16}}{\frac{7}{12}}\right| = \frac{1}{28} = 3.57\cdot 10^{-2}$$

und

$$\left|\frac{E_2^{\left[\frac{1}{4},1\right]}(f)}{I(f)}\right| = \left|\frac{0.58\overline{3} - 0.582785}{0.58\overline{3}}\right| = 9.405 \cdot 10^{-4},$$

also ca. 38 mal kleiner mit der Simpson-Regel.

Bemerkung 4.6. Es gilt immer $\sum_{i=0}^{n} w_i = b - a$.

Wir wollen nun den Genauigkeitsgrad der Newton-Cotes Formeln betrachten (o. B. d. A. nur für $[a, b] = [0, 1]$).

Beispiel 4.7. (a) Trapezregel: $Q_1(f) = \frac{1}{2}(f(0) + f(1))$

r	x^r	$I(x^r)$	$Q_1(x^r)$	$E_1(x^r)$
0	1	1	$\frac{1}{2}(1+1) = 1$	0
1	x	$\frac{1^2}{2} = \frac{1}{2}$	$\frac{1}{2}(0+1) = \frac{1}{2}$	0
2	x^2	$\frac{1^2}{3} = \frac{1}{3}$	$\frac{1}{2}(0+1) = \frac{1}{2}$	$-\frac{1}{6}$

Der Genauigkeitsgrad beträgt also $d = 1 \ \ (= n)$.

(b) Simpsonregel: $Q_2(f) = \frac{1}{6}\left(f(0) + 4f\left(\frac{1}{2}\right) + f(1)\right)$

r	x^r	$I(x^r)$	$Q_2(x^r)$	$E_2(x^r)$
0	1	1	$\frac{1}{6}(1+4+1) = 1$	0
1	x	$\frac{1}{2}$	$\frac{1}{6}\left(0 + 4\frac{1}{2} + 1\right) = \frac{1}{2}$	0
2	x^2	$\frac{1}{3}$	$\frac{1}{6}\left(0 + 4\frac{1}{2^2} + 1\right) = \frac{1}{3}$	0
3	x^3	$\frac{1}{4}$	$\frac{1}{6}\left(0 + 4\frac{1}{2^3} + 1\right) = \frac{1}{4}$	0
4	x^4	$\frac{1}{5}$	$\frac{1}{6}\left(0 + 4\frac{1}{2^4} + 1\right) = \frac{5}{24}$	$\frac{1}{5} - \frac{5}{24} = -\frac{1}{120}$

Der Genauigkeitsgrad beträgt also $d = 3 \ \ (= n + 1)$

Diese Beobachtung lässt sich folgendermaßen verallgemeinern.

Proposition 4.8. *Für alle $n \in \mathbb{N}$ hat die (abgeschlossene) Newton-Cotes Formel Q_n den Genauigkeitsgrad*

$$d = \begin{cases} n & \text{für } n \text{ ungerade,} \\ n+1 & \text{für } n \text{ gerade.} \end{cases}$$

Aus Definition 4.3 folgt sofort folgendes Resultat:

Proposition 4.9. *Hat Q_n einen Genauigkeitsgrad d, so gilt für alle p in $\mathcal{P}_d$*

$$E_n^{[a,b]}(p) = 0$$

Beweis. Folgt direkt aus Definition 4.3 und der Linearität von $I(f)$ und $Q_n(f)$, d. h. für beliebige $f, g \in \mathcal{C}[a,b]$ und $\alpha, \beta \in \mathbb{R}$:

$$I(\alpha f + \beta g) = \alpha I(f) + \beta I(g) \qquad \text{und} \qquad Q_n(\alpha f + \beta g) = \alpha Q_n(f) + \beta Q_n(g).$$

□

Um nun eine Fehlerdarstellung für die Newton-Cotes Formeln herzuleiten, benötigen wir den folgenden Satz (siehe z.B. *Analysis I* oder *Mathe für Informatiker II*):

Satz 4.10 (Mittelwertsatz der Integralrechnung)**.** *Gegeben seien $f : [a,b] \to \mathbb{R}$ stetig und $g : [a,b] \to \mathbb{R}$ integrierbar mit $g(x) \geq 0$, für $x \in [a,b]$. Dann existiert ein $\xi \in [a,b]$, sodass*

$$\int_a^b f(x)g(x)\,\mathrm{d}x \;=\; f(\xi)\int_a^b g(x)\,\mathrm{d}x.$$

Außerdem brauchen wir:

Lemma 4.11. *Sei Q_n die Newton-Cotes Formel mit $n+1$ Stützstellen und Genauigkeitsgrad d (also $d = n+1$ für n gerade und $d = n$ für n ungerade). Dann existiert eine* **Kernfunktion (Peano-Kern)** *$K : [0,1] \to \mathbb{R}$ mit $K(x) \geq 0$ für alle $x \in [0,1]$ oder $K(x) \leq 0$ für alle $x \in [0,1]$ (kein Vorzeichenwechsel), sodass für alle $f \in \mathcal{C}^{(d+1)}[0,1]$ gilt*

$$E_n^{[0,1]}(f) \;=\; \int_0^1 f^{(d+1)}(x)K(x)\,\mathrm{d}x\,. \tag{4.5}$$

Beweis. Für den vollständigen Beweis siehe [SB, Kapitel 3]. Wir zeigen hier nur den Spezialfall $n = 1$ (Trapezregel) und wählen $K(x) = \frac{x(x-1)}{2}$.

Wie in Kapitel 3 setzen wir $e(x) = f(x) - p_1(x)$. Dann gilt

$$\begin{aligned}\int_0^1 f''(x)K(x)\,\mathrm{d}x \;&=\; \int_0^1 f''\frac{x(x-1)}{2}\,\mathrm{d}x \\ &=\; \int_0^1 e''\frac{x(x-1)}{2}\,\mathrm{d}x\end{aligned}$$

$$\stackrel{\text{P.I.}}{=} \underbrace{\left[e'(x)\frac{x(x-1)}{2}\right]_0^1}_{=0} - \int_0^1 e'(x)\left(x-\frac{1}{2}\right)\,\mathrm{d}x$$

$$\stackrel{\text{P.I.}}{=} \underbrace{\left[e(x)\left(x-\frac{1}{2}\right)\right]_0^1}_{=0 \text{ (weil } e(0)=e(1)=0)} + \int_0^1 e(x)\,\mathrm{d}x = E_1^{[0,1]}(f),$$

wobei zweimal partielle Integration (P.I.) angewendet wurde. □

Satz 4.12. *Sei Q_n die Newton-Cotes Formel mit $n+1$ Stützstellen und Genauigkeitsgrad d (also $d = n+1$ für n gerade und $d = n$ für n ungerade). Dann gibt es ein $\xi \in [0,1]$, sodass für alle $f \in \mathcal{C}^{(d+1)}[0,1]$ gilt*

$$E_n^{[0,1]}(f) \;=\; \frac{E^{[0,1]}(x^{d+1})}{(d+1)!} f^{(d+1)}(\xi)\,. \tag{4.6}$$

Beweis. Lemma 4.11 garantiert die Existenz einer Funktion $K : [0,1] \to \mathbb{R}$ mit $K(x) = |K(x)|$ oder $K(x) = -|K(x)|$ für alle $x \in [0,1]$, sodass

$$\begin{aligned} E_n^{[0,1]}(f) &= \pm \int_0^1 |K(x)| f^{(d+1)}(x)\,\mathrm{d}x \\ &= \left(\pm \int_0^1 |K(x)|\,\mathrm{d}x\right) f^{(d+1)}(\xi) = \left(\int_0^1 K(x)\,\mathrm{d}x\right) f^{(d+1)}(\xi)\,, \end{aligned} \tag{4.7}$$

wobei wir im zweiten Schritt Satz 4.10 verwendet haben. Speziell gilt für $f(x) = x^{d+1}$ und für alle $\xi \in \mathbb{R}$, dass

$$f^{(d+1)}(\xi) = (d+1)!\,.$$

Damit folgt aus (4.7), dass

$$E_n^{[0,1]}\left(x^{d+1}\right) \;=\; \left(\int_0^1 K(x)\,\mathrm{d}x\right)(d+1)!$$

bzw.

$$\int_0^1 K(x)\,\mathrm{d}x \;=\; \frac{E_n^{[0,1]}\left(x^{(d+1)}\right)}{(d+1)!}$$

woraus nach Substitution in (4.7) das Resultat folgt. □

Korollar 4.13. *Es existiert ein $\eta \in [a,b]$, sodass*

$$E_n^{[a,b]}(f) = (b-a)^{d+2}\left(\frac{E_n^{[0,1]}\left(x^{d+1}\right)}{(d+1)!}\right) f^{(d+1)}(\eta) \tag{4.8}$$

für alle $f \in \mathcal{C}^{(d+1)}[a,b]$.

Beweis. Übungsaufgabe unter Verwendung der Substitution

MMS

$$g(t) = (b-a)f\left(a + (b-a)t\right), \quad t \in [0,1].$$

□

Beispiel 4.14. Nach obigem Korollar und Beispiel 4.7 existiert (jeweils) ein $\eta \in [a,b]$, sodass für die

(a) Trapezregel auf $[a,b]$ gilt:

$$E_1^{[a,b]}(f) = (b-a)^3 \left(\frac{E_1^{[0,1]}\left(x^2\right)}{2!}\right) f''(\eta) = -\frac{(b-a)^3}{12} f''(\eta).$$

(b) Simpsonregel auf $[a,b]$ gilt:

$$E_2^{[a,b]}(f) = (b-a)^5 \left(\frac{E_2^{[0,1]}\left(x^4\right)}{4!}\right) f^{(4)}(\eta) = -\frac{(b-a)^5}{2880} f^{(4)}(\eta).$$

Erhöhen des Polynomgrades zur Verbesserung des Quadraturfehlers ergibt jedoch keinen Sinn:

1. Man benötigt dafür mehr und mehr Ableitungen.
2. Wir haben die Probleme mit der fehlenden punktweisen Konvergenz. der Lagrange-Interpolation bereits ausführlich diskutiert (z. B. Runge-Funktion).
3. Ab $n = 7$ treten negative Gewichte auf — Gefahr der Auslöschung.
4. Es gibt eine sehr einfache und effiziente Alternative:

4.3 Summierte Newton-Cotes Formeln

Idee: Um das Integral $\int_a^b f(x)\,\mathrm{d}x$ zu berechnen, unterteilen wir das Intervall $[a,b]$ in die Teilintervalle $[y_j, y_{j+1}]$, $j = 0, \ldots, N-1$:

und verwenden eine **fixe** Newton-Cotes Formel niedriger Ordnung, z. B. $n = 1$ oder $n = 2$, definiert und vorberechnet für das Einheitsintervall $[0,1]$, d. h.

$$Q_n^{[0,1]}(f) = \sum_{i=0}^{n} \hat{w} f(\hat{x}_i) \approx \int_0^1 f(x)\,\mathrm{d}x,$$

auf jedem der Teilintervalle, zum Beispiel (aber nicht notwendigerweise) äquidistant:

$$y_j = a + jh, \quad j = 0, \ldots, N, \quad \text{mit } h = \frac{b-a}{N}.$$

Es gilt damit

$$\int_a^b f(x)\,\mathrm{d}x = \int_{y_0}^{y_1} f(x)\,\mathrm{d}x + \int_{y_1}^{y_2} f(x)\,\mathrm{d}x + \cdots + \int_{y_{N-1}}^{y_N} f(x)\,\mathrm{d}x\,.$$

Durch die Transformation $x = y_j + h_j\hat{x}$, für $\hat{x} \in [0,1]$, mit $h_j := y_{j+1} - y_j$, können wir aus $Q_n^{[0,1]}(f)$ eine Quadraturformel $Q_n^{[y_j,y_{j+1}]}(f)$ für $\int_{y_j}^{y_{j+1}} f(x)\,\mathrm{d}x$ definieren und erhalten die summierte Formel

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{j=0}^{N-1} Q_n^{[y_j,y_{j+1}]}(f) = \sum_{j=0}^{N-1} h_j \left(\sum_{i=0}^{n} \hat{w}_i f(y_j + h_j\hat{x}_i) \right). \tag{4.9}$$

Beispiel 4.15. (a) Summierte Trapezregel ($n = 1$):

$$Q_{1,N}^{[a,b]}(f) := \sum_{j=0}^{N-1} Q_1^{[y_j,y_{j+1}]}(f) = \sum_{j=0}^{N-1} \frac{h_j}{2} \left(f\left(y_j\right) + f\left(y_{j+1}\right) \right). \tag{4.10}$$

Im Falle eines uniformen Gitters mit $h = \frac{b-a}{N}$:

$$Q_{1,N}^{[a,b]}(f) := h \left(\frac{f(a)}{2} + \sum_{j=1}^{N-1} f(y_j) + \frac{f(b)}{2} \right).$$

(b) Summierte Simpson-Regel ($n = 2$):

$$\begin{aligned} Q_{2,N}^{[a,b]}(f) &:= \sum_{j=0}^{N-1} Q_1^{[y_j,y_{j+1}]}(f) \\ &= \sum_{j=0}^{N-1} \frac{h_j}{6} \left(f(y_j) + 4f\left(\frac{y_j + y_{j+1}}{2} \right) + f(y_{j+1}) \right), \end{aligned} \tag{4.11}$$

sowie auf dem uniformen Gitter:

$$Q_{2,N}^{[a,b]}(f) := \frac{h}{6} \left(f(a) + 2\sum_{j=1}^{N-1} f(y_j) + 4\sum_{j=0}^{N-1} f\left(\frac{y_j + y_{j+1}}{2} \right) + f(b) \right).$$

Aus Korollar 4.13 folgt sofort:

Korollar 4.16. *Sei $f \in \mathcal{C}^{d+1}[a,b]$. Dann existieren $\eta_j \in [y_j, y_{j+1}]$, $j = 0, \ldots, N-1$, sodass*

$$E_{n,N}^{[a,b]}(f) := I(f) - Q_{n,N}^{[a,b]}(f) = \left(\frac{E_n^{[0,1]}(x^{d+1})}{(d+1)!} \right) \sum_{j=0}^{N-1} h_j^{d+2} f^{(d+1)}(\eta_j),$$

(wobei d wieder der Genauigkeitsgrad von $Q_n^{[0,1]}$ ist), und

$$\left|E_{n,N}^{[a,b]}(f)\right| \leq Ch^{d+1}$$

mit

$$h := \max_{j=0}^{N-1} h_j \quad \text{und} \quad C := (b-a)\left(\frac{E_n^{[0,1]}(x^{d+1})}{(d+1)!}\right)\left\|f^{(d+1)}\right\|_{\infty,[a,b]}.$$

Beispiel 4.17. Für die summierte Trapez- bzw. Simpsonregel erhalten wir

$$\left|E_{1,N}^{[a,b]}(f)\right| \leq \frac{b-a}{12}\|f''\|_{\infty,[a,b]}\, h^2$$

und

$$\left|E_{2,N}^{[a,b]}(f)\right| \leq \frac{b-a}{2880}\left\|f^{(4)}\right\|_{\infty,[a,b]}\, h^4. \tag{4.12}$$

Wir diskutieren nun praktische Aspekte der in der Praxis wichtigen Simpson-Regel:

1. Die **a priori** Abschätzung (4.12) ist in der Praxis nur bedingt nützlich, weil es meist schwer ist die vierte Ableitung abzuschätzen. Wir können den Fehler aber näherungsweise aus den tatsächlich berechneten Werten durch eine sogenannte **a posteriori** Fehlerabschätzung bestimmen:

   Nehmen wir an $f \in \mathcal{C}^{(4)}[a,b]$. Dann gilt wegen (4.12), dass

   $$I(f) = Q_{2,N}(f) + \omega_f h^4 + \mathcal{O}(h^5).$$

   Zur Bestimmung von ω_f betrachten wir die Quadraturformel mit halber Schrittweite $\frac{h}{2}$ und $2N$ Teilintervallen, d.h. $Q_{2,2N}(f)$, für die gilt

   $$I(f) = Q_{2,2N}(f) + \omega_f \left(\frac{h}{2}\right)^4 + \mathcal{O}(h^5),$$

   woraus folgt, dass

   $$Q_{2,2N}(f) - Q_{2,N}(f) = \omega_f \left(h^4 - \left(\frac{h}{2}\right)^4\right) + \mathcal{O}(h^5),$$

   bzw.

   $$\begin{aligned}\omega_f &= h^{-4}\left(\frac{Q_{2,2N}(f) - Q_{2,N}(f)}{1-2^{-4}} + \mathcal{O}(h^5)\right)\\ &= \frac{16}{15}\,\frac{Q_{2,2N}(f) - Q_{2,N}(f)}{h^4} + \mathcal{O}(h)\end{aligned}$$

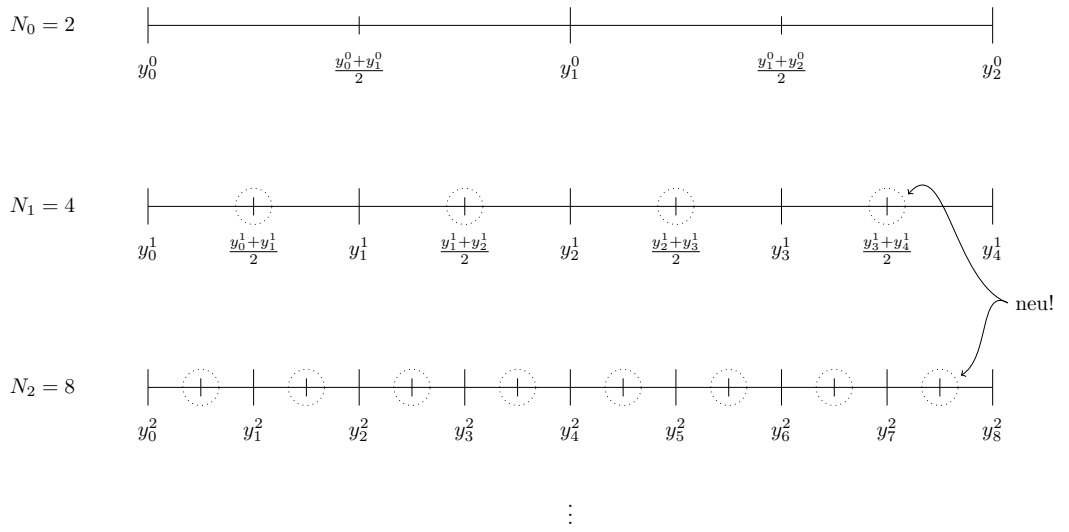


Abbildung 4.2: Darstellung der neu hinzukommenden Stellen, an denen die Funktion ausgewertet werden muss.

Wenn wir also eine Fehlertoleranz TOL vorgegeben haben, können wir mit N_0 Teilintervallen starten und so lange die Anzahl der Intervalle verdoppeln, d. h.

$$N_j = 2N_{j-1}, \quad j = 1, 2, \ldots,$$

bis

$$\frac{16}{15} \left| Q_{2,N_j}(f) - Q_{2,N_{j-1}}(f) \right| \ \leq \ \text{TOL}.$$

Dann gilt

$$\left| E_{2,N_j}(f) \right| \ \leq \ \text{TOL} + \mathcal{O}(h^5)$$

und die dazu nötige Schrittweite erfüllt

$$h_j = \left(\frac{\text{TOL}}{\omega_f} \right)^{1/4}$$

(aber ohne das ω_f a priori bekannt sein muss).

Bei äquidistanten Unterteilungen können die bisherigen Funktionsauswertungen alle wiederverwendet werden und es muss nur an den Intervallmittelpunkten neu ausgewertet werden (siehe Abbildung 4.2)

2. Oft ist das Verhalten der vierten Ableitung auf verschiedenen Teilen von $[a, b]$ sehr unterschiedlich. Hat man dazu a priori Informationen (wie z. B. bei $f(x) = \sqrt{x}$ auf $[0, 1]$), so kann man die $\{y_i\}$ a priori geeignet (d. h. **nicht uniform**) wählen.

3. Man kann aber auch adaptiv durch a posteriori Fehlerabschätzung (wie oben) auf jedem Teilintervall den Fehler mittels **a posteriori Fehlerschätzer** $\eta_{f,j}^k$

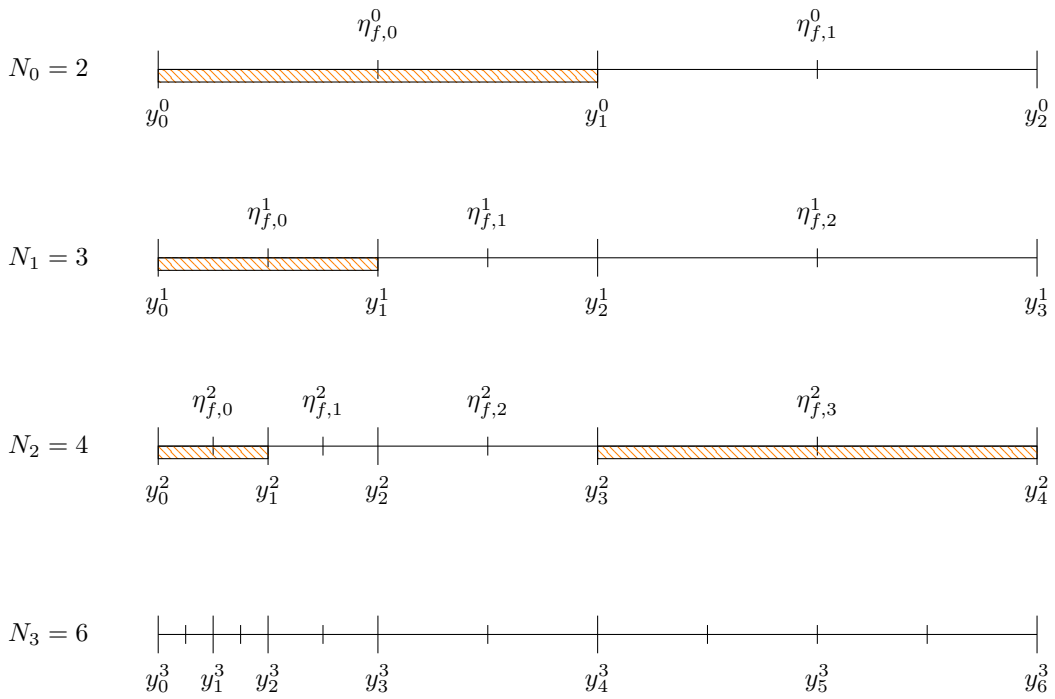


Abbildung 4.3: Adaptive summierte Simpsonregel (die schraffierten Flächen entsprechen den Intervallen, die verfeinert werden sollen).

abschätzen und im $k$ten Schritt nur die Intervalle $[y_j^k, y_{i+1}^k]$ verfeinern, wo der Fehler groß ist, was zur **adaptiven summierten Simpson-Regel** führt. Diese ist bspw. in der Matlab-Funktion `quad` implementiert. Wir wählen also z. B. die 30% der Intervalle, wo der a-posteriori Fehlerschätzer $\eta_{f,j}^k$ am größten ist (siehe Abbildung 4.3). Es gilt

$$\eta_f^k = \sum_{j=0}^{N_k-1} \eta_{f,j}^k$$

und so liefert dieser adaptive Algorithmus natürlich auch gleich wieder einen globalen Fehlerschätzer zur Bestimmung des Abbruchkriteriums.

4.4 Das Romberg-Verfahren

Wir können wieder das Prinzip der **Extrapolation zum Limes** $h = 0$ verwenden, um die summierte Newton-Cotes Formeln zu verbessern. Wir beschränken uns auf den Fall $n = 1$ (Trapezregel) und äquidistante $y_j = a + jh,\ j = 0, \ldots, N$ mit $h = \frac{b-a}{N}$. Dieses Integrationsverfahren geht auf Romberg zurück.

Wir definieren

$$a(h^2) := h\left(f(a) + \sum_{j=1}^{N-1} f(y_j) + f(b)\right).$$

Wir wissen bereits, dass

$$a(h^2) = I(f) - h^2\left(\frac{b-a}{12}\right) f''(\eta)$$

(Mittelwertsatz angewandt auf die Formel in Korollar 4.16).

Die Grundlage der Berechnung von $\lim_{h^2\to 0} a(h^2)$ durch Extrapolation ist wieder eine asymptotische Entwicklung von $a(h^2)$ nach Potenzen der Gitterweite h. Da die Koeffizienten der ungerade Potenzen 0 sind, führen wir diese Entwicklung (wie in Kapitel 3.3 aber) bzgl. $\widetilde{h} := h^2$ durch.

Satz 4.18. *Für $f \in \mathcal{C}^{2m+2}[a,b]$ gilt die sogenannte* **„Euler-Maclaurinsche Summenformel“**

$$\begin{aligned} a(h^2) = \int_a^b f(x)\,\mathrm{d}x + \sum_{k=1}^{m} h^{2k}\frac{B_{2k}}{(2k)!}\left(f^{(2k-1)}(b) - f^{(2k-1)}(a)\right) \\ + h^{2m+2}\frac{b-a}{(2m+2)!}B_{2m+2}f^{(2m+2)}(\eta) \end{aligned} \tag{4.13}$$

für ein $\eta \in [a,b]$, wobei B_l die sog. **Bernoulli-Zahlen** *sind, die der Rekursion*

$$B_0 = 1, \quad B_l = -\sum_{j=0}^{l-1}\binom{l}{j}\frac{B_j}{l-j+1}, \quad l = 1, 2, \cdots$$

genügen (mit $B_l = 0$ für $l > 1$ ungerade).

Beweis. Siehe [SB, Kapitel 3.3]. □

Die ersten Bernoulli-Zahlen sind

$$B_0 = 1,\ B_1 = -\frac{1}{2},\ B_2 = \frac{1}{6},\ B_4 = -\frac{1}{30},\ B_6 = \frac{1}{42}, \cdots$$

Dem Extrapolationsprinzip folgend funktioniert das Romberg-Verfahrens wie folgt:

1. Für die Folge von Schrittweiten $h_i = 2^{-i}h, \quad i = 0, \ldots, m$ berechnen wir

$$a(h_i^2) = Q_{1,N_i}^{[a,b]}(f) \quad \text{mit} \quad N_i = \frac{b-a}{h_i}.$$

2. Mit den Stützpunkten $\left(\widetilde{h}_i, a(\widetilde{h}_i)\right) = (h_i^2, a(h_i^2))$ konstruieren wir das Extrapolationstableau wie in Kapitel 3.3:

i	$\widetilde{h}_i = h_i^2$	$a_{i,0} = a(h_i^2)$						
0	h_0^2	$a_{0,0}$						
1	h_1^2	$a_{1,0}$	$\searrow$ $\rightarrow$	$a_{1,1}$				
2	h_2^2	$a_{2,0}$	$\searrow$ $\rightarrow$	$a_{2,1}$	$\searrow$ $\rightarrow$	$a_{2,2}$		
$\vdots$	$\vdots$	$\vdots$		$\vdots$			$\ddots$	
m	h_m^2	$a_{m,0}$	$\searrow$ $\rightarrow$	$a_{m,1}$	$\searrow$ $\rightarrow$	$a_{m,2}$	$\cdots$	$a_{m,m}$

wobei wieder

$$a_{i,k} = \frac{\widetilde{h}_{i-k} a_{i,k-1} - \widetilde{h}_i a_{i-1,k-1}}{\widetilde{h}_{i-k} - \widetilde{h}_i}.$$

Aus Satz 3.16 (mit $\widetilde{h} = h^2$ anstelle von h) folgt dann direkt

Satz 4.19. *Sei $f \in \mathcal{C}^{2m+2}[a,b]$. Dann gilt für die durch das Romberg-Verfahren berechnete Näherung $a_{m,m}$, dass*

$$\int_a^b f(x)\,\mathrm{d}x - a_{m,m} = \mathcal{O}(\widetilde{h}^{m+1}) = \mathcal{O}(h^{2m+2}). \tag{4.14}$$

Bemerkung 4.20 (Trapezregel für periodische Funktionen).

(a) Sei $f \in \mathcal{C}^{2m+2}(-\infty,\infty)$ periodisch auf $[a,b]$. Weil $f(a) = f(b)$, reduziert sich in diesem Fall die Trapezregel zur **summierten Rechtecksregel**

$$Q_{1,N}^{[a,b]}(f) = h \sum_{j=0}^{N-1} f(y_j) \qquad \text{(der einfachsten Quadraturregel).}$$

Da außerdem für alle $0 \le k \le 2m+2$ gilt $f^{(k)}(a) = f^{(k)}(b)$, ergibt sich aus Satz 4.18 sofort (ohne Extrapolation):

$$\left| E_{1,N}^{[a,b]}(f) \right| = \mathcal{O}\left(h^{2m+2}\right).$$

(b) Für den Spezialfall $f \in \mathcal{C}^\infty(-\infty,\infty)$ und f periodisch auf $[a,b]$, konvergiert $Q_{1,N}^{[a,b]}(f)$ schneller gegen $\int_a^b f(x)\mathrm{d}x$ als jede Potenz von h.

Aber Achtung! Diese Methode ist höchst sensibel auf kleinste Datenfehler, z. B. wenn b nur leicht gestört ist (bspw. $b \to b + 0.0001$), reduziert sich die Konvergenz sofort wieder zu $\mathcal{O}(h^2)$ und es kann sogar zu katastrophaler Auslöschung kommen.

4.5 Gaußsche Quadraturformeln

Eine zweite Alternative den Genauigkeitsgrad und die Konvergenzordnung interpolatorischer Quadraturformeln

$$Q_n(f) = \int_a^b p_n^f(x)\,\mathrm{d}x = \sum_{i=0}^{n} w_i f(x_i) \tag{4.15}$$

zu erhöhen ist eine gezielte Wahl der $n+1$ Stützstellen $x_0, \ldots, x_n \in [a, b]$.

Wir haben bereits in Definition 4.3 gesehen, dass $Q_n(f)$ mindestens einen Genauigkeitsgrad von n hat. Der folgende Satz beantwortet die Frage des maximalen Genauigkeitsgrads für beliebige Stützstellen $x_0, \ldots, x_n$:

Satz 4.21. *Der Genauigkeitsgrad einer interpolatorischen Quadraturformel ist kleiner oder gleich* $2n+1$.

Beweis. Nehmen wir an, die Formel $Q_n(f)$ hat Genauigkeitsgrad $d \geq 2n+2$. Dann müsste insbesondere

$$E_n(p) = I(p) - Q_n(p) = 0 \qquad \text{für} \qquad p(x) = \prod_{i=0}^{n} (x - x_i)^2 \in \mathcal{P}_{2n+2}.$$

Aber

$$0 = \sum_{i=0}^{n} w_i \underbrace{p(x_i)}_{=0} = \int_a^b p(x)\,\mathrm{d}x > 0,$$

was auf einen Widerspruch führt. Also ist $d \leq 2n+1$ □

Wir wollen nun zeigen, dass es tatsächlich interpolatorische Quadraturformeln gibt, die maximalen Genauigkeitsgrad $d = 2n+1$ haben. Der Einfachheit halber arbeiten wir in diesem Kapitel zuerst mit $[a, b] = [-1, 1]$ (wie in Kapitel 3.6 zur Gauß-Approximation).

Lemma 4.22. *Sei* $w_{n+1}(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$. *Die Regel* (4.15) *hat genau dann Genauigkeitsgrad* $d = 2n+1$, *wenn für alle* $q \in \mathcal{P}_n$ *gilt, dass*

$$\int_{-1}^{1} w_{n+1}(x) q(x)\,\mathrm{d}x = 0. \tag{4.16}$$

Beweis. Regel (4.15) hat Genauigkeitsgrad $d > n$ genau dann, wenn $E_n(x^{d+1}) \neq 0$ und wenn für alle $f \in \mathcal{P}_d$ und das zugehörige Interpolationspolynom $p_n^f \in \mathcal{P}_n$ zu den

Stützstellen $x_0, \dots, x_n$ gilt

$$\int_{-1}^{1} f(x) - p_n^f(x)\,\mathrm{d}x = 0 \tag{4.17}$$

Es sei $f \in \mathcal{P}_d$ beliebig und $p_n^f \in \mathcal{P}_n$ das zugehörige Interpolationspolynom. Dann gilt $f - p_n^f \in \mathcal{P}_d$ und aus den Interpolationsbedingungen

$$(f - p_n^f)(x_i) = 0, \quad i = 0, \dots, n.$$

folgt durch $(n+1)$-maliges Anwenden des Restpolynomsatzes (siehe Beweis zu Satz 3.8):

$$f - p_n^f = w_{n+1}\, r \quad \text{für ein } r \in \mathcal{P}_{d-n-1}. \tag{4.18}$$

Da $f \in \mathcal{P}_d$ beliebig war, erhält man auf diese Weise alle $r \in \mathcal{P}_{d-n-1}$ und so ergibt sich durch Substitution, dass (4.17) äquivalent ist zu

$$\int_{-1}^{1} w_{n+1}(x) r(x)\,\mathrm{d}x = 0, \quad \text{für alle } r \in \mathcal{P}_{d-n-1}.$$

Für $d = 2n+1$ ist das genau die Bedingung (4.16) und es folgt aus Lemma 4.21, dass $E_n(x^{d+1}) \neq 0$, womit beide Bedingungen dafür, dass Q_n Genauigkeitsgrad d hat, erfüllt sind. □

Die Suche nach einer Regel (4.15) mit Genauigkeitsgrad $d = 2n+1$ ist daher äquivalent zur Suche nach einem Polynom w_{n+1}, das Bedingung (4.16) erfüllt. Dazu muss w_{n+1} nicht notwendigerweise in faktorisierter Form, wie in Lemma 4.22, vorliegen, aber der führende Koeffizient (d. h. vor x^{n+1}) muss ungleich 0 sein (o.B.d.A. gleich 1).

Bevor wir das allgemein untersuchen, betrachten wir zwei Beispiele:

Beispiel 4.23. (a) **Ein Punkt** ($n = 0$). Es ist $w_{n+1}(x) = x - a$ für ein zu bestimmendes $a \in \mathbb{R}$. Weil $d = 2n+1 = 1$ und $n = 0$ reduziert sich (4.16) auf

$$0 = \int_{-1}^{1} (x - a)\,\mathrm{d}x = \left[\frac{x^2}{2} - ax\right]_{-1}^{1} = -2a,$$

also $w_{n+1}(x) = x$ und $x_0 = 0$. Das heißt die einzige 1-Punkt Regel mit Genauigkeitsgrad $d = 1$ ist die **Mittelpunktsformel** (oder **1-Punkt Gauß-Formel**)

$$Q_{G,0}^{[-1,1]}(f) := \int_{-1}^{1} p_0(x)\,\mathrm{d}x = 2f(0) \tag{4.19}$$

wobei p_0 die konstante Interpolierende an $x_0 = 0$ ist.

(b) **Zwei Punkte** ($n = 1$). Hier ist $w_2(x) = x^2 + ax + b$ für gewisse $a, b \in \mathbb{R}$ und (4.16) ist äquivalent zu

$$\int_{-1}^{1} \left(x^2 + ax + b\right) x^j \, \mathrm{d}x = 0, \quad \text{für } j = 0, 1.$$

Weil für jedes ungerade $n \in \mathbb{N}$ gilt $\int_{-1}^{1} x^n = 0$, erhalten wir einfach

$$\left[\frac{x^3}{3} + bx\right]_{-1}^{1} = 2\left(\frac{1}{3} + b\right) = 0 \qquad \Rightarrow b = -\frac{1}{3}$$
$$\left[a\frac{x^3}{3}\right]_{-1}^{1} = \frac{2}{3}a = 0 \qquad \Rightarrow a = 0$$

und deshalb

$$w_{n+1}(x) = x^2 - \frac{1}{3} = \left(x + \frac{1}{\sqrt{3}}\right)\left(x - \frac{1}{\sqrt{3}}\right),$$

also

$$x_0 = -\frac{1}{\sqrt{3}} \quad \text{und} \quad x_1 = \frac{1}{\sqrt{3}}.$$

Die einzige 2-Punkt Regel mit Genauigkeitsgrad $d = 2n + 1 = 3$ (d. h. exakt für $1, x, x^2, x^3$) ist die **2-Punkt Gauß-Formel**

$$Q_{G,1}^{[-1,1]}(f) := \int_{-1}^{1} p_1(x) \, \mathrm{d}x = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right), \tag{4.20}$$

d. h. $w_0 = w_1 = 1$, was man wie üblich durch Integration des interpolierenden Polynoms $p_1(x)$ an $x_0 = -\frac{1}{\sqrt{3}}$ und $x_1 = \frac{1}{\sqrt{3}}$ erhält.

MMS

Für den allgemeineren Fall blicken wir zurück auf die Gauß-Bestapproximation im Kapitel 3.6. Wie dort betrachten wir den Funktionenraum $\mathcal{C}[-1, 1]$ mit L_2-Skalarprodukt und L_2-Norm

$$\langle f, g \rangle := \int_{-1}^{1} f(x)g(x) \, \mathrm{d}x \quad \text{und} \quad \|f\| := \langle f, f \rangle^{1/2}.$$

Das Polynom $w_{n+1} \in \mathcal{P}_{n+1}$ in Lemma 4.22 soll orthogonal (bzgl. $\langle \cdot, \cdot \rangle$) auf alle Polynome $q \in \mathcal{P}_n$ stehen. Dafür können wir uns wieder der bereits besprochenen (nicht-normalisierten) **Legendre-Polynome** aus Beispiel 3.28 bedienen. Das bereits erwähnte **Gram-Schmidt Verfahren** zu deren Konstruktion ergibt, angewandt auf die Monombasis $\{1, x, \ldots, x^{n+1}\}$ von $\mathcal{P}_{n+1}$, die Formel:

$$p_0(x) := 1 \quad \text{und} \quad p_k(x) := x^k - \sum_{i=0}^{k-1} \frac{\langle x^k, p_i \rangle}{\|p_i\|^2} p_i(x), \quad k = 1, \ldots, n+1. \tag{4.21}$$

Eine notwendige Konsequenz dieser speziellen Konstruktion ist:

$$\{p_0, p_1, \ldots, p_k\} \text{ ist Basis von } \mathcal{P}_k \text{ für alle } k \leq n+1, \tag{4.22}$$

und $\langle p_j, p_k \rangle = 0$ für alle $j \neq k$. Daraus folgt speziell, dass für alle $q \in \mathcal{P}_n$ gilt:

$$\int_{-1}^{1} p_{n+1}(x) q(x) \, \mathrm{d}x = 0 \tag{4.23}$$

und wir können $w_{n+1} = p_{n+1}$ setzen. Aufrgund der Orthogonalität der Legendre Polynome folgt auch, dass w_{n+1} eindeutig ist.

Wir benötigen aber noch die Nullstellen von p_{n+1} um die Stützstellen $x_0, \ldots, x_n$ der Quadraturformel 4.15 zu definieren.

Satz 4.24. *Alle Nullstellen des Legendre-Polynoms p_{n+1}, $n \geq 0$, definiert in (4.21), sind einfach und liegen im offenen Intervall $(-1, 1)$.*

Beweis. Es seien $-1 < x_0 < \cdots < x_{k-1} < 1$ die Nullstellen von p_{n+1}, an denen das Vorzeichen wechselt (d.h. die reellen Nullstellen ungerader Vielfachheit in $(-1, 1)$). Falls diese Menge leer ist, gilt $k = 0$. (Wir zeigen: $k = n + 1$).

Unter der Annahme, dass $k < n + 1$, sei

$$q(x) = \begin{cases} \prod_{j=0}^{k-1} (x - x_j), & \text{falls } k > 0, \\ 1, & \text{falls } k = 0. \end{cases}$$

Da $q \in \mathcal{P}_k$, folgt aus (4.23) $\langle p_{n+1}, q \rangle = 0$. Nach Definition von q ändert das Produkt $p_{n+1}(x)q(x)$ auf $(-1, 1)$ sein Vorzeichen nicht, sodass

$$0 = \langle p_{n+1}, q \rangle = \int_{-1}^{1} p_{n+1}(x) q(x) \, \mathrm{d}x \neq 0,$$

was einen Widerspruch darstellt. Also ist $k = n + 1$, d. h. alle Nullstellen sind reell, einfach und liegen in $(-1, 1)$. □

Lemma 4.25. *Für beliebige $-1 \leq x_0 < x_1 < \cdots < x_n \leq 1$ ist die $(n+1) \times (n+1)$-Matrix*

$$A = \begin{pmatrix} p_0(x_0) & \cdots & p_0(x_n) \\ \vdots & \ddots & \vdots \\ p_n(x_0) & \cdots & p_n(x_n) \end{pmatrix} \tag{4.24}$$

nicht-singulär.

Beweis. Angenommen A wäre singulär. Dann existiert ein $0 \neq c \in \mathbb{R}^{n+1}$, sodass $c^\intercal A = 0$, d.h. das Polynom

$$q := \sum_{i=0}^{n} c_i p_i \in \mathcal{P}_n$$

hat $n+1$ unterschiedliche Nullstellen $x_0 < x_1 < \cdots < x_n$ in $[-1,1]$. Damit ist also $q|_{(-1,1)} \equiv 0$. Aber für alle $j = 0, \ldots, n$ gilt

$$\langle q, p_k \rangle = \sum_{i=0}^{n} c_i \langle p_i, p_k \rangle = c_k \|p_k\|^2 ,$$

woraus folgt, dass $c = 0$, und damit ein Widerspruch zu unserer Annahme. □

Wir kommen nun zum Hauptresultat des Kapitels:

Satz 4.26 (Gauß-Quadratur)**.** *Die interpolatorische Quadraturformel Q_n in* (4.15) *hat* **genau dann** *Genauigkeitsgrad $d = 2n+1$,* **wenn** *$x_0, \ldots, x_n$ die Nullstellen des $(n+1)$-ten Legendre-Polynoms p_{n+1} sind. Die zugehörigen Gewichte $w = (w_0, \ldots, w_n)^\intercal$ erhält man als Lösung des linearen Gleichungssystems*

$$Aw = (2, 0, \ldots, 0)^\intercal \tag{4.25}$$

mit A definiert in (4.24)*. Außerdem gilt $w_i > 0$ für alle $i = 0, \ldots, n$.*

Beweis. Nach Satz 4.24 sind die Nullstellen des $(n+1)$-ten Legendre Polynoms p_{n+1} unterschiedliche reelle Zahlen $x_0 < \ldots < x_n$ in $(-1,1)$. Deshalb gilt $p_{n+1}(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$ und p_{n+1} ist auch das einzige Polynom in $\mathcal{P}_{n+1}$, das die Orthogonalitätsbedingung (4.23) erfüllt.

Zusammenfassend folgt aus Lemma 4.22 (mit der Wahl $w_{n+1} = p_{n+1}$, dass die Quadraturformel Q_n genau dann Genauigkeitsgrad $d = 2n+1$ hat, wenn $x_0, \ldots, x_n$ die Nullstellen des $(n+1)$-ten Legendre-Polynoms p_{n+1} sind.

Zur Berechnung der Gewichte beachte man zunächst, dass wegen der Exaktheit von Q_n für Polynome

$$\sum_{i=0}^{n} w_i p_k(x_i) = \int_{-1}^{1} p_k(x)\,\mathrm{d}x = \langle p_k, p_0 \rangle = \begin{cases} 2, & k = 0, \\ 0, & 1 \leq k \leq n, \end{cases}$$

d. h. die Gewichte müssen (4.25) erfüllen. Mit Lemma 4.25 folgt auch die eindeutige Lösbarkeit von (4.25).

Schließlich gilt mit $f = \left(L_k^{(n)}\right)^2 \in \mathcal{P}_{2n}$, dass

$$w_k = \sum_{i=0}^{n} w_i \left(L_k^{(n)}(x_i)\right)^2 = \int_{-1}^{1} \left(L_k^{(n)}(x)\right)^2 \mathrm{d}x > 0, \quad \text{für } \; k = 0, \ldots, n$$

(wobei $L_k^{(n)}$ die Lagrange-Basispolynome sind). □

Bemerkung 4.27. Für Integrale $\int_a^b f(x)\,\mathrm{d}x$ über von $[-1,1]$ verschiedenen Intervallen $[a,b]$ können wir einfach die Transformation $\varphi : [-1,1] \to [a,b]$ mit

$$\widehat{x} = \varphi(x) := c + hx, \quad \text{wobei } \; c := \frac{b+a}{2} \; \text{ und } \; h := \frac{b-a}{2}\,.$$

Dann gilt $\dfrac{\mathrm{d}\varphi}{\mathrm{d}x} = h$ und es folgt : MMS

$$Q_{G,n}^{[a,b]}(f) = \sum_{i=0}^{n} \widehat{w}_i f(\widehat{x}_i)$$

mit $\widehat{x}_i = c + hx_i$ und $\widehat{w}_i = hw_i$.

Beispiel 4.28 (**Drei Punkte**, $n = 2$)**.** Wir wollen $Q_{G,2}^{[a,b]}(f)$ mit Genauigkeitsgrad $d = 5$ berechnen. Mittels der rekursiven Formel (3.45) in Beispiel 3.28 erhalten wir:

$$\begin{aligned} p_0(x) = 1, \quad p_1(x) = x, \quad p_2(x) &= xp_1(x) - \frac{1}{3}p_0(x) = x^2 - \frac{1}{3} \; \text{ und} \\ p_3(x) &= xp_2(x) - \frac{4}{16-1}p_1(x) = x^3 - \frac{1}{3}x - \frac{4}{15}x \\ &= x^3 - \frac{3}{5}x = x\left(x^2 - \frac{3}{5}\right) = x\left(x + \sqrt{\frac{3}{5}}\right)\left(x - \sqrt{\frac{3}{5}}\right) \end{aligned}$$

Also

$$x_0 = -\sqrt{\frac{3}{5}}, \quad x_1 = 0, \quad x_2 = \sqrt{\frac{3}{5}}.$$

Um die Gewichte zu finden, erhalten wir das Gleichungssystem

$$\begin{pmatrix} 1 & 1 & 1 \\ -\sqrt{\frac{3}{5}} & 0 & \sqrt{\frac{3}{5}} \\ \frac{4}{15} & -\frac{1}{3} & \frac{4}{15} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}$$

und lösen dies mittels Gauß-Elimination (vgl. Kapitel 5):

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ -\sqrt{\frac{3}{5}} & 0 & \sqrt{\frac{3}{5}} & 0 \\ \frac{4}{15} & -\frac{1}{3} & \frac{4}{15} & 0 \end{array}\right) \rightsquigarrow \left(\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & \sqrt{\frac{3}{5}} & 2\sqrt{\frac{3}{5}} & 2\sqrt{\frac{3}{5}} \\ 0 & -\frac{3}{5} & 0 & -\frac{8}{15} \end{array}\right) \begin{array}{l} (I) \\ (II)+\sqrt{\frac{3}{5}}\cdot(I) \\ (III)-\frac{4}{15}\cdot(I) \end{array}$$

$$\rightsquigarrow \left(\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & \frac{6}{5} & \frac{2}{3} \end{array}\right) \begin{array}{l} (I) \\ \sqrt{\frac{5}{3}}\cdot(II) \\ (III)-\sqrt{\frac{3}{5}}\cdot\cdot(II) \end{array}$$

Durch Rückwärtseinsetzen erhält man

$$w_2 = \frac{5}{6}\cdot\frac{2}{3} = \frac{5}{9}, \quad w_1 = 2 - \frac{10}{9} = \frac{8}{9}, \quad w_0 = 2 - \frac{8}{9} - \frac{5}{9} = \frac{5}{9}.$$

Um die Stützstellen und Gewichte auf $[a,b]$ zu erhalten, setzen wir wieder $c = \frac{1}{2}(a+b)$ und $h = \frac{1}{2}(b-a)$. Dann gilt $\widehat{x}_i = c + hx_i$ und $\widehat{w}_i = hw_i$ und somit ist

$$Q_{G,2}^{[a,b]}(f) = \frac{h}{9}\left[5f\left(c - \sqrt{\frac{3}{5}}h\right) + 8f(c) + 5f\left(c + \sqrt{\frac{3}{5}}h\right)\right].$$

Wir können auch hier wieder den Quadraturfehler ausrechnen:

Satz 4.29. *Für $f \in \mathcal{C}^{2n+2}[a,b]$ existiert ein $\xi \in [a,b]$, sodass*

$$E_{G,n}^{[a,b]}(f) := \int_a^b f(x)\,\mathrm{d}x - Q_{G,n}^{[a,b]}(f) = \frac{\|p_{n+1}\|^2}{(2n+2)!}\left(\frac{b-a}{2}\right)^{2n+3} f^{(2n+2)}(\xi).$$

Beweis. Siehe [Ran17], Satz 3.4 und Erläuterungen danach. □

Wir wissen, dass

$$\|p_{n+1}\|^2 = \frac{((n+1)!)^4\, 2^{2n+3}}{((2n+2)!)^2\,(2n+3)}$$

und somit gilt

$$E_{G,n}^{[a,b]}(f) = \frac{((n+1)!)^4}{((2n+2)!)^2}\frac{(b-a)^{2n+3}}{2n+3} f^{(2n+2)}(\xi),$$

d.h.

n	# Punkte	d	$E_{G,n}^{[a,b]}(f)$
0	1	1	$\frac{(b-a)^3}{12} f''(\xi)$
1	2	3	$\frac{(b-a)^5}{720} f^{(4)}(\xi)$
2	3	5	$\frac{(b-a)^7}{2016000} f^{(6)}(\xi)$
3	4	7	$\frac{(b-a)^9}{1778112000} f^{(8)}(\xi)$

Also wachsen nicht nur Genauigkeitsgrad und Konvergenzrate sehr schnell mit n; auch die Konstante im Fehler fällt viel schneller ab als bei Newton-Cotes!

Wir beenden das Kapitel mit einem wichtigen Satz, der besagt, dass die Gauß-Quadratur **nicht nur** für glatte Funktionen konvergiert:

Satz 4.30. *Für jedes* $f \in \mathcal{C}[a,b]$ *gilt* $\lim_{n\to\infty} E_{G,n}^{[a,b]}(f) = 0$.

Beweis. Seien $f \in \mathcal{C}[a,b]$ und $\epsilon > 0$ beliebig. Nach dem Approximationssatz von Weierstraß existiert ein $m_0 \in \mathbb{N}$, sodass für jedes $m \geq m_0$ ein $p^\epsilon \in \mathcal{P}_m$ existiert mit

$$\|f - p^\epsilon\|_{\infty,[a,b]} \leq \frac{\epsilon}{2(b-a)}.$$

Im Speziellen gilt für jedes $n \in \mathbb{N}$ mit $n \geq \frac{m}{2} - 1$, dass

$$\begin{aligned}
\left|E_{G,n}^{[a,b]}(f)\right| &= \left|I(f) - I(p^\epsilon) + I(p^\epsilon) - Q_{G,n}^{[a,b]}(p^\epsilon) + Q_{G,n}^{[a,b]}(p^\epsilon) - Q_{G,n}^{[a,b]}(f)\right| \\
&\leq |I(f-p^\epsilon)| + \left|E_{G,n}^{[a,b]}(p^\epsilon)\right| + \left|Q_{G,n}^{[a,b]}(f-p^\epsilon)\right| \\
&\leq \left|\int_a^b \frac{\epsilon}{2(b-a)}\,\mathrm{d}x\right| + 0 + \left|\underbrace{\sum_{i=0}^{n} w_i}_{=b-a} \frac{\epsilon}{2(b-a)}\right| = \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon
\end{aligned}$$

Wir haben also gezeigt, dass für alle $\epsilon > 0$ ein $m_0 \in \mathbb{N}$ existiert, sodass

$$\forall n \geq \frac{m_0}{2} - 1 \;:\; \left|E_{G,n}^{[a,b]}(f)\right| \leq \epsilon.$$

□

Der Approximationssatz von Weierstraß liefert aber leider keine Konvergenzrate. (Das selbe gilt natürlich für alle interpolatorischen Quadraturformeln und der Beweis verläuft identisch.)

5 Lineare Gleichungssysteme – Direkte Verfahren

Gegeben sei im Folgenden eine Matrix $A \in \mathbb{R}^{m\times n}$ und ein Vektor $b \in \mathbb{R}^m$
(Der Einfachheit halber betrachten wir wieder nur Gleichungssysteme in $\mathbb{R}$.)

$$A = (a_{jk})_{j,k=1}^{m,n} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, \quad b = (b_j)_{j=1}^{m} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}.$$

Gesucht ist dann ein Vektor $x = (x_k)_{k=1}^{n} \in \mathbb{R}^n$ mit der Eigenschaft

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

bzw. kompakt

$$Ax = b. \tag{5.1}$$

Definition 5.1. Das Gleichungssystem (GLS) (5.1) heißt

- **unterbestimmt**, falls $m < n$,
- **quadratisch**, falls $m = n$,
- **überbestimmt**, falls $m > n$.

Aus der linearen Algebra ist bekannt, dass (5.1) genau dann lösbar ist, wenn

$$\operatorname{rang}(A) = \operatorname{rang}(A \mid b),$$

wobei

$$(A \mid b) := \begin{pmatrix} a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} & b_m \end{pmatrix}.$$

Der Rang einer Matrix ist definiert als die Dimension des Spaltenraums bzw. des Zeilenraums. Das folgende Resultat aus der linearen Algebra liefert äquivalente Bedingungen für die eindeutige Lösbarkeit von (5.1).

Proposition 5.2. *Im Fall $m = n$ (A quadratisch) sind folgende Aussagen äquivalent:*

(i) $Ax = b$ *ist für jedes* b *eindeutig lösbar*

(ii) $\operatorname{rang}(A) = n$

(iii) $\det(A) \neq 0$

(iv) Alle Eigenwerte von A *sind von Null verschieden.*

Wir beschäftigen uns zunächst mit dem Fall $m = n$.

5.1 Störungstheorie (Konditionierung)

Bei der Lösung eines quadratischen Gleichungssystems $Ax = b$ treten zwei Fehlereinflüsse auf:

(a) Eingangsfehler in den Elementen von A und b

(b) Rundungsfehler im Verlauf des Lösungsprozesses

Zur Untersuchung dieser Fehler benötigen wir zunächst einige Definitionen.

5.1.1 Vektor- und Matrixnormen

Wir betrachten im Folgenden den endlichdimensionalen Vektorraum $\mathbb{R}^n$ (mit der üblichen Vektoraddition und Skalarmultiplikation). Die folgenden Definitionen und Sätze gelten jedoch meist ähnlich für $\mathbb{C}^n$.

Definition 5.3. Eine Abbildung $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}_+$ heißt **Norm**, wenn sie die folgenden Eigenschaften besitzt:

(N1) $\|x\| > 0$ für alle $x \in \mathbb{R}^n \setminus \{0\}$ (Definitheit)

(N2) $\|\alpha x\| = |\alpha| \|x\|$ für alle $x \in \mathbb{R}^n, \alpha \in \mathbb{R}$ (Homogenität)

(N3) $\|x + y\| \leq \|x\| + \|y\|$ für alle $x, y \in \mathbb{R}^n$ (Dreiecksungleichung)

Beispiel 5.4. Übliche Beispiele von Vektornormen sind

$$\|x\|_2 := \Big(\sum_{i=1}^n |x_i|^2\Big)^{1/2} \qquad \text{(Euklidische Norm oder } \ell_2\text{-Norm)}$$

$$\|x\|_\infty := \max_{i=1}^n |x_i| \qquad \text{(Maximumnorm oder } \ell_\infty\text{-Norm)}$$

$$\|x\|_1 := \sum_{i=1}^n |x_i| \qquad (\ell_1\text{-Norm})$$

(Vergleiche die Supremumsnorm $\|\cdot\|_{\infty,[a,b]}$ und die L_2-Norm $\|\cdot\|_2$ auf dem Funktionenraum $\mathcal{C}[a,b]$ aus Kapitel 3.)

Aus (N3) lässt sich auch leicht die ebenso wichtige *inverse Dreiecksungleichung* ableiten : Für alle $x, y \in \mathbb{R}^n$ gilt

MMS

$$\big|\|x\| - \|y\|\big| \le \|x - y\|\,. \tag{5.2}$$

Satz 5.5 (Normäquivalenz)**.** *Auf dem endlichdimensionalen Vektorraum $\mathbb{R}^n$ sind alle Normen* ***äquivalent****, d. h. für beliebige Normen $\|\cdot\|, \|\cdot\|'$ auf $\mathbb{R}^n$ existieren $m, M > 0$, sodass*

$$m\|x\|' \;\le\; \|x\| \;\le\; M\|x\|' \quad \textit{für alle } x \in \mathbb{R}^n\,. \tag{5.3}$$

Beweis. Es genügt $\|\cdot\|' = \|\cdot\|_\infty$ und $\|\cdot\|$ beliebig zu betrachten. Die Menge

$$S := \{y \in \mathbb{R}^n \;:\; \|y\|_\infty = 1\}$$

ist beschränkt und abgeschlossen (und deshalb kompakt). Da aufgrund der Normeigenschaften (N1) und (N2)

$$\|x\| = \left\|\sum_{i=1}^n x_i e_i\right\| \le \sum_{i=1}^n |x_i|\|e_i\| \le \gamma\|x\|_\infty$$

mit $\gamma := \sum_{i=1}^n \|e_i\|$ gilt, ist $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}_+$ gleichmäßig stetig und nimmt auf S sein Maximum/ Minimum

$$M := \max_{y\in S}\|y\| \qquad \text{und} \qquad m := \min_{y\in S}\|y\|$$

an. Da $0 \notin S$ folgt aus (N1), dass $M \ge m > 0$. Daher gilt für beliebiges $0 \ne x \in \mathbb{R}^n$, dass

$$m \le \left\|\frac{x}{\|x\|_\infty}\right\| \overset{\text{(N2)}}{=} \frac{1}{\|x\|_\infty}\|x\| \le M,$$

also

$$m\|x\|_\infty \leq \|x\| \leq M\|x\|_\infty.$$

Für $x = 0$ gilt die Ungleichung trivialerweise. □

Aus diesem Satz folgt, dass Konvergenz bzgl. einer beliebigen Norm im $\mathbb{R}^n$ auch der komponentenweisen Konvergenz entspricht, d. h.

$$\left\|x^{(t)} - x\right\| \xrightarrow{t\to\infty} 0 \qquad \Leftrightarrow \qquad x_i^{(t)} \xrightarrow{t\to\infty} x_i \quad \text{für alle } i = 1, \ldots, n. \tag{5.4}$$

Dies ist für *nicht* endlichdimensionale Vektorräume im Allgemeinen falsch!

Auch für den Vektorraum $(\mathbb{R}^{n\times n}, +, \cdot)$ aller quadratischen $n \times n$-Matrizen kann man Normen $\|\cdot\|$ einführen. Weil $\mathbb{R}^{n\times n}$ in natürlicher Weise mit $\mathbb{R}^{n^2}$ identifiziert werden kann, müssen Matrixnormen wieder die Bedingungen (N1) — (N3) erfüllen. Auch für die Konvergenz von Matrizen gilt dann

$$\left\|A^{(t)} - A\right\| \xrightarrow{t\to\infty} 0 \qquad \Leftrightarrow \qquad a_{jk}^{(t)} \xrightarrow{t\to\infty} a_{jk} \quad \text{für alle } j, k = 1, \ldots, n,$$

d. h. die Konvergenz von Matrizen ist äquivalent zu komponentenweiser Konvergenz.

Definition 5.6. Seien $A, B \in \mathbb{R}^{n\times n}$ und $x \in \mathbb{R}^n$ beliebig.

(a) Die Matrixnorm $\|\cdot\|_M$ heißt **verträglich** mit einer Vektornorm $\|\cdot\|_V$, wenn

$$\|Ax\|_V \;\leq\; \|A\|_M \|x\|_V \,. \tag{5.5}$$

(b) Eine Matrixnorm $\|\cdot\|$ auf $\mathbb{R}^{n\times n}$ nennt man **submultiplikativ**, wenn

$$\|AB\| \;\leq\; \|A\|\|B\| \,. \tag{5.6}$$

(c) Für jede beliebige Vektornorm $\|\cdot\|$ auf $\mathbb{R}^n$ definiert

$$\|A\| := \max_{x\neq 0} \frac{\|Ax\|}{\|x\|} \tag{5.7}$$

eine mit $\|\cdot\|$ verträgliche Matrixnorm auf $\mathbb{R}^{n\times n}$. Sie wird die **natürliche Matrixnorm** (oder **Grenznorm**) genannt (engl. *induced norm*).

Bemerkung. Offensichtlich ist die natürliche Matrixnorm die kleinste aller mit $\|\cdot\|$ verträglichen Matrixnormen, d. h. für eine beliebige Matrixnorm $\|\cdot\|_M$ (die mit $\|\cdot\|$ verträglich ist) gilt, dass falls

$$\|Ax\| \leq \|A\|_M \|x\|$$

für alle $x \in \mathbb{R}^n$ gilt, dann folgt

$$\|A\| \leq \|A\|_M$$

und es gilt

$$\|I\| = 1.$$

Jede natürliche Matrixnorm ist submultiplikativ, denn für beliebige $A, B \in \mathbb{R}^{n\times n}$ gilt:

$$\|AB\| = \max_{x\neq 0} \frac{\|ABx\|}{\|x\|} \leq \max_{x\neq 0} \frac{\|A\|\|Bx\|}{\|x\|} = \|A\|\|B\|.$$

Beispiel 5.7. (a) Die **Frobenius-Norm**

$$\|A\|_{\mathrm{Fr}} := \Big(\sum_{j,k=1}^{n} |a_{jk}|^2 \Big)^{1/2}$$

ist eine mit der Euklidischen Norm $\|\cdot\|_2$ verträgliche, submultiplikative Norm. Sie ist aber keine natürliche Norm.

MMS

(b) Die **Maximumnorm**

$$\|A\|_{\max} = \max_{j,k=1}^{n} |a_{jk}|$$

hingegen ist nicht submultiplikativ.

Lemma 5.8. *Die natürlichen Matrixnormen zu $\|\cdot\|_\infty$ und $\|\cdot\|_1$ sind die **Zeilensummennorm** bzw. die **Spaltensummennorm***

$$\|A\|_\infty := \max_{j=1}^{n} \sum_{k=1}^{n} |a_{jk}| \qquad bzw. \qquad \|A\|_1 := \max_{k=1}^{n} \sum_{j=1}^{n} |a_{jk}|. \tag{5.8}$$

Beweis. Wir führen den Beweis für $\|\cdot\|_\infty$. Der Beweis für $\|\cdot\|_1$ ist analog. Es gilt

$$\|Ax\|_\infty = \max_{j=1}^{n} \left| \sum_{k=1}^{n} a_{jk}x_k \right| \leq \left(\max_{k=1}^{n} |x_k| \right) \left(\max_{j=1}^{n} \sum_{j=1}^{n} |a_{jk}| \right) \leq \|A\|_\infty \|x\|_\infty,$$

woraus folgt

$$\max_{x\neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} \leq \|A\|_\infty. \tag{5.9}$$

Falls $\|A\|_\infty = 0$, dann ist $A = 0$, also $\max_{x\neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} = 0 = \|A\|_\infty$. Sei also $\|A\|_\infty > 0$ und $j^* \in \{1, \ldots, n\}$ ein Index, für den gilt

$$\|A\|_\infty = \max_{j=1}^{n} \sum_{k=1}^{n} |a_{jk}| = \sum_{k=1}^{n} |a_{j^*k}|. \tag{5.10}$$

Um Gleichheit der Normen zu zeigen wählen wir den speziellen Vektor $x^* = (x_k^*)_{k=1}^n \in \mathbb{R}^n$ mit Einträgen

$$x_k^* = \begin{cases} \frac{a_{j^*k}}{|a_{j^*k}|}, & \text{für } k \text{ mit } a_{j^*k} \neq 0, \\ 0, & \text{sonst.} \end{cases}$$

Dann gilt $\|x^*\| = 1$ und es folgt aus (5.10), dass

$$\begin{aligned} \|A\|_\infty &= \sum_{k=1}^n |a_{j^*k}| = \sum_{k=1}^n a_{j^*k} x_k^* = (Ax^*)_{j^*} \\ &\leq \|Ax^*\|_\infty = \frac{\|Ax^*\|_\infty}{\|x^*\|_\infty} \leq \max_{x \neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty}. \end{aligned}$$

□

5.1.2 Symmetrische Matrizen, Eigenwerte und die Spektralnorm

Definition 5.9. Sei $A \in \mathbb{R}^{n \times n}, \lambda \in \mathbb{C}$ und $v \in \mathbb{C}^n \setminus \{0\}$, so dass

$$Av = \lambda v. \tag{5.11}$$

Dann heißt λ **Eigenwert** von A und v ein zugehöriger **Eigenvektor**. Das Paar (λ, v) wird auch Eigenpaar genannt. Die Menge aller Eigenwerte heißt **Spektrum**.

Aus der linearen Algebra ist bekannt, dass $\lambda \in \mathbb{C}$ genau dann Eigenwert von A ist, falls $p_A(\lambda) := \det(A - \lambda I) = 0$, wobei $p_A \in \mathcal{P}_n$ das sog. **charakteristische Polynom** von A ist. Weiterhin ist bekannt, dass p_A über $\mathbb{C}$ genau n Nullstellen hat (Vielfachheiten mitgezählt), welche somit genau den Eigenwerten $\lambda_1, \ldots, \lambda_n$ von A entsprechen. Zu jedem λ_j existiert mindestens ein Eigenvektor $v_j \in \mathbb{C}^n \setminus \{0\}$.

Proposition 5.10. *Für alle (verträglichen) Matrixnormen $\|\cdot\|$ und für alle Eigenwerte λ von A gilt:*

$$|\lambda| \ \leq \ \|A\|, \tag{5.12}$$

d. h. alle Eigenwerte von A liegen in einer Kreisscheibe in $\mathbb{C}$ mit Mittelpunkt $0 \in \mathbb{C}$ und Radius $\|A\|$.

Beweis. Sei λ ein Eigenwert von A mit zugehörigem Eigenvektor v, der normiert sein soll, sodass $\|v\|_V = 1$. Sei $\|\cdot\|$ eine mit $\|\cdot\|_V$ verträgliche Matrixnorm. Dann gilt

$$|\lambda| = |\lambda| \|v\|_V = \|\lambda v\|_V = \|Av\|_V \leq \|A\| \|v\|_V = \|A\|.$$

□

Man könnte also bspw. $\|A\|_\infty$ zur praktischen Abschätzung der Eigenwerte verwenden. Es existieren auch noch schärfere Abschätzungen des Spektrums, z. B. die sog. Gerschgorin-Kreise.

Ein wichtiger Spezialfall sind **symmetrische Matrizen** $A \in \mathbb{R}^{n\times n}$ mit

$$A = A^\intercal,$$

wobei $(A^\intercal)_{ij} = A_{ji}$ die *transponierte Matrix* von A ist. Um das Spektrum symmetrischer Matrizen zu analysieren verwenden wir das Euklidische Skalarprodukt, definiert für alle $x, y \in \mathbb{C}^n$ durch

$$\langle x, y\rangle_2 := \sum_{i=1}^{n} x_i \overline{y_i}, \tag{5.13}$$

wobei $\overline{y_i}$ die komplex konjugierte Zahl zu y_i ist. Es hat die bekannten Eigenschaften eines Skalarprodukts, also Symmetrie, Linearität und Definitheit (vgl. Appendix A). Außerdem gilt

$$\|x\|_2^2 = \langle x, x\rangle_2$$

und für symmetrische Matrizen $A \in \mathbb{R}^{n\times n}$ und alle $x, y \in \mathbb{C}^n$ gilt

$$\langle Ax, y\rangle_2 = \langle x, Ay\rangle_2 . \tag{5.14}$$

Satz 5.11. *Die Eigenwerte $\lambda_1, \dots, \lambda_n$ einer symmetrischen Matrix $A = A^\intercal \in \mathbb{R}^{n\times n}$ sind alle reell und es existieren Eigenvektoren $v_1, \dots, v_n \in \mathbb{R}^n$, sodass*

$$\langle v_i, v_j\rangle_2 = \begin{cases} 1, & \text{für } i = j, \\ 0, & \text{sonst,} \end{cases} \tag{5.15}$$

d. h. $\{v_1, \dots, v_n\}$ ist eine Orthonormalbasis von $\mathbb{R}^n$.

Beweis. Wir zeigen nur, dass $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ und $v_1, \dots, v_n \in \mathbb{R}^n$ (für den Beweis der Orthonormalität siehe bspw. [Fis14]). Sei also $i = 1, \dots, n$ beliebig. Dann gilt

$$\overline{\lambda_i} = \overline{\lambda_i}\|v_i\|_2 = \langle v_i, \lambda_i v_i\rangle_2 = \langle v_i, Av_i\rangle_2 = \langle Av_i, v_i\rangle_2 = \langle \lambda_i v_i, v_i\rangle_2 = \lambda_i \|v_i\|_2 = \lambda_i.$$

Da A und λ_i reell sind, sind auch die Eigenvektoren $v_i \in \mathbb{R}^n$. □

Aus Satz 5.11 können einige wichtige Eigenschaften symmetrischer Matrizen abgeleitet werden. Dazu erinnern wir zunächst daran, dass eine Matrix $Q \in \mathbb{R}^{n\times n}$ **orthogonal** heißt, wenn

$$Q^{-1} = Q^\intercal.$$

Daraus folgert man leicht : $\|Qx\|_2 = \|x\|_2$ für alle $x \in R^n$. MMS

Definition 5.12. Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt **positiv definit**, wenn für alle $x \in \mathbb{R}^n \setminus \{0\}$ gilt:

$$\langle Ax, x\rangle_2 > 0.$$

Proposition 5.13. *Sei $A = A^\intercal \in \mathbb{R}^{n\times n}$.*

(a) *A besitzt eine* **Spektralzerlegung** *$A = QDQ^\intercal$ mit*[1]

$$D := \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix} \quad \text{und} \quad Q := \left(\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array}\right) \quad \text{orthogonal.}$$

(b) *Für die von der Euklidischen Norm $\|\cdot\|_2$ erzeugte natürliche Norm von A gilt*

$$\|A\|_2 := \max_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \max_{i=1}^{n} |\lambda_i|. \tag{5.16}$$

Deshalb wird $\|\cdot\|_2$ auch **Spektralnorm** *genannt.*

(c) *Es gilt für alle $x \in \mathbb{R}^n$*

$$\langle Ax, x\rangle_2 \geq \left(\min_{i=1}^{n} \lambda_i\right) \|x\|_2^2$$

(d) *A ist genau dann* **symmetrisch positiv definit (SPD)**, *wenn alle Eigenwerte λ_i größer Null sind.*

(e) *Alle Hauptdiagonalelemente einer SPD Matrix sind positiv.*

Beweis.

(a) Für alle $i = 1, \ldots, n$ gilt

$$QDQ^\intercal v_i = QDe_i = \lambda_i v_i = Av_i$$

und da $\{v_1, \ldots, v_n\}$ eine Basis des $\mathbb{R}^n$ ist, folgt das Resultat.

(b) Man kann leicht zeigen, dass $\|D\|_2 = \max_{i=1}^n |\lambda_i|$. MMS

Sei nun $x \in \mathbb{R}^n$ beliebig. Dann gilt

$$\|Ax\|_2 = \|QDQ^\intercal x\|_2 = \|DQ^\intercal x\|_2 \leq \|D\|_2 \|Q^\intercal x\|_2 \leq \max_{i=1}^{n} |\lambda_i| \|x\|_2 ,$$

[1] Im Folgenden werden Nulleinträge in Matrizen zu Gunsten der Übersichtlichkeit weggelassen.

woraus folgt, dass $\|A\| \leq \max_{i=1}^{n} |\lambda_i|$.

Sei $i^* \in \{1, \ldots, n\}$ sodass $|\lambda_{i^*}| = \max_{i=1}^{n} |y_i|$. Dann gilt für den zugehörigen normierten Eigenvektor

$$\frac{\|Av_{i^*}\|_2}{\|v_{i^*}\|_2} = \|Av_{i^*}\|_2 = \|\lambda_{i^*} v_{i^*}\|_2 = |\lambda_{i^*}| \|v_{i^*}\|_2 = \max_{i=1}^{n} |\lambda_i| \,.$$

weshalb $\|A\| \geq \max_{i=1}^{n} |\lambda_i|$.

(c) Sei $x \in \mathbb{R}^n$ beliebig. Dann gilt

$$\begin{aligned} \langle Ax, x \rangle_2 &= \langle QDQ^\intercal x, x \rangle_2 = \langle DQ^\intercal x, Q^\intercal x \rangle_2 \\ &\geq \left(\min_{i=1}^{n} \lambda_i \right) \|Q^\intercal x\|_2^2 = \left(\min_{i=1}^{n} \lambda_i \right) \|x\|_2^2. \end{aligned}$$

(d) Angenommen, A ist SPD und es existiert ein nicht-positiver Eigenwert $\lambda_{i^*} \leq 0$. Dann folgt

$$\langle Av_{i^*}, v_{i^*} \rangle_2 = \lambda_{i^*} \|v_{i^*}\|_2^2 \leq 0 \,,$$

was einen Widerspruch zur positiven Definitheit von A darstellt. Die Umkehrung folgt aus (c).

(e) Es gilt $0 < \langle Ae_k, e_k \rangle_2 = \left\langle (a_{jk})_{j=1}^{n}, e_k \right\rangle_2 = a_{kk}$.

□

Man kann auch leicht die folgenden Aussagen über die inverse A^{-1} symmetrischer Matrizen beweisen:

Proposition 5.14. *Sei $A = A^\intercal \in \mathbb{R}^{n \times n}$.*

(a) *A ist genau dann invertierbar, wenn für alle $i = 1, \ldots, n$ gilt: $\lambda_i \neq 0$.*

(b) *Die Matrix A^{-1}, wenn sie existiert, hat Eigenwerte*

$$\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_n}$$

und die selben Eigenvektoren.

(c) $\left\| A^{-1} \right\|_2 = \dfrac{1}{\min_{i=1}^{n} |\lambda_i|}$.

(d) *A ist genau dann symmetrisch positiv definit, wenn A^{-1} symmetrisch positiv definit ist.*

Beweis. Übungsaufgabe. □ MMS

5.1.3 Fehleranalyse

Wir betrachten nun das lineare Gleichungssystem (5.1) mit $A \in \mathbb{R}^{n\times n}$ invertierbar (oder regulär) und das gestörte System

$$\tilde{A}\tilde{x} = \tilde{b} \quad \text{mit} \quad \tilde{A} := A + \Delta A,\ \tilde{b} := b + \Delta b \ \text{ und } \ \tilde{x} := x + \Delta x, \tag{5.17}$$

d. h. wir nehmen an, dass A und b fehlerbehaftet sind und wollen den Fehler Δx in der Lösung in Abhängigkeit von ΔA und Δb schätzen. Dazu bezeichne im Folgenden $\|\cdot\|$ eine beliebige Vektornorm, sowie die zugehörige natürliche Matrixnorm.

Definition 5.15. Die Zahl

$$\operatorname{cond}(A) := \|A\|\|A^{-1}\|$$

heißt **Konditionszahl** von A (bzgl. der Norm $\|\cdot\|$).

Lemma 5.16. *Sei $B \in \mathbb{R}^{n\times n}$ mit $\|B\| < 1$. Dann ist die Matrix $I - B$ regulär und es gilt*

$$\left\|(I-B)^{-1}\right\| \ \le\ \frac{1}{1-\|B\|}. \tag{5.18}$$

Beweis. Da für alle $n \in \mathbb{N}$ gilt $\|B^n\| \le \|B\|^n$ und $\|B\| < 1$, folgt

$$\lim_{n\to\infty}\left\|\sum_{i=0}^{n} B^i\right\| \le \lim_{n\to\infty}\sum_{i=0}^{n}\left\|B^i\right\| \ \le\ \sum_{i=0}^{\infty}\|B\|^i = \frac{1}{1-\|B\|}.$$

Deshalb konvergiert die sog. **Neumannsche Reihe** $\displaystyle\sum_{i=0}^{\infty} B^i$.

Man überzeugt sich leicht, dass

$$(I-B)(I+B+\cdots+B^{n-1}) = (I+B+\cdots+B^{n-1})(I-B) = I - B^n \xrightarrow{n\to\infty} I,$$

sodass

$$(I-B)^{-1} = \sum_{i=0}^{\infty} B^i \tag{5.19}$$

womit auch (5.18) folgt. □

Satz 5.17 (Störungssatz)**.** *Die Matrix $A \in \mathbb{R}^{n\times n}$ sei regulär und wir nehmen an, es existiert $\gamma \in [0,1)$ mit*

$$\|\Delta A\| \;\leq\; \frac{\gamma}{\|A^{-1}\|}. \tag{5.20}$$

Dann ist die gestörte Matrix $\tilde{A} = A + \Delta A$ ebenfalls regulär und für den relativen Lösungsfehler gilt

$$\frac{\|\Delta x\|}{\|x\|} \;\leq\; \frac{\operatorname{cond}(A)}{1-\gamma}\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right). \tag{5.21}$$

Beweis. Man schreibe $(A + \Delta A) = (I + \Delta A A^{-1})\, A$ und setze $B := -\Delta A A^{-1}$. Es folgt aus (5.20), dass

$$\|B\| \;=\; \left\|\Delta A A^{-1}\right\| \;\leq\; \|\Delta A\|\|A^{-1}\| \;\leq\; \gamma < 1.$$

Weil A als regulär vorausgesetzt wurde, folgt die Regularität von $\tilde{A}$ aus Lemma 5.16. Außerdem gilt

$$\left\|(A+\Delta A)^{-1}\right\| = \left\|A^{-1}(I-B)^{-1}\right\| \leq \left\|A^{-1}\right\|\frac{1}{1-\gamma}.$$

Um (5.21) zu zeigen, subtrahieren wir (5.1) von (5.17) und erhalten so

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

Deshalb gilt

$$\begin{aligned}
\|\Delta x\| &\leq \left\|(A+\Delta A)^{-1}\right\|\left(\|\Delta b\| + \|\Delta A\|\|x\|\right) \\
&\leq \frac{\|A^{-1}\|\|A\|\|x\|}{1-\gamma}\left(\frac{\|\Delta b\|}{\|A\|\|x\|} + \frac{\|\Delta A\|}{\|A\|}\right) \\
&\leq \frac{\operatorname{cond}(A)}{1-\gamma}\left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|}\right)\|x\|,
\end{aligned}$$

wobei im letzten Schritt die Ungleichung

$$\|b\| = \|Ax\| \leq \|A\|\|x\|$$

verwendet wurde. □

Bemerkung. Für $\gamma \ll 1$ gilt $\frac{1}{1-\gamma} = 1 + \mathcal{O}(\gamma)$ und der Fehlerverstärkungsfaktor in (5.21) ist genau die Konditionszahl $\operatorname{cond}(A)$. Daraus folgt, dass eine Konditionszahl $\operatorname{cond}(A) \approx 10^s, s \in \mathbb{N}$, zum Verlust von s Stellen Genauigkeit in $\frac{\|\Delta x\|}{\|x\|}$ im Bezug auf

die Datenfehler $\frac{\|\Delta A\|}{\|A\|}$ und $\frac{\|\Delta b\|}{\|b\|}$ führt. Die Konditionszahl $\mathrm{cond}(A)$ hängt offenbar von der gewählten Matrixnorm ab.

Für symmetrische Matrizen gilt nach Proposition 5.13 (b) und 5.14 (c), dass

$$\mathrm{cond}_2(A) := \|A\|_2 \left\|A^{-1}\right\|_2 = \frac{\max_{i=1}^n |\lambda_i|}{\min_{j=1}^n |\lambda_j|}. \tag{5.22}$$

Beispiel 5.18. Für die Matrix

$$A = \begin{pmatrix} 1.2968 & 0.4322 \\ 0.4322 & 0.1441 \end{pmatrix} \quad \text{mit} \quad A^{-1} = \frac{5}{36} \cdot 10^4 \begin{pmatrix} 0.1441 & -0.4322 \\ -0.4322 & 1.2968 \end{pmatrix}$$

gilt

$$\|A\|_\infty = 1.729, \ \left\|A^{-1}\right\|_\infty \approx 2.4014 \cdot 10^4,$$

also

$$\mathrm{cond}_\infty(A) \approx 4.152 \cdot 10^4$$

und

$$\|A\|_2 = \max_{i=1,2} |\lambda_i| = 1.44085, \ \left\|A^{-1}\right\|_2 = \frac{1}{\min_{i=1,2} |\lambda_i|} = 20000,$$

also

$$\mathrm{cond}_2(A) = 2.8817 \cdot 10^4.$$

Die Abschätzung in Satz 5.17 ist im Allgemeinen scharf. Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch positiv definit mit $\lambda_1 \leq \ldots \leq \lambda_n$ sowie zugehörigen normierten Eigenvektoren $v_1, \ldots, v_n$. Wir wählen $\Delta A \equiv 0, b = v_n$ und $\Delta b = \epsilon v_1$ für $\epsilon > 0$. Dann gilt

$$x = \frac{1}{\lambda_n} v_n \quad \text{und} \quad \tilde{x} = x + \epsilon \frac{1}{\lambda_1} v_1.$$

Folglich gilt

$$\frac{\|\Delta x\|_2}{\|x\|_2} = \epsilon \frac{\delta_n}{\delta_1} \frac{\|v_1\|_2}{\|v_n\|_2} = \mathrm{cond}_2(A) \frac{\|\Delta b\|}{\|b\|}.$$

5.2 Gaußsches Eliminationsverfahren – LR Zerlegung

Ein **direktes** Lösungsverfahren für (reelle) quadratische lineare Gleichungssysteme der Form (5.1) ist ein Verfahren, das in endlich vielen Schritten (unter Vernachlässigung von Rundungsfehlern) die Lösung x liefert. Wir beschreiben jetzt das bekannteste und meist verwendete direkte Lösungsverfahren, das **Gaußsche Eliminationsverfahren**.

Dazu betrachten wir zunächst **gestaffelt** Systeme oder **Dreieckssysteme**

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ \ddots \qquad\quad &\vdots \\ a_{nn}x_n &= b_n \end{aligned} \tag{5.23}$$

d. h. $a_{jk} = 0$ für alle $j > k$, oder in Matrixform hat A eine obere Dreiecksform

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

(untere Dreieckssysteme kann man ähnlich lösen). A ist genau dann invertierbar, wenn $a_{jj} \neq 0$, für alle $j = 1, \ldots, n$. Die Lösung eines solchen Systems erhält man durch sog. **Rückwärtseinsetzen**:

$$x_n = \frac{b_n}{a_{nn}} \quad \text{und} \quad x_j = \frac{1}{a_{jj}} \left(b_j - \sum_{k=j+1}^{n} a_{jk} x_k \right), \quad \text{für } j = n-1, \ldots, 1\,. \tag{5.24}$$

Betrachten wir die Komplexität, so bemerken wir zunächst, dass man für das Rückwärtseinsetzen für die j-te Gleichung

$$n - j + 1 \text{ Operationen}$$

benötigt. Substituiert man $k = n - j + 1$ erhält man gesamt

$$\sum_{j=1}^{n} n - j + 1 = \sum_{k=1}^{n} k = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \mathcal{O}(n) \quad \text{Operationen.} \tag{5.25}$$

Die Lösungsidee beim Gaußschen Eliminationsverfahren ist es, das gegebene System $Ax = b$ schrittweise in ein oberes Dreieckssystem $Rx = c$ umzuformen, welches dieselbe Lösung x besitzt und durch Rückwärtseinsetzen gelöst werden kann. Dazu verwendet man die folgenden elementaren Umformungen:

(1) Vertauschen zweier Gleichungen (Zeilen);

(2) Addition des Vielfachen einer Gleichung (Zeile) zu einer anderen.

In der praktischen Durchführung des Gaußschen Eliminationsverfahren wendet man diese Operationen auf die **erweiterte Koeffizientenmatrix**

$$[A,b] = \left(\begin{array}{ccc|c} a_{11} & \cdots & a_{n1} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} & b_n \end{array}\right)$$

an. Wir nehmen im Folgenden an, dass A regulär ist. Da Gaußsche Eliminationsverfahren ist folgendermaßen definiert.

Schritt 1: Setze $A^{(0)} \equiv A, b^{(0)} \equiv b$. Verwende nun (1) und (2), um $\left[A^{(0)}, b^{(0)}\right]$ zu $\left[A^{(1)}, b^{(1)}\right]$ umzuformen mit

$$a_{j1}^{(1)} = 0 \quad \text{für alle } j > 1.$$

(a) Pivotsuche: Wähle $r \in \{1, \ldots, n\}$ mit $a_{r1}^{(0)} \neq 0$ (solch ein Element existiert, da A sonst singulär wäre).

$$\text{Pivotelement} \;\rightarrow \left(\begin{array}{ccc|c} a_{11}^{(0)} & \cdots & a_{1n}^{(0)} & b_1^{(0)} \\ \vdots & & \vdots & \vdots \\ a_{r1}^{(0)} & \cdots & a_{rn}^{(0)} & b_r^{(0)} \\ \vdots & & \vdots & \vdots \\ a_{n1}^{(0)} & \cdots & a_{nn}^{(0)} & b_n^{(0)} \end{array}\right)$$

(b) Vertausche erste und r-te Zeile:

$$\left(\begin{array}{ccc|c} a_{11}^{(0)} & \cdots & a_{1n}^{(0)} & b_1^{(0)} \\ \vdots & & \vdots & \vdots \\ a_{r1}^{(0)} & \cdots & a_{rn}^{(0)} & b_r^{(0)} \\ \vdots & & \vdots & \vdots \\ a_{n1}^{(0)} & \cdots & a_{nn}^{(0)} & b_n^{(0)} \end{array}\right) \begin{array}{l} \leftarrow \\ \\ \\ \\ \leftarrow \end{array} \rightsquigarrow \left(\begin{array}{ccc|c} a_{r1}^{(0)} & \cdots & a_{rn}^{(0)} & b_r^{(0)} \\ \vdots & & \vdots & \vdots \\ a_{11}^{(0)} & \cdots & a_{1n}^{(0)} & b_1^{(0)} \\ \vdots & & \vdots & \vdots \\ a_{n1}^{(0)} & \cdots & a_{nn}^{(0)} & b_n^{(0)} \end{array}\right) =: \left[\tilde{A}^{(0)}, \tilde{b}^{(0)}\right]$$

(c) Elimination: Für $j = 2, \ldots, n$ setze

$$q_{j1} := \frac{\tilde{a}_{j1}^{(0)}}{\tilde{a}_{11}^{(0)}} \quad \left(= \frac{a_{j1}^{(0)}}{a_{r1}^{(0)}}\right)$$

und ziehe von der j-ten Zeile das q_{j1}-fache der r-ten Zeile ab, d. h.

$$\left(\begin{array}{cccc|c} \tilde{a}_{11}^{(0)} & \tilde{a}_{12}^{(0)} & \cdots & \tilde{a}_{1n}^{(0)} & \tilde{b}_1^{(0)} \\ \tilde{a}_{21}^{(0)} & \tilde{a}_{22}^{(0)} & \cdots & \tilde{a}_{2n}^{(0)} & \tilde{b}_2^{(0)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \tilde{a}_{n1}^{(0)} & \tilde{a}_{n2}^{(0)} & \cdots & \tilde{a}_{nn}^{(0)} & \tilde{b}_n^{(0)} \end{array}\right) \quad \rightsquigarrow \quad \left(\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ & \vdots & \ddots & \vdots & \vdots \\ & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right) =: \left[A^{(1)}, b^{(1)}\right] .$$

wobei für alle $j = 2, \ldots, n$ und $k = 1, \ldots, n$ gilt:

$$\begin{aligned} a_{1k}^{(1)} &:= \tilde{a}_{1k}^{(0)} = a_{rk}^{(0)}, \quad a_{jk}^{(1)} := \tilde{a}_{jk}^{(0)} - q_{j1}\tilde{a}_{1k}^{(0)}, \\ b_1^{(1)} &:= \tilde{b}_1^{(0)} = b_r^{(0)}, \quad b_j^{(1)} := \tilde{b}_j^{(0)} - q_{j1}\tilde{b}_1^{(0)} . \end{aligned} \tag{5.26}$$

Schritt i, für $\mathbf{i = 2, \ldots, n-1}$: $[A^{(i-1)}, b^{(i-1)}] \rightsquigarrow [A^{(i)}, b^{(i)}]$ mit

$$a_{ji}^{(i)} = 0 \quad \text{für alle } \; j > i \, .$$

(a) Pivotsuche: Wähle $r \in \{1, \ldots, n\}$ mit $a_{ri}^{(i)} \neq 0$.

(b) Vertausche i-te und r-te Zeile:

$$\left(\begin{array}{ccccccc|c} a_{11}^{(i-1)} & a_{12}^{(i-1)} & & \cdots & & a_{1n}^{(i-1)} & b_1^{(i-1)} \\ & a_{22}^{(i-1)} & & \cdots & & a_{2n}^{(i-1)} & b_2^{(i-1)} \\ & & \ddots & & & \vdots & \vdots \\ & & & a_{ii}^{(i-1)} & \cdots & a_{in}^{(i-1)} & b_i^{(i-1)} \\ & & & \vdots & & \vdots & \vdots \\ & & & a_{ri}^{(i-1)} & \cdots & a_{rn}^{(i-1)} & b_r^{(i-1)} \\ & & & \vdots & & \vdots & \vdots \\ & & & a_{ni}^{(i-1)} & \cdots & a_{nn}^{(i-1)} & b_n^{(i-1)} \end{array}\right) \quad \begin{array}{c} \longleftarrow \\ \\ \\ \\ \\ \longleftarrow \\ \\ \\ \end{array} \quad \rightsquigarrow \cdots$$

$$\cdots \quad \rightsquigarrow \quad \left(\begin{array}{ccccccc|c} a_{11}^{(i-1)} & a_{12}^{(i-1)} & & \cdots & & a_{1n}^{(i-1)} & b_1^{(i-1)} \\ & a_{22}^{(i-1)} & & \cdots & & a_{2n}^{(i-1)} & b_2^{(i-1)} \\ & & \ddots & & & \vdots & \vdots \\ & & & a_{ri}^{(i-1)} & \cdots & a_{rn}^{(i-1)} & b_r^{(i-1)} \\ & & & \vdots & & \vdots & \vdots \\ & & & a_{ii}^{(i-1)} & \cdots & a_{in}^{(i-1)} & b_i^{(i-1)} \\ & & & \vdots & & \vdots & \vdots \\ & & & a_{ni}^{(i-1)} & \cdots & a_{nn}^{(i-1)} & b_n^{(i-1)} \end{array}\right) =: \left[\tilde{A}^{(i-1)}, \tilde{b}^{(i-1)}\right] .$$

(c) Elimination: Für $j = i+1, \dots, n$ setze

$$q_{ji} \coloneqq \frac{\tilde{a}_{ji}^{(i-1)}}{\tilde{a}_{ii}^{(i-1)}} \quad \left(= \frac{a_{ji}^{(i-1)}}{a_{ri}^{(i-1)}} \right)$$

und ziehe von der j-ten Zeile das q_{ji}-fache der i-ten Zeile ab:

$$\left(\begin{array}{cccc|c} \tilde{a}_{11}^{(i-1)} & & \cdots & \tilde{a}_{1n}^{(i-1)} & \tilde{b}_1^{(i-1)} \\ & \ddots & & \vdots & \vdots \\ & \tilde{a}_{ii}^{(i-1)} & \tilde{a}_{i,i+1}^{(i-1)} \cdots & \tilde{a}_{in}^{(i-1)} & \tilde{b}_i^{(i-1)} \\ & \tilde{a}_{i+1,i}^{(i-1)} & \tilde{a}_{i+1,i+1}^{(i-1)} \cdots & \tilde{a}_{i+1,n}^{(i-1)} & \tilde{b}_{i+1}^{(i-1)} \\ & \vdots & \vdots & \vdots & \vdots \\ & \tilde{a}_{n,i}^{(i-1)} & \tilde{a}_{n,i+1}^{(i-1)} \cdots & \tilde{a}_{n,n}^{(i-1)} & \tilde{b}_n^{(i-1)} \end{array}\right) \rightsquigarrow \cdots$$

$$\cdots \rightsquigarrow \left(\begin{array}{cccc|c} a_{11}^{(i)} & & \cdots & a_{1n}^{(i)} & b_1^{(i)} \\ & \ddots & & \vdots & \vdots \\ & a_{ii}^{(i)} & a_{i,i+1}^{(i)} \cdots & a_{in}^{(i)} & b_i^{(i)} \\ & & a_{i+1,i+1}^{(i)} \cdots & a_{i+1,n}^{(i)} & b_{i+1}^{(i)} \\ & & \vdots & \vdots & \vdots \\ & & a_{n,i+1}^{(i)} \cdots & a_{n,n}^{(i)} & b_n^{(i)} \end{array}\right) =: \left[A^{(i)}, b^{(i)}\right],$$

wobei für $k \geq j$ und

$$\begin{aligned} j < i: &\quad a_{jk}^{(i)} = a_{jk}^{(i-1)}, && b_j^{(i)} = b_j^{(i-1)}, \\ j = i: &\quad a_{ik}^{(i)} = \tilde{a}_{ik}^{(i-1)} = a_{rk}^{(i-1)}, && b_i^{(i)} = \tilde{b}_i^{(i-1)} = b_r^{(i-1)}, \\ j > i: &\quad a_{jk}^{(i)} = \tilde{a}_{jk}^{(i-1)} - q_{ji}\tilde{a}_{ik}^{(i-1)}, && b_j^{(i)} = \tilde{b}_j^{(i-1)} - q_{ji}\tilde{b}_i^{(i-1)}. \end{aligned} \tag{5.27}$$

Nach $n-1$ Schritten hat $A^{(n-1)}$ *obere Dreiecksgestalt* und wir setzen

$$R \coloneqq A^{(n-1)} \qquad \text{und} \qquad c \coloneqq b^{(n-1)}.$$

Das Dreieckssystem kann dann wie in (5.24) durch Rückwärtseinsetzen gelöst werden.

Beispiel 5.19. Wir wollen das Gaußsche Eliminationsverfahren anwenden auf das Gleichungssystem $Ax = b$ mit

$$A = \begin{pmatrix} 0 & 4 & -2 & -3 \\ -2 & -6 & 3 & 1 \\ 3 & 7 & 4 & -1 \\ 1 & 3 & 0 & -2 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 2 \end{pmatrix}.$$

Es ist also

$$[A^{(0)}, b^{(0)}] = \left(\begin{array}{cccc|c} 0 & 4 & -2 & -3 & 1 \\ -2 & -6 & 3 & 1 & 2 \\ 3 & 7 & 4 & -1 & 3 \\ 1 & 3 & 0 & -2 & 2 \end{array}\right).$$

Es ist (z. B.) $a_{41} \neq 0$, wir setzen also $r = 4$. Tauschen der ersten und vierten Zeile und Elimination in der ersten Spalte führt auf

$$[\tilde{A}^{(0)}, \tilde{b}^{(0)}] = \left(\begin{array}{cccc|c} 1 & 3 & 0 & -2 & 2 \\ -2 & -6 & 3 & 1 & 2 \\ 3 & 7 & 4 & -1 & 3 \\ 0 & 4 & -2 & -3 & 1 \end{array}\right) \begin{array}{l} \\ \\ \\ \\ \end{array} \begin{array}{l} \\ \left|\cdot 2\right. \\ \\ \left|\cdot(-3)\right. \end{array} \begin{array}{l} \\ \longleftarrow + \\ \\ \longleftarrow + \end{array} \rightsquigarrow \ldots$$

$$\ldots \rightsquigarrow \left(\begin{array}{cccc|c} 1 & 3 & 0 & -2 & 2 \\ 0 & 0 & 3 & -3 & 6 \\ 0 & -2 & 4 & 5 & -3 \\ 0 & 4 & -2 & -3 & 1 \end{array}\right) = [A^{(1)}, b^{(1)}],$$

wobei $q_{21} = -2,\ q_{31} = 3,\ q_{41} = 0$. Wir setzen $r = 3$ und erhalten

$$[\tilde{A}^{(1)}, \tilde{b}^{(1)}] = \left(\begin{array}{cccc|c} 1 & 3 & 0 & -2 & 2 \\ 0 & -2 & 4 & 5 & -3 \\ 0 & 0 & 3 & -3 & 6 \\ 0 & 4 & -2 & -3 & 1 \end{array}\right) \begin{array}{l} \\ \\ \left|\cdot 2\right. \\ \\ \end{array} \begin{array}{l} \\ \\ \longleftarrow + \\ \\ \end{array} \rightsquigarrow \ldots$$

$$\ldots \rightsquigarrow \left(\begin{array}{cccc|c} 1 & 3 & 0 & -2 & 2 \\ 0 & -2 & 4 & 5 & -3 \\ 0 & 0 & 3 & -3 & 6 \\ 0 & 0 & 6 & 7 & -5 \end{array}\right) = [A^{(2)}, b^{(2)}],$$

wobei $q_{32} = 0,\ q_{42} = -2$. Im letzten Schritt ist kein Zeilentausch notwendig, d. h. $[\tilde{A}^{(2)}, \tilde{b}^{(2)}] = [A^{(2)}, b^{(2)}]$. Mit $q_{43} = 2$ erhalten wir dann

$$[\tilde{A}^{(2)}, \tilde{b}^{(2)}] \quad \rightsquigarrow \quad \left(\begin{array}{cccc|c} 1 & 3 & 0 & -2 & 2 \\ 0 & -2 & 4 & 5 & -3 \\ 0 & 0 & 3 & -3 & 6 \\ 0 & 0 & 0 & 13 & -17 \end{array}\right) =: [R, c].$$

Den Übergang

$$[A^{(i-1)}, b^{(i-1)}] \quad \rightsquigarrow \quad [\tilde{A}^{(i-1)}, \tilde{b}^{(i-1)}] \quad \rightsquigarrow \quad [A^{(i)}, b^{(i)}]$$

kann man mit Hilfe von Matrixmultiplikationen beschreiben:

$$[\tilde{A}^{(i-1)}, \tilde{b}^{(i-1)}] = P_i[A^{(i-1)}, b^{(i-1)}] \quad \text{bzw.} \quad [A^{(i)}, b^{(i)}] = G_i[\tilde{A}^{(i-1)}, \tilde{b}^{(i-1)}]\,,$$

wobei

$$P_i := \begin{array}{c} \\ \\ \\ \\ i \\ \\ \\ \\ r \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccccc} & & & & i & & & & r \end{array} \\ \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & 0 & \cdots & 1 & & & \\ & & & & 1 & & & & \\ & & & \vdots & \ddots & \vdots & & & \\ & & & & & 1 & & & \\ & & & 1 & \cdots & 0 & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{pmatrix} \end{array} \tag{5.28}$$

eine sog. **Permutationsmatrix** ist (d. h. $(P_i)_{ri} = (P_i)_{ir} = 1,\ (P_i)_{jj} = 1\ \forall j \neq i, r$ und $(P_i)_{jk} = 0$ sonst) und

$$G_i := \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & & -q_{i+1,i} & & \\ & & \vdots & \ddots & \\ & & -q_{n,i} & & 1 \end{pmatrix} \tag{5.29}$$

eine sog. **Frobeniusmatrix** ist (d. h. $(G_i)_{jj} = 1,\ (G_i)_{ji} = -q_{ji}\ \forall j > i$ und $(G_i)_{jk} = 0$ sonst). Das Endresultat

$$[R, c] = G_{n-1}P_{n-1}G_{n-2}\cdots G_1P_1[A, b] \tag{5.30}$$

entspricht einem oberen Dreieckssystem $Rx = c$, das dieselbe Lösung wie das Ausgangssystem $Ax = b$ besitzt.

Wir nehmen zunächst an, dass $a_{ii}^{(i-1)} \neq 0$ für alle $i = 1, \ldots, n-1$, d. h. wir können in jedem Schritt $r = i$ wählen und es ist kein Zeilentausch notwendig; es ist somit

$$P_i = I, \quad \text{für alle } i = 1, \ldots, n-1\,.$$

Lemma 5.20. *Für alle $k = 1, \ldots, n-1$ sei G_k eine Frobeniusmatrix und sei*

$$G'_k = G_k - I = \begin{pmatrix} 0 & & & & \\ & \ddots & & & \\ & & 0 & & \\ & & g_{k+1,k} & \ddots & \\ & & \vdots & & \\ & & g_{n,k} & & 0 \end{pmatrix}.$$

Dann gilt

(a)

$$G'_k G'_l \equiv 0 \quad \text{für alle } k \leq l \tag{5.31}$$

(b)

$$G_k^{-1} = I - G'_k \tag{5.32}$$

Beweis.

(a) Sei $k \leq l$. Für alle $i, j = 1, \ldots, n$ gilt

$$(G'_k G'_l)_{ij} = \sum_{m=1}^{n} (G'_k)_{im} (G'_l)_{mj} = (G'_k)_{ik} (G'_l)_{kj}.$$

Aber $(G'_l)_{kj} \neq 0$ nur für $k > l$, d. h. für alle $k \leq l$:

$$G'_k G'_l \equiv 0.$$

(b) Aus (a) folgt $G'_k G'_k \equiv 0$. Deshalb ist

$$G_k(I - G'_k) = (I + G'_k)(I - G'_k) = I - G'_k G'_k = I.$$

Analog folgt $(I - G'_k)G_k = I$ und damit $G_k^{-1} = I - G'_k$.

□

Satz 5.21 (LR-Zerlegung ohne Pivotierung)**.** *Die Matrizen*

$$L := \begin{pmatrix} 1 & & & \\ q_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ q_{n1} & \cdots & q_{n,n-1} & 1 \end{pmatrix} \quad \text{und} \quad R := \begin{pmatrix} a_{11}^{(n-1)} & \cdots & a_{1n}^{(n-1)} \\ & \ddots & \vdots \\ & & a_{nn}^{(n-1)} \end{pmatrix} \tag{5.33}$$

bilden eine sog. ***LR-Zerlegung*** *der Matrix A, d. h.*

$$A = LR \tag{5.34}$$

und diese Zerlegung ist eindeutig bestimmt.

Beweis. Aus (5.30) folgt

$$G_1^{-1} \cdots G_{n-1}^{-1} R = A$$

mit den Frobeniusmatrizen, definiert in (5.29). Weiterhin folgt aus Lemma 5.20:

$$\begin{aligned} G_1^{-1} \cdots G_{n-1}^{-1} &\overset{(5.32)}{=} (I - G_1')(I - G_2') \cdots (I - G_{n-1}') \\ &\overset{(5.31)}{=} I - G_1' - G_2' - \ldots - G_{n-1}' = L. \end{aligned}$$

Der Beweis der Eindeutigkeit ist Übungsaufgabe. □ MMS

Korollar 5.22 (LR-Zerlegung)**.** *Im allgemeinen Fall sei*

$$P = P_{n-1} \cdots P_1 \,. \tag{5.35}$$

Dann existiert eine LR-Zerlegung der Matrix PA, d. h.

$$PA = LR \tag{5.36}$$

mit R wie in (5.30) *und mit*

$$L \coloneqq I + \sum_{i=1}^{n-1} \tilde{G}_i' \,, \tag{5.37}$$

wobei

$$\tilde{G}_i' \coloneqq P_{n-1} \cdots P_{i+1} \begin{pmatrix} 0 & & & & \\ & \ddots & & & \\ & & 0 & & \\ & & q_{i+1,i} & \ddots & \\ & & \vdots & & \ddots \\ & & q_{n,i} & & 0 \end{pmatrix}.$$

Beweis. Folgt aus Satz 5.21. (Wir geben keinen detailierten Beweis.) □ MMS
Wie auch in Satz 5.21 enthält die i-te Spalte von L auch im Falle $P \neq I$ die Werte $q_{i+1,i}, \ldots, q_{n,i}$, im Allgemeinen aber permutiert.

Was ist der Vorteil, die LR-Zerlegung (5.36) explizit zu berechnen, anstatt das Gaußsche Eliminationsverfahren auf $[A, b]$ anzuwenden?

Wenn P, L, R bekannt sind, können Systeme $Ax = b$ mit *beliebigen* rechten Seiten b gelöst werden. Denn

$$\begin{aligned} Ax = b \quad &\Leftrightarrow \quad PAx = Pb \\ &\Leftrightarrow \quad LRx = Pb\,. \end{aligned}$$

Setzen wir $y \coloneqq Rx$ und $c \coloneqq Pb$ führt das zu folgendem Lösungsverfahren

$$\begin{aligned} &\text{(i)} \quad c = Pb && \text{(rechte Seite permutieren)} \\ &\text{(ii)} \quad Ly = c && \text{(vorwärts einsetzen)} \\ &\text{(iii)} \quad Rx = y && \text{(rückwärts einsetzen)} \end{aligned} \tag{5.38}$$

Die Komplexität dieser Lösungsprozedur ist aufgrund von (5.25)

$$n^2 + \mathcal{O}(n) \;\; \text{Operationen}\,. \tag{5.39}$$

Bemerkung (Praktische Implementierung). Da $a_{i+1,i}^{(l)} = \ldots = a_{n,i}^{(l)} = 0$ für alle $l \geq i$, können diese Werte in $A^{(i)}$ mit $q_{i+1,i}, \ldots, q_{n,i}$ überschrieben werden. In Teil (b) der Schritte $(i+1)$ bis $(n-1)$ werden diese Einträge in L dann korrekt permutiert.

Die Matrix P speichert man in der Praxis als Vektor $p \in \{1, \ldots, n\}^n$. Zu Beginn initialisiert man $p = (1, \ldots, n)^\top$ und setzt im i-ten Schritt $p_r = i$ und $p_i = r$.

Beispiel 5.23. Die LR-Zerlegung $PA = LR$ aus Beispiel 5.19

$$\underset{P}{\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}} \underset{A}{\begin{pmatrix} 0 & 4 & -2 & -3 \\ -2 & -6 & 3 & 1 \\ 3 & 7 & 4 & -1 \\ 1 & 3 & 0 & -2 \end{pmatrix}} = \underset{L}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 0 & -2 & 2 & 1 \end{pmatrix}} \underset{R}{\begin{pmatrix} 1 & 3 & 0 & -2 \\ 0 & -2 & 4 & 5 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 13 \end{pmatrix}}$$

speichert man in der Praxis als

$$\begin{pmatrix} 1 & 3 & 0 & -2 \\ 3 & -2 & 4 & 5 \\ -2 & 0 & 3 & -3 \\ 0 & -2 & 2 & 13 \end{pmatrix}$$

wobei die roten Einträge aus $L - I$ entstehen und die restlichen Einträge die Einträge aus R sind. Die Permutationsmatrix P speichert man als Vektor

$$p = (4\ 3\ 2\ 1)\,.$$

Die LR-Zerlegung ist in Algorithmus 5.1 zusammengefasst.

Algorithmus 5.1 (LR-Zerlegung).

```
function LR(A)                          ▷ Eingabe: A ∈ ℝ^{n×n} (wird überschrieben)
                                        ▷ Ausgabe: L, R ∈ ℝ^{n×n} (gespeichert in A) und
                                        ▷ p ∈ {1,...,n}^n (Permutationsvektor)
    p = (1,...,n)                       ▷ Initialisieren des Permutationsvektors
    for i = 1,...,n−1 do
        r = i
        for j = i+1,...,n do
            if |a_ji| > |a_ri| then
                r = j                   ▷ Pivotsuche: Spaltenpivotisierung
            end if
        end for
        if r > i then
            for k = 1,...,n do
                t = a_ik; a_ik = a_rk; a_rk = t   ▷ Zeilenvertauschen
            end for
            t = p_i; p_i = p_r; p_r = t
        end if
        for j = i+1,...,n do
            a_ji = a_ji/a_ii            ▷ Berechnung des Eintrags q_ji in L
            for k = i+1,...,n do
                a_jk = a_jk − a_ji a_ik ▷ Eliminationsschritt
            end for
        end for
    end for
end function
```

Lemma 5.24. *Die zur LR-Zerlegung einer Matrix $A \in \mathbb{R}^{n\times n}$ erforderliche Anzahl arithmetischer Operationen ist*

$$\frac{2}{3}n^3 + \mathcal{O}(n^2)\,. \tag{5.40}$$

Beweis. Im i-ten Schritt benötigt man

$n-i$	Divisionen	zur Berechnung von $q_{i+1,i},\dots,q_{n,i}$
$(n-i)^2$	Multiplikationen	} Eliminationsschritt
$(n-i)^2$	Subtraktionen	} Eliminationsschritt

Damit ergibt sich eine Gesamtzahl von:

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

□

Bemerkung 5.25. In der Praxis wird die LR-Zerlegung einer Matrix in der sog. **Blockform** durchgeführt, weil dadurch der Arbeitsspeicher am Computer besser genutzt wird (vgl. [Gv96, Kapitel 3.4.7]).

Definition 5.26. Eine Matrix $A \in \mathbb{R}^{n\times n}$, für die gilt:

$$\exists m < n : \; a_{ij} = 0 \quad \text{für alle } |i-j| > m$$

nennt man eine **Bandmatrix** mit **Bandbreite** $2m+1$, d. h. nur die Hauptdiagonalelemente und $2m$ Nebendiagonalen der Matrix enthalten Nichtnulleinträge und A hat die Form

$$\begin{array}{c} \\ 1 \\ \\ m{+}1 \\ \\ \end{array}\!\!\begin{array}{c} \begin{array}{cccccc} 1 & & & m{+}1 & & \end{array} \\ \begin{pmatrix} * & \cdots & * & & & \\ \vdots & \ddots & & \ddots & & \\ * & & * & & * & \\ & \ddots & & \ddots & \vdots & \\ & & * & \cdots & * & \end{pmatrix} \end{array}.$$

Kann die LR-Zerlegung einer Bandmatrix ohne Zeilenvertauschung durchgeführt werden, so gilt für die Einträge L und R

$$l_{ij} = 0 \;\text{ und }\; r_{ji} = 0\,,$$

falls $i - j > m$. Die erforderliche Anzahl arithmetischer Operationen reduziert sich auf

$$\frac{2}{3}nm^2 + \mathcal{O}(nm)\,. \tag{5.41}$$

5.2.1 Stabilitätsanalyse der Gauß-Elimination

Sei rd : $\mathbb{R} \to \mathbb{F}$ die Rundungsoperation ins Fließgitter $\mathbb{F}$ und `eps` die relative Maschinengenauigkeit in `eps`, definiert wie in Kapitel 2.3.

Definition 5.27. Eine Algorithmus $f' = \varphi^{(n)} \circ \cdots \circ \varphi^{(1)}$ zur Lösung einer numerischen Aufgabe $y = f(x)$ heißt **rückwärtsstabil**, wenn zu jeder Eingabe x eine modifizierte Eingabe $\tilde{x}$ existiert, mit

$$f'(x) = f(\tilde{x})$$

und

$$|x - \tilde{x}| \leq C|x - \mathrm{rd}(x)|\,.$$

Stabilität und *Rückwärtsstabilität* können folgendermaßen interpretiert werden:

- Stabilität: Die in jedem Schritt eines Algorithmus auftretenden Rundungsfehler akkumulieren sich zu einem Fehler in der Lösung der den Rundungsfehler nicht übersteigt (Konditionierung).
- Rückwärtsstabilität: Der Algorithmus löst das gestellte Problem, aber für eine leicht gestörte Eingabe (von der Ordnung des Rundungsfehlers); die Konstante C hängt von der Konditionszahl ab.

Bemerkung 5.28. Man kann zeigen, dass Rückwärtsstabilität Stabilität impliziert. Die Umkehrung gilt im Allgemeinen nicht.

Beispiel 5.29. Es sei $f(x_1, x_2) = x_1 + x_2$ und $g(x) = 1 + x$. In Kapitel 2.1 haben wir gesehen, dass die Addition stabil ist, d. h. $f'(x_1, x_2) = x_1 \oplus x_2$ und $g'(x) = 1 \oplus x$ sind stabil.

- f' ist auch rückwärtsstabil:

  $$x_1 \oplus x_2 = (x_1 + x_2)(1 + \epsilon) = \underbrace{x_1(1 + \epsilon)}_{=:\tilde{x}_1} + \underbrace{x_2(1 + \epsilon)}_{=:\tilde{x}_2}\,,$$

  und

  $$|x_i - \tilde{x}_i| = |x_i||\epsilon| \leq |x_i|\texttt{eps}\,.$$

- g' ist jedoch nicht rückwärtsstabil:

  $$1 \oplus x = (1 + x)(1 + \epsilon) = 1 + \underbrace{x(1 + (1 + x^{-1})\epsilon)}_{=:\tilde{x}}$$

  und für $\texttt{eps} \leq |x| \leq 1/2$

  $$|x - \tilde{x}| = |x||1 + x^{-1}||\epsilon| = |\tilde{x}| \underbrace{\left|\frac{1 + x^{-1}}{1 + (1 + x^{-1})\epsilon}\right|}_{\leq 3/2|x|^{-1}} |\epsilon| \leq \frac{3}{2|x|}|\tilde{x}|\texttt{eps}\,,$$

  d. h. für $|x| \ll 1$ wird der relative Fehler in $\tilde{x}$ beliebig groß.

Lemma 5.30. *Das Skalarprodukt $f(x,z) = x^\intercal z$ für $x, z \in \mathbb{R}^n$ ist rückwärtsstabil (bzgl. Rundungsfehler), d. h. es existiert ein $\Delta x \in \mathbb{R}^n$, sodass für den Algorithmus*

$$f_n'(x,z) = x_1 \odot z_1 \oplus \ldots \oplus x_n \odot z_n$$

gilt

$$f_n'(x,z) = (x+\Delta x)^\intercal z \quad \textit{und} \quad \frac{|\Delta x_i|}{|x_i|} \leq n\mathtt{eps} + \mathcal{O}(\mathtt{eps}^2)\,.$$

Beweis. Per Induktion über n. Für $n = 1$ ist

$$f_1'(x,z) = x_1 \odot z_1 = x_1 z_1 (1+\epsilon_1) = (x_1 + \underbrace{1+\epsilon_1}_{=:\Delta x_1}) z_1 = f_1(x+\Delta x_1, z)$$

und

$$|\Delta x_1| \leq |\epsilon||x_1| \leq \mathtt{eps}|x_1| + C_1 \mathtt{eps}^2 \quad \text{mit} \quad C_1 := 0\,.$$

Es gelte nun die Induktionsvoraussetzung für $n-1$ und es seien $\bar{x} := (x_1, \ldots, x_{n-1})^\intercal$, $\bar{z} := (z_1, \ldots, z_{n-1})^\intercal \in \mathbb{R}^{n-1}$. Dann existiert nach Induktionsannahme ein $\Delta\bar{x} \in \mathbb{R}^{n-1}$ mit

$$f_{n-1}'(\bar{x}, \bar{z}) = (\bar{x} + \Delta\bar{x})^\intercal \bar{z} = f_{n-1}(\bar{x} + \Delta\bar{x})^\intercal, \bar{z})$$

und

$$|\Delta\bar{x}_i| \leq (n-1)\mathtt{eps}|x_i| + C_{n-1}\mathtt{eps}^2\,.$$

Deshalb ist

$$\begin{aligned} f_n'(x,z) &= (\bar{x} + \Delta\bar{x})^\intercal \bar{z} \oplus x_n \odot z_n \\ &= ((\bar{x} + \Delta\bar{x})^\intercal \bar{z} + x_n \cdot z_n(1+\epsilon_{2n}))\,(1+\epsilon_{2n+1}) \\ &= \underbrace{\left(x + \epsilon_{2n+1}x + (1+\epsilon_{2n+1})\begin{pmatrix}\Delta\bar{x}\\ \epsilon_{2n}x_n\end{pmatrix}\right)^\intercal}_{=:\Delta x} z = f_n(x+\Delta x, z) \end{aligned}$$

und für alle $i < n$ gilt aufgrund der Induktionsannahme, dass

$$\begin{aligned} |\Delta x_i| \leq |\epsilon_{2n+1}||x_i| + |1+\epsilon_{2n+1}||\Delta\bar{x}_i| &\leq \Big(\mathtt{eps} + (1+\mathtt{eps})\big((n-1)\mathtt{eps} + C_{n-1}\mathtt{eps}^2\big)\Big)|x_i| \\ &= \Big(n\mathtt{eps} + \underbrace{\big((n-1) + (1+\mathtt{eps})C_{n-1}\big)}_{=:C_n}\mathtt{eps}^2\Big)|x_i|\,. \end{aligned}$$

Außerdem ist

$$|\Delta x_n| = |\epsilon_{2n+1} + (1+\epsilon_{2n+1})\epsilon_{2n}||x_n| \leq (2\mathtt{eps} + \mathtt{eps}^2)|x_n| \leq (n\mathtt{eps} + C_n\mathtt{eps}^2)|x_n|\,.$$

Man kann leicht nachrechnen, dass $C_n = \sum_{j=1}^{n-1} j + \mathcal{O}(\mathtt{eps}) = \frac{n(n-1)}{2} + \mathcal{O}(\mathtt{eps})$ und C_n ist beschränkt für $\mathtt{eps} \to 0$, unabhängig von $\mathtt{eps}$. □

MMS

Der Verstärkungsfaktor n ist der schlechteste Fall und sehr pessimistisch, weil Vorzeichenwechsel in den Rundungsfehlern nicht berücksichtigt werden. Die Rundungsfehler heben sich im Durchschnitt größtenteils auf! Besser wäre eine statistische Betrachtung. Auf ähnliche Weise kann man folgendes Resultat induktiv beweisen:

Satz 5.31. *Seien $\hat{L}, \hat{R} \in \mathbb{F}^{n\times n}$ die numerisch berechnete LR-Zerlegung von $A \in \mathbb{R}^{n\times n}$. Sei weiter $\hat{y} \in \mathbb{F}^n$ die numerische Lösung von $\hat{L}y = Pb$ und $\hat{x} \in \mathbb{F}^n$ die numerische Lösung von $\hat{R}\hat{x} = \hat{y}$. Dann gilt*

$$(A + E)\hat{x} = b$$

mit

$$|e_{ij}| \leq n\mathtt{eps}\left(3|a_{ij}| + 5\sum_{k=1}^{n}\left|(P^\intercal \hat{L})_{ik}\right||\hat{r}_{kj}|\right) + \mathcal{O}(\mathtt{eps}^2)\,. \tag{5.42}$$

Beweis. Siehe [Gv96, Kapitel 3.3]. □

Wie auch in Lemma 5.30 ist der Verstärkungsfaktor n pessimistisch. Für moderates n ist der erste Term in (5.42) aber wieder gutartig. Der zweite Term dagegen kann problematisch werden, da $\hat{L}$ Einträge der Form

$$\frac{a_{ji}^{(i-1)}}{a_{ii}^{(i-1)}}$$

enthält (und $a_{ii}^{(i-1)}$ beliebig klein werden kann).

Die Gauß-Elimination (und somit die LR-Zerlegung) ist in dieser Form **nicht** rückwärtsstabil (und daher auch nicht vorwärtsstabil).

Beispiel 5.32. Es sei

$$A = \begin{pmatrix} \epsilon & 1 \\ 1 & 0 \end{pmatrix}, \quad A^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & -\epsilon \end{pmatrix}$$

mit $0 < \epsilon \ll 1$ und $\mathrm{cond}_\infty(A) = (1+\epsilon)^2 \approx 1$, also gut konditioniert. Die *exakte* LR-Zerlegung ist gegeben durch

$$L = \begin{pmatrix} 1 & 0 \\ \epsilon^{-1} & 1 \end{pmatrix}, \quad R = \begin{pmatrix} \epsilon & 1 \\ 0 & -\epsilon^{-1} \end{pmatrix}$$

und somit

$$\|\hat{L}\|_\infty \approx \|L\|_\infty = \epsilon^{-1} + 1 \gg 1$$
$$\|\hat{R}\|_\infty \approx \|R\|_\infty = \epsilon^{-1} \gg 1$$

und für $i = j = 2$ gilt für den zweiten Term in (5.42) (ohne Pivotisierung)

$$5 \sum_{k=1}^{2} |\hat{l}_{2k}||\hat{r}_{k2}| = 10\epsilon^{-1} \gg 1\,,$$

sodass die Schranke in (5.42) beliebig groß werden kann.

Lemma 5.33. *Wählt man bei der Gauß-Elimination den Index r im i-ten Schritt so wie in Algorithmus 5.1, d. h. man führt eine* ***Spaltenpivotisierung*** *durch, sodass für alle $i \leq j \leq n$*

$$\left|a_{ri}^{(i)}\right| \geq \left|a_{ji}^{(i)}\right|, \tag{5.43}$$

dann gilt

$$\left|\hat{l}_{ij}\right| \leq 1 \quad \textit{und deshalb} \quad \left\|\hat{L}\right\|_\infty \leq n\,.$$

Beweis. Übungsaufgabe. □ MMS

Beispiel 5.34. Das Gauß-Verfahren (ohne Pivotisierung) für das lineare Gleichungssystem

$$\begin{pmatrix} -10^{-5} & 1 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{5.44}$$

führt in exakter Arithmetik auf

$$\begin{pmatrix} -10^{-5} & 1 \\ 0 & 1 + 2 \times 10^5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \times 10^5 \end{pmatrix}$$

mit Lösung $x = (0.4999975,\ 0.999995)^\intercal \approx (-\frac{1}{2},\ 1)^\intercal$. In $\mathbb{F}(10, 4, 1)$ ergibt sich zwar ebenfalls $\hat{l}_{21} = 2 \times 10^5$, das dadurch entstehende System ist aber

$$\begin{pmatrix} -0.1 \times 10^{-4} & 1 \\ 0 & 0.2 \times 10^6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \begin{pmatrix} 1 \\ 0.2 \times 10^6 \end{pmatrix}.$$

Nur der Eintrag $\hat{r}_{22}$ ist (als einziger) leicht fehlerbehaftet. Als Ergebnis erhält man die völlig inakzeptable Lösung

$$\hat{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

der relative Fehler $\frac{\|x-\hat{x}\|_\infty}{\|x\|_\infty}$ beträgt also 50%, obwohl die Matrix sehr gut konditioniert ist.

Wir führen nun die Rückwärtsanalyse durch: Tatsächlich ist $\hat{x}$ die exakte Lösung des gestörten Problems

$$\begin{pmatrix} -10^{-5} & 1 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

d. h.

$$E = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix} \quad \text{mit} \quad \|E\|_\infty = 1!$$

Spaltenpivotisierung führt in $\mathbb{F}(10, 4, 1)$ in diesem Fall auf das System

$$\begin{pmatrix} 2 & 7 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

wo wieder nur der Eintrag $\hat{r}_{22}$ leicht gestört ist (exakt: $r_{22} = 1.000005$). Das führt auf die völlig akzeptable Näherungslösung

$$\hat{x} = \begin{pmatrix} -0.5 \\ 1 \end{pmatrix}.$$

Mit Spaltenpivotisierung kann man folgende Abschätzung aus Satz 5.31 und Lemma 5.33 ableiten:

Satz 5.35. *Die Matrix $A \in \mathbb{R}^{n\times n}$ sei regulär und das Gleichungssystem $Ax = b$ werde mit Gauß-Elimination mit Spaltenpivotisierung gelöst. Die numerische Lösung $\hat{x}$ löst das gestörte Problem*

$$(A + E)\hat{x} = b$$

und

$$\frac{\|E\|_\infty}{\|A\|_\infty} \leq C\rho n^3 \texttt{eps} + \mathcal{O}(\texttt{eps}^2), \tag{5.45}$$

wobei

$$\rho := \frac{\max_{i,j,k} \left|\hat{a}_{jk}^{(i)}\right|}{\|A^{-1}\|_\infty}$$

der sog. Wachstumsfaktor *ist.*

Beweis. Siehe [Wil61] und [Gv96]. □

Bemerkung. Man kann Matrizen konstruieren für die $\rho = 2^{n-1}$ (Wilkinson-Matrix). In der Praxis gilt jedoch üblicherweise $\rho \approx 10$ und so kann die Gauß-Elimination mit Spaltenpivotisierung als stabil angesehen werden.

Korollar 5.36. *Unter den Voraussetzungen von Satz 5.35 und unter der Bedingung* $\|E\|_\infty \|A^{-1}\|_\infty \leq \gamma < 1$ *gilt:*

$$\frac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} \leq \mathrm{cond}_\infty(A) \frac{C\rho n^3}{1-\gamma} \mathtt{eps} + \mathcal{O}(\mathtt{eps}^2)\,.$$

Beweis. Folgt direkt aus Satz 5.17 (Störungssatz) und Satz 5.35. □

Bemerkung 5.37. Achtung! Der positive Effekt der Spaltenpivotisierung ist nur gesichert, wenn die (betragsmäßigen) Zeilensummen in etwa gleich groß sind. Das Gleichungssystem

$$\begin{pmatrix} -4 & 4 \times 10^5 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \times 10^5 \\ 0 \end{pmatrix}$$

ist äquivalent zu (5.44) in Beispiel 5.34. Da das betragsgrößte Element in der ersten Spalte a_{11} ist, führt Gauß-Elimination mit Spaltenpivotisierung in $\mathbb{F}(10, 4, 1)$ wieder auf die inakzeptable Näherungslösung $\hat{x} = (0,\ 1)^\intercal$.

Um das zu verhindern führt man daher vor der Rechnung eine sog. **Äquilibrierung** durch, d. h. man multipliziert mit der Diagonalmatrix D

$$d_{ii} := \left(\sum_{j=1}^{n} |a_{ij}| \right)^{-1} :$$
$$Ax = b \quad \rightsquigarrow \quad DAx = Db\,,$$

die alle Zeilensummen auf 1 normiert. Eine verbesserte Stabilisierung im Fall stark unterschiedlicher Größenordnungen der Matrixeinträge liefert die **totale Pivotisierung**, bei der man in jedem Schritt der Gauß-Elimination sowohl Zeilen- als auch Spaltenvertauschung zulässt und die Bedingung (5.43) durch

$$\left| a_{rs}^{(i)} \right| \ \geq \ \left| a_{kj}^{(i)} \right| \quad \text{für } i \leq k, j \leq n$$

ersetzt.

5.2.2 Weitere Anwendungen der Gauß-Elimination

Determinantenberechnung. Für quadratische Matrizen $A, b \in \mathbb{R}^{n \times n}$ gilt

$$\det(AB) = \det(A) \det(B)\,.$$

Sei $K < n$ die Anzahl der Zeilenvertauschungen. Dann gilt

$$\det(P) = \det(P^{-1}) = (-1)^K$$

und

$$\det(A) = \det(P^{-1}LR) = (-1)^K \underbrace{\det(L)}_{=:1} \underbrace{\det R}_{=:\prod_{j=1}^n r_{jj}}$$

bzw.

$$\det(A) = (-1)^K \prod_{j=1}^{n} r_{jj}\,. \tag{5.46}$$

Inversenberechnung. Um die Inverse einer Matrix A zu berechnen, gehe man wie folgt vor:

(i) Berechnung der LR-Zerlegung von PA.

(ii) Lösung der gestaffelten Systeme

$$Ly^{(i)} = Pe^{(i)},\ Rx^{(i)} = y^{(i)}, \quad \text{für } i = 1, \ldots, n$$

mit den kartesischen Basisvektoren $e^{(i)}$ des $\mathbb{R}^n$.

(iii) Die Inverse ist dann gegeben durch

$$A^{-1} = \left(x^{(1)}\ x^{(2)}\ \cdots\ x^{(n)}\right).$$

Wie in Bemerkung 5.25 löst man die gestaffelten Matrixsysteme in Schritt (ii) in der Praxis in Blockform: Finde $X, Y \in \mathbb{R}^{n\times n}$, sodass

$$LY = P \quad \text{und} \quad RX = Y\,.$$

Es gilt dann: $A^{-1} = X$.

5.3 Die Cholesky-Zerlegung für symmetrisch positiv definite Matrizen

Wir erinnern zunächst daran, dass für eine symmetrisch positiv definite (SPD) Matrix $A = A^\intercal \in \mathbb{R}^{n\times n}$ gilt

$$\langle Ax, x\rangle_2 > 0 \qquad \forall x \in \mathbb{R}^n \setminus \{0\}.$$

Satz 5.38. *Für symmetrisch positiv definite Matrizen $A \in \mathbb{R}^{n\times n}$ ist die Gauß-Elimination ohne Zeilenvertauschungen durchführbar und alle auftretenden Pivotelemente $r_{ii} = a_{ii}^{(i-1)}$ sind positiv.*

Beweis. Aus Proposition 5.13 (e) folgt, dass $a_{11} > 0$. Sei nun

$$A^{(1)} = \begin{pmatrix} a_{11} & * & \cdots & * \\ 0 & & & \\ \vdots & & \tilde{A}^{(1)} & \\ 0 & & & \end{pmatrix}$$

die im ersten Eliminationsschritt erzeugt Matrix mit Untermatrix

$$\tilde{A}^{(1)} := (a_{jk})_{j,k=2,\dots,n} \in \mathbb{R}^{(n-1)\times(n-1)}\,.$$

Wegen $A = A^\intercal$ gilt für alle $j, k = 2, \dots, n$, dass

$$a_{jk}^{(1)} = a_{jk} - \frac{a_{j1}}{a_{11}} a_{1k} = a_{kj} - \frac{a_{k1}}{a_{11}} a_{1j} = a_{kj}^{(1)}\,. \tag{5.47}$$

Sei nun $\tilde{x} = (x_2, \dots, x_n)^\intercal \in \mathbb{R}^{n-1} \setminus \{0\}$ beliebig und $x = (x_1, \tilde{x}^\intercal)^\intercal \in \mathbb{R}^n$, sodass

$$x_1 := -\frac{1}{a_{11}} \sum_{k=2}^{n} a_{1k} x_k\,.$$

Dann ist

$$\sum_{k=1}^{n} a_{jk} x_k = \sum_{k=2}^{n} a_{jk} x_k - \frac{a_{j1}}{a_{11}} \sum_{k=2}^{n} a_{1k} x_k = \begin{cases} 0 & \text{für } j = 1 \\ \sum_{k=2}^{n} a_{jk}^{(1)} x_k & \text{sonst,} \end{cases}$$

woraus folgt

$$0 < \langle Ax, x\rangle_2 = \sum_{j,k=1}^{n} x_j a_{jk} x_k = \sum_{jk=2}^{n} x_j a_{jk}^{(1)} x_k = \langle \tilde{A}^{(1)} \tilde{x}, \tilde{x}\rangle_2\,.$$

Daraus folgt, zusammen mit (5.47), dass $\tilde{A}^{(1)}$ symmetrisch positiv definit ist. Das Resultat folgt deshalb per Induktion. □

MMS

Mit $D = \operatorname{diag}(R)$ gilt $A = LD(D^{-1}R)$ und wegen der Symmetrie muss $D^{-1}R = L^\intercal$ sein, d. h.

$$A = LDL^\intercal\,.$$

Da $d_{ii} = r_{ii} = a_{ii}^{(i-1)} > 0$ ist die Diagonalmatrix $D^{1/2}$ mit

$$\left(D^{1/2}\right)_{ii} = d_{ii}^{1/2}$$

wohldefiniert und es gilt

$$A = \tilde{L}\tilde{L}^\intercal \quad \text{mit} \quad \tilde{L} := LD^{1/2}\,.$$

Diese spezielle Form der Zerlegung für SPD Matrizen heißt **Cholesky-Zerlegung** und kann (aufgrund der Symmetrie) in $\frac{1}{3}n^3 + \mathcal{O}(n^2)$ Operationen berechnet werden.

5.4 Nicht-reguläre Systeme

Es sei nun das Gleichungssystem

$$Ax = b \tag{5.48}$$

mit (nicht notwendig quadratischer) Matrix $A \in \mathbb{R}^{m\times n}$ und Vektor $b \in \mathbb{R}^m$ gegeben mit $\operatorname{rang}(A)$ beliebig. Wir wiederholen zunächst einige Definitionen und Eigenschaften aus der linearen Algebra:

Definition 5.39 (Wiederholung)**.** Es sei $A \in \mathbb{R}^{m\times n}$ beliebig gegeben.

(a) Das **Bild** von A ist die Menge $\operatorname{im}(A) := \{y \in \mathbb{R}^m \ : \ \exists x \in \mathbb{R}^n \text{ mit } y = Ax\}$.

(b) Der **Kern** von A ist die Menge $\ker(A) := \{x \in R^n \ : \ Ax = 0\}$.

(c) Der Rang von A ist definiert als $\operatorname{rang}(A) = \dim(\operatorname{im}(A))$. Man kann zeigen, dass auch $\operatorname{rang}(A) = \operatorname{rang}(A^\intercal) = \dim(\operatorname{im}(A^\intercal))$.

(d) Es gilt $\operatorname{im}(A) \perp \ker(A^\intercal)$, d. h. für alle $y \in \operatorname{im}(A), z \in \ker(A^\intercal)$ gilt

$$\langle y, z\rangle_2 = 0\,.$$

(e) Aus (c) und (d) folgt der **Rangsatz**:

$$\begin{aligned}\dim(\operatorname{im}(A)) + \dim(\ker(A^\intercal)) &= m\\ \dim(\ker(A)) + \dim(\operatorname{im}(A^\intercal)) &= n.\end{aligned}$$

Satz 5.40 („Least-Squares“ Lösung)**.** *Es existiert ein $\overline{x} \in \mathbb{R}^n$, die sog.* ***Least-Squares Lösung*** *von* (5.48)*, sodass*

$$\|A\overline{x} - b\|_2 = \min_{x\in\mathbb{R}^n} \|Ax - b\|_2\,. \tag{5.49}$$

Das ist äquivalent dazu, dass $\overline{x}$ die Lösung der sog. ***Normalengleichung***

$$A^\intercal A\overline{x} = A^\intercal b \tag{5.50}$$

ist. Falls $\operatorname{rang}(A) = n$*, so ist $\overline{x}$ eindeutig. Andernfalls hat jede weitere Lösung die Form*

$$\overline{x} + z \qquad \text{mit } z \in \ker(A)\,.$$

Beweis.

(i) Wir zeigen zuerst die Äquivalenz von (5.49) und (5.50). Sei dazu $\overline{x}$ eine Lösung von (5.50) und $x \in \mathbb{R}^n$ beliebig. Wir definieren $\Phi(x) := \|Ax - b\|_2^2$. Dann gilt

$$\begin{aligned}\Phi(x) &= \|(A\overline{x} - b) + A(x - \overline{x})\|_2^2 \\ &= \Phi(\overline{x}) + 2 \underbrace{\langle A\overline{x}b, A(x - \overline{x})\rangle_2}_{= \langle \underbrace{A^\mathsf{T}A\overline{x} - A^\mathsf{T}b}_{=0}, x - \overline{x}\rangle_2} + \underbrace{\|A(x - \overline{x})\|_2^2}_{\geq 0} \geq \Phi(\overline{x})\,.\end{aligned}$$

Umgekehrt gilt für eine Minimallösung $\overline{x}$ von (5.49) notwendigerweise

$$\nabla\Phi(\overline{x}) = 0\,. \tag{5.51}$$

Aber

$$\begin{aligned}\frac{\partial}{\partial x_i}\Phi(x) &= \frac{\partial}{\partial x_i}\left(\sum_{j=1}^{n}\left(\sum_{k=1}^{n} a_{jk}x_k - b_j\right)^2\right) \\ &= 2\sum_{j=1}^{n} a_{ji}\left(\sum_{k=1}^{n} a_{jk}x_k - b_j\right) = 2\left(A^\mathsf{T}Ax - A^\mathsf{T}b\right)\,,\end{aligned}$$

sodass (5.51) äquivalent zu (5.50) ist.

(ii) Wir untersuchen nun die Lösbarkeit von (5.50). Aus der Wiederholung 5.39 (d) und (e) folgt, dass b sich eindeutig schreiben lässt als

$$b = s + r \tag{5.52}$$

wobei $s \in \mathrm{im}(A)$ und $r \in \ker(A^\mathsf{T})$. D. h. es existiert ein $\overline{x} \in \mathbb{R}^n$ mit $A\overline{x} = s$ und

$$A^\mathsf{T}A\overline{x} = A^\mathsf{T}s = A^\mathsf{T}s + \underbrace{A^\mathsf{T}r}_{=0} = A^\mathsf{T}b\,,$$

d. h. $\overline{x}$ löst (5.50). Sei nun x_1 eine weitere Lösung von (5.50). Dann gilt wegen (5.52)

$$b = \underbrace{Ax_1}_{\in\mathrm{im}(A)} + \underbrace{(b - Ax_1)}_{\in\ker(A^\mathsf{T})} = s + r\,,$$

also gilt notwendigerweise $Ax_1 = A\overline{x}$ und somit $x_1 - \overline{x} \in \ker(A)$. Falls $\mathrm{rang}(A) = n$, dann folgt $\ker(A) = \{0\}$ und $x_1 = \overline{x}$.

□

Für $x \in \mathbb{R}^n$ gilt

$$x^\mathsf{T}A^\mathsf{T}Ax = (Ax)^\mathsf{T}(Ax) = \|Ax\|_2^2 \geq 0\,. \tag{5.53}$$

Falls $\operatorname{rang}(A) = n$ impliziert $\|Ax\|_2 = 0 \Rightarrow x = 0$ und $A^\intercal A$ ist symmetrisch positiv definit. Man könnte die Normalengleichung (5.50) also über die Cholesky-Zerlegung lösen. Dieses Problem ist im Allgemeinen aber sehr schlecht konditioniert; im Fall $m = n$ gilt bspw.

$$\operatorname{cond}_2(A^\intercal A) = \operatorname{cond}_2(A)^2 .$$

Stattdessen betrachten wir eine Methode, die die Cholesky-Zerlegung

$$A^\intercal A = L^\intercal L$$

zur Verfügung stellt, jedoch ohne $A^\intercal A$ explizit aufzustellen.

5.4.1 QR-Zerlegung — Das Householder-Verfahren

Definition 5.41 (Householdertransformation)**.** Sei $0 \neq v \in \mathbb{R}^m$ gegeben. Dann heißt die Matrix

$$H_v = I - \frac{2}{v^\intercal v} v v^\intercal \in \mathbb{R}^{m \times m}$$

Householdermatrix oder **Householdertransformation**, wobei $vw^\intercal$ das *äußere* (oder *dyadische*) Produkt der Vektoren v und $w \in \mathbb{R}^m$ (also eine Matrix vom Rang 1) ist, d. h.

$$vw^\intercal = \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} \cdot \begin{pmatrix} w_1 & \cdots & w_m \end{pmatrix} = \begin{pmatrix} v_1 w_1 & \cdots & v_1 w_m \\ \vdots & \ddots & \vdots \\ v_m w_1 & \cdots & v_m w_m \end{pmatrix} .$$

Lemma 5.42. *Die Householdermatrix H_v hat die folgenden Eigenschaften:*

(i) *H_v ist orthogonal, d. h. $H_v^\intercal = H_v^{-1} = H_v$.*

(ii) *Für alle $x \in \mathbb{R}^m$ gilt $H_v x = x - 2\beta v$, wobei $\beta = \dfrac{v^\intercal x}{v^\intercal v}$.*

(iii) *Für $x = \alpha v$ gilt $H_v x = -x$.*

(iv) *Aus $\langle x, v \rangle_2 = 0$ folgt $H_v x = x$.*

(v) *Als Operator von $\mathbb{R}^m \to \mathbb{R}^m$ bewirkt H_v eine Reflexion an der Ebene mit Normale v.*

Beweis. (i) – (iv) folgen durch einfaches Nachrechnen . MMS
(v) Für alle $x \in \mathbb{R}^m$ existiert ein $\alpha \in \mathbb{R}$ und $x^\perp \in \mathbb{R}^m$ mit $\langle x^\perp, v \rangle_2 = 0$, sodass $x = \alpha v + x^\perp$. Dann folgt aus (iii) und (iv): $H_v x = -\alpha v + x^\perp$. □

Im Speziellen gilt: Sei $0 \neq x \in \mathbb{R}^m$ und $v \coloneqq x \pm \|x\|_2 e_i$. Dann ist $H_v x = \mp \|x\|_2 e_i$, d. h. Householdermatrizen H_v können wie Frobeniusmatrizen G_k benutzt werden, um eine Matrix $A \in \mathbb{R}^{m\times n}$ mit $m \geq n$ und $\operatorname{rang}(A) = n$ auf Dreiecksgestalt zu bekommen, was auf das sog. **Householder-Verfahren** führt:

Sei $A \in \mathbb{R}^{m\times n}$ mit $m \geq n$ und $\operatorname{rang}(A) = n$.

Schritt 1: Seien a_k die Spaltenvektoren der Matrix A.

(a) Wähle $v_1 \coloneqq a_1 + \operatorname{sgn}(a_{11})\|a_1\|_2 e_1 \in \mathbb{R}^m$, wobei

$$\operatorname{sgn}(x) = \begin{cases} 1 & , \text{für } x \geq 0\,, \\ -1 & , \text{für } x < 0\,. \end{cases}$$

(b) Setze $A^{(1)} = H_{v_1} A$ mit Spaltenvektoren

$$a_1^{(1)} = -\operatorname{sgn}(a_{11})\|a_1\|_2 e_1, \quad a_k^{(1)} = a_k - 2\frac{\langle a_k, v_1\rangle_2}{\|v_1\|_2^2} v_1\,,$$

für $k = 2, \ldots, n$.

Schritt i: Nach $i-1$ Schritten haben wir

$$A^{(i-1)} = \begin{pmatrix} * & \cdots & * & \cdots & * \\ & \ddots & \vdots & & \vdots \\ & & * & \cdots & * \\ & & & \tilde{A}^{(i-1)} & \end{pmatrix},$$

wobei

$$\tilde{A}^{(i-1)} = \begin{pmatrix} \tilde{a}_i^{(i-1)} & \cdots & \tilde{a}_n^{(i-1)} \end{pmatrix} \in \mathbb{R}^{(m-i+1)\times(n-i+1)}\,,$$

mit $\tilde{a}_k^{(i-1)} = \begin{pmatrix} a_{ik}^{(i-1)} & \cdots & a_{mk}^{i-1} \end{pmatrix}^{\mathsf{T}} \in \mathbb{R}^{m-i+1}$.

(a) Wähle

$$v_i = \begin{pmatrix} 0 \\ \tilde{v}_i \end{pmatrix} \coloneqq \begin{pmatrix} 0 \\ \tilde{a}_i^{(i-1)} \end{pmatrix} + \operatorname{sgn}(a_{ii}^{(i-1)})\left\|\tilde{a}_i^{(i-1)}\right\| e_i\,.$$

(b) Setze $A^{(i)} = H_{v_i} A^{(i-1)}$. Wegen $(v_i)_j = 0$ für $j < i$ bleiben die ersten $i-1$ Zeilen und Spalten von $A^{(i-1)}$ unverändert. Es gilt dann

$$\begin{aligned} a_k^{(i)} &= a_k^{(i-1)}, \qquad \text{für } k = 1, \ldots, i-1\,, \\ a_i^{(i)} &= \left(a_{1i}^{(i-1)}, \ldots, a_{i-1,i}^{(i-1)}, -\operatorname{sgn}(a_{ii}^{(i-1)})\left\|\tilde{a}_i^{(i-1)}\right\|_2, 0, \ldots, 0\right)^{\mathsf{T}}, \\ a_k^{(i)} &= a_k^{(i-1)} - 2\frac{\langle \tilde{a}_k^{(i-1)}, \tilde{v}_i\rangle_2}{\|\tilde{v}_i\|_2^2} v_i\,, \qquad \text{für } k = i+1, \ldots, n\,. \end{aligned}$$

Nach n Schritten gilt

$$\underbrace{H_{v_n} \cdots H_{v_1}}_{=:Q^\intercal \in \mathbb{R}^{m\times m}} A = A^{(n-1)} =: R = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ 0 & \ddots & \vdots \\ \vdots & \ddots & r_{nn} \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{m\times n},$$

d. h. die letzten $m-n$ Zeilen von R sind Nullzeilen.

Als Produkt orthogonaler Matrizen ist $Q^\intercal$ selbst auch orthogonal, sodass gilt

$$A = QR.$$

Das Householder-Verfahren liefert einen konstruktiven Beweis des folgenden Satzes:

Satz 5.43 (QR-Zerlegung)**.** *Zu jeder Matrix $A \in R^{m\times n}$ mit $m \geq n$ und* $\operatorname{rang}(A) = n$ *existiert eine orthogonale Matrix $Q \in \mathbb{R}^{m\times m}$ und eine obere Dreiecksmatrix $R \in \mathbb{R}^{m\times n}$ mit $r_{ii} \neq 0$, sodass*

$$A = QR.$$

Die ersten n Spalten von Q bilden eine Orthonormalbasis von $\operatorname{im}(A)$.

Bemerkung 5.44. (i) In der Praxis wird Q **nicht** explizit aufgestellt, sondern man speichert $v_1, \ldots, v_n$.

(ii) Für $m = n$ reichen $n-1$ Schritte und die QR-Zerlegung benötigt den **doppelten** Aufwand der LR-Zerlegung, nämlich

$$\frac{4}{3}n^3 + \mathcal{O}(n^2) \text{ Operationen.}$$

Die QR-Zerlegung ist in Algorithmus 5.2 zusammengefasst.

Frage: Wir kann man nun die QR-Zerlegung verwenden um (5.48) zu lösen?

(a) **Fall $m \geq n$ (überbestimmt oder quadratisch) und** $\operatorname{rang}(A) = n$. Es gilt

$$A^\intercal A = (QR)^\intercal QR = R^\intercal Q^\intercal QR = R^\intercal R.$$

Also ist

$$A^\intercal A\overline{x} = A^\intercal b \;\Leftrightarrow\; R^\intercal R\overline{x} = R^\intercal Q^\intercal b. \tag{5.54}$$

Wir setzen

$$R = \begin{bmatrix} \tilde{R} \\ 0 \end{bmatrix}_{\substack{n \\ m-n}}^{} \quad \text{und} \quad Q = \underset{n}{\begin{bmatrix} \tilde{Q} \end{bmatrix}} \underset{m-n}{\begin{bmatrix} Q^* \end{bmatrix}}_{m}. \tag{5.55}$$

Algorithmus 5.2 (QR-Zerlegung).

function QR(A, V) ▷ Eingabe: $A \in \mathbb{R}^{m\times n}$ mit $m \geq n$ (wird überschrieben)
▷ Ausgabe: $V \in \mathbb{R}^{m\times n}, R \in \mathbb{R}^{m\times n}$ (gespeichert in A), sodass mit
▷ $v_k :=$ k-te Spalte von V gilt $A = H_{v_1} H_{v_2} \dots H_{v_n} R =: QR$

$v_k = 0$ ▷ Initialisieren der $k$ten Spalte von V
for $k = 1, \dots, n$ **do**
▷ Berechnung der k-ten Householder-Matrix
$\alpha = \sum_{i=k,\dots,m} |a_{ik}|^2$
$v_{kk} = a_{kk} + \operatorname{sgn}(a_{kk})\sqrt{\alpha}$
$\beta = |v_{kk}|^2$
for $i = k+1, \dots, m$ **do**
$v_{ik} = a_{ik}$
$\beta = \beta + |v_{ik}|^2$
end for
▷ Multiplikation der k-ten Householdermatrix mit $A^{(k-1)}$
$a_{kk} = -\operatorname{sgn}(a_{kk})\sqrt{\alpha}$
for $i = k+1, \dots, m$ **do**
$a_{ik} = 0$
end for
for $j = k+1, \dots, n$ **do**
$\gamma = \frac{2}{\beta} \sum_{i=k,\dots,m} v_{ik} a_{ij}$
for $i = k, \dots, m$ **do**
$a_{ij} = a_{ij} - \gamma v_{ik}$
end for
end for
end for
end function

Dann ist $\tilde{R} \in \mathbb{R}^{n\times n}$ regulär und (5.54) ist äquivalent zu

$$\tilde{R}^\intercal \tilde{R}\overline{x} = \tilde{R}^\intercal \tilde{Q}^\intercal b \;\Leftrightarrow\; \tilde{R}\overline{x} = \tilde{Q}^\intercal b$$

und wir erhalten als Lösungsverfahren

(i) $c = Q^\intercal b$ (n Householdertransformationen)

(ii) $\tilde{R}\overline{x} = c$ (rückwärts einsetzen)

Falls $m = n$, dann folgt aus $\operatorname{rang}(A) = n$, dass

$$A^\intercal A\overline{x} = A^\intercal b \;\Leftrightarrow\; A\overline{x} = b$$

und die Least-Squares-Lösung $\overline{x}$ ist die „normale“ Lösung x des Systems (5.49) und $\tilde{R} = R,\ \tilde{Q} = Q$.

(b) **Fall** $m < n$ **(unterbestimmt) und** $\operatorname{rang}(A) = m$. Nach Satz 5.43 existiert eine QR-Zerlegung $A^\intercal = QR$ und

$$A^\intercal A\overline{x} = A^\intercal b \;\Leftrightarrow\; QRR^\intercal Q^\intercal \overline{x} = QRb \;\Leftrightarrow\; RR^\intercal Q^\intercal \overline{x} = Rb\,.$$

Wegen (5.55) enthält dieses Gleichungssystem nur $m < n$ Gleichungen zur Bestimmung von $\overline{x} \in \mathbb{R}^n$. Aufgrund der Regularität von $\tilde{R} \in \mathbb{R}^{m\times m}$ folgt

$$A^\intercal A\overline{x} = A^\intercal b \;\Leftrightarrow\; \tilde{R}^\intercal \tilde{Q}^\intercal \overline{x} = b$$

mit $\tilde{Q} \in \mathbb{R}^{n\times m}$. Wir erhalten hier als Lösungsverfahren:

(i) Finde $\tilde{y} \in \mathbb{R}^m :\; \tilde{R}^\intercal \tilde{y} = b$ (vorwärts einsetzen)

(ii) Berechne $\overline{x} = \tilde{Q}\overline{y}$ (n Householdertransformationen)

Bemerkung 5.45. Sei $y^\perp \in \mathbb{R}^{n-m}$ beliebig und sei $z = Q^* y^\perp$. Dann gilt

$$R^\intercal Q^\intercal(\overline{x} + z) = \begin{bmatrix}\tilde{R}^\intercal & 0\end{bmatrix}\begin{bmatrix}\tilde{Q}^\intercal \\ Q^{*\intercal}\end{bmatrix}(\overline{x} + z) = \begin{bmatrix}\tilde{R}^\intercal & 0\end{bmatrix}\begin{bmatrix}\tilde{y} \\ y^\perp\end{bmatrix} = \tilde{R}^\intercal \tilde{y} = b\,.$$

Also ist $\overline{x} + z$ auch eine Lösung von (5.50), aber

$$\|\overline{x} + z\|_2 = \|Q^\intercal(\overline{x} + z)\|_2 = \left\|\begin{bmatrix}\tilde{y} \\ y^\perp\end{bmatrix}\right\|_2 = \|\tilde{y}\|_2 + \left\|y^\perp\right\|_2$$

und deshalb ist $\overline{x}$ die Lösung von (5.50) mit minimaler Norm $\|\overline{x}\|_2$.

Für Systeme mit maximalem $\operatorname{rang}(A) = \min(m, n)$ gibt es also immer eine eindeutig bestimmte Least-Squares-Lösung $\overline{x}$, für die $\|\overline{x}\|_2$ minimal ist und die mittels QR-Zerlegung berechnet werden kann.

(c) **Fall** $\operatorname{rang}(A) < \min(m, n)$ **(rangdefizitär)**. Auch hier gibt es immer noch eine eindeutig bestimmte Least-Squares-Lösung $\overline{x}$ mit minimaler Norm $\|x\|_2$, die sog. **Minimallösung**. Diese hängt allerdings nicht mehr stetig von der Problemstellung ab. Kleine Störungen in A und b führen zu beliebig großen Fehlern in $\overline{x}$. Man nennt so ein Problem ein **schlecht gestelltes Problem**. Stabilitätsprobleme treten jedoch bereits auf, wenn A nur *fast rangdefizitär* (oder schlecht konditioniert) ist.

5.4.2 Die Singulärwertzerlegung (SVD)

Die derzeit zuverlässigste Technik zur Behandlung (nahezu) rangdefizitärer lineare Gleichungssysteme ist die **Singulärwertzerlegung** oder **SVD** (engl.: **S**ingular **V**alue **D**ecomposition). Trotz ihrer praktischen Relevanz gehen wir nur kurz darauf ein. Den folgenden Satz geben wir ohne Beweis (siehe z.B. [Ran17, Satz 4.11]).

Satz 5.46 (Singulärwertzerlegung). *Sei $A \in \mathbb{R}^{m\times n}$. Dann existieren orthogonale Matrizen $U \in \mathbb{R}^{m\times m}$ und $V \in \mathbb{R}^{n\times n}$, sodass*

$$U^\intercal A V = \Sigma := \operatorname{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m\times n} \tag{5.56}$$

mit $p = \min(m, n)$ und $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$.

Je nachdem, ob $m \leq n$ oder $m \geq n$ ist, hat Σ die Gestalt

$$\left(\begin{array}{ccc|c} \sigma_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & \sigma_p & \end{array}\right) \quad \text{oder} \quad \left(\begin{array}{ccc} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_p \\ \hline & 0 & \end{array}\right).$$

Die Einträge σ_i von Σ heißen **Singulärwerte**. Die zugehörigen Spalten u_i und v_i von U und V heißen (linke bzw. rechte) **Singulärvektoren**. Aus (5.56) folgt unmittelbar für $i = 1, \ldots, p$:

$$A v_i = \sigma_i u_i \quad \text{und} \quad A^\intercal u_i = \sigma_i v_i\,,$$

woraus folgt

$$A^\intercal A v_i = \sigma_i^2 v_i \quad \text{und} \quad A A^\intercal u_i = \sigma_i^2 u_i\,,$$

d. h. die Singulärwerte sind gerade die (positiven) Wurzeln der (von 0 verschiedenen) Eigenwerte von $A^\intercal A$ und $AA^\intercal$. Daraus folgt auch

MMS

$$\|A\|_2 = \sigma_1\,. \tag{5.57}$$

Falls ein $r < p$ existiert, sodass $\sigma_{r+1} = \ldots = \sigma_p = 0$, dann ist

- $\operatorname{rang}(A) = r < \min(m, n)$ (rangdefizitär),
- $\ker(A) = \operatorname{span}\{v_{r+1}, \ldots, v_n\}$,
- $\operatorname{im}(A) = \operatorname{span}\{u_1, \ldots, u_r\}$.

Mit Hilfe der Singulärwertzerlegung lassen sich solche rangdefizitären Probleme jetzt elegant lösen. Sei o. B. d. A. wieder $m \geq n$ und $\operatorname{rang}(A) = r < n$.

Satz 5.47 (Minimallösung). *Es sei $A = U\Sigma V^\intercal$ die Singulärwertzerlegung der Matrix $A \in \mathbb{R}^{m\times n}$ mit $m \geq n$ und $\operatorname{rang}(A) = r$. Dann ist*

$$\overline{x} = A^\dagger b = \sum_{i=1}^{r} \frac{u_i^\intercal b}{\sigma_i} v_i \tag{5.58}$$

mit

$$A^\dagger := V\Sigma^\dagger U^\intercal \quad \textit{und} \quad \Sigma^\dagger := \operatorname{diag}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0\right) \in \mathbb{R}^{m\times n} \tag{5.59}$$

die eindeutig bestimmte Lösung der Normalengleichung (5.50) *mit minimaler euklidischer Norm. Für das Residuum gilt:*

$$\|A\overline{x} - b\|_2 = \left(\sum_{i=r+1}^{m} (u_i^\intercal b)^2\right)^{1/2}.$$

Beweis. Übungsaufgabe. □ MMS

Die Matrix $A^\dagger$ heißt **Pseudo-Inverse** (oder auch *Moore-Penrose-Inverse*) von A. Aus Stabilitätsgründen sollte man bei der Definition von $A^\dagger$ allerdings den sog. *numerischen Rang* r_{num} anstelle von r wählen. Man bezeichnet die Matrix als *numerisch rangdefizitär*, wenn $r_{\text{num}} \leq p$ existiert, sodass

$$\sigma_{r_{\text{num}}} \geq \|A\|_2 \texttt{eps} = \sigma_1 \texttt{eps} \geq \sigma_{r_{\text{num}+1}}.$$

Wenn $r_{\text{num}} = p$ (nicht defizitär), gilt:

$$A^\dagger = \begin{cases} (A^\intercal A)^{-1} A^\intercal & \text{falls } m > n, \\ A^{-1} & \text{falls } m = n, \\ A^\intercal (AA^\intercal)^{-1} & \text{falls } m < n. \end{cases}$$

Die Singulärwertzerlegung ist aber sehr aufwendig, die Anzahl der benötigten Operation ist

$$12n^3 + \mathcal{O}(n^2).$$

Sie wird nur angewandt, wo der hohe Aufwand aus Stabilitätsgründen nötig ist.

5.4.3 Anwendung: Gaußsche Ausgleichsrechnung

Wir betrachten folgende Problemstellung: Gegeben seien

(i) n Funktionen $u_1, \ldots, u_n : \mathbb{R} \to \mathbb{R}$,

(ii) $m \geq n$ Datenpunkte $(x_i, y_i) \in R^2$, $i = 1, \ldots, m$.

Gesucht sind n Koeffizienten $c_1, \ldots, c_n$, sodass

$$\sum_{i=1}^{m} (u(x_i) - y_i)^2 \to \min \qquad \text{mit} \quad u(x) := \sum_{k=1}^{n} c_k u_k(x)\,,$$

d. h. wir suchen ein „Modell“ $u(x)$ für die Abhängigkeit der Größe y vom Parameter x mit Ansatzraum $\{u_1, \ldots, u_n\}$. Im Rahmen der linearen Algebra können wir das Problem folgendermaßen formulieren:

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, \; y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}, \; A \in \mathbb{R}^{m \times n} \text{ mit } a_{ij} = u_j(x_i)\,.$$

Dann gilt:

$$\sum_{i=1}^{m} \left(u(x_i) - y_i\right)^2 = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} x_j u_j(x_i) - y_i \right)^2 = \|Ac - y\|_2^2\,.$$

Die optimale (Least-Squares-) Lösung $\overline{c}$ kann also (laut Satz 5.40) über die Normalengleichung

$$A^\intercal A \overline{c} = A^\intercal y$$

berechnet werden. Gilt $\operatorname{rang}(A) = n$ so ist $\overline{c}$ eindeutig.

Für den Spezialfall $u_k(x) = x^{k-1}$, d. h. *Polynomapproximation* (vgl. Kapitel 3.6 und 4.5) gilt

$$A = \begin{pmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \cdots & x_m^{n-1} \end{pmatrix} \quad \text{(Vandermonde-Matrix)}$$

und $\operatorname{rang}(A) = n$ (Lemma 4.25). Das heißt, $\overline{c}$ und $\overline{u}(x) = \sum_{k=1}^{n} \overline{c}_k x^{k-1}$ sind eindeutig bestimmt (für ein Beispiel, siehe Übungsblatt).

6 Iterative Verfahren

In diesem Kapitel betrachten wir **iterative Näherungsverfahren** für Systeme **nichtlinearer (oder auch linearer) algebraischer Gleichungen**:

$$\text{Finde } x^* \in \mathbb{R}^n : \; f(x^*) = 0\,. \tag{6.1}$$

Die Lösung x^* ist, wenn sie existiert, im Allgemeinen nicht eindeutig.

Wir beschränken uns auf den Fall $f : \mathbb{R}^n \to \mathbb{R}^n$, d. h.

$$f(x) = \begin{bmatrix} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_n(x_1, \ldots, x_n) \end{bmatrix}.$$

6.1 Kontraktion – Der Banachsche Fixpunktsatz

Ausser im linearen Fall ist man dabei im der Regel auf iterative Näherungsverfahren angewiesen, d. h. ausgehend von einem **Startwert** x_0 berechnet man weitere Näherungswerte $x_1, x_2, \ldots$ für x^* mit Hilfe einer **Iterationsfunktion**

$$\Phi : \mathbb{R}^n \to \mathbb{R}^n\,,$$

indem man

$$x_{i+1} = \Phi(x_i), \quad i = 0, 1, 2, \ldots \tag{6.2}$$

setzt. Wenn x^* ein **Fixpunkt** von Φ ist, d. h.

$$\Phi(x^*) = x^*\,, \tag{6.3}$$

und Φ in einer Umgebung U von x^* eine sog. **Kontraktion** ist, dann gilt für alle $x_0 \in U$, dass

$$x_i \longrightarrow x^*\,.$$

Um diese Aussage zu präzisieren, benötigen wir einige Begriffe. Im Folgenden bezeichne $\|\cdot\|$ eine beliebige Vektornorm im $\mathbb{R}^n$ sowie die zugehörige natürliche Matrixnorm.

Definition 6.1. Sei $\emptyset \neq U \subset \mathbb{R}^n$ eine abgeschlossene Menge. Eine Abbildung $g : U \to \mathbb{R}^n$ heißt **Lipschitz-stetig**, wenn es ein $L > 0$ gibt, sodass für alle $x, y \in U$

$$\|g(x) - g(y)\| \;\leq\; L\|x - y\| \,. \tag{6.4}$$

Ist die sog. *Lipschitz-Konstante* $L < 1$, so nennt man g eine **Kontraktion** auf U.

Satz 6.2 (Banachscher Fixpunktsatz)**.** *Sei $\tilde{x}_0 \in \mathbb{R}^n$ und sei*

$$B_r(\tilde{x}_0) := \{x \in \mathbb{R}^n \;\mid\; \|x - \tilde{x}_0\| < r\} \,,$$

d. h. der Ball mit Mittelpunkt $\tilde{x}_0$ und Radius r. Es gelte

(i) $\Phi : \overline{B_r(\tilde{x}_0)} \to \overline{B_r(\tilde{x}_0)}$

(ii) Φ *ist eine Kontraktion auf* $\overline{B_r(\tilde{x}_0)}$*, d. h. es gebe* $q \in (0, 1)$*, sodass für* $x, y \in \overline{B_r(\tilde{x}_0)}$ *gelte:* $\|\Phi(x) - \Phi(y)\| \leq q\|x - y\|$.

Dann gilt für jedes $x_0 \in \overline{B_r(\tilde{x}_0)}$, dass

(a) *die Fixpunktiteration $x_{k+1} = \Phi(x_k)$ zu einem $x^* \in \overline{B_r(\tilde{x}_0)}$ konvergiert, und dass*

(b) *x^* der eindeutige Fixpunkt von Φ in $\overline{B_r(\tilde{x}_0)}$ ist.*

Beweis. Sei $x_0 \in \overline{B_r(\tilde{x}_0)}$ beliebig. Aus Bedingung (i) folgt $x_k \in \overline{B_r(\tilde{x}_0)}$ für alle $k \in \mathbb{N}$ und

$$\begin{aligned} \|x_{k+1} - x_k\| &= \|\Phi(x_k) - \Phi(x_{k-1})\| \\ &\leq q\|x_k - x_{k-1}\| \\ &\leq q^k\|x_1 - x_0\| \,. \end{aligned}$$

Mit der Dreiecksungleichung folgt daraus für $k > \ell \geq 0$:

$$\begin{aligned} \|x_k - x_\ell\| &\leq \|x_k + x_{k-1}\| + \ldots + \|x_{\ell+1} - x_\ell\| \\ &\leq \left(q^{k-1} + q^{k-2} + \ldots + q^\ell\right) \|x_1 - x_0\| \\ &\leq \frac{q^\ell}{1-q}\|x_1 - x_0\| \,. \end{aligned}$$

Die rechte Seite konvergiert für $\ell \to \infty$ gegen 0, sodass $(x_k)_{k\geq 0}$ eine Cauchyfolge im abgeschlossenen Ball $\overline{B_r(\tilde{x}_0)} \subset \mathbb{R}^n$ ist, woraus folgt, dass der Limes $x^* \in \overline{B_r(\tilde{x}_0)}$ existiert.

Außerdem folgt aus (ii), dass

$$\begin{aligned}\|x^* - \Phi(x^*)\| &\leq \left\|x^* - *^{k+1}\right\| + \left\|\Phi(x^k) - \Phi(x^*)\right\| \\ &\leq \left\|x^* - x^{k+1}\right\| + q\left\|x^* - x^k\right\| \longrightarrow 0 \quad (k \to \infty)\,,\end{aligned}$$

und deshalb $\Phi(x^*) = x^*$, d. h. x^* ist ein Fixpunkt von Φ. Um die Eindeutigkeit zu zeigen, sei $x^* \neq \hat{x} \in \overline{B_r(\tilde{x}_0)}$ mit $\Phi(\hat{x}) = \hat{x}$. Da $q < 1$, gilt dann

$$\|\hat{x} - x^*\| = \|\Phi(\hat{x}) - \Phi(x^*)\| \leq q\|\hat{x} - x^*\| < \|\hat{x} - x^*\|\,,$$

was auf einen Widerspruch führt. Also gilt $\hat{x} = x^*$. □

Bemerkung. Für den Startwert $x_0 = \tilde{x}_0 \in \mathbb{R}^n$ und eine Iterationsfunktion Φ, die die Bedingung (ii) aus Satz 6.2 erfüllt, kann man Bedingung (i) durch folgende, leichter zu verifizierende Bedingung ersetzen, sodass Aussagen (a) und (b) aus Satz 6.2 weiterhin gelten:

$$\|x_0 - \Phi(x_0)\| \leq (1-q)r\,. \tag{6.5}$$

Es stellen sich nun folgende Fragen:

1. Wie findet man passende Iterationsfunktionen?
2. Unter welchen Bedingungen konvergiert die Folge x_i?
3. Wie schnell konvergiert die Folge der x_i?

Um diese Fragen zu beantworten, benötigen wir zunächst die folgende

Definition 6.3 (Konvergenzordnung)**.** Sei $(x_k)_{k\geq 0} \subset \mathbb{R}^n$ eine konvergente Folge mit Grenzwert $x^* \in \mathbb{R}^n$. Wir sagen

(i) x_k konvergiert gegen x^* mit **Konvergenzordnung** $p > 1$, falls $C \geq 0$ und $N \in \mathbb{N}$ existieren, sodass für alle $k \geq N$ gilt

$$\|x_{k+1} - x^*\| \leq C\|x_k - x^*\|^p\,.$$

(ii) x_k konvergiert linear gegen x^* mit **Konvergenzfaktor** $C \in [0,1)$, falls $N \in \mathbb{N}$ existiert, sodass für alle $k \geq N$ gilt

$$\|x_{k+1} - x^*\| \leq C\|x_k - x^*\|\,.$$

Für $p = 2$ spricht man von **quadratischer Konvergenz**.

Beispiel 6.4. (a) Bei linearer Konvergenz wird der Fehler $e_k = \|x_k - x^*\|$ in jedem Schritt mindestens um den Faktor $C < 1$ reduziert. Die Konvergenz ist umso schneller, je kleiner der Konvergenzfaktor C ist. Eine typische Zahlenfolge für $C = 0.2$ ist z. B.

$$e_0 = 0.2,\ e_1 = 0.04,\ e_2 = 0.008,\ e_3 = 0.0016,\ \ldots$$

(b) Bei quadratischer Konvergenz ist das Verhalten völlig anders. Eine typische Zahlenfolge für $C = 1$ ist z. B.

$$e_0 = 0.2,\ e_1 = 0.04,\ e_2 = 0.0016,\ e_3 = 0.00000256,\ \ldots$$

Die Konvergenz in Satz 6.2 ist linear mit Konvergenzfaktor $q < 1$, da

$$\|x_{k+1} - x^*\| = \|\Phi(x_k) - \Phi(x^*)\| \leq q\|x_k - x^*\|\,, \tag{6.6}$$

d. h. wir haben in Satz 6.2 bereits hinreichende Bedingungen für lineare Konvergenz eines Iterationsverfahrens. Wir betrachten nun die Frage, wie man geeignete Iterationsvorschriften Φ finden kann:

Beispiel 6.5. Für die Gleichung $f(x) := x - e^{-x} = 0$ liegt es nahe, die Iterationsvorschrift

$$x_{i+1} = e^{-x_i}, \quad i = 1, 2, \ldots, \text{ d. h. } \Phi(x) := e^{-x}\,,$$

zu verwenden. Die Konvergenz ist aber nur linear und relativ langsam mit $C = 0.57$:

i	x_i	$\lvert x_i - x^*\rvert$	$\dfrac{\lvert x_i - x^*\rvert}{\lvert x_{i-1} - x^*\rvert}$
0	1	0.43286	—
1	0.36788	0.19926	0.46034
2	0.69220	0.12505	0.62757
3	0.50047	0.06667	0.53318
4	0.60624	0.03910	0.58647
5	0.54540	0.02175	0.55627
6	0.57961	0.01247	0.57333

Für allgemeine Funktionen f betrachten wir nun einen systematischen Weg Iterationsfunktionen zu finden, die schneller konvergieren.

6.2 Das Newton-Verfahren

Sei vorerst $n = 1$ und sei x^* eine Nullstelle von $f : \mathbb{R} \to \mathbb{R}$. Wenn f genügend oft in einer Umgebung $U(x^*)$ von x^* differenzierbar ist, dann erhält man durch Taylorentwicklung (Satz 1.3) um einen Punkt $x_0 \in U(x^*)$

$$0 = f(x^*) = f(x_0) + (x^* - x_0)f'(x_0) + R_2(x^* - x_0) \tag{6.7}$$

mit

$$R_2(x^* - x_0) = (x^* - x_0)^2 \frac{f''(x_0 + \theta(x^* - x_0))}{2!}, \quad \theta \in (0, 1)\,.$$

Wenn $|x^* - x_0|$ klein genug ist, kann man den quadratischen Restterm vernachlässigen und erhält eine Gleichung, der die Nullstelle x^* näherungsweise genügt, d. h. es existiert $\tilde{x}^* \approx x^*$, sodass

$$0 = f(x_0) + (\tilde{x}^* - x_0)f'(x_0)\,,$$

bzw.

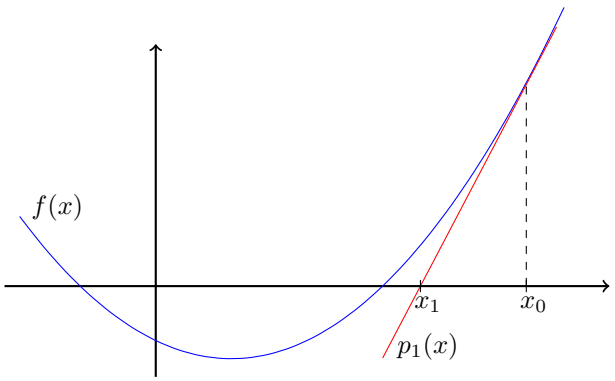
$$\tilde{x} = x_0 - \frac{f(x_0)}{f'(x_0)}\,.$$

Diese Näherungslösung kann man auch demselben Schema wieder verbessern und erhält so das **klassische Newton-Verfahren**

$$x_{i+1} = \Phi(x_i) \qquad \text{mit} \qquad \Phi(x) := x - \frac{f(x)}{f'(x)}\,. \tag{6.8}$$

Auf ähnliche Weise kann man auch Iterationsverfahren höherer Ordnung konstruieren, indem man die Taylorentwicklung in (6.7) erst nach dem Glied $(x^* - x_0)^3$ oder $(x^* - x_0)^4$ abbricht; das wird jedoch in der Praxis selten gemacht.

Geometrisch läuft das Newton-Verfahren (6.8) darauf hinaus, dass man die Funktion f am Punkt x_0 *linearisiert*, d. h. man berechnet die *Tangente* zu f an der Stelle x_0 und wählt als nächste Näherung x_1 den Punkt, wo die Tangente die x-Achse schneidet:



Der Vorteil des Newton-Verfahrens ist, dass die Iteration (nahe der Lösung) quadratisch konvergiert, was wir in Satz 6.9 beweisen werden.

Beispiel 6.6. Betrachten wir noch einmal $f(x) = x - e^{-x}$. Das Newton-Verfahren führt zu

$$\Phi(x) := x - \frac{x - e^{-x}}{1 + e^{-x}} = \frac{1 + x}{1 + e^{x}}$$

und zu folgenden Näherungslösungen:

i	x_i	$\lvert x_i - x^*\rvert$	$\dfrac{\lvert x_i - x^*\rvert}{\lvert x_i - x^*\rvert^2}$
0	1	4.329×10^{-1}	—
1	0.537883	2.926×10^{-2}	0.15617
2	0.566987	1.563×10^{-4}	0.18256
3	0.567143	4.421×10^{-9}	0.18096

Also eine **viel** schnellere Konvergenz als die Fixpunktiteration als im Beispiel 6.5.

Durch die selbe Vorgehensweise wie im $\mathbb{R}$, lässt sich das Newton-Verfahren für Gleichungssysteme (6.1) im $\mathbb{R}^n$ herleiten: Sei $x^* \in \mathbb{R}^n$ eine Nullstelle von $f : \mathbb{R}^n \to \mathbb{R}^n$ und sei $x_0 \in \mathbb{R}^n$ ein Näherungswert. Ferner sei f für $x = x_0$ differenzierbar. Die Matrix aller partiellen Ableitungen

$$\mathrm{D}f(x) := \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \cdots & \frac{\partial f_n}{\partial x_n}(x) \end{bmatrix} \in \mathbb{R}^{n \times n} \tag{6.9}$$

wir **Jacobi-Matrix** von f an x genannt. Taylorentwicklung im $\mathbb{R}^n$ (siehe Analysis) ergibt

$$0 = f(x^*) = f(x_0) + \mathrm{D}f(x_0)(x^* - x_0) + R_2(x^* - x_0)$$

mit

$$\|R_2(x^* - x_0)\| = \mathcal{O}\left(\|x^* - x_0\|^2\right).$$

Wie im eindimensionalen Fall können wir für $\|x^* - x_0\|$ klein genug den quadratischen Restterm vernachlässigen. Falls $\mathrm{D}f(x_0)$ nicht-singulär ist, kann dann die Gleichung

$$0 = f(x_0) + \mathrm{D}f(x_0)(\tilde{x}^* - x_0)$$

nach $\tilde{x}^*$ aufgelöst werden

$$\tilde{x}^* = x_0 - \mathrm{D}f(x_0)^{-1} f(x_0)$$

und wir erhalten das **Newton-Verfahren im** $\mathbb{R}^n$:

$$x_{k+1} = \underbrace{x_k - \mathrm{D}f(x_k)^{-1} f(x_k)}_{=:\Phi(x_k)}, \quad k = 0, 1, 2, \dots \tag{6.10}$$

Die Herleitung basierte auf zwei wesentlichen Voraussetzungen:

(i) $\mathrm{D}f(x_k)$ ist nicht-singulär für alle k;

(ii) x_0 ist genügend nahe an der Nullstelle x^*.

Wir werden diese Voraussetzungen nun präzisieren und die lokale quadratische Konvergenz des Newton-Verfahrens zeigen. Der folgende Lemma ist im Prinzip nur ein Korollar des Störungssatzes 5.17.

Lemma 6.7. *Es seien $R, L > 0$ und $x^* \in \mathbb{R}^n$, so dass f auf $\overline{B_R(x^*)}$ differenzierbar ist und für alle $x, y \in \overline{B_R(x^*)}$*

$$\|\mathrm{D}f(x) - \mathrm{D}f(y)\| \le L\|x - y\|\,, \tag{6.11}$$

d. h. $\mathrm{D}f : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ ist Lipschitz-stetig. Wenn $\mathrm{D}f(x^*)$ nicht-singulär ist, dann existiert ein $r \in (0, R]$, sodass $\mathrm{D}f(x)^{-1}$ für alle $x \in \overline{B_r(x^*)}$ existiert und

$$\left\|\mathrm{D}f(x)^{-1}\right\| \le 2\left\|\mathrm{D}f(x^*)^{-1}\right\|. \tag{6.12}$$

Beweis. Ähnlich wie im Beweis von Satz 5.17, setzen wir

$$B = \mathrm{D}f(x^*)^{-1}(\mathrm{D}f(x^*) - \mathrm{D}f(x))$$

und wenden darauf Lemma 5.16 an. Dazu muss $\|B\| < 1$ sein. Sei $\sigma := \|\mathrm{D}f(x^*)^{-1}\|$. Dann folgt aus (6.11), dass für alle $x \in \overline{B_R(x^*)}$:

$$\|B\| \le \|\mathrm{D}f(x^*)^{-1}\|\|\mathrm{D}f(x^*) - \mathrm{D}f(x)\| \le \sigma L\|x^* - x\|\,,$$

d. h. um zu garantieren, dass $\|B\| < 1$, reicht es

$$r = \min\left(R, \frac{1}{2L\sigma}\right)$$

zu wählen. Dann ist $\|B\| \le \frac{1}{2}$,

$$(I - B)^{-1} = \left(\mathrm{D}f(x^*)^{-1}\mathrm{D}f(x)\right)^{-1} = \mathrm{D}f(x)^{-1}\mathrm{D}f(x^*)$$

und somit existiert auch $\mathrm{D}f(x)^{-1}$ und

$$\left\|\mathrm{D}f(x)^{-1}\right\| = \left\|(I - B)^{-1}\mathrm{D}f(x^*)^{-1}\right\| \le \frac{1}{1 - \|B\|}\left\|\mathrm{D}f(x^*)^{-1}\right\| \le 2\left\|\mathrm{D}f(x^*)^{-1}\right\|. \quad \square$$

Wenn also x^* eine Nullstelle von f ist, folgt aus 6.7, dass das Newton-Verfahren (zumindest) in $\overline{B_r(x^*)}$ wohldefiniert ist. Um die Konvergenz des Newton-Verfahrens zu beweisen, benötigen wir noch den folgenden Mittelwertsatz im $\mathbb{R}^n$:

Lemma 6.8. *Es seien $x^* \in \mathbb{R}^n$ und $r > 0$. Wenn $f : \mathbb{R}^n \to \mathbb{R}^n$ auf $\overline{B_r(x^*)}$ stetig differenzierbar ist, dann gilt für alle $x, x' \in \overline{B_r(x^*)}$*

$$f(x') = f(x) + \left(\int_0^1 \mathrm{D}f\left(x + t(x' - x)\right) \,\mathrm{d}t \right) (x' - x) \,.$$

Beweis. Sei $\Phi(t) := f(x + t(x' - x))$. Dann folgt aus der Kettenregel, dass

$$\begin{aligned} \Phi_i'(t) &= \frac{\mathrm{d}}{\mathrm{d}t} f_i(x + t(x' - x)) = \sum_{j=1}^n \frac{\partial f_i}{\partial x_j}(x + t(x'x))(x_j' - x_j) \\ &= \big(\mathrm{D}f(x + t(x' - x))(x' - x)\big)_i \,, \end{aligned}$$

also

$$\Phi'(t) = \mathrm{D}f(x + t(x' - x))(x' - x) \,,$$

woraus mit dem Hauptsatz der Differential- und Integralrechnung folgt

$$\begin{aligned} f(x') - f(x) &= \Phi(1) - \Phi(0) \\ &= \int_0^1 \Phi'(t) \,\mathrm{d}t \\ &= \left(\int_0^1 \mathrm{D}f(x + t(x' - x)) \,\mathrm{d}t \right) (x' - x) \,. \end{aligned}$$

□

Satz 6.9 (Quadratische Konvergenz des Newton-Verfahrens)**.** *Es seien $U \subset \mathbb{R}^n$ offen und $f : U \to \mathbb{R}^n$ stetig differenzierbar. sodass $\mathrm{D}f$ auf U Lipschitz-stetig mit Lipschitz-Konstante $L > 0$ ist. Wenn ein $x^* \in U$ existiert mit $f(x^*) = 0$ und $\mathrm{D}f(x^*)$ nicht-singulär ist, dann gibt es ein $r > 0$, so dass für alle $x_0 \in \overline{B_r(x^*)}$ das Newton-Verfahren wohldefiniert ist und quadratisch gegen x^* konvergiert.*

Beweis. Es sei wieder $\sigma := \|\mathrm{D}f(x^*)^{-1}\|$. Unter den gegeben Voraussetzungen garantiert Lemma 6.7 die Existenz eines $0 < r \leq \frac{1}{2L\sigma}$, sodass $\mathrm{D}f$ auf $\overline{B_r(x^*)} \subset U$ Lipschitz-stetig und nicht-singulär ist mit $\|\mathrm{D}f(x)^{-1}\| \leq 2\sigma$.

Nehmen wir an, dass $x_k \in \overline{B_r(x^*)}$. Dann folgt aus der Definition des Newton-Verfahrens in (6.10), dass

$$\mathrm{D}f(x_k)(x_{k+1} - x^*) = \mathrm{D}f(x_k)(x_k - x^*) - f(x_k)$$

$$= \mathrm{D}f(x_k)(x_k - x^*) - (f(x_k) - f(x^*)) \,. \tag{6.13}$$

Aus Lemma 6.8 folgt, weil $x^* + t(x_k - x^*) \in \overline{B_r(x^*)}$ für alle $t \in [0,1]$, dass

$$f(x_k) - f(x^*) = \left(\int_0^1 \mathrm{D}f(x^* + t(x_k - x^*)) \, \mathrm{d}t \right) (x_k - x^*) \,.$$

Zusammen mit (6.13) ergibt das

$$\mathrm{D}f(x_k)(x_{k+1} - x^*) = \left(\int_0^1 (\mathrm{D}f(x_k) - \mathrm{D}f(x^* + t(x_k - x^*))) \, \mathrm{d}t \right) (x_k - x^*) \,.$$

Multipliziert man beide Seiten mit $\mathrm{D}f(x_k)^{-1}$ und nimmt die Norm davon, so gilt

$$\begin{aligned}
\|x_{k+1} - x^*\| &= \left\| \mathrm{D}f(x_k)^{-1} \left(\int_0^1 (\mathrm{D}f(x_k) - \mathrm{D}f(x^* + t(x_k - x^*))) \, \mathrm{d}t \right) (x_k - x^*) \right\| \\
&\leq \left\| \mathrm{D}f(x_k)^{-1} \right\| \left\| \int_0^1 (\mathrm{D}f(x_k) - \mathrm{D}f(x^* + t(x_k - x^*))) \, \mathrm{d}t \right\| \|x_k - x^*\| \\
&\overset{\text{Lemma 6.7}}{\leq} 2\sigma \int_0^1 \|\mathrm{D}f(x_k) - \mathrm{D}f(x^* + t(x_k - x^*))\| \, \mathrm{d}t \|x_k - x^*\| \\
&\overset{(6.11)}{\leq} 2\sigma L \int_0^1 \|(1-t)(x_k - x^*)\| \, \mathrm{d}t \|x_k - x^*\| \\
&= \sigma L \|x_k - x^*\|^2 \,.
\end{aligned} \tag{6.14}$$

Weil $r \leq \frac{1}{2L\sigma}$ folgt daraus, dass

$$\|x_{k+1} - x^*\| \leq \frac{1}{2} \|x_k - x^*\| \,.$$

Daraus ersieht man per Induktion, dass die vom Newton-Verfahren erzeugte Folge von Näherungslösungen $(x_k)_{k \geq 0} \subseteq \overline{B_r(x^*)}$ sofern $x_0 \in \overline{B_r(x^*)}$. Außerdem gilt

$$\|x_{k+1} - x^*\| \leq \frac{1}{2} \|x_k - x^*\| \leq \cdots \leq \frac{1}{2}^{k+1} \|x_0 - x^*\| \leq r \frac{1}{2}^{k+1} \,,$$

sodass $x_k \longrightarrow x^*$ wenn $k \to \infty$. Wegen (6.14) ist die Konvergenz quadratisch. □

Bemerkung 6.10. (a) Es gibt andere Beweise, die mit leicht schwächeren Voraussetzungen auskommen.

(b) Wenn es konvergiert, ist das Newton-Verfahren ein extrem effizientes Verfahren zur Lösung nichtlinearer Gleichungen.

(c) Das Newton-Verfahren konvergiert auch noch, wenn $\mathrm{D}f(x^*)$ singulär ist, allerdings nur mehr linear.

(d) Man kann das Newton-Verfahren zu einem global konvergenten Verfahren machen, indem man einen Dämpfungsfaktor λ_k einführt, der geeignet gesteuert

werden muss, d. h.

$$x_{k+1} = x_k - \lambda_k \mathrm{D}f(x_k)^{-1} f(x_k)$$

mit $\lambda_k \in (0, 1]$, sodass zu Beginn der Iteration Divergenz vermieden wird und nach endlich vielen Schritten $\lambda_k = 1$ gewählt werden kann.

(e) Ein weiteres Problem sind die Kosten, in jedem Schritt die Jacobi-Matrix zu berechnen und zu invertieren. Es gibt Varianten des Newton-Verfahrens, wo das vermieden wird, z. B. Quasi-Newton-Verfahren wie das BFGS Verfahren[1], die sehr effizient sind.

Wir werden aber auf keinen dieser Punkte genauer eingehen, sondern betrachten nun iterative Verfahren für lineare Gleichungssysteme.

6.3 Lineare Gleichungssysteme

Wir betrachten nun den Spezialfall

$$f(x^*) = b - Ax^* = 0\,, \tag{6.15}$$

d. h. lineare Gleichungssysteme mit $b \in \mathbb{R}^n$ und $A \in \mathbb{R}^{n\times n}$, und besprechen verschiedene Wahlen der Iterationsvorschrift $x_{k+1} = \Phi(x_k)$:

- **Newton.** Wegen $\frac{\partial f_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left(b_i - \sum_{l=1}^n a_{il}x_l\right) = -a_{ij}$ ist $\mathrm{D}f(x) = -A$ (für alle $x \in \mathbb{R}^n$) und das Newton-Verfahren reduziert sich auf

  $$x_{k+1} = \Phi(x_k) = x_k + A^{-1}(b - Ax_k)$$

  d. h. mit $x_0 = 0$ erhält man

  $$x_1 = A^{-1}b \tag{6.16}$$

  und das Verfahren konvergiert in *einer* Iteration. Das ist auch nicht verwunderlich, da in diesem Falle für das quadratische Restglied in der Taylorentwicklung zur Motivation des Newton-Verfahrens gilt $R_2(x^* - x_0) \equiv 0$. Aber für die Vorschrift (6.16) benötigt man A^{-1}, was wieder auf die direkten Faktorisierungsverfahren aus Kapitel 5 zurückführt.

- **Fixpunktiteration.** Dadurch kann man das berechnen von A^{-1} vermeiden. Die Konvergenz ist aber *nur mehr linear*. Wenn n groß ist und die Anzahl der Iterationen kleiner als n, dann zahlt sich das trotzdem aus im Bezug auf die Anzahl der arithmetischen Operationen — speziell für dünnbesetzte Matrizen, wo die Anzahl der Einträge $a_{ij} \neq 0$ viel kleiner als n^2 ist.

[1] Broyden-Fletcher–Goldfarb-Shanno Verfahren

Die einfachste Fixpunktiteration ist die **Richardson-Iteration**

$$\Phi(x) := x + \alpha(b - Ax) \tag{6.17}$$

für geeignet gewähltes $\alpha > 0$, sodass Φ eine Kontraktion ist. Allgemeiner lassen sich iterative Verfahren dieses Typs mittels der Vorschrift

$$\Phi(x) := x + C^{-1}(b - Ax) \tag{6.18}$$

konstruieren, mit geeignet gewähltem $C \in \mathbb{R}^{n\times n}$ regulär. Sei $A = L + D + R$ mit

$$L := \begin{pmatrix} 0 & & & \\ a_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix}, \; D := \begin{pmatrix} a_{11} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & a_{nn} \end{pmatrix}, \; R := \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & \ddots & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ & & & 0 \end{pmatrix}$$

dann führen die Wahlen $C = D$ bzw. $C = D+L$ auf das klassische **Jacobi-Verfahren**

$$x_{k+1} = x_k + D^{-1}(b - Ax_k) \tag{6.19}$$

bzw. auf das klassische **Gauß-Seidel-Verfahren**

$$x_{k+1} = x_k + (D + L)^{-1}(b - Ax_k)\,. \tag{6.20}$$

Beide Verfahren führen auf sehr einfache Iterationsvorschriften, wenn man sie komponentenweise betrachtet, z. B. für das Jacobi-Verfahren gilt

$$\begin{aligned} (x_{k+1})_i &= (x_k)_i + \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{n} a_{ij} (x_k)_j \right) \\ &= \frac{1}{a_{ii}} \left(b_i - \sum_{j\neq i} a_{ij} (x_k)_j \right) \qquad \text{für alle } i = 1, \ldots, n\,. \end{aligned} \tag{6.21}$$

und mit Vorwärtseinsetzen kann man für das Gauß-Seidel-Verfahren zeigen

MMS

$$(x_{k+1})_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} (x_{k+1})_j - \sum_{j>i} a_{ij} (x_k)_j \right) \quad \text{für alle } i = 1, \ldots, n\,. \tag{6.22}$$

Ein wichtiger Unterschied zwischen (6.21) und (6.22) ist, dass in (6.21) alle Komponenten unabhängig von- und parallel zueinander gelöst werden können, während in (6.22) zur Lösung der i-ten Gleichung die ersten $(i-1)$ Gleichungen bereits gelöst sein müssen. Das Gauß-Seidel-Verfahren lässt sich nicht so leicht parallelisieren und die Lösung hängt von der Reihenfolge der Gleichungen in (6.15) ab.

Die Konvergenz in (6.18) folgt direkt aus dem Banachschen Fixpunktsatz (Satz 6.2). Wir schreiben dazu

$$\Phi(x) = Bx + c \quad \text{mit} \quad B := I - C^{-1}A \text{ und } c := C^{-1}b\,, \tag{6.23}$$

sodass

$$x_{k+1} = Bx_k + c \tag{6.24}$$

und

$$e_{k+1} = Be_k \quad \text{mit} \quad e_k := x_k - x^*\,. \tag{6.25}$$

Korollar 6.11. *Falls $\|B\| < 1$, dann konvergiert x_k linear gegen x^* mit Konvergenzfaktor $\rho \leq \|B\|$.*

Beweis. Für beliebige $x, y \in \mathbb{R}^n$ gilt

$$\|\Phi(x) - \Phi(y)\| = \|B(x-y)\| \leq \|B\|\|x-y\|$$

und die Voraussetzungen von Satz 6.2 sind erfüllt (mit r groß genug). □

Ob die Voraussetzung $\|B\| < 1$ erfüllt ist, kann für eine konkrete Matrix sehr wohl von der speziellen Wahl der Matrixnorm abhängen.

Definition 6.12. Der **Spektralradius** einer Matrix $A \in \mathbb{R}^{n\times n}$ ist

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

Aus Proposition 5.10 wissen wir, dass

$$\operatorname{spr}(B) \leq \|A\| \tag{6.26}$$

für alle natürlichen Matrixnormen. Für symmetrisches B folgt aus (5.16), dass

$$\operatorname{spr}(B) = \|B\|_2 \quad (\text{die Spektralnorm})\,.$$

Satz 6.13. *Die Fixpunktiteration* (6.24) *konvergiert* ***genau dann*** *für jeden Startwert $x_0 \in \mathbb{R}^n$,* ***wenn***

$$\operatorname{spr}(B) < 1\,. \tag{6.27}$$

Beweis. Für B symmetrisch folgt das Resultat aus Korollar 6.11 und (6.26). Dass (6.24) *nicht* konvergiert, wenn $\operatorname{spr}(B) \geq 1$ sieht man, indem man

$$x_= := x^* - v_{\max}$$

wählt, wobei $v_{\max}$ Eigenvektor zum größten Eigenwert $\lambda_{\max}$ von B ist. Dann folgt für alle $k \in \mathbb{N}$

$$\|e_k\|_2 = \|Be_{k-1}\|_2 = \ldots = \left\|B^k e_0\right\|_2 = |\lambda_{\max}|^k \|v_{\max}\| = \operatorname{spr}(B)^k \geq 1\,.$$

Für den sehr technischen Beweis für nicht-symmetrische $B \in \mathbb{R}^{n\times n}$ siehe [Ran17, Satz 6.1 und Hilfssatz 6.1]. □

Wir betrachten nun unter welchen Bedingungen an A das Jacobi- und das Gauß-Seidel-Verfahren konvergieren. Dazu benötigen wir die folgende

Definition 6.14. Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt **diagonaldominant**, wenn

$$\sum_{j\neq i} |a_{ij}| \leq |a_{ii}|, \quad \text{für alle } i = 1, \ldots, n\,, \tag{6.28}$$

und **strikt diagonaldominant**, wenn

$$\sum_{j\neq i} |a_{ij}| < |a_{ii}|, \quad \text{für alle } i = 1, \ldots, n\,. \tag{6.29}$$

Für das Jacobi- bzw. Gauß-Seidel-Verfahren gilt

$$B_{\mathrm{J}} := -D^{-1}(L+R) \quad \text{bzw.} \quad B_{\mathrm{GS}} := -(D+L)^{-1}R\,. \tag{6.30}$$

Satz 6.15. *Ist $A \in \mathbb{R}^{n\times n}$ strikt diagonaldominant, so gilt*

$$\operatorname{spr}(B_{\mathrm{J}}) < 1 \quad \textit{und} \quad \operatorname{spr}(B_{\mathrm{GS}}) < 1\,.$$

Beweis. Sei $\lambda \in \sigma(B_{\mathrm{J}})$ und v der zugehörige Eigenvektor. Aus (6.28) folgt $a_{ii} \neq 0$, d. h.

$$\lambda v = B_{\mathrm{J}} v = -D^{-1}(L+R)v\,.$$

Daraus folgt

$$|\lambda| \leq \left\|D^{-1}(L+R)\right\|_\infty = \max_{i=1}^{n} \left\{ \frac{1}{|a_{ii}|} \sum_{j\neq i} |a_{ij}| \right\} < 1\,.$$

Sei nun $\lambda \in \sigma(B_{\mathrm{GS}})$ mit zugehörigem Eigenvektor v. Dann gilt

$$\begin{aligned} \lambda v = -(D+L)^{-1}Rv \quad &\Leftrightarrow \quad \lambda(D+L)v = -Rv \\ &\Leftrightarrow \quad \lambda v = -D^{-1}(\lambda L + R)v\,, \end{aligned}$$

woraus folgt

$$|\lambda| \leq \left\| D^{-1}(\lambda L + R) \right\|_\infty \leq \max_{i=1}^{n} \left\{ \frac{1}{|a_{ii}|} \left(|\lambda| \sum_{j<i} |a_{ij}| + \sum_{j>i} |a_{ij}| \right) \right\} .$$

Im Falle $|\lambda| \geq 1$ ergäbe sich der Widerspruch

$$|\lambda| \leq |\lambda| \left\| D^{-1}(L + R) \right\|_\infty < |\lambda| ,$$

sodass $|\lambda| < 1$ sein muss. □

Strikte Diagonaldominanz ist nur ein hinreichendes Kriterium. In der Praxis ist es zu restriktiv und schließt z. B. typische Modellprobleme im Bereich der numerischen Lösung von PDEs aus. Um ein schwächeres Kriterium anzugeben, benötigen wir

Definition 6.16. Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt **reduzibel**, wenn es eine Permutationsmatrix $P \in \mathbb{R}^{n\times n}$ gibt, sodass

$$PAP^\intercal = \begin{bmatrix} \tilde{A}_{11} & 0 \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} \tag{6.31}$$

mit $\tilde{A}_{11} \in \mathbb{R}^{p\times p}$, $\tilde{A}_{22} \in \mathbb{R}^{q\times q}$, $\tilde{A}_{21} \in \mathbb{R}^{q\times p}$ und $p+q = n$. Eine Matrix $A \in \mathbb{R}^{n\times n}$, die nicht reduzibel ist, heißt **irreduzibel**.

Bemerkung 6.17. Die Reduzibilität von A lässt sich auch wie folgt formulieren: Es existiert eine (nicht–triviale) Zerlegung $J \cup K$ der Indexmenge $\{1, ..., n\}$, mit $J, K \neq \emptyset$ und $J \cap K = \emptyset$, so dass $a_{jk} = 0$ für alle Paare $(j, k) \in J \times K$.

Umgekehrt bedeutet das zwingendermaßen, dass es für jede irreduzible Matrix A und für beliebige Indizes $j \neq k \in \{1, \ldots, n\}$, eine Kette von Indizes $j = i_1, i_2, \ldots, i_m = k$ gibt, so dass

$$a_{i_1 i_2} \neq 0, \quad a_{i_1 i_2} \neq 0, \quad \ldots , \quad a_{i_{m-1} i_m} \neq 0. \tag{6.32}$$

Satz 6.18. *Ist $A \in \mathbb{R}^{n\times n}$ eine irreduzible, diagonaldominante Matrix und existiert ein $r \in \{1, \ldots, n\}$, sodass*

$$\sum_{j\neq r} |a_{rj}| \ < \ |a_{rr}| , \tag{6.33}$$

so ist A regulär und es gilt

$$\mathrm{spr}(B_{\mathrm{J}}) < 1 \quad \textit{und} \quad \mathrm{spr}(B_{\mathrm{GS}}) < 1 .$$

Beweis. Aus der Irreduzibilät von A folgt notwendigerweise

$$\sum_{k=1}^{n} |a_{jk}| > 0, \quad \text{für alle } \; 1 \le j \le n,$$

da sonst für dieses j, mit $J = \{j\}$ und $K = \{1, \ldots, n\} \backslash J$ und $k \in K$ gilt $a_{jk} = 0$, im Widerspruch zur Irreduzibilität von A. Wegen der Diagonaldominanz folgt daraus notwendigerweise $a_{jj} \neq 0$, für alle $1 \le j \le n$, was die Wohlgestelltheit des Jacobi- und des Gauß-Seidel Verfahrens garantiert.

Analog zum Beweis von Satz 6.15 zeigt man

$$\mathrm{spr}(B_{\mathrm{J}}) \le 1 \quad \text{und} \quad \mathrm{spr}(B_{\mathrm{GS}}) \le 1\,.$$

Um zu zeigen, daß $\mathrm{spr}(B_{\mathrm{J}}) < 1$, nehmen wir an, dass $\lambda \in \mathbb{C}$ ein Eigenwert von B_{J} ist, mit $|\lambda| = 1$ und zugehörigem Eigenvektor $v \in \mathbb{C}^n$, normiert so dass $|v_s| = \|v\|_\infty = 1$ und $|v_j| \le |v_s|$ für $j \neq s$. Für dieses Eigenpaar gilt

$$|v_j| = |\lambda v_j| = |(B_{\mathrm{J}} v)_j| = \left| \frac{1}{a_{jj}} \sum_{k \neq j} a_{jk} v_k \right| \le \frac{1}{|a_{jj}|} \sum_{k \neq j} |a_{jk}||v_k|. \tag{6.34}$$

Aufgrund der Irreduzibilität gibt es nun Indizes $s = i_1, i_2, \ldots, i_m = r$, so dass $a_{i_\ell i_{\ell+1}} \neq 0$ für alle $1 \le \ell \le m-1$, wobei r der Index aus Bedingung (6.33) ist. Aus (6.34) und (6.33) folgt

$$|v_r| \le \frac{1}{|a_{rr}|} \sum_{k \neq r} |a_{rk}||v_k| \le \left(\frac{1}{|a_{rr}|} \sum_{k \neq r} |a_{rk}| \right) \|v\|_\infty < \|v\|_\infty\,.$$

Aber nun folgt (rekursiv) für jedes $1 \le \ell \le m-1$ auch wieder aus (6.34), dass

$$\begin{aligned}
|v_{i_\ell}| &\le \frac{1}{|a_{i_\ell i_\ell}|} \left(\sum_{k \neq i_\ell, i_{\ell+1}} |a_{i_\ell k}||v_k| + |a_{i_\ell i_{\ell+1}}||v_{i_{\ell+1}}| \right) \\
&\le \frac{1}{|a_{i_\ell i_\ell}|} \left(\sum_{k \neq i_\ell, i_{\ell+1}} |a_{i_\ell k}| \|v\|_\infty + \underbrace{|a_{i_\ell i_{\ell+1}}|}_{\neq 0} |v_{i_{\ell+1}}| \right) \\
&< \underbrace{\left(\frac{1}{|a_{i_\ell i_\ell}|} \sum_{k \neq i_\ell} |a_{i_\ell k}| \right)}_{\le 1} \|v\|_\infty \le \|v\|_\infty\,.
\end{aligned}$$

Für $\ell = 1$, führt das auf $\|v\|_\infty = |v_s| = |v_{i_1}| < \|v\|_\infty$ und deshalb zu einem Widerspurch. Also muss $\mathrm{spr}(B_{\mathrm{J}}) < 1$ sein. Wegen $A = D(I - B_{\mathrm{J}})$ folgt daraus auch die Regularität von A. Der Beweis für B_{GS} folgt analog. □

Um die Konvergenzrate zu verbessern, kann man sowohl beim Jacobi-, als auch beim Gauß-Seidel-Verfahren auch noch einen Relaxationsparameter einführen, d. h.

$$x_{k+1} = x_k + \omega C^{-1}(b - Ax_k)\,. \tag{6.35}$$

Beim Gauß-Seidel-Verfahren, $C = D + L$, führt das auf das sog. **SOR-Verfahren**[2] mit

MMS

$$B_{\mathrm{SOR}} := (D + \omega L)^{-1}((1-\omega)D + \omega R)\,, \tag{6.36}$$

welches für symmetrisch positiv definite Matrizen für alle $\omega \in (0, 2)$ konvergiert, d. h. $\mathrm{spr}(B_{\mathrm{SOR}}) < 1$. Wir werden das aber nicht weiter untersuchen, weil die Konvergenz für typische, in der Praxis auftretende Gleichungssysteme trotzdem viel zu langsam ist. Der Relaxationsparameter ω spielt jedoch eine wichtige Rolle, wenn man das Jacobi- bzw. Gauß-Seidel-Verfahren als sog. **Glätter** in einem **Mehrgitterverfahren** verwenden will.

6.4 Anwendung: Numerische Lösung von Differentialgleichungen

Ein typisches Modellproblem in der Numerik ist die Lösung der Poisson-Gleichung

$$\begin{aligned} -\Delta u(x) &= f(x) \quad && \text{für alle } x \in \Omega \subset \mathbb{R}^d \\ u(x) &= 0 && \text{für alle } x \in \partial\Omega\,, \end{aligned} \tag{6.37}$$

wobei Ω ein beschränktes Gebiet im $\mathbb{R}^d$ ist und $f : \Omega \to \mathbb{R}$ eine gegebene Funktion. Im Fall $d = 1$, für $\Omega = (0, 1)$, reduziert sich (6.37) auf

$$\begin{aligned} -u''(x) &= f(x) \quad && \text{für alle } x \in (0,1) \\ u(0) = u(1) &= 0\,. \end{aligned} \tag{6.38}$$

Taylorentwicklung an einem Punkt $x \in (0, 1)$ ergibt

$$\begin{aligned} u(x+h) &= u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u^{(3)}(x) + \mathcal{O}(h^4) \\ u(x-h) &= u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u^{(3)}(x) + \mathcal{O}(h^4)\,, \end{aligned}$$

also

$$u(x+h) + u(x-h) = 2u(x) + h^2u''(x) + \mathcal{O}(h^4)$$

und somit

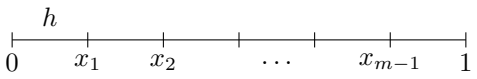
$$u''(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + \mathcal{O}(h^2)\,, \tag{6.39}$$

[2] engl. **S**uccessive **O**ver **R**elaxation Method

der sog. **zentrale Finite Differenzenquotient**. Betrachten wir also (6.38) nur an den Gitterpunkten

$$x_i = ih \quad \text{für } i = 1, \dots, m-1 \quad \text{und } h = \frac{1}{m}$$

d. h.



Dann lässt sich unter Vernachlässigung des $\mathcal{O}(h^2)$ Terms in (6.39) die Differentialgleichung (6.38) an x_i approximieren durch

$$\frac{1}{h^2}(-u_{i-1} + 2u_i - u_{i+1}) = f(x_i)\,,$$

wobei $u_i \approx u(x_i)$ und $u_0 = u_m = 0$ aufgrund der Randbedingungen. Das führt auf das lineare Gleichungssystem

$$Au := \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{m-1} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{m-1} \end{pmatrix} := \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{m-1}) \end{pmatrix}. \tag{6.40}$$

Mann kann zeigen[3], dass

$$\max_{1 \le i \le m-1} |u_i - u(x_i)| = \mathcal{O}(h^2),$$

d.h. um den Diskretisierungsfehler unter eine vorgegebene Schranke zu reduzieren, muss m genügend gross gewählt werden. Wir diskutieren das aber nicht mehr weiter.

Die Matrix A in (6.40) ist offensichtlich irreduzibel, diagonaldominant und für $r = 1$ oder $r = m-1$ gilt (6.33). Deshalb konvergieren laut Satz 6.18 sowohl das Jacobi- als auch das Gauß-Seidel-Verfahren. (Man kann auch leicht zeigen, dass A symmetrisch positiv definit ist , sodass auch das SOR-Verfahren für $\omega \in (0, 2)$ konvergiert.) Durch Nachrechnen sieht man, dass die Eigenwerte und -vektoren von A

MMS

$$\lambda_k = 4m^2 \sin^2\left(\frac{k\pi}{2m}\right) \quad \text{und} \quad v_k = \left(\sin\frac{k\pi j}{m}\right)_{j=1}^{m-1}$$

sind (Übungsaufgabe mithilfe der trigonometrischen Additionssätze).

MMS

Daraus folgert man:

[3]Siehe die Vorlesungen *Numerik 1 – Numerik Gewöhnlicher Differentialgleichungen.*

Satz 6.19. *Sei A die Matrix aus* (6.40)*. Dann gilt*

$$\|B_{\mathrm{J}}\|_2 = \mathrm{spr}(B_{\mathrm{J}}) = \cos\left(\frac{\pi}{m}\right) = 1 - \frac{1}{2}\left(\frac{\pi}{m}\right)^2 + \mathcal{O}(m^{-4})\,.$$

Beweis. Sei $\mu_k \in \sigma(B_{\mathrm{J}})$ mit Eigenvektor w_k. Da $D = 2m^2 I$ gilt

$$\begin{aligned} B_{\mathrm{J}} w_k = \mu_k w_k \quad &\Leftrightarrow \quad (L+R)w_k = -2m^2\mu_k w_k \\ &\Leftrightarrow \quad Aw_k = 2m^2(1-\mu_k)w_k \end{aligned}$$

und somit

$$w_k = v_k \quad \text{und} \quad 2m^2(1-\mu_k) = \lambda_k$$

bzw.

$$\mu_k = 1 - 2\sin^2\left(\frac{k\pi}{2m}\right)\,.$$

Das Maximum wird für $k = 1$ angenommen, sodass

$$\mathrm{spr}(B_{\mathrm{J}}) = \max_{k=1}^{m-1} |\mu_k| = 1 - 2\sin^2\left(\frac{k\pi}{2m}\right) = \cos\left(\frac{\pi}{m}\right)\,.$$

Durch Taylorentwicklung bei $x = 0$ sieht man, dass

$$\begin{aligned} \cos\left(\frac{\pi}{m}\right) &= \cos(0) - \frac{\pi}{m}\sin(0) - \frac{1}{2}\left(\frac{\pi}{m}\right)^2\cos(0) + \frac{1}{6}\left(\frac{\pi}{m}\right)^3\sin(0) + \mathcal{O}(m^{-4}) \\ &= 1 - \frac{1}{2}\left(\frac{\pi}{m}\right)^2 + \mathcal{O}(m^{-4}). \end{aligned}$$

□

Die Konvergenzrate des Jacobi-Verfahrens ist also sehr schlecht. Man kann zeigen, dass $\mathrm{spr}(B_{\mathrm{GS}}) = \mathrm{spr}(B_{\mathrm{J}})^2$ (siehe dafür [Ran17, Satz 6.5]), also konvergiert das Gauß-Seidel-Verfahren zwar doppelt so schnell wie das Jacobi-Verfahren, aber immer noch sehr schlecht.

m	$\mathrm{spr}(B_{\mathrm{J}})$	$\mathrm{spr}(B_{\mathrm{GS}})$
20	0.98769	0.97553
40	0.99692	0.99384
80	0.99923	0.99846
160	0.99981	0.99961

Tabelle 6.1: Spektralradien für die Matrizen B_{J} und B_{GS} aus Beispiel 6.20.

Beispiel 6.20. Nach k Iterationen mit $u_0 = 0$ gilt

$$\|u - u_k\|_2 \leq \mathrm{spr}(B)\|u - u_{k-1}\|_2 \leq \cdots \leq (\mathrm{spr}(B))^k\|u\|_2\,,$$

sodass für den relativen Fehler gilt

$$\frac{\|u - u_k\|_2}{\|u\|_2} \;\leq\; \mathrm{spr}(B)^k \,.$$

Um zu garantieren, dass der relative Fehler kleiner $\epsilon = 10^{-6}$ ist, muss

$$\mathrm{spr}\,(B)^k \leq \epsilon \,,$$

also

$$k \geq \frac{\log \epsilon}{\log\left(\mathrm{spr}(B)\right)}$$

sein, d. h. für $m = 160$ (siehe Tabelle 6.1)

$$\begin{aligned} k_{\mathrm{J}} \;&\geq\; \frac{\log(10^{-6})}{\log(0.99981)} \approx 72706.3 \\ k_{\mathrm{GS}} \;&\geq\; \frac{\log(10^{-6})}{\log(0.99961)} \approx 36353.1 \,, \end{aligned}$$

Bemerkung 6.21. (a) Wie bereits erwähnt, sind das ω-Jacobi- und das SOR-Verfahren gute **Glätter**, d. h. durch geeignete Wahl von ω (z. B. $\omega = \frac{2}{3}$ für Jacobi) kann man garantieren, dass die hochfrequenten Anteile im Fehler $e_0 = u - u_0$ in jeder Iteration stark gedämpft werden (z. B. um einen Faktor $\frac{1}{3}$ bei Jacobi), sodass nach wenigen Iterationen e_k glatt ist und auf einem gröberen Gitter approximativ berechnet werden kann. Das ist die Basis des **Mehrgitterverfahrens**, einem sehr effizienten Iterationsverfahren für (6.38) und vor allem für (6.37) in höheren Dimensionen[4].

(b) Man kann auch Blockvarianten des Jacobi- und Gauß-Seidel-Verfahrens herleiten, welche die Basis der ebenfalls sehr effizienten **Gebietszerlegungsverfahren**[5] bilden.

[4] Siehe die Vorlesungen *Numerik 2 – Finite Elemente* und *Parallel Solution of Large Linear Systems*.
[5] Siehe Vorlesung *Parallel Solution of Large Linear Systems*.

6.5 Abstiegsverfahren (nicht prüfungsrelevant)

Eine viel effizientere Familie von iterativen Verfahren zur Lösung von (quadratischen) linearen Gleichungssystemen $Ax = b$ im $\mathbb{R}^n$, vor allem von dünnbesetzten Systemen, die bei der Diskretisierung von Differentialgleichungen auftreten, basiert auf der Minimisierung eines geeigneten quadratischen Funktionals. Für SPD Matrizen ist es das Funktional

$$Q(x) := \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle.$$

Wir skizzieren die Verfahren nur kurz und verweisen für Details auf [Ran17, Kap. 6.2].

Zum Minimisieren eines solchen quadratischen Funktionals eignen sich sog. **Abstiegsverfahren** bei denen, ausgehend von einem Startwert x_0, in jeder Iteration eine **Abstiegsrichtung** r_k gewählt und die nächste Näherungslösung x_{k+1} durch eindimensionale Minimisierung entlang dieser Abstiegsrichtung bestimmt wird, d.h.

$$x_{k+1} = x_k + \alpha_k r_k \quad \text{mit} \quad \alpha_k = -\frac{\langle g_k, r_k\rangle}{\langle Ar_k, r_k\rangle},$$

wobei

$$g_k := \mathrm{grad}Q(x_k) = Ax_k - b \quad \text{und} \quad \mathrm{grad}Q(y) = \left(\frac{\partial Q}{\partial y_i}(y)\right)_{i=1}^{n}$$

gerade der **Gradient** des quadratischen Funktionals Q ist.

Wählt man in jedem Schritt die Richtung $r_k = -g_k$ des stärksten (lokalen) Abfalls des Funktionals $Q(\cdot)$, so erhält man das **Gradientenverfahren**. Die lineare Konvergenz des Gradientenverfahrens lässt sich relativ leicht zeigen [Ran17, Satz 6.7]. Die Konvergenzrate hängt von der Konditionszahl $\kappa(A)$ ab [Ran17, Satz 6.8]. Vor allem bei Matrizen mit weit auseinander liegenden Eigenwerten, wie z.B. bei Gleichungssytem(6.40), kann das wieder zu einer sehr langsamen Konvergenz führen.

Das Gradientenverfahren nutzt die Struktur des quadratischen Funktionals $Q(\cdot)$, und im Speziellen die Verteilung der Eigenwerte der Matrix A, nur lokal von einem Schritt zum nächsten aus. Es ist besser, bei der Wahl der Abstiegsrichtungen die bereits gewonnen Informationen über die globale Struktur von $Q(\cdot)$ mit zu berücksichtigen. Das führt auf das **CG-Verfahren** (engl.: **C**onjugate **G**radient method = "Verfahren der konjugierten Gradienten"). Anstatt einer eindimensionalen "Liniensuche" entlang der gewählten Abstiegsrichtung, minimiert man beim CG-Verfahren im k-ten Iterationsschritt in einem k-dimensionalen Unterraum des $\mathbb{R}^n$, dem sog. **Krylov Unterraum**

$$\mathcal{K}_k(A, b) := \mathrm{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}.$$

(Hier der Einfachheit halber nur für den Fall $x_0 = 0$.) Das garantiert die A-Orthogonalität der Abstiegsrichtungen, d.h. Orthogonalität im Skalarprodukt $\langle x, y\rangle_A := \langle Ax, y\rangle_2$ und verbessert die Konvergenzeigenschaften signifikant.

Algorithmus 6.1 (CG-Verfahren).

function CG(A, b, x_0, x_K) ▷ Eingabe: $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ und Startpunkt $x_0 \in \mathbb{R}^n$
▷ Ausgabe: Näherung $x_K \in \mathbb{R}^n$ nach K Iterationen
$d_0 = -g_0 = b - Ax_0$ ▷ Initiale Abstiegsrichtung
for $k = 0, \dots, K-1$ **do**
$\alpha_k = \dfrac{\|g_k\|^2}{\langle Ad_k, d_k\rangle}$
$x_{k+1} = x_k + \alpha_k d_k$
$g_{k+1} = g_k + \alpha_k A d_k$
$\beta_k = \dfrac{\|g_{k+1}\|^2}{\|g_k\|^2}$
$d_{k+1} = -g_k + \beta_k d_k$
end for
end function

Die Orthogonalisierung der neuen Abstiegsrichtung zu den vorangegangenen Richtungen muss nicht explizit durchegführt werden, sondern die nächste Richtung kann mittels einer einfachen Rekursionsformel aus dem Gradienten und der vorhergehenden Richtung berechnet werden [Ran17, Seite 206], wie in Algorithmus 6.1 zusammengefasst ist. In der Praxis, stoppt man nicht nach einer fixen Anzahl von Iterationsschritten sondern man baut auch noch ein Konvergenzkriterium ein.

Die folgenden beiden Sätze fassen das prinzipielle Konvergenzverhalten des CG-Verfahrens zusammen. Für Beweise verweisen wir wieder auf [Ran17].

Satz 6.22 ([Ran17, Satz 6.9]). *Für eine beliebige SPD Matrix $A \in \mathbb{R}^{n\times n}$, eine beliebige rechte Seite $b \in \mathbb{R}^n$ und einen beliebigen Startpunkt $x_0 \in \mathbb{R}^n$ konvergiert das CG Verfahren (bei rundungsfreier Rechnung) in maximal n Iterationen zur exakten Lösung $x \in \mathbb{R}^n$, d.h. $x_n = x$.*

Satz 6.23 ([Ran17, Satz 6.10]). *Für das CG-Verfahren gilt die Fehelerabschätzung*

$$\|x_k - x\|_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|x_0 - x\|_A\,,$$

wobei $\kappa = cond_2(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$ die Konditionszahl von A bzgl. der Spektralnorm ist und $\lambda_{\max}(A)$ bzw. $\lambda_{\min}(A)$ den grössten bzw. kleinsten Eigenwert von A bezeichnen. Die Anzahl an Iterationsschritten, um den Anfangsfehler um den Faktor ε zu reduzieren, ist beschränkt durch

$$K(\varepsilon) \leq \tfrac{1}{2}\sqrt{\kappa}\log(2/\varepsilon) + 1.$$

Da die Konvergenzrate von der Konditionszahl der Matrix A abhängt bietet es sich an, das Gleichungssystem zuerst mit einer geeigneten SPD Matrix $P \in \mathbb{R}^{n\times n}$ zu multiplizieren, d.h.

$$\widetilde{A}x := PAx = Pb =: \widetilde{b}$$

und dann das CG-Verfahren auf $\widetilde{A}x = \widetilde{b}$ anzuwenden. Das ist möglich, da dieses System symmetrisch positiv definit bzgl. des Skalarprodukts $\langle x, y\rangle_{P^{-1}} := \langle P^{-1}x, y\rangle$ ist. Diese Vorgehensweise nennt man **Vorkonditionierung** und das daraus resultierende Verfahren nennt man das **PCG-Verfahren** (engl. **P**reconditioned **C**onjugate **G**radients). Wenn P billig invertiert werden kann und $\mathrm{cond}_2(\widetilde{A}) \ll \mathrm{cond}_2(A)$, dann konvergiert dieses Verfahren in wesentlich weniger Iterationen und ist viel effizienter.

Krylov Unterraumverfahren kann man auch für allgemeine quadratische Gleichungssysteme mit unsymmetrischer, aber regulärer Systemmatrix $A \in \mathbb{R}^{n\times n}$ herleiten. Üblicherweise betrachtet man hierfür wieder die Normalengleichung $A^TAx = A^Tb$, arbeitet aber nicht direkt mit der schlechter konditionierten Matrix A^TA, sondern baut dann zwei Krylovräume, bzgl. A und A^T auf. Wir behandeln das aber nicht mehr weiter sondern verweisen auf das

Proseminar/Seminar Krylov Methoden (Numerik),

das im SS 2021 in Heidelberg angeboten wird.

7 Eigenwertprobleme (nicht prüfungsrelevant)

Eine weitere wichtige Aufgabe der Numerik ist die Berechnung von Eigenwerten und Eigenvektoren von quadratischen Matrizen $A \in \mathbb{R}^{n\times n}$, sowie von Singulärwerten von allgemeinen Matrizen $A \in \mathbb{R}^{m\times n}$. Letzteres kann i. Allg. auf die Berechnung von Eigenwerten zurückgeführt werden, und wir besprechen nur kurz Verfahren zur Lösung von Eigenwertproblemen. Für Details verweisen wir wieder auf [Ran17, Kapitel 7].

Da die Nullstellenberechnung von Polynomen ein hochgradig schlecht konditioniertes Problem ist während das eigentliche Eigenwertproblem meist gut konditioniert ist, ist es i. Allg. nicht angeraten, Eigenwerte direkt auf Grund des charakteristischen Polynoms zu berechnen. Nur für Tridiagonalmatrizen bzw. Hessenberg-Matrizen, d.h.

$$\begin{pmatrix} a_{11} & a_{12} & & \\ a_{21} & \ddots & \ddots & \\ & \ddots & \ddots & a_{n-1,n} \\ & & a_{n,n-1} & a_{n,n} \end{pmatrix} \quad \text{bzw.} \quad \begin{pmatrix} a_{11} & a_{12} & \vdots & a_{1,n} \\ a_{21} & \ddots & & \dots \\ & \ddots & \ddots & a_{n-1,n} \\ & & a_{n,n-1} & a_{n,n} \end{pmatrix}$$

ist dies stabil und effizient durchführbar. Deshalb ist es besser die Matrix zuerst durch Reduktionsverfahren auf Hessenberg-Form zu transformieren.

Das derzeit effizienteste und stabilste Verfahren zur Lösung des Eigenwertproblems für Hessenberg-Matrizen ist das **QR-Verfahren** (siehe [Ran17, Kapitel 7.4]).

In vielen Anwendungen in der Praxis ist man nur an einem oder an einer kleinen Anzahl von Eigenwerten der Matrix A und den zugehörigen Eigenvektoren interessiert (siehe Beispiel 7.1 unten).

Zur Berechnung des betragsgrössten (auch **dominant** genannten) Eigenwerts $\lambda_n \in \mathbb{C}$ und des zugehörigen Eigenvektors $v_n \in \mathbb{C}^n$ mit $\|v_n\| = 1$, sodass

$$Av_n = \lambda_n v_n,$$

gibt es ein besonders einfaches Verfahren, die sog. **Potenzmethode** in Alg. 7.1.

Seien $\lambda_1, \dots, \lambda_n$ die Eigenwerte der Matrix A (mit Vielfachheiten gezählt), sodass $|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_n|$, und seien $v_1, \dots, v_n$ die zugehörigen Eigenvektoren. Für eine

Algorithmus 7.1 (Potenzmethode).

function CG(A, w_0, w_K, μ_K) ▷ Eingabe: $A \in \mathbb{R}^{n\times n}$ und Startvektor $w_0 \in \mathbb{C}^n$
▷ Ausgabe: Näherungslösung $(\mu_K, w_K) \in \mathbb{C} \times \mathbb{C}^n$
▷ von (λ_n, v_n) nach K Iterationen

for $k = 1, \dots, K$ **do**

$y_k = Aw_{k-1}$

$w_k = \dfrac{y_k}{\|y_k\|}$

end for

$\mu_K = \langle Aw_K, w_K \rangle$

end function

dagonalisierbare Matrix A (siehe Appendix A) bilden diese Eigenvektoren eine Basis von $\mathbb{C}^n$. Sofern der betragsgrösste Eigenwert einfach und deshalb separiert vom Rest des Spektrums ist, d.h. $|\lambda_n| > |\lambda_{n-1}|$, kann man sehr leicht zeigen, dass

$$\mu_K \to \lambda_n \quad \text{und} \quad \|w_K - \sigma_K v_n\| \to 0 \quad \text{mit} \quad K \to \infty,$$

wobei $\sigma_K \in \mathbb{C}$, $|\sigma_K| = 1$. Die Konvergenzrate hängt vom Abstand zwischen dem dominanten und dem nächstkleineren Eigenwert ab, d.h.

$$|\mu_K - \lambda_n| = \mathcal{O}\left(\left|\frac{\lambda_{n-1}}{\lambda_n}\right|^K\right). \tag{7.1}$$

Details und einen Beweis findet man in [Ran17, Satz 7.3].

Das Prinzip der Potenzmethode lässt sich sehr leicht erweitern, um damit auch beliebige (einfache) Eigenwerte und die zugehörigen Eigenvektoren zu berechnen. Die Potenzmethode wird in diesem Falle anstatt auf die Matrix A auf die Matrix $(A - \tilde{\lambda}I)^{-1}$ angewandt. Diese Methode nennt sich **Inverse Iteration**. Wenn man eine gute Näherung $\tilde{\lambda}$ eines beliebigen Eigenwerts λ_j der Matrix A kennt, sodass

$$|\lambda_j - \tilde{\lambda}| < |\lambda_i - \tilde{\lambda}|, \quad \text{für alle} \quad i \neq j,$$

so konvergiert die Inverse Iteration gegen das Eigenpaar (λ_j, v_j) und die Konvergenzrate ist $|\lambda_j - \tilde{\lambda}| / \min_{i\neq j} |\lambda_i - \tilde{\lambda}|$.

So wie das Gradientenverfahren in Kapitel 6.5, verwendet auch die Potenzmethode in jedem Schritt nur lokale Information. Ein effizienteres Verfahren lässt sich daraus wieder durch das Aufbauen eines Krylov Unterraumes konstruieren. Das entsprechende Verfahren nennt man die **Arnoldi Iteration**. Aber auch hier werden wir nicht mehr weiter darauf eingehen, sondern verweisen auf das

Proseminar/Seminar Eigenwertprobleme (Numerik),

das auch im SS 2021 in Heidelberg angeboten wird.

Zum Abschluss geben wir noch ein Beispiel eines Eigenwertproblems, dessen Lösung Sie alle schon einmal in der Praxis verwendet haben.

Beispiel 7.1 (Google PageRank)**.** In den späten 90er Jahren haben Informatiker festgestellt, dass die Hyperlinkstruktur des Internets genutzt werden kann, um die Ergebnisse von Suchmaschinen zu verbessern. Sie verwendeten Ideen aus der Wahrscheinlichkeitstheorie, im Speziellen Markovkettenmodelle, um eine Rangordnung von Webseiten basierend auf deren relativer Wichtigkeit zu bestimmen, die rein auf der Zusammenhangsstruktur des Netzes beruht. Der kommerziell und mathematisch erfolgreichste Algorithmus dieser Art ist der **PageRank Algorithmus** von Sergey Brin und Larry Page, zweier Informatik-Doktoranden aus Stanford aus 1998[1], woraus der Internetriese Google entstand und der aus Brin und Page Multmillionäre machte.

Die Grundidee von PageRank ist einfach. Man betrachtet die Netztruktur des Internets als einen gerichteten Graphen $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, wobei $\mathcal{N}$ die Menge der Knoten – die Webseiten (oder eine Untermenge davon) – und $\mathcal{E}$ die Menge der gerichteten Kanten – die Hyperlinks zwischen den Webseiten – sind. Jede Webseite wird durch eine ganze Zahl von 1 bis n repräsentiert, d.h. $\mathcal{N} = \{1, \dots, n\}$. Wenn Webseite i einen Link zur Seite j enthält, dann ist die Kante $(i, j) \in \mathcal{E}$. Wir stellen den Graphen durch die Transponierte A der Inzidenzmatrix dar, d.h. $A_{i,j} = 1$, wenn es einen Link von Seite j nach Seite i gibt. Da die Kanten gerichtet und Links nicht symmetrisch sind, ist auch A normalerweise unsymmetrisch.

Die Philosophie von PageRank ist nun, dass die Wichtigkeit einer Webseite dadurch definiert ist, wieviele Webseiten auf sie verlinkt sind, d.h. wenn wir mit π_i den sog. PageRank der Seite i bezeichnen, dann gilt

$$\pi_i \;=\; \sum_{j\,:\,(j,i)\,\in\,\mathcal{E}} \frac{1}{c_j}\,\pi_j \tag{7.2}$$

wobei $c_j = \sum_{i=1}^{n} A_{i,j}$ den **Grad** des Knotens j bezeichnet, d.h. die Anzahl der Links auf Seite j. Je höher der PageRank einer Webseite, desto wichtiger ist sie. Eine Möglichkeit, um den PageRank Vektor $\boldsymbol{\pi} = (\pi_i)_{i=1,\dots,n}$ in (7.2) zu berechnen, ist es, ausgehend von einem Startvektor die Webseiten zu durchlaufen und die Komponenten des Vektors nacheinander anzupassen, bis sich der Vektor nicht mehr ändert. Es stellt sich die Frage ob dieser iterative Prozess auch wirklich konvergiert und ob die Lösung eindeutig und stabil ist, d.h., dass man von jedem beliebigen Startvektor immer auf die selbe Lösung kommt.

[1] Die Originalarbeit ist [L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", Technical Report, Stanford University, 1998] (erhältlich unter `http://dbpubs.stanford.edu:8090/pub/1999-66`).

Wir formulieren das Problem in Matrixform und betrachten es als Beispiel einer Markovkette. In dieser Interpretation, sind die Näherungslösungen $\boldsymbol{\pi}^{(k)}$ in der $k$ten Iteration, **Zustände** der Markovkette und der Graph $\mathcal{G}$ ist dargestellt durch eine $n \times n$ Matrix P, wobei $P_{i,j} \in [0,1]$ die **Übergangswahrscheinlichkeit** vom Knoten j zum Knoten i im Graphen $\mathcal{G}$ darstellt.

Eine Matrix P heisst **(spalten-)stochastisch** wenn $\sum_{i=1}^{n} P_{i,j} = 1$ für alle $j \in 1, \ldots, n$. Wenn ausserdem noch alle Einträge der Matrix strikt zwischen 0 und 1 liegen, so heisst P **irreduzibel**. Man kann relativ leicht zeigen, dass eine stochastische Matrix P einen Spektralradius $\mathrm{spr}(P) = 1$ hat und dass 1 ein Eigenwert von P ist. Wenn P auch noch irreduzibel ist, dann besagt der wichtige **Perron-Frobenius Satz**, dass 1 ein einfacher Eigenwert ist und betragsmässig strikt grösser als alle anderen Eigenwerte. Ausserdem hat der zugehörige Eigenvektor $\boldsymbol{\pi}$ strikt positive Einträge $\pi_i > 0$, sodass

$$P\,\boldsymbol{\pi} \;=\; \boldsymbol{\pi} \quad \text{und} \quad \|\boldsymbol{\pi}\|_1 = \sum_{i=1}^{n} \pi_i = 1, \tag{7.3}$$

(normalisiert bzgl. der $\|\cdot\|_1$-Norm), d.h. $\boldsymbol{\pi}$ ist der eindeutige, stationäre Verteilungsvektor der Markovkette.[2]

Zurück zum PageRank Algorithmus. Wir schreiben (7.2) jetzt als ein Markovkettenmodell. Es sei $P_{i,j} = A_{i,j}/c_j$, d.h. wir nehmen an, dass es gleich wahrscheinlich ist, jeden der Links auf einer Webseite zu verfolgen. Dann ist der stationäre Verteilungsvektor $\boldsymbol{\pi}$ in (7.3) auch die Lösung von (7.2). Das soll das Verhalten eines "zufälligen Surfers" modellieren mit unendlich viel Zeit. Es bleibt allerdings noch ein Problem mit diesem Modell, da üblicherweise Webseiten in $\mathcal{G}$ sein werden ohne jegliche Links, sog. *"dangling pages"* oder baumelnde Seiten. Das heisst der zufällige Surfer könnte in einem dieser Knoten hängen bleiben und nicht mehr weiterkommen. Vom Standpunkt der Markovkettentheorie bedeutet das, dass die Matrix P nicht stochastisch ist. Um dieses Problem zu beheben, setzen wir für jede baumelnde Seite j die Einträge $A_{i,j} = 1$ für alle $1 \leq i \leq n$, d.h. wir setzen einen Link zu jeder beliebigen anderen Seite. Das bedeutet, dass der Surfer, wenn er auf so einer baumelnden Seite landet, zufällig die Adresse einer anderen Webseite eintippt um weiterzusurfen.

Das garantiert, dass P eine stochastische Matrix ist. Die Matrix ist aber im Allgemeinen noch nicht irreduzibel. Um die Matrix irreduzibel zu machen, ermöglichen wir es dem Surfer auch noch zu jedem Zeitpunkt die Links auf der aktuellen Webseite zu ignorieren und mit Wahrscheinlichkeit $(1-\alpha)$, für $\alpha \in (0,1)$, auf eine völlig beliebige Webseite zu springen. (Google verwendet anscheinend $\alpha = 0.85$.)

[2] Die meisten Bücher zu Markovketten verwenden die Transponierte von P. In diesem Falle ist die stationäre Verteilung der linke und nicht der rechte Eigenvektor zum Eigenwert 1.

Die endgültige Form der Übergangsmatrix ist

$$P = \alpha A\check{C} + (1-\alpha)\tfrac{1}{n}ee^T\,, \tag{7.4}$$

wobei $\check{C}$ die $n \times n$ Diagonalmatrix mit Einträgen $\check{C}_{i,i} = 1/c_i$ ist und e der Vektor mit Einträgen $e_i = 1$.

Diese Matrix ist nun stochastisch und irreduzibel und das Problem (7.3) hat wie besprochen eine eindeutige Lösung, den PageRank Vektor $\boldsymbol{\pi}$. Da der Eigenwert 1 dominant und einfach ist, garantiert [Ran17, Satz 7.3], dass $\boldsymbol{\pi}$ mit Hilfe der Potenzmethode berechnet werden kann. Da der zweitgrösste Eigenwert von P gleich α ist, folgt aus (7.1), dass die Konvergenzrate α ist.

Mehr Informationen findet man in der Originalarbeit von Page, Brin et al oder in

> K. Bryan and T. Leise, "The $ 25,000,000,000 Eigenvector: The Linear Algebra behind Google", *SIAM Review*, **48**:569–581, 2006.

Appendices

A Grundbegriffe der linearen Algebra

Wir beschränken uns auf den Körper $\mathbb{K} = \mathbb{R}$ der reelen Zahlen mit den üblichen Operationen $+$ und $*$. Alle Sätze in diesem Kapitel werden ohne Beweise gegeben; die Aussagen wurden in *Mathe für Informatiker I* oder *Lineare Algebra 1* gezeigt, ansonsten siehe bspw. [Fis14].

Definition A.1 (Vektorraum)**.** (a) Eine Menge V mit einer (inneren) Verknüpfung (*Addition*)

$$+ : V \times V \to V, \quad (v, w) \mapsto v + w$$

und einer (äußeren) Verknüpfung (*Multiplikation mit Skalaren*)

$$* : \mathbb{R} \times V \to V, \quad (\lambda, v) \mapsto \lambda \cdot v$$

heißt **Vektorraum** (über $\mathbb{R}$), wenn gilt:

(V1) $(V, +)$ ist eine abelsche Gruppe (d. h. kommutativ) mit *Nullvektor* 0 und *Negativem*, bezeichnet mit $-v$.

(V2) Für beliebige $\lambda, \mu \in \mathbb{R}$ und $v, w \in V$ gilt

$$\begin{aligned} (\lambda + \mu) * v &= \lambda * v + \mu * v \\ \lambda * (v + w) &= \lambda * v + \lambda * w \\ \lambda * (\mu * v) &= (\lambda\mu) * v \\ 1 * v &= v \end{aligned}$$

(b) Eine Menge $\emptyset \neq W \subset V$ ist ein **Untervektorraum**, wenn für alle $\lambda \in \mathbb{R}$ und $v, w \in W$:

$$\lambda \cdot (v + w) \in W\,.$$

- Wir bezeichnen als **Linearkombination** einer Familie $(v_i)_{i=1}^r \subseteq V$ ein Element der Menge

$$\text{span } (v_i)_{i=1}^r = \left\{ \sum_{i=1}^r \lambda_i v_i \ : \lambda_1, \ldots, \lambda_r \in \mathbb{R} \right\}.$$

Man sieht leicht, dass der *Spann* ein Untervektorraum von V ist.

- Eine Familie $(v_i)_{i=1}^r$ ist *linear unabhängig* (kurz: l.u.), falls aus

$$\lambda_1 v_1 + \ldots + \lambda_r v_r = 0 \qquad \Rightarrow \qquad \lambda_1 = \ldots = \lambda_r = 0\,.$$

- Eine Familie $(v_i)_{i=1}^r$ ist ein *Erzeugendensystem* von V, wenn

$$\operatorname{span}(v_i)_{i=1}^r = V\,.$$

- Ein linear unabhängiges Erzeugendensystem $(v_i)_{i=1}^r$ heißt **Basis** von V. Ist $(v_i)_{i=1}^r$ eine Basis von V, dann gibt es zu jedem $v \in V$ *eindeutig* bestimmte $\lambda_1, \ldots, \lambda_r \in \mathbb{R}$, sodass

$$v = \lambda_1 v_1 + \ldots + \lambda_r v_r\,.$$

Satz A.2 (Basisauswahlsatz)**.** *Aus jedem endlichen Erzeugendensystem kann man eine Basis auswählen.*

Satz A.3 (Austauschsatz)**.** *Gegeben seien eine Basis $(v_i)_{i=1}^r$ von V und eine l.u. Familie $(w_j)_{j=1}^n$ mit $n < r$. Dann existiert eine Permutation $(i_1, \ldots, i_r)$ von $(1, \ldots, r)$, sodass $(w_1, \ldots, w_n, v_{i_{n+1}}, \ldots, v_{i_r})$ eine Basis von V ist.*

Definition A.4. Die **Dimension** von V ist definiert als

$$\dim V := \begin{cases} \infty & \text{falls keine endliche Basis existiert,} \\ r & \text{falls eine Basis der Länge } r \text{ existiert.} \end{cases}$$

Satz A.5 (Basisergänzungssatz)**.** *Es sei V ein endlichdimensionaler Vektorraum mit $\dim V = r < \infty$ und $(w_i)_{i=1}^n$ eine linear unabhängige Familie mit $n < r$. Dann existieren $w_{n+1}, \ldots, w_r \in V$, sodass $(w_i)_{i=1}^r$ eine Basis von V ist.*

Definition A.6. Für zwei Vektorräume W_1 und W_2, heißt $W_1 + W_2$ die **Summe** der beiden Vektorräume. Es gilt

$$\dim(W_1 + W_2) \leq \dim W_1 + \dim W_2\,.$$

$V = W_1 \bigoplus W_2$ heißt **direkte Summe** von W_1 und W_2, wenn

$$V = W_1 + W_2 \quad \text{und} \quad W_1 \cap W_2 = \{0\}\,.$$

Beachte: Gegeben W_1, ist W_2 nicht eindeutig! Man kann aber Basen von W_1 und W_2 einfach kombinieren.

Definition A.7. Das (kanonische) **Skalarprodukt** im $\mathbb{R}^n$ ist die Abbildung

$$\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \quad (x, y) \mapsto \langle x, y \rangle$$

mit

$$\langle x, y \rangle := x_1 y_1 + x_2 y_2 + \ldots + x_n y_n \,.$$

Das Skalarprodukt ist linear in beiden Argumenten, symmetrisch und es gilt $\langle x, x \rangle \geq 0$ mit $\langle x, x \rangle = 0$ genau dann, wenn $x = 0$.

- Die vom (kanonischen) Skalarprodukt induzierte Abbildung

  $$\|\cdot\| : \mathbb{R}^n \to \mathbb{R}, \quad x \mapsto \|x\| := \sqrt{\langle x, x \rangle}$$

  ist die (Euklidische) **Norm** auf $\mathbb{R}^n$, mit den folgenden Eigenschaften:

  (N1) $\|x\| = 0 \quad \Leftrightarrow \quad x = 0$

  (N2) $\|\lambda x\| = |\lambda| * \|x\|$

  (N3) $\|x + y\| \leq \|x\| + \|y\|$ (“Dreiecksungleichung”)

- Für beliebige $x, y \in \mathbb{R}^n$ gilt die **Cauchy-Schwarz-Ungleichung**

  $$|\langle x, y \rangle| \leq \|x\| * \|y\| \,,$$

  wobei Gleichheit genau dann gilt, wenn x und y linear abhängig sind. In diesem Fall schreiben wir $x \parallel y$.

- Zwei Vektoren $v, w \in W$ heißen **orthogonal**, kurz $v \perp w$, wenn $\langle v, w \rangle = 0$.

- Gegeben $U \subset V$ nennt man den Untervektorraum

  $$U^\perp := \{v \in V \ : \ v \perp u \quad \forall u \in U\}$$

  das **orthogonale Komplement** von U (im Gegensatz zur direkten Summe ist $U^\perp$ hier eindeutig).

- Eine Familie $(v_i)_{i=1}^n$ heißt **orthonormal**, falls alle v_j paarweise orthogonal sind und zusätzlich für alle $i = 1, \ldots, n$ gilt: $\|v_i\| = 1$. Ist $(v_i)_{i=1}^n$ eine Basis von V, so nennt man sie eine **Orthonormalbasis** (ONB).

Satz A.8 (Orthonormalisierungssatz)**.** *Sei V ein euklidischer Vektorraum mit $\dim V = n < \infty$. Gegeben sei eine orthonormale Familie $(w_i)_{i=1}^n \subset V$ mit $m < n$. Dann existieren*

$$w_{m+1}, \ldots, w_n \in V\,, \quad \textit{sodass} \;\; (w_i)_{i=1}^n \;\textit{ONB von}\; V\,.$$

Ein konstruktiver Beweis dieses Satzes ist das *Gram-Schmidt-Verfahren*, welches eine effiziente Methode darstellt, um ONB zu bauen.

Definition A.9 (Lineare Abbildung)**.** (a) Eine Abbildung $F : V \to W$ zwischen zwei Vektorräumen V und W heißt **linear** (oder **Homomorphismus**), wenn für alle $v, w \in V$ und $\lambda, \mu \in \mathbb{R}$ gilt

$$F(\lambda v + \mu w) = \lambda F(v) + \mu F(w)\,.$$

(b) Falls $V = W$, so nennt man F **Endomorphismus**; ist F zusätzlich bijektiv heißt die Abbildung **Isomorphismus**.

(c) Man nennt $\mathrm{im}(F) = F(V)$ das **Bild** von F.

(d) Man nennt $\ker(F) = F^{-1}(\{0\})$ den **Kern** von F.

Proposition A.10. *Seien $F : V \to W$, $G : W \to V$ lineare Abbildungen. Dann ist auch die Komposition $F \circ G$ linear.*

Proposition A.11. *Sei $F : V \to W$ linear. Dann gilt*

(a) F ist genau dann surjektiv, wenn $\mathrm{im}(F) = W$.

(b) F ist genau dann injektiv, wenn $\ker(F) = \{0\}$.

Satz A.12. *Sei $F : V \to W$ eine lineare Abbildung zwischen zwei endlichdimensionalen Vektorräumen V und W. Dann gilt*

$$F \textit{ injektiv} \quad \Leftrightarrow \quad F \textit{ surjektiv} \quad \Leftrightarrow \quad F \textit{ bijektiv.}$$

Satz A.13. *Zu jeder linearen Abbildung $F : \mathbb{R}^n \to \mathbb{R}^m$ gibt es **genau eine** Matrix $A \in \mathbb{R}^{m \times n}$, sodass*

$$F(x) = Ax, \qquad \textit{für alle } x \in \mathbb{R}^n\,.$$

Definition A.14 (Eigenwerte)**.** Sei $F : V \to V$ ein Endomorphismus. Ein Skalar $\lambda \in \mathbb{C}$ heißt **Eigenwert** von F, falls $v \in V \setminus \{0\}$ existiert, sodass

$$F(v) = \lambda v\,.$$

Jedes $v \in V \setminus \{0\}$, das obige Gleichung erfüllt heißt **Eigenvektor** von F (zum Eigenwert λ). Man nennt $\mathrm{Eig}(F;\lambda) = \{v \in V \ : \ F(v) = \lambda v\}$ den **Eigenraum**.

- Falls eine Basis aus Eigenvektoren von F existiert, nennt man F **diagonalisierbar**. Ist $\dim(V) = n < \infty$ existiert in diesem Fall eine Matrix $S \in \mathbb{R}^{n\times n}$, sodass
$$SAS^{-1} = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix},$$
wobei A die Matrix aus Satz A.13 ist und $\lambda_1, \dots, \lambda_n$ die Eigenwerte von F sind.

- Ist A **symmetrisch**, d. h. $A = A^\intercal$, dann sind alle Eigenwerte $\lambda_1, \dots, \lambda_n$ reell und die Matrix S ist orthogonal, d. h. es existiert eine Zerlegung
$$QAQ^\intercal = \begin{pmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \lambda_n \end{pmatrix},$$
mit einer orthogonalen Matrix $Q \in \mathbb{R}^{n\times n}$ (also $Q^{-1} = Q^\intercal$).

- Ist λ ein Eigenwert von F, dann gilt definitionsgemäß, dass
$$\mathrm{Eig}(F;\lambda) = \ker(F - \lambda\,\mathrm{id}_V) \neq \{0\}\,,$$
was genau dann erfüllt ist, wenn $\det(F - \lambda\,\mathrm{id}_V) = 0$. Im Fall $V = \mathbb{R}^n$ und $F(v) = Av$ bezeichnet man
$$p_A(t) \coloneqq \det(A - tI_n)$$
als **charakteristisches Polynom** von A. Wir könnten also Eigenwerte von A berechnen, indem wir Nullstellen von $p_A(t)$ bestimmen; dies ist jedoch **numerisch instabil** und ineffizient.

Übungsaufgabe A.15. Wiederholen Sie folgende Themen der Linearen Algebra:

(a) Rechenregeln für Matrizen;

(b) Gauss'sches Eliminationsverfahren zur Lösung lineaer Gleichungssysteme;

(c) Definition und Eigenschaften der Determinante einer Matrix;

(d) Polynome über $\mathbb{R}$, Division von Polynomen mit Rest, Faktorisierbarkeit und Restpolynomsatz.

Literatur

[Bas20] Peter Bastian. *Einführung in die Numerik*. Vorlesungsskript. Universität Heidelberg, 2020. URL: `https://conan.iwr.uni-heidelberg.de/teaching/numerik0_ss2020/`.

[Fis14] Gerd Fischer. *Lineare Algebra*. Wiesbaden: Springer, 2014.

[For16] Otto Forster. *Analysis 1*. Wiesbaden: Springer, 2016.

[Gv96] Gene Golub und Charles F. van Loan. *Matrix computations*. Baltimore: Johns Hopkins University Press, 1996.

[Ran17] Rolf Rannacher. *Numerik 0: Einführung in die Numerische Mathematik*. Heidelberg: Heidelberg University Publishing, 2017. URL: `https://heiup.uni-heidelberg.de/heiup/catalog/book/206`.

[SB] Josef Stoer und Roland Bulirsch. *Einführung in die Numerische Mathematik*. Teil I (1. Auflage 1971, Josef Stoer), Teil II (1. Auflage 1993, Josef Stoer und Roland Bulirsch). Springer.

[Wil61] James Hardy Wilkinson. „Error Analysis of Direct Methods of Matrix Inversion“. In: *J. ACM* 8 (Juli 1961), S. 281–330.

[WS72] Helmut Werner und Robert Schaback. *Praktische Mathematik II*. Berlin, Heidelberg: Springer, 1972.