Numerik I
Wintersemester 2006/07

Anita Schöbel

12. November 2007

Vorwort

Das vorliegende Skriptum ist anhand der von mir im Wintersemester 2006/2007 gehaltenen Vorlesung *Numerische Mathematik I* entstanden. Diese Vorlesung sowie ihre Fortsetzung *Numerische Mathematik II* führt in die Grundlagen der Numerischen und Angewandten Mathematik ein und bietet damit einen Einstieg für weiterführende Vorlesungen auf diesem Gebiet. Neben Grundlagen in der Fehleranalyse und der Funktionalanalysis werden in dem hier vorliegenden ersten Teil lineare Gleichungssysteme, nichtlineare Gleichungssysteme und die Interpolation von Funktionen behandelt. Die eingeführten Methoden beschreiben Eliminationsverfahren, Orthogonalisierungsverfahren und iterative Verfahren sowie klassische Verfahren der Interpolation. Diese Themen werden in *Numerik II* mit Integration, Approximation und der numerische Lösung von gewöhnlichen Differentialgleichungen fortgesetzt, es wird dort auch die Lösung von Eigenwertaufgaben und von Optimierungsproblemen kurz gestreift.

Die Zielgruppe der *Numerik I* sind neben Mathematik-Studierenden ab dem dritten Semester auch interessierte Hörerinnen und Hörer aus der Informatik und der Physik. Es sei bemerkt, dass die Vorlesung durch theoretische Übungen und Programmieraufgaben (mit Matlab) ergänzt wurde. Auch beim Selbststudium des Textes ist die eigenständige Beschäftigung mit dem Stoff und die Implementation des einen oder anderen Verfahrens unbedingt zu empfehlen!

Ein großer Anteil des vorliegenden Skriptes wurde mit sehr viel Sorgfalt von Anke Uffmann und Michael Siebert erstellt: An beide dafür ein herzliches Dankeschön! Frau Uffmann bin ich besonders dankbar für die Erstellung der Graphiken; Michael Siebert beeindruckte mich durch seine oft trickreichen Ideen, mit denen es ihm gelang, alle anfallenden Problemen (nicht nur in LaTeX) zu lösen.

Göttingen, September 2007
Anita Schöbel

Inhaltsverzeichnis

1 The name of the game: Numerik **3**
- 1.1 Einleitung . . . 3
- 1.2 Algorithmen . . . 6
- 1.3 Aufwand . . . 6
- 1.4 Fehlerabschätzung und Gleitkommazahlen . . . 11
- 1.5 Fehlerfortpflanzung, Kondition und Stabilität . . . 15

2 Lineare Gleichungssysteme: Eliminationsverfahren **19**
- 2.1 Begriffe und Grundlagen . . . 19
- 2.2 Gauß-Verfahren und LU-Zerlegung . . . 22
- 2.3 Das Cholesky-Verfahren . . . 37
- 2.4 Schwachbesetzte Matrizen . . . 41

3 Störungsrechnung **45**
- 3.1 Metrische und normierte Räume . . . 45
- 3.2 Normen für Abbildungen und Matrizen . . . 52
- 3.3 Kondition . . . 61

4 Orthogonalisierungsverfahren **65**
- 4.1 Die QR-Zerlegung . . . 65
- 4.2 Lineare Ausgleichsprobleme . . . 76
- 4.3 Singulärwertzerlegung . . . 82
- 4.4 Anwendung der Singulärwertzerlegung auf lineare Ausgleichsprobleme 83

5 Iterationsverfahren **86**
- 5.1 Das Verfahren der sukzessiven Approximation . . . 86
- 5.2 Der Banach'sche Fixpunktsatz . . . 90
- 5.3 Iterative Verfahren für lineare Gleichungssysteme . . . 95
- 5.4 Iterative Verfahren für nichtlineare Gleichungssysteme . . . 112

6 Interpolation **122**
- 6.1 Polynomiale Interpolation . . . 122
- 6.2 Abschätzung des Interpolationsfehlers und Konvergenzanalyse . . . 135

6.3 Spline Interpolation 138
6.4 Trigonometrische Interpolation 151

Kapitel 1

The name of the game: Numerik

1.1 Einleitung

In der Wikipedia ist der Begriff *Numerik* folgendermaßen definiert:

> Die numerische Mathematik, kurz Numerik genannt, beschäftigt sich als Teilgebiet der Mathematik mit der Konstruktion und Analyse von Algorithmen für kontinuierliche mathematische Probleme.
>
> Interesse an solchen Algorithmen besteht meist aus einem der beiden folgenden Gründe:
>
> - Es gibt zu dem Problem keine explizite Lösungsdarstellung (so zum Beispiel bei den Navier-Stokes-Gleichungen oder bei Integralen ohne Stammfunktion) oder
> - die Lösungsdarstellung existiert, ist jedoch nicht geeignet, um die Lösung schnell auszurechnen beziehungsweise sie liegt in einer Form vor, in der Rechenfehler sich stark bemerkbar machen (zum Beispiel bei vielen Potenzreihen).

In der angewandten Mathematik setzt man dabei noch einen Schritt früher an: Beschäftigt man sich mit Anwendungen, so liegt zunächst noch kein mathematisches Problem vor, sondern einzig eine von Praktikern formulierte Problembeschreibung. Die erste Aufgabe besteht also in der *Modellierung* d.h. in der Formalisierung eines in der Natur beobachteten Phänomens oder eines ökonomischen Problems durch ein sogenanntes *mathematisches Modell*. Mathematische Modelle sind Systeme von Gleichungen oder Ungleichungen, durch die Beziehungen zwischen bekannten und unbekannten Größen dargestellt werden. Dabei können algebraische Gleichungen (oder Ungleichungen), Differentialgleichungen oder Integralgleichungen verwendet werden und es dürfen dabei alle Arten von Formeln oder Grenzwerten auftreten. Gibt es zusätzlich noch ein Kriterium zur

Beurteilung der Lösung (eine *Zielfunktion*), so liegt ein Problem der mathematischen Optimierung vor und das Modell wird auch als *mathematisches Programm* bezeichnet.

Klassische Disziplinen der "reinen" Mathematik wie die Algebra und Analysis beschäftigen sich mit Fragen der Existenz und Eindeutigkeit der Lösungen mathematischer Modelle. Dagegen besteht das Ziel der numerischen Mathematik darin, Verfahren zu entwickeln, mit denen sich die Lösungen mathematischer Modelle praktisch (auf derzeit verfügbaren Rechenanlagen) ermitteln lassen. Wir veranschaulichen das an einigen Beispielen:

- Der Fundamentalsatz der Algebra besagt, dass ein reelles Polynom vom Grad n auch n Nullstellen in der Menge der komplexen Zahlen besitzt. Der Existenzbeweis ist jedoch nicht konstruktiv, d.h. man erhält kein Verfahren, wie man die entsprechenden Nullstellen bestimmen kann. Das liefert die numerische Mathematik.

- Die Lösung linearer, nicht singulärer Gleichungssysteme kann durch die Cramersche Regel aufgeschrieben werden. Für die praktische Berechnung ist sie aber bei mehr als drei Variablen unbrauchbar.

- Der Satz von Weierstrass liefert die Aussage, dass stetige Funktionen auf kompakten Mengen ihre Minima (oder Maxima) annehmen. Wie aber soll man sie berechnen?

- Für Anfangswertprobleme einer gewöhnlichen Differentialgleichung liefert der Existenzbeweis von Picard-Lindelöf unter bestimmten Glattheitsvoraussetzungen ein konstruktives Iterationsverfahren. Bei der Realisierung auf Computern ist dieses Verfahren aber nicht sonderlich effektiv.

In der vorliegenden Vorlesung: Numerik I sollen Verfahren für die Berechnung mathematischer Modelle vorgestellt und diskutiert werden. Die Vorlesung richtet sich an Studierende der Mathematik ab dem dritten Semester, und gerne auch an interessierte Physik oder Informatik-Studierende. Die in der Vorlesung verwendeten Grundlagen sind so ausgewählt, dass sie auch aus der "Mathematik für Informatik-Anfänger" Vorlesung bekannt sein sollten. Die Vorlesung bietet den Einstieg in den Bereich numerische Mathematik, wissenschaftliches Rechnen und Optimierung und ist als Grundlage der meisten Vorlesungen aus diesem Bereich zu verstehen. Die Vorlesung kann mit *Numerik II* oder mit *Optimierung* fortgesetzt werden.

Um die numerische Mathematik richtig zu verstehen, sollte man natürlich auch einiges selbst programmiert und implementiert haben. Es werden in den Übungen daher theoretische und praktische Aufgaben gestellt, wobei die meisten Programmieraufgaben mit MATLAB zu bearbeiten sind. MATLAB ist eine Skriptsprache,

in der viele numerische Verfahren, Operationen und die entsprechenden Datenstrukturen bereits zur Verfügung stehen. Um die Schwierigkeiten zu verstehen, auf die man stößt, wenn man ein Verfahren von Grund auf neu implementiert, sind bei einigen Aufgaben auch "klassische" Programmiersprachen zu verwenden. Computeralgebrasysteme (wie MuPAD, Maple, Mathematica, Singular) werden ebenfalls gestreift.

Es gibt zahlreiche Lehrbücher und Skripten über numerische Mathematik, von denen die folgenden Quellen erwähnt werden sollen:

- J. Stoer, Numerische Mathematik I, Springer, 1989.
- J. Werner. Numerische Mathematik 1, Vieweg, Braunschweig, 1992.
- P. Deuflhard und A. Hohmann. Numerische Mathematik I. Walter de Gruyter, Berlin, New York, 2nd edition, 1993.
- R. Kreß. Numerical Analysis. Springer, New York, 1998.
- M. Hanke-Bourgeois. Grundlagen der Numerischen Mathematik und des wissenschaftlichen Rechnens. Teubner, Stuttgart, 2002.
- R. Schaback, H. Wendland. Numerische Mathematik. Springer, Berlin, 2004.
- Skriptum *Numerik I* von G. Lube, siehe
  http://www.num.math.uni-goettingen.de/lube/NM1-04akt.pdf
- Skriptum *Numerik I* von T. Hohage, siehe
  http://www.num.math.uni-goettingen.de/hohage/Numerik1/numerik1.html

In der Vorlesung Numerik I werden die folgenden Themen behandelt:

- Lineare Gleichungssysteme
- Ausgleichsprobleme
- Nullstellensuche (eine nichtlineare Gleichung oder ein System nichtlinearer Gleichungen)
- Interpolation

Die „klassische" Numerik beschäftigt sich vorrangig mit kontinuierlichen Problemen. Dagegen sind *diskrete Probleme* durch eine endliche Menge an möglichen Lösungen gekennzeichnet. Auch sie sind ein wichtiger Bestandteil der angewandten Mathematik, insbesondere bei ökonomischen Fragestellungen. In diesem Skript werden wir auf diskrete Probleme aber nicht eingehen.

1.2 Algorithmen

Ein Algorithmus für ein Problem (P) ist ein durch eine Abfolge von (Rechen-)-Vorschriften beschriebenes Verfahren, das zu einer "Lösung" des Problems (P) führt. Je nach Qualität der erzielten Lösung, unterscheidet man zwischen exakten Verfahren, konstruktiven Verfahren und Heuristiken.

Exakte Verfahren sind streng genommen nur bei diskreten mathematischen Aufgaben möglich, in denen unter endlich vielen Möglichkeiten eine Lösung auszuwählen ist. Dazu gehört beispielsweise das Gebiet der ganzzahligen Programmierung sowie die meisten Aufgaben in Netzwerken. Dagegen ist bei kontinuierlichen Problemen aufgrund der beschränkten Genauigkeit der Darstellung reeller Zahlen durch den Computer jede Lösung eine Näherungslösung.
Ein **konstruktives oder direktes Verfahren** ist eine Rechenvorschrift, mit deren Hilfe die numerische Lösung einer mathematischen Aufgabe in endlich vielen Rechenschritten beliebig genau ermittelt werden kann.
Lässt sich gar keine Genauigkeit angeben oder sind dieser Genauigkeit Grenzen gesetzt, so spricht man von einer **Heuristik**. Kann wie im letzteren Fall zwar eine Genauigkeit angegeben werden, diese ist aber beschränkt, so hat die Heuristik eine *Gütegarantie*; man spricht dann auch von einer *Approximation*. Heuristiken werden vor allem bei sehr schweren Problemen der diskreten Optimierung verwendet und führen dort häufig zu empirisch sinnvollen Ergebnissen.

Neben der Wohldefiniert eines numerischen Verfahrens sollte man bei jedem Verfahren die folgenden Punkte diskutieren:

- Aufwand
- Fehleranalyse
- Stabilität.

1.3 Aufwand

Ein Verfahren kann nur dann sinnvoll eingesetzt werden, wenn es auch in praktikabler Zeit eine Lösung ermittelt. Daher ist es wichtig, den Aufwand verschiedener Verfahren für die gleiche Aufgabenstellung vergleichend zu diskutieren. Man spricht auch von der **Komplexität** eines Verfahrens und bezeichnet damit den Aufwand an wesentlichen Rechenoperationen in Abhängigkeit einer sinnvoll gewählten Eingangsgröße.

Als wesentliche Rechenoperationen zählen wir Additionen, Subtraktionen, Multiplikationen, Divisionen, Vergleiche und davon getrennt Funktionsauswertungen. (Zuweisungen werden nicht gezählt.)

Die Eingangsgröße kann meistens auf verschiedene Weise gewählt werden. Sie sollte die Größe des Problems repräsentieren.

Beispiele:

- Bestimme das Minimum von n Zahlen $x_1, \ldots, x_n$: Hier bestimmt man die Anzahl der Rechenoperationen in Abhängigkeit der Zahl n. Für den einfachen Algorithmus

  $Min := \infty$; For $i := 1$ to n do: If $x_i < Min$ then $Min := x_i$; Output: Min

  ergeben sich n Vergleiche, also eine Anzahl von $A_1(n) = n$ wesentlichen Rechenoperationen.

- Bei der Addition von zwei Vektoren x und y der Dimension k bietet sich als sinnvolle Eingangsgröße die Dimension k an. Das Verfahren

  For $i := 1$ to k do: $z_i := x_i + y_i$; Output: $z_1, \ldots, z_n$

  benötigt k Additionen, hat also einen Aufwand von $A_2(k) = k$

- Addition von zwei Matrizen A, B der Dimension $k \times k$ (mit Elementen $a_{ij}, b_{ij}, i, j = 1, \ldots, k$): Hier kann man als Eingabegröße k oder k^2 wählen. Das kanonische Verfahren ist das folgende.

  For $i := 1$ to k do:

  For $j = 1$ to k do: $c_{ij} := a_{ij} + b_{ij}$;

  Output: c_{ij}, $i, j = 1, \ldots, n$

  Die Anzahl der wesentlichen Rechenoperationen beträgt k^2. Normalerweise wird man den Aufwand in Abhängigkeit der Anzahl der Matrixelemente $m = k^2$ mit $A_3(m) = m$ als linear angeben. In vielen Anwendungen macht es aber Sinn, die Dimension k als Eingabegröße zu wählen, was zu einem quadratischen Aufwand $A_4(k) = k^2$ führt.

Bei komplizierteren Problemen sind meist verschiedene Verfahren mit jeweils unterschiedlichem Aufwand möglich. Hat man also z.B. einen Algorithmus A und einen Algorithmus B zur Auswahl, und ist der Aufwand beider Verfahren durch Funktionen $A(n)$ beziehungsweise $B(n)$ bekannt so wird man für die Problemgröße n das Verfahren mit dem jeweils kleineren Aufwand wählen.

Um für steigende Problemgrößen den Aufwand von zwei Verfahren auf einfache Methode zu vergleichen, bieten sich die *Landau-Symbole* an. Diese sind nicht nur in der Analyse von Aufwandsabschätzungen sondern allgemeiner zur quantitativen Beschreibung von Grenzprozessen ein wichtiges Hilfsmittel.

Die Landau-Symbole geben dabei an, wie sich die Größe von zwei Funktionen $A(n), B(n) : \mathbb{N} \to \mathbb{R}$ im Verhältnis zueinander entwickelt, wenn $n \to \infty$ geht.

Beispiel: Tabelle mit der Entwicklung von verschiedenen Funktionen in der Vorlesung.

Definition 1.1 *Seien $(a_n), (b_n)$ reelle Zahlenfolgen. Die* **Landau-Symbole** *sind wie folgt definiert.*

1. $a_n = \mathrm{O}(b_n)$ *falls es ein* $C \in \mathbb{R}, C > 0$ *und ein* $N \in \mathbb{N}$ *gibt mit*

$$|a_n| \leq C|b_n| \text{ für alle } n \geq N.$$

2. $a_n = \Omega(b_n)$ *falls es ein* $C \in \mathbb{R}, C > 0$ *und ein* $N \in \mathbb{N}$ *gibt mit*

$$|a_n| \geq C|b_n| \text{ für alle } n \geq N.$$

3. $a_n = \mathrm{o}(b_n)$ *falls es zu jedem* $\varepsilon > 0$ *ein* $N \in \mathbb{N}$ *gibt mit*

$$|a_n| \leq \varepsilon|b_n| \text{ für alle } n \geq N.$$

4. $a_n = \Theta(b_n)$ *falls* $a_n = \mathrm{O}(b_n)$ *und* $a_n = \Omega(b_n)$.

Um die Bedeutung dieser Symbole zu verdeutlichen, formulieren wir die Aussagen um, indem wir die Entwicklung des Quotienten $\frac{a_n}{b_n}$ betrachten, um die Wachstumsraten der beiden Folgen zu vergleichen. Nehmen wir dazu an, dass $b_n \neq 0$ für alle $n \in \mathbb{N}$. Dann erhält man:

$$\begin{aligned}
a_n = \mathrm{O}(b_n) &\iff \left|\frac{a_n}{b_n}\right| \leq C \text{ für alle } n \geq N \text{ und ein } C > 0. \\
a_n = \Omega(b_n) &\iff \left|\frac{a_n}{b_n}\right| \geq C \text{ für alle } n \geq N \text{ und ein } C > 0. \\
a_n = \mathrm{o}(b_n) &\iff \left|\frac{a_n}{b_n}\right| \to 0 \\
a_n = \Theta(b_n) &\iff C_1 \leq \left|\frac{a_n}{b_n}\right| \leq C_2 \text{ für alle } n \geq N \text{ und } C_1, C_2 > 0.
\end{aligned}$$

Die Bedeutung der Landau-Symbole kann hier nun abgelesen werden: Ist $a_n = \mathrm{O}(b_n)$ so wächst a_n nicht schneller als b_n, im Fall $a_n = \Omega(b_n)$ wächst a_n nicht langsamer als b_n. Weiterhin bedeutet $a_n = \Theta(b_n)$, dass beide Folgen annähernd gleich schnell wachsen, und bei $a_n = \mathrm{o}(b_n)$ wächst b_n viel schneller als a_n.

Einfache Beispiele:

$$\begin{aligned} n^2 &= \mathrm{O}(n^3) \\ n^2 &= \mathrm{O}(\tfrac{1}{1000}n^3) \\ n^3 &= \Omega(n^2) \\ n^2 &= \mathrm{O}(\tfrac{1}{3}n^2) \\ n^2 &= \Omega(\tfrac{1}{3}n^2) \\ n^2 &= \Theta(\tfrac{1}{3}n^2) \\ n^2 &= \mathrm{o}(\tfrac{1}{1000}n^3) \\ n^2 &\neq \mathrm{o}(\tfrac{1}{3}n^2) \end{aligned}$$

Lemma 1.2 *Die folgenden Aussagen gelten:*

1. *Alle vier Begriffe sind transitiv, d.h.*

   $$a_n = \mathrm{O}(b_n), b_n = \mathrm{O}(c_n) \Longrightarrow a_n = \mathrm{O}(c_n),$$

   analog für $\Omega, \mathbf{o}$ *und* Θ.

2. Θ *ist eine Äquivalenzrelation*

3. $a_n = \mathrm{O}(b_n)$ *genau dann wenn* $b_n = \Omega(a_n)$.

4. $a_n = \mathrm{o}(b_n) \Longrightarrow a_n = \mathrm{O}(b_n)$.

Beweis: Lässt sich leicht nachrechnen, Übungen!

Weiterhin sollte man sich klar machen, dass

$$\begin{aligned} a_n = \mathrm{O}(b_n) &\iff a_n = \mathrm{O}(\alpha b_n) \text{ für alle } \alpha \in \mathbb{R} \setminus \{0\} \\ a_n = \mathrm{O}(b_n) \text{ und } a'_n = \mathrm{O}(b_n) &\Longrightarrow a_n + a'_n = \mathrm{O}(b_n). \end{aligned}$$

Diese Aussagen gelten auch für $\mathrm{o}, \Omega, \Theta$ gilt.

Von großer praktischer Bedeutung ist (wie in der Tabelle am Anfang gezeigt), dass

- Logarithmisches Wachstum langsamer ist als polynomiales, in Formeln:

  $$(\log_\beta(n))^\gamma = \mathbf{o}(n^\alpha) \text{ für alle } \alpha > 0, \beta > 1, \gamma > 0$$

- Polynomiales Wachstum schwächer ist als exponentielles,

  $$n^\alpha = \mathbf{o}(\beta^n) \text{ für alle } \alpha > 0, \beta > 1$$

- Exponentielles Wachstum schwächer ist als fakultatives,

$$\beta^n = \mathbf{o}(n!) \text{ für alle } \beta > 1$$

Diese Aussagen lassen sich nun auf die Analyse von Algorithmen anwenden. Dazu betrachten wir die folgenden beiden schematischen Algorithmen-Bruchstücke:

- Algorithmus 1:

  Schritt 1: Führe Verfahren A aus

  Schritt 2: Führe Verfahren B aus

Hat Verfahren A einen Aufwand von $\mathrm{O}(a_n)$ und Verfahren B einen Aufwand von $\mathrm{O}(b_n)$, und gilt $a_n = \mathrm{O}(b_n)$, so ergibt sich für Algorithmus 1 ein Aufwand von $\mathrm{O}(b_n)$, das heißt, bei der Hintereinanderausführung von Algorithmenteilen ist immer der größere Aufwand maßgebend.

- Algorithmus 2:

  Schritt 1: Für $m = 1, \ldots, M$ führe Verfahren A aus

Hat Verfahren A einen Aufwand von $\mathrm{O}(a_n)$, und lässt sich die Größe der Zahl M in Abhängigkeit von der Eingabegröße n durch $M = \mathrm{O}(c_n)$ abschätzen, so ergibt sich für Algorithmus 2 ein Aufwand von $\mathrm{O}(a_n \cdot c_n)$.

Abschließend erweitern wir Definition 1.1 auf beliebige Funktionen. Vor allem O und o werden in dieser Formulierung auch häufig für die Abschätzung von Restgliedern verwendet.

Definition 1.3 *Seien $f, g : \mathbb{K} \to \mathbb{K}$. Dann definiert man*

- $f = \mathbf{O}(g)$ *für* $x \to x_0$ *falls* $f(x_n) = \mathbf{O}(g(x_n))$ *für jede Folge* $x_n \to x_0$.
- *Analog für* Ω, o, Θ.

Es lässt sich leicht zeigen, dass obige Definition äquivalent ist zu

- $f = \mathbf{O}(g)$ falls es eine Zahl $C > 0$ und eine Umgebung $U = U(x_0)$ von x_0 gibt, so dass $|f(x)| \leq C|g(x)|$ für alle $x \in U$.
- $f = \mathbf{o}(g)$ falls es zu jedem $\varepsilon > 0$ eine Umgebung $U = U(x_0)$ von x_0 gibt, so dass $|f(x)| \leq \varepsilon|g(x)|$ für alle $x \in U$.

Die Umformulierungen von Ω und Θ erhält man analog.

1.4 Fehlerabschätzung und Gleitkommazahlen

Hat man ein Verfahren zur Lösung eines mathematischen Problems entwickelt, so sind die folgenden Fehlerquellen zu diskutieren:

- **Verfahrensfehler:** Dazu gehören die folgenden beiden Typen

  Abbruchfehler: Sie entstehen beim Ersetzen eines unendlichen Prozesses durch ein endliches Verfahren, z.B. das Abbrechen beim Aufsummieren einer konvergenten unendlichen Reihe.

  Diskretisierungsfehler: Dagegen entstehen Diskretisierungsfehler, wenn man eine kontinuierliche Menge durch eine diskrete ersetzt. Das kann beispielsweise bei der Beschreibung einer Funktion durch endlich viele Koeffizienten oder der Auswertung an endlich vielen Gitterpunkten geschehen.

- **Eingangsfehler** sind Fehler der Eingangsgrößen (z.B. Datenfehler). Sie beziehen sich auf die Qualität der Eingabedaten, die aufgrund von Messfehlern oder aufgrund statistischer Schwankungen ungenau vorliegen können. Manchmal sind Eingabegrößen auch nicht hinreichend bekannt und man ist auf Schätzwerte (z.B. über Kundenverhalten) angewiesen.

- Wichtig sind außerdem **Rundungsfehler** die durch die jeweilige Maschinengenauigkeit bedingt werden. Rundungsfehler können schon bei der Übersetzung der (möglicherweise fehlerbehafteten) Eingangswerte auf maschinenkonforme Daten auftreten.

Eingangsfehler und bei der Eingabe entstehende Rundungsfehler sind zunächst unabhängig von der gewählten Rechenmethode, aber man muss besonders für konstruktive Verfahren unbedingt abschätzen, wie sich solche Fehler im Verlauf des Verfahrens weiterentwickeln. Dabei werden schon vorhandene Fehler in jedem Schritt des Verfahrens übertragen; Rundungsfehler können zusätzlich bei jeder numerischen Operation neu entstehen. Das Ziel ist, abzuschätzen, wie schlimm sich Eingangsfehler und Rundungsfehler auf die Qualität des Ergebnisses auswirken. Ein *gut konditioniertes Problem* liegt vor, wenn kleine Änderungen der Ausgangsgrößen auch nur kleine Lösungsänderungen bewirken. Das Problem wird dann auch *robust* genannt. Bei schlecht konditionierten Problemen muss man geeignete Verfahren wählen.

Im folgenden formalisieren wir, was man unter *Fehler* versteht.

Definition 1.4 *Sei $\tilde{x} = \tilde{f}(\tilde{y})$ die von einem Verfahren $\tilde{f}$ bei (fehlerhaften) Eingangsdaten $\tilde{y}$ ermittelte Lösung. Sei $x = f(y)$ die exakte Lösung des Problems mit exakten Eingangsdaten y. Dann bezeichnen wir mit $|\tilde{x} - x|$ den* **absoluten Fehler** *der Lösung. Im Fall $x \neq 0$ heißt $|\frac{\tilde{x}-x}{x}|$ der* **relative Fehler** *der Lösung.*

Der **absolute Verfahrensfehler** *eines Verfahrens $\tilde{f}$ ist $|\tilde{f}(y) - f(y)|$, der* **relative Verfahrensfehler** *für $f(y) \neq 0$ ist $\left|\frac{\tilde{f}(y)-f(y)}{f(y)}\right|$. Das Verfahren $\tilde{f}$ nennt man* **K-Approximation**, *wenn es für alle möglichen Instanzen (d.h. für alle möglichen Eingangsdaten) y eine Lösung ermittelt, deren relativer Fehler maximal K ist, wenn also für alle y gilt:*

$$\left|\frac{\tilde{f}(y) - f(y)}{f(y)}\right| \leq K.$$

Der durch fehlerhafte Eingangsdaten übertragene absolute Fehler ist $|f(\tilde{y}) - f(y)|$, der durch fehlerhafte Eingangsdaten entstehende relative Fehler ist $\left|\frac{f(\tilde{y})-f(y)}{f(y)}\right|$.

Mit dem letztgenannten Fehler werden wir uns in Abschnitt 1.5 näher beschäftigen.

Hier schauen wir uns zunächst die Abschätzung des Abbruchfehlers am Beispiel der Berechnung der Exponentialfunktion $\exp(x) = \sum_{j=0}^{\infty} \frac{x^j}{j!}$ an. Ein mögliches Verfahren zur Berechnung von $\exp(x)$ für $x \in \mathbb{R}$ besteht in der Auswerten der n-ten Partialsumme

$$P_n(x) = \sum_{j=0}^{n} \frac{x^j}{j!}.$$

(In den Bezeichnungen von Definition 1.4 ist $\tilde{f}(x) = P_n(x)$ und $f(x) = \exp(x)$.)
Wir nehmen an, dass $n > |x|$ gewählt wurde.
Für $x < 0$ erhält man dann den absoluten Abbruchfehler

$$\begin{aligned}
|\exp(x) - P_n(x)| &= \left|\sum_{j=n+1}^{\infty} \frac{x^j}{j!}\right| \\
&\leq \frac{|x|^{n+1}}{(n+1)!} \underbrace{-\frac{|x|^{n+2}}{(n+2)!} + \frac{|x|^{n+3}}{(n+3)!}}_{\leq 0} \underbrace{-\frac{|x|^{n+4}}{(n+4)!} + \frac{|x|^{n+5}}{(n+5)!}}_{\leq 0} \cdots \\
&\leq \frac{|x|^{n+1}}{(n+1)!}
\end{aligned}$$

und für $x \geq 0$ kann man

$$|\exp(x) - P_n(x)| = \sum_{j=n+1}^{\infty} \frac{x^j}{j!} \leq \sum_{j=0}^{\infty} \frac{x^j}{j!} \cdot \frac{x^{n+1}}{(n+1)!} = \exp(x) \frac{|x|^{n+1}}{(n+1)!}$$

abschätzen. In einer kleinen Umgebung von Null hat man also kleine absolute (und relative) Fehler. Das Verfahren, $\exp(x)$ durch Auswertung der n-ten Partialsumme zu bestimmen, ist also für positive reelle Zahlen x eine $K = \frac{|x|^{n+1}}{(n+1)!}$-Approximation. Weil $|x|^{n+1} = \mathrm{o}\left((n+1)!\right)$, kann man die gewünschte Zahl $\exp(x)$ beliebig genau approximieren, indem man n wachsen lässt.

Wenden wir uns nun den in der Numerik besonders wichtigen Rundungsfehlern und ihrer Fortpflanzung zu. Dazu müssen wir wissen, wie reelle Zahlen auf Rechenanlagen dargestellt werden. Üblich ist die Darstellung durch Gleitkommazahlen:

Definition 1.5 *Sei $B \geq 2$ eine ganze Zahl. Eine positive B-adische und m-stellige* **normalisierte Gleitkommazahl** *hat die Form $x = 0$ oder*

$$x = B^e \sum_{k=-m}^{-1} x_k B^k \quad \textit{mit } e \in \mathbb{Z}, x_{-1} \neq 0, x_k \in \{0, 1, \ldots, B-1\} \textit{ für } k = -m, \ldots, -1.$$

Man bezeichnet die ganze Zahl e als **Exponenten**, *$B \geq 2$ als* **Basis**, *die x_k, $k = -m, \ldots, -1$ als* **Ziffern** *und $\sum_{k=-m}^{-1} x_k B^k$ als* **Mantisse**. *Für festes m definieren wir $rd_m(x)$ als die auf m Stellen abgeschnittene Gleitkommadarstellung von x und $rd^m(x)$ als die auf m Stellen gerundete Gleitkommadarstellung von x.*

Beispiel: Die Zahl 123.45 lässt sich als Gleitkommazahl bezüglich der Basis $B = 10$ darstellen als

$$\begin{aligned} 123.45 &= 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0 + 4 \cdot 10^{-1} + 5 \cdot 10^{-2} \\ &= 10^3 \cdot (1 \cdot 10^{-1} + 2 \cdot 10^{-2} + 3 \cdot 10^{-3} + 4 \cdot 10^{-4} + 5 \cdot 10^{-5}) \\ &= 10^3 \cdot 0.12345 \end{aligned}$$

Der Exponent von 123.45 ist also $e = 3$, die Mantisse ist 0.12345 und im vorliegenden Fall reichen $m = 5$ Ziffern, um die Zahl exakt darzustellen. Für $m = 4$ ist

$$\begin{aligned} rd^4(123.45) &= 10^3 \cdot (1 \cdot 10^{-1} + 2 \cdot 10^{-2} + 3 \cdot 10^{-3} + 5 \cdot 10^{-4}) \\ &= 10^3 \cdot 0.1235 \\ rd_4(123.45) &= 10^3 \cdot (1 \cdot 10^{-1} + 2 \cdot 10^{-2} + 3 \cdot 10^{-3} + 4 \cdot 10^{-4}) \\ &= 10^3 \cdot 0.1234 \end{aligned}$$

die auf vier Stellen gerundete bzw. abgeschnittene Gleitkommadarstellung von 123.45.

In Rechnern verwendet man in der Regel $B = 2$ und eine Stellenzahl von $m = 52$, um mit einem weiteren Vorzeichenbit und 11 Bits für die Exponentendarstellung mit insgesamt 64 Bits pro Zahl aus zukommen. Gleitkommazahlen garantieren eine feste relative Genauigkeit der Zahldarstellung.

Satz 1.6 *Für die nach m Stellen abgeschnittene B-adische Darstellung von x gilt das* **Rundungsgesetz**

$$|x - rd_m(x)| \leq |x| eps$$

mit $eps = B^{1-m}$, d.h. der relative Fehler ist kleiner als $\frac{1}{B^{m-1}}$.

Beweis: Sei $x > 0$ und

$$x = \sum_{k=-\infty}^{n(x)} b_k B^k$$

die B-adische Darstellung von x mit Ziffern $b_k \in \{0, \ldots, B-1\}$, Basis $B \geq 2$ und einer führenden Ziffer $b_{n(x)} \neq 0$ mit einer ganzen Zahl $n(x)$. Normierung wie in Definition 1.5 ergibt

$$x = B^{1+n(x)} \sum_{k=-\infty}^{-1} b_{n(x)+1+k} B^k,$$

woraus wir berechnen können, dass

$$\begin{aligned}
|x - rd_m(x)| &= B^{1+n(x)} \sum_{k=-\infty}^{-1} b_{n(x)+1+k} B^k - B^{1+n(x)} \sum_{k=-m}^{-1} b_{n(x)+1+k} B^k \\
&= B^{1+n(x)} \sum_{k=-\infty}^{-m-1} b_{n(x)+1+k} B^k \\
&\leq B^{1+n(x)} \sum_{k=-\infty}^{-m-1} (B-1) B^k \quad \text{weil } b_i \in \{1, \ldots, B-1\}\ \forall i \\
&= B^{1+n(x)-m} \quad \text{siehe } (*) \\
&\leq |x| B^{1-m},
\end{aligned}$$

weil $x \geq b_{n(x)} B^{n(x)} \geq B^{n(x)}$. Um $(*)$ zu rechtfertigen, summieren wir auf:

$$\begin{aligned}
\sum_{k=-\infty}^{-m-1} (B-1) B^k &= (B-1) \sum_{k=m+1}^{\infty} \left(\frac{1}{B}\right)^k \\
&= (B-1) \left(\sum_{k=0}^{\infty} \left(\frac{1}{B}\right)^k - \sum_{k=0}^{m} \left(\frac{1}{B}\right)^k \right) \\
&= (B-1) \left(\frac{1}{1-\left(\frac{1}{B}\right)} - \frac{1-\left(\frac{1}{B}\right)^{m+1}}{\left(1-\frac{1}{B}\right)} \right) \\
&= (B-1) \frac{1}{1-\frac{1}{B}} \left(\frac{1}{B}\right)^{m+1} = \left(\frac{1}{B}\right)^m.
\end{aligned}$$

Den Fall $x < 0$ behandelt man analog. QED

Heutige Rechner stellen also alle reelle Zahlen mit einem maximalen relativen Fehler von $eps = 2^{-51} \approx 4.4409 \cdot 10^{-16}$ dar. Die 16te Dezimalstelle ist also bis auf 5 Einheiten genau. Erfreulicherweise ist auf den meisten Rechenanlagen gewährleistet, dass auch alle Einzeloperationen $(+.-, \cdot, /$ aber auch $\sin, \sqrt{}, \exp \ldots)$ auf

Gleitkommazahlen mit einem maximalen relativen Fehler $eps = 2^{-51}$ ausgeführt werden. Deshalb bezeichnet man eps auch als **Maschinengenauigkeit**. Dennoch können sich die entstehenden Rundungsfehler im Verlauf eines Verfahrens vergrößern.

Der **(gesamte) Rundungsfehler eines Verfahrens** entsteht

- durch die Rundung der Eingabedaten auf Gleitkommazahlen,
- durch die Rundungsfehler der einzelnen Gleitkommaoperationen, sowie
- durch Fortpflanzung der Eingabefehler und der einzelnen Rundungsfehler bei nachfolgenden Operationen.

Im nächsten Abschnitt werden wir uns daher mit der Übertragung von Fehlern beschäftigen.

1.5 Fehlerfortpflanzung, Kondition und Stabilität

In der numerischen Mathematik heißt ein Verfahren *stabil*, wenn es gegenüber kleinen Störungen der Daten unempfindlich ist. Insbesondere bedeutet dies, dass sich Rundungsfehler nicht zu stark auf die Berechnung auswirken.
Die Beziehung zwischen Kondition eines Problems und Stabilität lässt sich wie folgt beschreiben: Es sei $f(y)$ das mathematische Problem in Abhängigkeit einer Eingangsgröße y und es sei $\tilde{f}$ der numerische Algorithmus, sowie $\tilde{y}$ die gestörten Eingangsdaten. Nach Definition 1.4 interessieren wir uns für den folgenden (absoluten) Fehler:

$$|\tilde{f}(\tilde{y}) - f(y)|.$$

Mit der Dreiecksungleichung gilt:

$$\begin{aligned} |\tilde{f}(\tilde{y}) - f(y)| &= |\tilde{f}(\tilde{y}) - f(\tilde{y}) + f(\tilde{y}) - f(y)| \\ &\leq |\tilde{f}(\tilde{y}) - f(\tilde{y})| + |f(\tilde{y}) - f(y)|. \end{aligned}$$

Hierbei sagt der erste Fehler-Term aus, wie gut sich das Verfahren $\tilde{f}$ im Vergleich mit der exakten Lösung f des Problems bei gestörten Eingangsdaten $\tilde{y}$ verhält. Dieser Term ist klein, wenn das Verfahren *stabil* ist. Der zweite Term hängt dagegen nicht von dem Verfahren ab, sondern ausschließlich von dem Problem. Er ist klein, wenn das Problem *gut konditioniert* ist. Die Stabilität ist also eine Eigenschaft des Algorithmus und die Kondition eine Eigenschaft des Problems.

Im Anschluss an den letzten Abschnitt wollen wir nun den zweiten Term weiter untersuchen. Wir möchten also analysieren, wie sich relative Fehler (die z.B. durch Rundung entstanden sein können) durch verschiedene Operationen fortpflanzen, wenn diese exakt ausgeführt werden. Dazu betrachten wir zunächst die Operationen $+, \cdot, /$.

Lemma 1.7 *Seien $x, y \in \mathbb{R} \setminus \{0\}$ mit relativen Fehlern*

$$\varepsilon_x = \left|\frac{\tilde{x} - x}{x}\right|, \quad \varepsilon_y = \left|\frac{\tilde{y} - y}{y}\right|.$$

Für den relativen Fehler bei der Addition gilt

$$\left|\frac{\tilde{x} + \tilde{y} - (x + y)}{x + y}\right| \leq \varepsilon_x \left|\frac{x}{x + y}\right| + \varepsilon_y \left|\frac{y}{x + y}\right|.$$

Beweis: Nachrechnen zeigt, dass

$$\left|\frac{\tilde{x} + \tilde{y} - (x + y)}{x + y}\right| \leq \frac{|\tilde{x} - x| + |\tilde{y} - y|}{|x + y|} = \frac{|x|\varepsilon_x + |y|\varepsilon_y}{|x + y|} = \varepsilon_x \left|\frac{x}{x + y}\right| + \varepsilon_y \left|\frac{y}{x + y}\right|.$$

QED

Haben x und y das gleiche Vorzeichen, so ergibt sich also bei der Addition der beiden Zahlen ein relativer Fehler von höchstens $\varepsilon_x + \varepsilon_y$. Dagegen kann der relative Fehler bei der Subtraktion von zwei Zahlen x, y gleichen Vorzeichens (also der Addition von x und $-y$) den möglicherweise sehr großen Wert

$$\varepsilon_x \left|\frac{x}{x - y}\right| + \varepsilon_y \left|\frac{y}{x - y}\right|$$

erreichen.

Lemma 1.8 *Seien $x, y \in \mathbb{R} \setminus \{0\}$ mit relativen Fehlern*

$$\varepsilon_x = \left|\frac{\tilde{x} - x}{x}\right|, \quad \varepsilon_y = \left|\frac{\tilde{y} - y}{y}\right|.$$

Unter Vernachlässigung von Produkten von Fehlern lässt sich der relative Fehler bei der Multiplikation abschätzen durch

$$\left|\frac{\tilde{x}\tilde{y} - xy}{xy}\right| \leq \varepsilon_x + \varepsilon_y$$

und der relative Fehler bei der Division ebenfalls durch

$$\left|\frac{\frac{\tilde{x}}{\tilde{y}} - \frac{x}{y}}{\frac{x}{y}}\right| \leq \varepsilon_x + \varepsilon_y.$$

Beweis: Auch hier rechnen wir nach:

$$\begin{aligned}\left|\frac{\tilde{x}\tilde{y} - xy}{xy}\right| &= \left|\frac{(\tilde{x} - x)\tilde{y} + x(\tilde{y} - y)}{xy}\right| \leq \frac{|\tilde{x} - x||\tilde{y}| + |x||\tilde{y} - y|}{|xy|} \\ &= \varepsilon_x \left|\frac{\tilde{y}}{y}\right| + \varepsilon_y = \varepsilon_x\varepsilon_y + \varepsilon_x + \varepsilon_y,\end{aligned}$$

wobei im letzten Schritt ausgenutzt wurde, dass

$$\left|\frac{\tilde{y}}{y}\right| \leq \frac{|\tilde{y}-y|}{|y|} + \frac{|y|}{|y|} = \varepsilon_y + 1.$$

Vernachlässigen wir nun Produkte von Fehlern, erhalten wir das gewünschte Ergebnis.
Es fehlt noch die Fehlerübertragung bei der Division:

$$\begin{aligned}\left|\frac{\frac{\tilde{x}}{\tilde{y}} - \frac{x}{y}}{\frac{x}{y}}\right| &= \left|\frac{\tilde{x}y - x\tilde{y}}{y\tilde{y}} \cdot \frac{y}{x}\right| = \left|\frac{(\tilde{x}-x)y + x(y-\tilde{y})}{x \cdot y \cdot \tilde{y}}\right| \cdot |y| \\ &\leq \varepsilon_x \left|\frac{y}{\tilde{y}}\right| + \varepsilon_y \left|\frac{y}{\tilde{y}}\right| = (\varepsilon_x + \varepsilon_y),\end{aligned}$$

wobei hier im letzten Schritt verwendet wurde, dass

$$\begin{aligned}\left|\frac{y}{\tilde{y}}\right| &\leq \frac{|\tilde{y}-y| + |\tilde{y}|}{|\tilde{y}|} = 1 + \frac{|\tilde{y}-y|}{|y|} \cdot \frac{|y|}{|\tilde{y}|} \\ &\leq 1 + \varepsilon_y \frac{|y|}{|\tilde{y}|} \leq 1 + \varepsilon_y (1 + \varepsilon_y \frac{|y|}{|\tilde{y}|} \leq \\ &= \leq 1 + \varepsilon_y + \varepsilon_y^2 \frac{|y|}{|\tilde{y}|} \leq 1 + \varepsilon_y + \varepsilon_y^2 + \varepsilon^3 \frac{|y|}{|\tilde{y}|} \leq \ldots\end{aligned}$$

Vernachlässigen der Produkte von Fehlern ergibt auch hier das gewünschte Ergebnis. QED

Notation 1.9 *Die* **Kondition** *eines Problems ist der im ungünstigsten Fall auftretende Vergrößerungsfaktor für den Einfluss von relativen Eingangsfehlern auf relative Ergebnisfehler. Ist die Kondition eines Problems groß, so spricht man von einem* **schlecht konditionierten** *Problem.*

Ist das zu betrachtende Problem die Multiplikation oder Division von zwei Zahlen (d.h. $f(x,y) = x \cdot y$ oder $f(x,y) = \frac{x}{y}$) so haben wir bereits gezeigt, dass $\left|\frac{f(\tilde{x},\tilde{y})-f(x,y)}{f(x,y)}\right|$ klein ist: Im schlimmsten Fall ist für den relativen Fehler eine Addition der Beträge der relativen Fehler $\varepsilon_y + \varepsilon_y$ der Eingangsgrößen x und y zu erwarten. Beide Probleme sind also gut konditioniert. Betrachten wir nun die Addition: Haben die beiden zu addierenden Zahlen das gleiche Vorzeichen, so sind die Faktoren $|\frac{x}{x+y}|$ und $|\frac{y}{x+y}|$ aus Lemma 1.7 beide durch 1 beschränkt, so dass wir wieder eine gute Kondition erhalten. Bei der Addition von Zahlen verschiedenen Vorzeichens (also der Subtraktion von Zahlen gleichen Vorzeichens) können die beiden Faktoren $|\frac{x}{x-y}|$ und $|\frac{y}{x-y}|$ dagegen beliebig groß werden. Dieses Problem ist also schlecht konditioniert. Wir demonstrieren das an einem Beispiel:

Rechnen wir mit 6-stelliger Genauigkeit und subtrahieren die beiden 6-stelligen Zahlen $x = 1234.00$ und $y = 1233.99$. Angenommen, der relative Fehler von x beträgt nur $\varepsilon_x = 0.1$, also z.B. $\tilde{x} = 1357.40$ und der relative Fehler von y ist sogar Null ($\tilde{y} = y$). Dennoch ergibt sich ein relativer Fehler der Differenz von

$$\varepsilon_{x-y} = \frac{\tilde{x} - \tilde{y} - (x - y)}{x - y} = \frac{123.41 - 0.01}{0.01} = 12340.00$$

und das, obwohl wir exakt gerechnet haben!

An dem Beispiel sehen wir, dass die Subtraktion von zwei Zahlen schlecht konditioniert ist, wenn die beiden zu subtrahierenden Zahlen fast gleich groß sind. Der Effekt wird entsprechend auch als **Auslöschung** bezeichnet. Er ist ein ernstzunehmendes Problem im wissenschaftlichen Rechnen. Man sollte daher, wenn es irgendwie möglich ist, die Differenzenbildung von fast gleich großen Zahlen vermeiden, oder zumindest möglichst zum Schluss eines Verfahrens ausführen!

Genauso problematisch ist die Situation bei der numerischen Berechnung von Ableitungen einer Funktion $f : \mathbb{R} \to \mathbb{R}$: Der Ausdruck

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

führt bei kleinen Werten von h, $h > 0$ immer zu einer Auslöschung bei der Differenzenbildung. Ein relativer Fehler von maximal ε in der Berechnung der f-Werte hat bei der Differenzenbildung nach Lemma 1.7 schlimmstenfalls einen relativen Fehler von

$$\begin{aligned}
\varepsilon_{f(x+h)-f(x)} &\leq \left|\frac{f(x+h)}{f(x+h)-f(x)}\right| \varepsilon_{f(x+h)} + \left|\frac{f(x)}{f(x+h)-f(x)}\right| \varepsilon_{f(x)} \\
&\leq \left|\frac{f(x+h)}{f(x+h)-f(x)}\right| \varepsilon + \left|\frac{f(x)}{f(x+h)-f(x)}\right| \varepsilon \\
&= \frac{|f(x+h)| + |f(x)|}{|f(x+h)-f(x)|}\varepsilon \\
&\approx \frac{2\varepsilon|f(x)|}{|hf'(x)|}
\end{aligned}$$

zur Folge. Das Problem ist also für kleine Werte von h (oder betragsmäßig kleine Werte von $f'(x)$) schlecht konditioniert. Man ist hier in einer Zwickmühle: Für kleine Werte von h ist die Auslöschung groß, für große h ist dagegen der Diskretisierungsfehler groß. Einige weitere Betrachtungen führen zu der Faustregel $h \approx \sqrt{\varepsilon}$, die z.B. in [Schaback und Wendland, 2004] nachgelesen werden kann.

Kapitel 2

Lineare Gleichungssysteme: Eliminationsverfahren

2.1 Begriffe und Grundlagen

Wir wollen zunächst die nötigen Notationen einführen und dabei einige Begriffe und Ergebnisse aus der Linearen Algebra wiederholen.

Notation 2.1 *$A \in \mathbb{K}^{m,n}$ bezeichne eine reelle oder komplexe $m \times n$* **Matrix**, *d.h. eine Matrix mit m Zeilen und n Spalten. Wir schreiben*

$$A = (a_{ij})_{\substack{i=1,\dots,m,\\ j=1,\dots,n}} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} = \begin{pmatrix} A_1 & A_2 & \dots & A_n \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}.$$

Dabei bezeichnen a_{ij} die Elemente der Matrix A, A_j die Spalten der Matrix und a_i ihre Zeilen, $i = 1, \dots, m, j = 1, \dots, n$. Gilt $m = n$ so nennt man die Matrix **quadratisch**.

Matrizen kann man miteinander multiplizieren, allerdings ist die Matrixmultiplikation nicht kommutativ. Die Einheitsmatrix bezüglich der Multiplikation von quadratischen Matrizen ist $I \in \mathbb{K}^{n,n}$ mit Elementen $e_{ij} = 0$ für alle $i \neq j$, $e_{ii} = 1, i = 1, \dots, n$. Gibt es zu einer Matrix A eine Matrix A^{-1} mit $A \cdot A^{-1} = A^{-1} \cdot A = I$, so nennt man A invertierbar.
Wir können jetzt definieren, was ein lineares Gleichungssystem ist:

Definition 2.2 *Ein* **lineares Gleichungssystem**

$$Ax = b$$

ist gegeben durch eine Matrix $A \in \mathbb{K}^{m,n}$*, einen Vektor* $b = (b_1, \ldots, b_m)^T \in \mathbb{K}^m$ *und* n *Variablen* $x_1, \ldots, x_n$*, geschrieben als Vektor* $x = (x_1, \ldots, x_n)^T$*. Ausgeschrieben erhält man* m *Gleichungen*

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ \vdots & \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m. \end{aligned}$$

Falls $b = 0$ *nennt man das Gleichungssystem homogen.*
Ist $m < n$ *so heißt das Gleichungssystem* **unterbestimmt**.

Lineare Gleichungssysteme haben ausgesprochen viele Anwendungen. Einerseits tauchen sie direkt als praktische Probleme auf, andererseits sind sie ein wichtiger Baustein für viele numerische Verfahren, z.B. zur numerischen Lösung von Differentialgleichungen.

Wir wiederholen einige Begriffe aus der linearen Algebra. Seien $A_1, \ldots, A_p \in \mathbb{K}^n$ Vektoren. Dann bezeichne

$$\operatorname{span}\{A_1, \ldots, A_p\} = \left\{ \sum_{i=1}^{p} \alpha_i A_i : \alpha_i \in \mathbb{K} \right\}$$

die Menge der von $A_1, \ldots, A_p$ erzeugten Linearkombinationen (die **lineare Hülle** von $A_1, \ldots A_p$).
Sicherheitshalber erinnern wir noch an den Begriff der linearen Unabhängigkeit: Die Vektoren $A_1, \ldots, A_p$ heißen linear unabhängig, falls aus $\sum_{i=1}^{p} \alpha_i A_i = 0$ folgt, dass $\alpha_i = 0$ für $i = 1, \ldots, p$. Die Anzahl der linear unabhängigen Spalten einer Matrix A definiert den Spaltenrang der Matrix und dieser entspricht ihrem Zeilenrang, d.h. der Anzahl der linear unabhängigen Zeilen von A.

Satz 2.3 *Sei* $A \in \mathbb{K}^{n,n}$*. Die folgenden Aussagen sind äquivalent:*

(i) A *ist invertierbar.*

(ii) $\det(A) \neq 0$.

(iii) Die Spalten $A_1, \ldots, A_n$ *von* A *sind linear unabhängig.*

(iv) Die Zeilen $a_1, \ldots, a_n$ *von* A *sind linear unabhängig.*

Die Matrix A nennt man in obigem Fall auch *regulär* oder *nicht singulär*. Eine $m \times n$ Matrix A kann man als lineare Abbildung

$$\begin{aligned} A : \mathbb{K}^n &\to \mathbb{K}^m \\ x &\mapsto Ax \end{aligned}$$

auffassen und man kann dementsprechend z.B. vom *Kern*

$$\text{Kern}(A) = \{x \in \mathbb{K}^n : Ax = 0\}$$

der Matrix sprechen.

Bevor wir numerische Verfahren zur Lösung eines linearen Gleichungssystems entwickeln, fassen wir einige Ergebnisse (die alle schon bekannt sein sollten) über die Lösbarkeit linearer Gleichungssysteme zusammen.

Satz 2.4

- *Das Gleichungssystem $Ax = b$ hat genau dann mindestens eine Lösung, wenn $b \in \text{span}\{A_1, \ldots, A_n\}$.*
- *Das Gleichungssystem $Ax = b$ hat genau dann höchstens eine Lösung, wenn $A_1, \ldots, A_n$ linear unabhängig sind.*
- *Das Gleichungssystem $Ax = b$ ist genau dann eindeutig lösbar, wenn die Matrix A nicht singulär ist. In diesem Fall ist $x = A^{-1}b$ die eindeutige Lösung.*

Aufgabe: *Beweisen Sie Satz 2.4!*

Für unterbestimmte Gleichungssysteme gilt, dass sie – wenn sie überhaupt lösbar sind – niemals eindeutig lösbar sein können: Sei $\bar{x}$ eine Lösung des Gleichungssystems, also $A\bar{x} = b$. Betrachten wir nun das entsprechende homogene System

$$Ax = 0.$$

Weil $m < n$ sind die Vektoren $A_1, \ldots, A_n \in \mathbb{K}^m$ linear abhängig, also gibt es ein $y \neq 0$ mit $Ay = 0$. Dementsprechend gilt $\bar{x} + y \neq \bar{x}$, aber wegen

$$A(\bar{x} + y) = A\bar{x} + Ay = b + 0 = b$$

ist auch $\bar{x}+y$ eine Lösung des Gleichungssystems. Genauer lässt sich die Lösungsmenge durch $\{\bar{x} + y : y \in \text{Kern}(A)\}$ angeben.

Das Problem $Ax = b$ heißt *schlecht gestellt*, wenn es nicht eindeutig lösbar ist.

Übersicht über Verfahren zum Lösen von linearen Gleichungssystemen

Man unterscheidet zunächst zwischen *direkten* und *iterativen* Verfahren. Bei den direkten Verfahren erhält man nach endlich vielen Schritten eine Lösung des Problems. Die bekanntesten hiervon sind die sogenannten *Eliminationsverfahren*, bei denen in jedem Schritt eine der n Unbekannten eliminiert wird. Dazu gehören das Gauß-Verfahren (siehe Abschnitt 2.2) und das Cholesky-Verfahren (Abschnitt 2.3). Das QR-Verfahren ist ein *Orthogonalisierungsverfahren* zur Lösung linearer Gleichungssysteme oder zur Behandlung von linearen Ausgleichsproblemen. Es wird in Kapitel 4 besprochen. *Iterative Verfahren* starten mit einer Näherungslösung, die in jedem Schritt verbessert wird. Sie sind vor allem bei großen Gleichungssystemen oder bei Gleichungssystemen mit spezieller Struktur der Matrix A sinnvoll. Mit ihnen werden wir uns in Kapitel 5 beschäftigen.

2.2 Gauß-Verfahren und LU-Zerlegung

Idee: Betrachten wir als Beispiel ein Gleichungssystem mit

$$A = \begin{pmatrix} 1 & 3 & 2 \\ 0 & 5 & 4 \\ 0 & 0 & 6 \end{pmatrix}, b = \begin{pmatrix} 9 \\ 14 \\ 6 \end{pmatrix}.$$

Ausgeschrieben erhält man das folgende *gestaffelte* Gleichungssystem

$$\begin{aligned} 1x_1 + 3x_2 + 2x_3 &= 9 \\ 5x_2 + 4x_3 &= 14 \\ 6x_3 &= 6. \end{aligned}$$

Die dritte Gleichung $6x_3 = 6$ enthält nur eine Unbekannte; entsprechend lässt sich der Wert $x_3 = 1$ bestimmen. Setzt man diesen in die zweite Gleichung ein erhält man $5x_2 = 10$, also $x_2 = 2$. Setzt man abschließend die beiden gefundenen Werte in die erste Gleichung ein, ergibt sich $x_1 = 1$.

Die Idee des Gauß-Verfahrens nutzt nun diese einfache Lösbarkeit gestaffelter Gleichungssysteme aus: Ein gegebenes Gleichungssystem wird in ein gestaffeltes Gleichungssystem transformiert und dann gelöst. Formalisieren wir dazu zunächst, wie man solche gestaffelten Gleichungssysteme beschreiben und lösen kann.

Definition 2.5 *Eine quadratische Matrix $A \in \mathbb{K}^{n,n}$ heißt* **untere Dreiecksmatrix**, *falls $a_{ij} = 0$ für alle $i < j$. A heißt* **obere Dreiecksmatrix**, *falls $a_{ij} = 0$ für alle $i > j$. Eine Dreiecksmatrix heißt* **normiert**, *falls $a_{ii} = 1$ für $i = 1, \ldots, n$. Ist A eine Dreiecksmatrix, so bezeichnet man $Ax = b$ als* **gestaffeltes** *Gleichungssystem.*

Bemerkung: Eine $n \times n$-Dreiecksmatrix ist genau dann regulär, wenn $a_{ii} \neq 0$ für alle $i = 1, \ldots, n$.

Lemma 2.6 (Lösen durch Rückwärtselimination) *Sei A eine obere Dreiecksmatrix mit Diagonalelementen $a_{ii} \neq 0$ für $i = 1, \ldots, n$. Die Lösung von $Ax = b$ lässt sich dann sukzessive durch*

$$x_j = \frac{1}{a_{jj}} \left(b_j - \sum_{k=j+1}^{n} a_{jk} x_k \right), \qquad j = n, \ldots, 1$$

bestimmen.

Beweis: Die Gültigkeit der Formel überprüft man schnell (ausgehend von $j = n$). QED

Obiges Verfahren heißt *Lösen durch Rückwärtseinsetzen* oder *Rückwärtselimination* weil man mit der letzten Gleichung beginnt. Analog kann man gestaffelte Gleichungssysteme mit unterer Dreiecksmatrix durch *Vorwärtseinsetzen* lösen:

Lemma 2.7 (Lösen durch Vorwärtselimination) *Sei A eine untere Dreiecksmatrix mit Diagonalelementen $a_{ii} \neq 0$ für $i = 1, \ldots, n$. Die Lösung von $Ax = b$ lässt sich dann sukzessive durch*

$$x_j = \frac{1}{a_{jj}} \left(b_j - \sum_{k=1}^{j-1} a_{jk} x_k \right), \qquad j = 1, \ldots, n$$

bestimmen.

Aufwand: Beim Lösen durch Rückwärtseinsetzen (oder Lösen durch Vorwärtseinsetzen) benötigt man n Divisionen, $\frac{1}{2}n(n-1)$ Multiplikationen und $\frac{1}{2}n(n-1)$ Subtraktionen, also einen Gesamtaufwand von $\mathrm{O}(n^2)$.

Definition 2.8 *Eine Faktorisierung einer Matrix $A \in \mathbb{K}^{n,n}$ der Form $A = LU$ mit einer regulären unteren Dreiecksmatrix L und einer regulären oberen Dreiecksmatrix U heißt LU-Zerlegung von A.*

Ist eine LU-Zerlegung von A bekannt, so lässt sich die Lösung des Gleichungssystems $Ax = b$ durch das Lösen von zwei gestaffelten Gleichungssystemen bestimmen: Durch Vorwärtselimination löst man zuerst das Gleichungssystem

$$Lz = b$$

und anschließend durch Rückwärtselimination

$$Ux = z.$$

Die so erhaltene Lösung x erfüllt dann

$$Ax = LUx = Lz = b$$

und ist somit eine Lösung von $Ax = b$.

Bevor wir uns ansehen, wie man ein gegebenes Gleichungssystem in ein gestaffeltes Gleichungssystem verwandelt, zunächst noch folgende Beobachtung.

Satz 2.9 *Folgende Mengen sind Gruppen bezüglich der Matrixmultiplikation: Die Menge der regulären oberen Dreiecksmatrizen, die Menge der oberen normierten Dreiecksmatrizen, die Menge der regulären unteren Dreiecksmatrizen, die Menge der unteren normierten Dreiecksmatrizen.*

Beweis: Sei Δ eine der oben genannten Mengen. Zu zeigen ist

(0) $A, B \in \Delta \Rightarrow A \cdot B \in \Delta$.

(1) $A \cdot (B \cdot C) = (A \cdot B) \cdot C$ für alle $A, B, C \in \Delta$.

(2) Die Einheitsmatrix $I \in \Delta$.

(3) $A \in \Delta \Rightarrow A^{-1} \in \Delta$.

(1) ist für alle Matrizen richtig und (2) ist klar. Wir zeigen also (0) und (3) für die Menge Δ der (normierten) oberen Dreiecksmatrizen. (Für untere Dreiecksmatrizen verläuft der Beweis analog.)

ad (0): Seien $A, B \in \Delta$ und $C = (c_{ij}) = AB$. Dann ist

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} = \sum_{k=i}^{j} a_{ik} b_{kj},$$

weil $a_{ik} = 0$ für $i > k$ und $b_{kj} = 0$ für $k > j$. Für $i > j$ gilt also $c_{ij} = 0$ und damit ist C eine obere Dreiecksmatrix. (Man beachte, dass die Regularität der Matrizen A und B hierfür nicht nötig ist.)

Sind A, B weiterhin normiert, so auch C, denn

$$c_{ii} = a_{ii} b_{ii} = 1.$$

ad (3): Sei $A \in \Delta$ regulär und $A^{-1} = B = (b_{ij})$ die Inverse von A. Dann gilt für die Spalten $B_1, B_2, \ldots, B_n$ von der Inversen

$$AB_k = e_k.$$

Für jedes $k \in \{1, \ldots, n\}$ kann $B_k = (b_{1k}, b_{2k}, \ldots, b_{nk})^T$ also als Lösung von $Ax = e_k$ aufgefasst werden. Nach Lemma 2.6 folgt, dass $b_{jk} = 0$ für $j = n, n-1, \ldots, k+1$ und $b_{kk} = \frac{1}{a_{kk}}$, also ist B eine obere Dreiecksmatrix, die für eine normierte Matrix A auch wieder normiert ist.

QED

Man sieht hier schon direkt die folgende Aussage:

Lemma 2.10 *Hat eine reguläre Matrix A eine LU-Zerlegung mit normierter unterer Dreiecksmatrix L, so ist diese eindeutig.*

Beweis: Weil $\det(A) \neq 0$ sind auch die Matrizen L, U mit $A = LU$ regulär. Sei nun $A = L_1U_1 = L_2U_2$. Das ist wegen der Regularität aller beteiligten Matrizen äquivalent zu

$$U_1U_2^{-1} = L_1^{-1}L_2.$$

Wegen Satz 2.9 steht links eine obere und rechts eine untere Dreiecksmatrix. Um Gleichheit zu gewähren, muss also

$$U_1U_2^{-1} = I = L_1^{-1}L_2$$

gelten, und dementsprechend folgern wir $L_1 = L_2$ und $U_1 = U_2$. QED

Für Dreiecksmatrizen ist das Lösen von linearen Gleichungssystemen also einfach. Was aber macht man, wenn die Koeffizientenmatrix nicht in Dreiecksform vorliegt? Die Idee des Gauß-Verfahrens besteht dann darin, die Matrix durch *elementare Zeilenoperationen* in eine Matrix in Dreiecksform zu transformieren. Das kann man mit Hilfe der folgenden Matrizen formulieren:

Definition 2.11 *Für einen Vektor $l^{(k)} = (0, \ldots, 0, t_{k+1}, \ldots, t_n)^T \in \mathbb{K}^n$ mit $1 \leq k \leq n$ und dem k-ten Einheitsvektor $e_k \in \mathbb{K}^n$ ist die* **Gauß-Matrix** *M_k definiert durch*

$$M_k := I_n - l^{(k)}e_k^T = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ & & & -t_{k+1} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -t_n & & & 1 \end{pmatrix}.$$

Sammeln wir zunächst einige Eigenschaften der Gauß-Matrizen.

Lemma 2.12 *Sei M_k die Gauß-Matrix bezüglich eines Vektors $l^{(k)} = (0, \ldots, 0, t_{k+1}, \ldots, t_n)^T$.*

1. $\det(M_k) = 1$
2. $M_k^{-1} = I_n + l^{(k)}e_k^T$

Beweis: Da die Gauß-Matrizen untere Dreiecksmatrizen sind, folgt der erste Teil des Lemmas. Für den zweiten Teil rechnet man nach

$$\begin{aligned} M_kM_k^{-1} &= (I_n - l^{(k)}e_k^T)(I_n + l^{(k)}e_k^T) \\ &= I_n + l^{(k)}e_k^T - l^{(k)}e_k^T - l^{(k)}e_k^Tl^{(k)}e_k^T = I_n, \end{aligned}$$

wobei im letzten Schritt ausgenutzt wurde, dass $e_k^Tl^{(k)} = 0$ gilt. Analog erhält man $M_k^{-1}M_k = I_n$ QED

Multipliziert man eine Gauß-Matrix M_k von links mit einer Matrix A, so erhält man als Ergebnis eine Matrix A', die aus A entsteht, indem man das t_j-te Vielfache der k-ten Zeile a_k von A von der j-ten Zeile abzieht, für $j = k+1, \ldots, n$. In Formeln erhält man also:

$$M_k\ A = \begin{pmatrix} a_1 \\ \vdots \\ a_k \\ a_{k+1} - t_{k+1}a_k \\ \vdots \\ a_n - t_n a_k \end{pmatrix}.$$

Man nennt diese Operation auch die *Anwendung elementarer Zeilenoperationen.* Lemma 2.12 besagt dabei, dass die Anwendung von elementaren Zeilenoperationen die Determinante der Matrix nicht verändert.

Setzt man für einen Vektor $b = (b_1, \ldots, b_n)^T$ und eine Zahl $k \in \{1, \ldots, n\}$ mit $b_k \neq 0$

$$l^{(k)} = \left(0, \ldots, 0, \frac{b_{k+1}}{b_k}, \ldots, \frac{b_n}{b_k}\right)^T$$

so erhält man

$$M_k b = (b_1, b_2, \ldots, b_k, 0, \ldots, 0)^T. \tag{2.1}$$

Genau das wird im Gauß-Verfahren zur Transformation einer Matrix auf Dreiecksform ausgenutzt. Das folgende Verfahren ist in der angegebenen Form zur Implementierung allerdings ungeeignet, weil Matrixoperationen rechenzeitmäßig einen hohen Aufwand bedeuten. Eine effizientere Variante wird in Algorithmus 3 auf Seite 35 beschrieben.

Algorithmus 1: Gauß-Verfahren ohne Spaltenpivotsuche (Matrixversion)

Input: $A \in \mathbb{K}^{n,n}$

Schritt 1: $A^{(1)} := A$

Schritt 2: **For** $k = 1, \ldots, n-1$ **do**

$$\begin{aligned} l^{(k)} &:= \left(\underbrace{0, \ldots, 0}_{k \text{ mal}}, \frac{a^{(k)}_{k+1,k}}{a^{(k)}_{kk}}, \ldots, \frac{a^{(k)}_{n,k}}{a^{(k)}_{kk}}\right)^T \\ M_k &:= I_n - l^{(k)} e_k^T \\ A^{(k+1)} &:= M_k A^{(k)} \end{aligned}$$

Ergebnis: LU `Zerlegung von` A `mit`

$$\begin{aligned} U &:= A^{(n)} \\ L &:= M_1^{-1} \cdot M_2^{-1} \cdots M_{n-1}^{-1} \end{aligned}$$

Wir müssen nun zeigen, dass obiger Algorithmus hält, was er verspricht, d.h. dass wirklich $A = LU$ gilt, und L eine untere und U eine obere Dreiecksmatrix ist. Außerdem muss die **Durchführbarkeit** des Verfahrens untersucht werden, die nur dann gewährleistet ist, wenn $a_{kk}^{(k)} \neq 0$ für alle $k = 1, \ldots, n-1$. Dazu betrachten wir die **Hauptminoren** $A^{[k]}$ der Matrix A.

Satz 2.13 (Korrektheit des Gauß-Verfahrens) *Sei für $k = 1, \ldots, n-1$:*

$$\det(A^{[k]}) = \det \begin{pmatrix} a_{11} & \ldots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \ldots & a_{kk} \end{pmatrix} \neq 0. \tag{2.2}$$

Dann ist Algorithmus 1 korrekt. Genauer:

1. *Algorithmus 1 ist durchführbar, d.h.*

$$a_{kk}^{(k)} \neq 0 \text{ für alle } k = 1, \ldots, n-1. \tag{2.3}$$

2. *Für die Matrizen $A^{(k)}$, $k = 1, \ldots, n-1$ gilt:*

$$a_{ij}^{(k)} = 0 \text{ für alle } j < k \text{ und } i > j. \tag{2.4}$$

   Insbesondere ist U eine obere Dreiecksmatrix.

3. *L ist eine untere Dreiecksmatrix.*

4. *$A = LU$.*

Beweis:

ad 1. und 2. Wir zeigen zuerst, dass für jedes feste k (2.3) aus (2.4) folgt, d.h. dass gilt:

$$(2.4) \Longrightarrow (2.3).$$

Danach beweisen wir (2.4) für alle k per Induktion.

Sei also $a_{ij}^{(k)} = 0$ für alle $j < k$ und $i > j$. Wegen Lemma 2.12 wissen wir, dass

$$\begin{aligned} \det(A^{(k)}) &= \det(M_{k-1} A^{(k-1)}) = \det(M_{k-1}) \det(A^{(k-1)}) = \det(A^{(k-1)}) \\ &= \ldots = \det(A). \end{aligned}$$

Wendet man die elementaren Zeilenoperationen ausschließlich auf Submatrizen der Form (2.2) an, gilt diese Gleichung weiterhin, d.h.

$$\begin{aligned}\det\begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix} &= \det\begin{pmatrix} a_{11}^{(k)} & \dots & a_{1k}^{(k)} \\ \vdots & & \vdots \\ a_{k1}^{(k)} & \dots & a_{kk}^{(k)} \end{pmatrix} = \det\begin{pmatrix} a_{11}^{(k)} & a_{12}^{(k)} & \dots & a_{1k}^{(k)} \\ 0 & a_{22}^{(k)} & \dots & a_{2k}^{(k)} \\ \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & a_{kk}^{(k)} \end{pmatrix} \\ &= a_{11}^{(k)} \cdot a_{22}^{(k)} \cdot \ldots \cdot a_{kk}^{(k)},\end{aligned}$$

wobei wir im zweiten Schritt ausgenutzt haben, dass die Matrix $A^{(k)}$ Aussage (2.4) erfüllt. Nach Voraussetzung unseres Satzes ist

$$\det\begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix} \neq 0,$$

also $a_{kk}^{(k)} \neq 0$. Damit ist (2.3) für dieses k gezeigt.

Um (2.4) zu zeigen, nutzen wir diese Aussage in einem Induktionsbeweis. Für den Anfang $k = 1$ ist nichts zu zeigen. Für den Induktionsschritt $k \to k+1$ nehmen wir an, dass (2.4) für k richtig ist. Insbesondere gilt dann nach dem ersten Teil dieses Beweises, dass $a_{kk}^{(k)} \neq 0$. Der Vektor $l^{(k)}$ ist also definiert. Anwendung von (2.1) ergibt die geforderte Eigenschaft $a_{ik}^{(k+1)} = 0$ für alle $i > k$ für die k-te Spalte. Zusammen mit der Induktionsannahme folgt (2.4) für $A^{(k+1)}$.

ad 3. L ist definiert als Produkt der M_k^{-1}. Da die M_k alle untere Dreiecksmatrizen sind, sind nach Satz 2.9 auch ihre Inversen Dreiecksmatrizen, ebenso die Produkte ihrer Inversen, also auch L.

ad 4. Nach Algorithmus 1 gilt

$$U = A^{(n)} = M_{n-1}A^{(n-1)} = M_{n-1}M_{n-2} \cdot \ldots \cdot M_1 A.$$

Wegen $L = M_1^{-1} \cdot M_2^{-1} \cdots M_{n-1}^{-1}$ gilt weiter $L^{-1} = M_{n-1}M_{n-2} \cdots M_1$, also

$$U = L^{-1}A \quad \text{oder} \quad LU = A.$$

Aufgabe: *Eine $n \times n$ Matrix heißt streng diagonal-dominant, falls für alle $i = 1, \dots, n$ gilt:*

$$2|a_{ii}| > \sum_{j=1}^{n} |a_{ij}|.$$

Zeigen Sie, dass jede streng diagonal-dominante Matrix invertierbar ist und dass Algorithmus 1 auch in diesem Fall korrekt ist. Das heißt, die Aussagen von Satz 2.13 bleiben richtig, auch wenn man die bisherige Voraussetzung (2.2) durch die Forderung nach strenger Diagonal-Dominanz ersetzt.

Lemma 2.14 *Ist Algorithmus 1 durchführbar, so gilt für die Matrix L:*

$$L = I + \sum_{k=1}^{n-1} l^{(k)} e_k^T.$$

Beweis: Nach Definition von L und Lemma 2.12 ist

$$\begin{aligned} L &= M_1^{-1} \cdot M_2^{-1} \cdots M_{n-1}^{-1} \\ &= (I + l^{(1)} e_1^T)(I + l^{(2)} e_2^T) \cdots (I + l^{(n-1)} e_{n-1}^T). \end{aligned}$$

Zu zeigen bleibt also, dass für alle m gilt:

$$I + \sum_{k=1}^{m} l^{(k)} e_k^T = (I + l^{(1)} e_1^T)(I + l^{(2)} e_2^T) \cdots (I + l^{(m)} e_m^T).$$

Für $m = 1$ sieht man die Behauptung direkt. Per Induktion leitet man sie dann für beliebige m her: Gelte die Behauptung also für $m - 1$. Dann betrachte

$$\begin{aligned} & (I + l^{(1)} e_1^T)(I + l^{(2)} e_2^T) \cdots (I + l^{(m)} e_m^T) \\ = & \left(I + \sum_{k=1}^{m-1} l^{(k)} e_k^T \right) (I + l^{(m)} e_m^T) \\ = & I + l^{(m)} e_m^T + \sum_{k=1}^{m-1} l^{(k)} e_k^T + \sum_{k=1}^{m-1} l^{(k)} \underbrace{e_k^T l^{(m)}}_{=0} e_m^T \\ = & I + \sum_{k=1}^{m} l^{(k)} e_k^T. \end{aligned}$$

QED

Diese Beobachtung hilft uns, das Gauß-Verfahren effizient zu organisieren: Man speichert die Vektoren $l^{(1)}, l^{(2)}, \ldots, l^{(n-1)}$ über die erzeugten Nullen im unteren Teil der Matrix A, während der obere Teil die Matrix U enthält.

Bevor wir aber die effizientere Variante des Gauß-Verfahrens angeben, möchten wir das Verfahren so erweitern, dass wir es für alle regulären Matrizen anwenden können. Das ist in der Variante aus Algorithmus 1 leider nicht der Fall — sie scheitert schon an einer so einfachen regulären Matrix wie

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Ein weiteres Problem besteht darin, dass bei kleinen, aber von Null verschiedenen Elementen $a_{kk}^{(k)}$ große Rundungsfehler auftreten können, wie das folgende Beispiel zeigt:
Sei das Gleichungssystem

$$\begin{pmatrix} 0.001 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

gegeben. Die einzige nötige Umformung im Gauß-Verfahren führt zu dem System

$$\begin{pmatrix} 0.001 & 1 \\ 0 & -999 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -998 \end{pmatrix}$$

und entsprechend zu der exakten Lösung von

$$x_1 = \frac{1000}{999} \approx 1, x_2 = \frac{998}{999} \approx 1.$$

Angenommen, wir arbeiten mit zweistelliger Gleitkomma-Arithmetik. Dann erhält man nach der ersten Umformung das auf zwei Stellen gerundete Gleichungssystem

$$\begin{pmatrix} 0.10 \cdot 10^{-2} & 0.10 \cdot 10^{1} \\ 0 & -0.1 \cdot 10^{4} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.10 \cdot 10^{1} \\ -0.10 \cdot 10^{4} \end{pmatrix},$$

dessen Lösung sich (sogar bei exakter Rechnung) zu $x_1 = 0$ und $x_2 = 1$ ergibt, also weit von der echten Lösung entfernt liegt.

Erfreulicherweise lassen sich die beiden aufgeführten Schwierigkeiten durch das nun zu beschreibende Verfahren der *Pivotisierung* vermeiden. Im einfachsten Fall der *Zeilenpivotisierung* vertauscht man während des k-ten Schritts des Gauß-Verfahrens die k-te Zeile mit einer darunterliegenden, und zwar der, die den betragsmäßig größten Eintrag in der k-ten Spalte aufweist. Das Ziel dabei ist, dass nach der Vertauschung das neue Element $a_{kk}^{(k)}$ so groß wie möglich wird. Formal wählt man im k-ten Schritt ein $j \in \{k, k+1, \ldots, n\}$ so dass

$$|a_{jk}^{(k)}| \geq |a_{lk}^{(k)}| \text{ für alle } l = k, \ldots, n.$$

In diesem Fall nennt man $a_{jk}^{(k)}$ das **Pivotelement**. Zur formalen Beschreibung dieser Vertauschungen benötigen wir die folgenden Matrizen.

Definition 2.15 *Eine bijektive Abbildung* $\Pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ *heißt* **Permutation** *der Menge* $\{1, \ldots, n\}$. *Eine* $n \times n$ *Matrix* P *heißt* **Permutationsmatrix**, *falls es eine Permutation* Π *so gibt, dass*

$$Pe_i = e_{\Pi(i)} \text{ für alle } i = 1, \ldots, n.$$

P entsteht also durch Permutation der Spalten der Einheitsmatrix. Wir sammeln Eigenschaften von Permutationsmatrizen.

Satz 2.16 *Sei die $n \times n$-Matrix P eine Permutationsmatrix zur Permutation Π. Dann gilt:*

1. *P ist invertierbar.*
2. *P^{-1} ist die Permutationsmatrix, die zu der Permutation Π^{-1} gehört. (Dabei ist Π^{-1} die Umkehrabbildung von Π.)*
3. *P ist orthogonal, das heißt, es gilt $P^{-1} = P^T$.*

Beweis:

ad 1: P besteht aus einer Vertauschung von Spalten der Einheitsmatrix, ist also regulär.

ad 2: Zu zeigen ist $P^{-1}(e_i) = e_{\Pi^{-1}(i)}$: Dazu betrachten wir

$$\begin{aligned} P(e_{\Pi^{-1}(i)}) &= e_{\Pi(\Pi^{-1}(i))} \text{ nach Definition 2.15} \\ &= e_i. \end{aligned}$$

Multiplikation beider Seiten von links mit P^{-1} liefert das Ergebnis.

ad 3: In den Spalten von P stehen die permutierten Spalten der Einheitsmatrix, $P = (e_{\Pi(1)}, e_{\Pi(2)}, \ldots e_{\Pi(n)})$. Entsprechend ist die i-te Zeile von P^T durch $e^T_{\Pi(i)}$ gegeben. Um nachzuweisen, dass P orthogonal ist, müssen wir $PP^T = P^TP = I$ zeigen. Sei $Q = P^TP$. Dann ist

$$Q_{ij} = e^T_{\Pi(i)} e_{\Pi(j)} = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{sonst} \end{cases},$$

also ist $Q = I$. Analog gilt auch $PP^T = I$. QED

Eine Erweiterung der zweiten Aussage des Satzes soll noch erwähnt werden: Für zwei Permutationen Π_1, Π_2 mit zugehörigen Permutationsmatrizen P_1,P_2 gilt wegen

$$\begin{aligned} P_1P_2(e_i) &= P_1(e_{\Pi_2(i)}) \\ &= e_{\Pi_1(\Pi_2(i))} = e_{\Pi_1 \circ \Pi_2(i)}, \end{aligned}$$

dass P_1P_2 die Permutationsmatrix ist, die zu der Permutation $\Pi_1 \circ \Pi_2$ gehört, das heißt, die Verkettung von zwei Permutationen entspricht dem Produkt der entsprechenden Permutationsmatrizen.

Um das Gauß-Verfahren zu verbessern, benötigen wir spezielle Permutationen, nämlich solche, die genau zwei Elemente $r < s$ vertauschen. Zu so einer Permutation

$$\begin{aligned} \Pi(r) &= s, \\ \Pi(s) &= r, \\ \Pi(i) &= i \quad \text{für alle } i \notin \{r, s\} \end{aligned}$$

gehört entsprechend die Matrix

$$P_{rs} = (e_1, \ldots, e_{r-1}, e_s, e_{r+1}, \ldots, e_{s-1}, e_r, e_{s+1}, \ldots, e_n).$$

Solche Matrizen sind symmetrisch, das heißt $P_{rs} = P_{rs}^T$, und man kann sie auch als $P_{rs} = I - (e_r - e_s)(e_r - e_s)^T$ schreiben.

Man sollte sich das folgende einprägen:

- Die Linksmultiplikation $A \cdot P_{rs}$ einer Matrix A mit P_{rs} vertauscht die r-te mit der s-ten Spalte.
- Die Rechtsmultiplikation $P_{rs} \cdot A$ einer Matrix A mit P_{rs} vertauscht die r-te mit der s-ten Zeile.

Formal können wir nun die Matrixversion des Gauß-Verfahrens mit Spaltenpivotisierung folgendermaßen beschreiben.

Algorithmus 2: Gauß-Verfahren mit Spaltenpivotsuche (Matrixversion)

Input: $A \in \mathbb{K}^{n,n}$

Schritt 1: $\tilde{A}^{(1)} := A$

Schritt 2: **For** $k = 1, \ldots, n-1$ **do**

`Bestimme einen Pivotindex` $r \in \{k, \ldots, n\}$ `mit` $|\tilde{a}_{rk}^{(k)}| = \max_{i=k,\ldots,n} |\tilde{a}_{ik}^{(k)}|$.

$$\begin{aligned} P^{(k)} &:= P_{kr} \\ A^{(k)} &:= P^{(k)} \tilde{A}^{(k)} \\ l^{(k)} &:= \left(\underbrace{0, \ldots, 0}_{k \text{ mal}}, \frac{a_{k+1,k}^{(k)}}{a_{kk}^{(k)}}, \ldots \frac{a_{n,k}^{(k)}}{a_{kk}^{(k)}} \right)^T \\ M_k &:= I_n - l^{(k)} e_k^T \\ \tilde{A}^{(k+1)} &:= M_k A^{(k)} \end{aligned}$$

Ergebnis: $PA = LU$ `mit`

$$\begin{aligned} P &:= P^{(n-1)} \cdot \ldots \cdot P^{(1)} \text{ eine Permutationsmatrix} \\ U &:= \tilde{A}^{(n)} \text{ ist eine obere Dreiecksmatrix} \\ L &:= I + \sum_{k=1}^{n-1} \Theta^{(k)} e_k^T \text{ ist eine untere Dreiecksmatrix mit} \\ \Theta^{(k)} &:= P^{(n-1)} \cdot \ldots \cdot P^{(k+1)} l^{(k)}. \end{aligned}$$

Satz 2.17 *Für eine reguläre $n \times n$ Matrix A existiert eine Permutationsmatrizen $P \in \mathbb{R}^{n,n}$, eine normierte untere Dreiecksmatrix $L \in \mathbb{K}^{n,n}$ und eine obere Dreiecksmatrix $U \in \mathbb{K}^{n,n}$ so dass $PA = LU$, und diese Zerlegung wird von Algorithmus 2 gefunden.*

Beweis: Im Beweis zeigen wir zunächst die folgenden Eigenschaften:

1. Algorithmus 2 ist durchführbar, d.h.

$$a_{kk}^{(k)} \neq 0 \text{ für } k = 1, \ldots, n-1. \tag{2.5}$$

2. Für die Matrizen $A^{(k)}$, $k = 1, \ldots, n-1$ gilt:

$$a_{ij}^{(k)} = 0 \text{ für alle } j < k \text{ und } i > j. \tag{2.6}$$

$$A^{(k)} = P^{(k)} M_{k-1} P^{(k-1)} \cdot \ldots \cdot P^{(2)} M_1 P^{(1)} A \tag{2.7}$$

Ähnlich wie im Beweis zu Satz 2.13 zeigen wir, dass für jedes feste $k = 1, \ldots, n-1$ aus den beiden letztgenannten Eigenschaften (2.6) und (2.7) die erstgenannte Aussage $a_{kk}^{(k)} \neq 0$ folgt, und beweisen anschließend (2.6) und (2.7) für alle k per Induktion.

Gelte also (2.6) und (2.7) für k. Wir nehmen an, dass $a_{kk}^{(k)} = 0$. Nach der Definition von $P^{(k)}$ erfüllt die Matrix $A^{(k)} = P^{(k)} \tilde{A}^{(k)}$

$$|a_{kk}^{(k)}| \geq |a_{lk}^{(k)}| \text{ für alle } l = k, \ldots, n.$$

Wegen $a_{kk}^{(k)} = 0$ folgt daraus, dass die erste Spalte der Submatrix

$$C = \begin{pmatrix} a_{kk}^{(k)} & \ldots & a_{kn}^{(k)} \\ \vdots & & \vdots \\ a_{nk}^{(k)} & \ldots & a_{nn}^{(k)} \end{pmatrix}$$

eine Nullspalte ist und entsprechend $\det(C) = 0$ gilt. Wegen (2.6) gilt weiter, dass

$$A^{(k)} = \left(\begin{array}{cccc|c} a_{11}^{(k)} & \dots & & a_{1k-1}^{(k)} & \\ 0 & a_{22}^{(k)} & \dots & a_{2k-1}^{(k)} & \\ \vdots & 0 & \ddots & \vdots & * \\ 0 & \dots & & a_{k-1k-1}^{(k)} & \\ \hline & & 0 & & C \end{array}\right),$$

also folgt

$$|\det(A^{(k)})| = a_{11}^{(k)} \cdot a_{22}^{(k)} \cdot \ldots \cdot a_{k-1k-1}^{(k)} \cdot \det(C) = 0.$$

Wegen (2.7) folgt daraus $\det(A) = 0$, ein Widerspruch zur Regularität von A.

Jetzt zeigen wir (2.6) und (2.7) per Induktion.
Für beide Aussagen ist der Induktionsanfang $k = 1$ klar. Für den Übergang $k \to k+1$ nehmen wir an, dass die Aussagen für k schon gelten. Wegen dem ersten Teil des Beweises gilt $a_{kk}^{(k)} \neq 0$, also ist die Matrix $A^{(k+1)}$ definiert.
(2.6) gilt dann für $\tilde{A}^{(k+1)}$ wegen der Induktionsannahme für $A^{(k)}$ und wegen der alten Aussage (2.1) auf Seite 26. Für $A^{(k+1)}$ nutzen wir die Aussage für $\tilde{A}^{(k+1)}$ zusammen mit dem Argument, dass die Transformation $P^{(k+1)}$ die ersten k Zeilen von $\tilde{A}^{(k+1)}$ unberührt lässt. (2.7) ergibt sich schließlich durch Einsetzen

$$A^{(k+1)} = P^{(k+1)}\tilde{A}^{(k+1)} = P^{(k+1)}M_kA^{(k)}$$

und der Induktionsannahme.

Damit wissen wir, dass das Verfahren durchführbar ist und

$$U = \tilde{A}^{(n)} = M_{n-1}P^{(n-1)} \cdot \ldots \cdot P^{(2)}M_1P^{(1)}A$$

eine obere Dreiecksmatrix ist. Weil $M_j^{-1} = I + l^{(j)}e_j^T$ (Lemma 2.12) und $(P^{(k)})^{-1} = P^{(k)}$ (Lemma 2.16) erhält man daraus

$$A = P^{(1)}(I + l^{(1)}e_1^T)P^{(2)}(I + l^{(2)}e_2^T)P^{(3)} \cdot \ldots \cdot P^{(n-1)}(I + l^{(n-1)}e_{n-1}^T)U. \qquad (2.8)$$

Wir möchten nun beide Seiten der Gleichung von links mit

$$P = P^{(n-1)} \cdot \ldots \cdot P^{(2)}P^{(1)}$$

multiplizieren. Um den entstehenden Term zu vereinfachen, überlegen wir uns zunächst, dass für beliebige Vektoren l und alle $j > i$

$$P^{(j)}(I + le_i^T)P^{(j)} = (I + P^{(j)}l(P^{(j)}e_i)^T) = (I + P^{(j)}le_i^T)$$

gilt, weil $P^{(j)}e_i = e_i$, falls $j > i$.

Diese Aussage nutzen wir, um bei der Multiplikation von (2.8) mit $P = P^{(n)} \cdot \ldots \cdot P^{(2)} P^{(1)}$ die Permutationsmatrizen $P^{(i)}$ für $i \geq 3$ durch das Einfügen von Identitäten $I = P^{(i)} P^{(i)}$ bis zum entsprechenden Faktor $(I + l^{(i-1)} e_{i-1}^T) P^{(i)}$ "durchrutschen" zu lassen:

$$
\begin{array}{rcl}
P^{(1)}A & = & (I + l^{(1)} e_1^T) P^{(2)} (I + l^{(2)} e_2^T) P^{(3)} \cdot \ldots \quad \cdot P^{(n-1)} (I + l^{(n-1)} e_{n-1}^T) U \\
P^{(2)} P^{(1)} A & = & P^{(2)} (I + l^{(1)} e_1^T) P^{(2)} (I + l^{(2)} e_2^T) P^{(3)} \cdot \ldots \quad \cdot P^{(n-1)} (I + l^{(n-1)} e_{n-1}^T) U \\
 & = & (I + P^{(2)} l^{(1)} e_1^T)(I + l^{(2)} e_2^T) P^{(3)} \cdot \ldots \quad \cdot P^{(n-1)} (I + l^{(n-1)} e_{n-1}^T) U \\
P^{(3)} P^{(2)} P^{(1)} A & = & \underbrace{P^{(3)} (I + P^{(2)} l^{(1)} e_1^T) P^{(3)}}_{I + P^{(3)} P^{(2)} l^{(1)} e_1^T} \underbrace{P^{(3)} (I + l^{(2)} e_2^T) P^{(3)}}_{I + P^{(3)} l^{(2)} e_2^T} (I + l^{(3)} e_3^T) \cdot \ldots \\
 & & \qquad \ldots \quad \cdot P^{(n-1)} (I + l^{(n-1)} e_{n-1}^T) U \\
 & = & (I + P^{(3)} P^{(2)} l^{(1)} e_1^T)(I + P^{(3)} l^{(2)} e_2^T) \cdot \ldots \quad \cdot P^{(n-1)} (I + l^{(n-1)} e_{n-1}^T) U \\
 & \vdots & \qquad\qquad\qquad\qquad \vdots \\
\Longrightarrow \; PA & = & (I + \Theta^{(1)} e_1^T)(I + \Theta^{(2)} e_2^T) \cdot \ldots \cdot (I + \Theta^{(n-1)} e_{n-1}^T) U \\
 & = & I + \sum_{k=1}^{n-1} \Theta^{(k)} e_k^T,
\end{array}
$$

wobei der letzte Schritt per Induktion analog zu dem entsprechenden Schritt im Beweis von Lemma 2.14 (auf S. 29) gezeigt wird. QED

Für die praktische Implementierung empfiehlt es sich, auf Matrixoperationen zu verzichten, da diese aufwändig sind. Die folgende Variante ist effizienter. Wir geben sie gleich in einer Form an, die man verwenden würde, um ein Gleichungssystem $Ax = b$ zu lösen.

Algorithmus 3: Gauß-Verfahren mit Spaltenpivotsuche

Input: $A \in \mathbb{K}^{n,n}$, $b \in \mathbb{K}^n$.

Schritt 1: `For` $k = 1$ `to` $n - 1$ `do`

> **Schritt 1.1:** `Finde Pivotelement` a_{kr} `für Zeile` k
>
> **Schritt 1.2:** `Vertausche Zeilen` k `und` r `in` A `sowie` b_k `und` b_r.
>
> **Schritt 1.3:** `For` $i = k + 1$ `to` n `do`
>
> > **Schritt 1.3.1.** $a_{ik} = \frac{a_{ik}}{a_{kk}}$
> >
> > **Schritt 1.3.2.** `For` $j = k + 1$ `to` n `do` $a_{ij} = a_{ij} - a_{ik} \cdot a_{kj}$

Ergebnis: `Gleichungssystem` $LU = Pb$, `wobei` L `und` U `gegeben sind durch`

$$
l_{ij} = \begin{cases} a_{ij} & \text{für } i > j \\ 1 & \text{für } i = j \\ 0 & \text{für } i < j \end{cases} \qquad u_{ij} = \begin{cases} a_{ij} & \text{für } i \leq j \\ 0 & \text{für } i > j \end{cases}
$$

Das entstandene Gleichungssystem $LU = Pb$ kann man nun leicht durch Rückwärts - und Vorwärtselemination lösen. Man kann auch direkt während des Verfahrens alle elementaren Zeilenoperationen auf die rechte Seite Pb anwenden, und erhält dann als Ergebnis das Dreieckssystem $U = L^{-1}Pb$, so dass man sich den Schritt der Vorwärtselimination spart.

Aufwand der LU-Zerlegung nach Algorithmus 3:
Wir zählen hier noch einmal gründlich. Die äußere for-Schleife wird für jedes $k = 1, \ldots n-1$ durchlaufen. Darin werden folgende Operationen durchgeführt:

- Maximumsuche bei der Bestimmung des Pivotindexes: $n-1$ Vergleiche
- Vertauschungen sind Zuweisungen, die wir nicht mit zählen
- Innere for-Schleifen: $n-k$ Divisionen, $(n-k)(n-k)$ Multiplikationen, $(n-k)(n-k)$ Additionen

Zusammen beträgt die Anzahl der benötigten Operationen also

$$(n-1)(n-1) + \sum_{k=1}^{n-1}(n-k) + 2(n-k)^2 = \mathrm{O}(n^3).$$

Aufgabe: *Rechnen Sie die Anzahl der Operationen exakt (also ohne Abschätzung durch* O*) aus. Bestimmen Sie außerdem die Anzahl der in der Matrixversion (Algorithmus 2) benötigten Operationen exakt und durch* O*. Vergleichen Sie!*

Zwei einfache Anwendungen

Hat man eine LU-Zerlegung gefunden, so kann man diese für die folgenden beiden Anwendungen nutzen:

Anwendung 1: Bestimmung der Inversen A^{-1} einer Matrix A.
Sei A^{-1} gegeben durch ihre Spalten $A^{-1} = (B_1, B_2, \ldots, B_n)$. Dann gilt

$$AB_k = e_k,$$

und B_k ergibt sich als Lösung x von $Ax = e_k$. Kennt man die LU-Zerlegung der Matrix A, so bestimmt man also für $k = 1, \ldots, n$ zunächst die Lösung y_k des Gleichungssystems $Ly_k = e_k$ und löst anschließend das Gleichungssystem $Ux_k = y_k$ zur Bestimmung von $B_k := x_k$.

Anwendung 2: Bestimmung der Determinante von A.
Ist $A = LU$ eine LU-Zerlegung von A, so gilt $\det(A) = \det(L) \cdot \det(U) = u_{11} \cdot \ldots \cdot u_{nn}$. Die Determinante lässt sich also als Produkt der Diagonalelemente von U direkt berechnen.

2.3 Das Cholesky-Verfahren

Wir betrachten auch in diesem Abschnitt Gleichungssysteme $Ax = b$, allerdings nehmen wir nun an, dass die Matrix A eine symmetrische und positiv definite Matrix ist.

Definition 2.18 *Eine Matrix $A \in \mathbb{R}^{n,n}$ heißt* **hermitesch** *falls $A^* = A$, wobei $A^* = (\overline{a}_{ji})$ die konjugiert komplexe Matrix zu A ist. Ist $\mathbb{K} = \mathbb{R}$ so nennt man A auch symmetrisch. Eine hermitesche Matrix heißt*

- **positiv definit** *falls $x^T Ax > 0$ für alle $x \in \mathbb{R}^n \setminus \{0\}$,*
- **positiv semi-definit** *falls $x^T Ax \geq 0$ für alle $x \in \mathbb{R}^n$.*

Lemma 2.19 *Die folgenden Aussagen gelten:*

1. *Eine symmetrische Matrix ist genau dann positiv definit, wenn alle ihre Eigenwerte echt positiv sind.*
2. *Eine symmetrische Matrix ist genau dann positiv semi-definit, wenn alle ihre Eigenwerte größer oder gleich Null sind.*
3. *Eine symmetrische Matrix ist genau dann positiv definit, wenn ihre Hauptminoren positiv sind, d.h. wenn für alle ihre linken oberen $k \times k$-Teilmatrizen*

$$A^{[k]} := \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix}, \quad k = 1, \dots, n$$

gilt: $\det(A^{[k]}) > 0$.

Positiv definite Matrizen sind also regulär (weil $\det(A^{[n]}) = \det(A) \neq 0$). Wir betrachten die folgenden beiden Zerlegungen:

Definition 2.20 *Eine Faktorisierung einer symmetrischen Matrix $A \in \mathbb{R}^{n,n}$ der Form $A = LL^T$ mit einer (regulären) unteren Dreiecksmatrix L heißt* **Cholesky-Zerlegung** *von A .*

Definition 2.21 *Eine Faktorisierung einer symmetrischen Matrix $A \in \mathbb{R}^{n,n}$ der Form $A = LDL^T$ mit einer normierten unteren Dreiecksmatrix L und einer Diagonalmatrix D heißt* **LDL-Zerlegung** *von A .*

Im folgenden werden wir uns u.a. mit Diagonalmatrizen beschäftigen, die wir wie folgt bezeichnen.

Notation 2.22 *Für einen Vektor $a \in \mathbb{K}^n$ ist die* **Diagonalmatrix** *bezüglich a gegeben durch*

$$diag(a) = \begin{pmatrix} a_1 & & & & \\ & a_2 & & & \\ & & \ddots & & \\ & & & a_{n-1} & \\ & & & & a_n \end{pmatrix}.$$

Für positive definite symmetrische Matrizen sind die folgenden Aussagen bekannt.

Satz 2.23 *Sei $A \in \mathbb{R}^{n,n}$ eine positiv definite symmetrische Matrix. Dann existiert eine eindeutig bestimmte LDL-Zerlegung von A.*

Beweis: Zunächst bestätigen wir, dass A eine eindeutige LU-Zerlegung mit normierter unterer Dreiecksmatrix L hat: Nach Satz 2.13 kann man eine LU-Zerlegung (ohne Pivotisierung) finden, wenn die Teilmatrizen

$$A^{[k]} := \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix}$$

die Bedingung (2.2) erfüllen, d.h. wenn $\det(A^{[k]}) \neq 0$, $k = 1, \dots, n-1$. Weil A positiv definit ist, gilt das nach Lemma 2.19 und zusätzlich sogar $\det(A^{[n]}) > 0$, also sind A, L und U regulär. Entsprechend ist L eine untere normierte Dreiecksmatrix und die Zerlegung ist eindeutig nach Lemma 2.10.

Sei daher

$$A = LU \tag{2.9}$$

mit normierter unterer Dreiecksmatrix L und oberer Dreiecksmatrix U. Wir setzen $D = diag(u_{11}, \dots, u_{nn})$ als die Diagonalmatrix mit den Einträgen aus der Hauptdiagonalen von U. Da U regulär ist, ist auch D regulär, so dass wir

$$\tilde{U} := D^{-1}U$$

definieren können. Es gilt $LD\tilde{U} = LU = A$. Wir möchten zeigen, dass $\tilde{U} = L^T$: Betrachte dazu

$$A = A^T = (LD\tilde{U})^T = \tilde{U}^T D^T L^T = \tilde{U}^T \cdot (D^T L^T). \tag{2.10}$$

$\tilde{U}$ ist nach Konstruktion eine normierte obere Dreiecksmatrix, also ist $\tilde{U}^T$ eine normierte untere Dreiecksmatrix. Weiter ist $D^T L^T$ eine obere Dreiecksmatrix, also ist (2.10) auch eine LU-Zerlegung von A mit normierter unterer Dreiecksmatrix. Wegen Lemma 2.10 ist die LU-Zerlegung von A eindeutig, also folgern wir aus dem Vergleich von (2.9) und (2.10) dass

$$L = \tilde{U}^T$$

und haben damit die LDL-Zerlegung von A gefunden.

Sei nun $A = L'D'(L')^T$ eine weitere LDL-Zerlegung von A mit normierter unterer Dreiecksmatrix L, so kann man wiederum

$$A = L' \cdot (D'(L')^T)$$

als LU-Zerlegung auffassen. Da die LU-Zerlegung nach Lemma 2.10 eindeutig ist, folgt

$$L' = L \quad \text{und} \quad D'(L')^T = DL^T,$$

wobei sich aus letzterem wegen der Invertierbarkeit von $L = L'$ auch $D' = D$ ergibt. QED

Der Beweis des Satzes zeigt außerdem, dass die LU-Zerlegung einer symmetrischen und positiv definiten Matrix A ohne Pivotisierung gefunden werden kann. Wir kommen nun auf die Cholesky-Zerlegung zurück.

Satz 2.24 *Sei $A \in \mathbb{R}^{n,n}$ eine positiv definite symmetrische Matrix. Dann existiert eine Cholesky-Zerlegung von $A = LL^T$ mit positiven Diagonalelementen von L. Unter dieser Nebenbedingung ist L eindeutig bestimmt.*

Beweis: Nach Satz 2.23 gibt es eine eindeutige LDL-Zerlegung

$$A = LDL^T$$

von A. Bezeichnen wir mit $A^{[k]}$ und $D^{[k]}$ wieder die linken oberen $k \times k$ Teilmatrizen von A und D. Weil L eine untere Dreiecksmatrix ist, gilt $A^{[k]} = L^{[k]}D^{[k]}(L^T)^{[k]}$, also

$$det(A^{[k]}) = det(D^{[k]}). \tag{2.11}$$

Wegen der positiven Definitheit von A (siehe Lemma 2.19) gilt $\det(A^{[k]}) > 0$. Zusammen mit (2.11) erhalten wir

$$d_{11} \cdot \ldots \cdot d_{kk} = \det(D^{[k]}) = \det(A^{[k]}) > 0.$$

Diese Aussage gilt für alle k, also sind die Diagonalelemente von D positiv. Jetzt setzen wir

$$\tilde{L} = L \cdot diag(\sqrt{d_{11}}, \ldots, \sqrt{d_{nn}}) \tag{2.12}$$

und erhalten aus der normierten unteren Dreiecksmatrix L eine untere Dreiecksmatrix $\tilde{L}$ mit positiven Diagonalelementen, für die

$$\tilde{L}\tilde{L^T} = L \cdot diag(\sqrt{d_{11}}, \ldots, \sqrt{d_{nn}}) \cdot diag(\sqrt{d_{11}}, \ldots, \sqrt{d_{nn}}) \cdot L^T = LDL^T = A$$

gilt. Die entsprechende Cholesky-Zerlegung ist also gefunden.

Um die Eindeutigkeit zu zeigen, sei neben $A = \tilde{L}\tilde{L}^T$

$$A = \tilde{L}'(\tilde{L}')^T$$

eine weitere Cholesky-Zerlegung mit Diagonalelementen $\lambda_1, \lambda_2, \ldots, \lambda_n > 0$. Mit $D' := diag(\lambda_1^2, \ldots, \lambda_n^2)$ und

$$L' := \tilde{L}' \cdot diag\left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_n}\right)$$

erhält man $A = L'D'(L')^T$, also eine weitere LDL-Zerlegung von A mit normierter unterer Dreiecksmatrix L'. Aus der Eindeutigkeit der LDL-Zerlegung (nach Satz 2.23) folgt $L = L'$ und $D = D'$. Letzteres bedeutet $d_{ii} = \lambda_i^2$ für $i = 1, \ldots, n$ und wegen der Positivität der λ_i und d_{ii} also

$$\lambda_i = \sqrt{d_{ii}} \text{ für } i = 1, \ldots, n.$$

Zusammen erhält man

$$\begin{aligned} \tilde{L}' &= L' \cdot diag(\lambda_1, \ldots, \lambda_n) \\ &= L \cdot diag(\sqrt{d_{11}}, \ldots, \sqrt{d_{nn}}) = \tilde{L} \text{ nach (2.12).} \end{aligned}$$

QED

Um eine Cholesky-Zerlegung effizient ausrechnen zu können, betrachten wir die Gleichung $A = LL^T$ komponentenweise. Das ergibt ein Gleichungssystem mit Unbekannten l_{ij} für $i \geq j$. Bezeichnen wir dazu im folgenden mit l_{ij}^T die Elemente der Matrix L^T. Dann ergibt sich

$$a_{ik} = \sum_{j=1}^{n} l_{ij} l_{jk}^T = \sum_{j=1}^{n} l_{ij} l_{kj} = \sum_{j=1}^{k} l_{ij} l_{kj} \text{ für } k = 1, \ldots, n,\ i = k+1, \ldots, n \quad (2.13)$$

und

$$a_{kk} = \sum_{j=1}^{n} l_{kj} l_{jk}^T = \sum_{j=1}^{n} l_{kj} l_{kj} = \sum_{j=1}^{k} l_{kj}^2 \text{ für } k = 1, \ldots, n. \quad (2.14)$$

Wählt man die Reihenfolge geschickt aus, lassen sich die Werte l_{ij} effizient berechnen: Zunächst ergibt sich l_{11} aus (2.14) für $k = 1$ zu $l_{11} = \sqrt{a_{11}}$. Danach lassen sich nacheinander die Werte $l_{21}, \ldots, l_{n1}$ der ersten Spalte von L durch (2.13) bestimmen, dann das Diagonalelement der zweiten Spalte durch (2.14) und so weiter. Es ergibt sich das folgende Verfahren, in dem wir nur das untere Dreieck der Matrix A benutzen und die Elemente von L gleich über die Werte von A schreiben.

Algorithmus 4: Cholesky-Verfahren

Input: $A \in \mathbb{R}^{n,n}$ `symmetrisch und positiv definit`
`gegeben durch Werte` a_{ij} `für` $i \geq j$.

Schritt 1: `For` $k = 1$ `to` n `do`

Schritt 1.1: $a_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} |a_{kj}|^2}$

Schritt 1.2: `For` $i = k+1$ `to` n `do`

$$a_{ik} = \frac{1}{a_{kk}} \left(a_{ik} - \sum_{j=1}^{k-1} a_{ij} a_{kj} \right)$$

Ergebnis: L `ist gegeben durch` $l_{ij} = \begin{cases} a_{ij} & \text{für } i \geq j \\ 0 & \text{für } i < j \end{cases}$

2.4 Schwachbesetzte Matrizen

Die bisher beschriebenen Verfahren sind bei sehr großen Matrizen leider ineffizient. Daher versucht man, die LU-Zerlegung an Matrizen mit spezieller Struktur anzupassen. Einen ersten Ansatz haben wir im letzten Abschnitt bei symmetrischen Matrizen kennengelernt. In Anwendungen treten oft *schwachbesetzte* Matrizen auf, in denen für die meisten Elemente $a_{ij} = 0$ gilt. Leider sind die bei der LU-Zerlegung von schwachbesetzten Matrizen entstehenden Dreiecksmatrizen L und U im allgemeinen nicht auch wieder schwach besetzt. Als Beispiel sei die Matrix

$$A = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 1 & 0 & 0 & 0 \\ 0.1 & 0 & 1 & 0 & 0 \\ 0.1 & 0 & 0 & 1 & 0 \\ 0.1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

aus dem Skriptum von G. Lube genannt, bei der die Dreiecksmatrizen ihrer LU-Zerlegung voll besetzt sind. Es gibt aber eine Klasse von Matrizen, bei der sich die Struktur der Matrix A auf die Struktur der Matrizen L und U ihrer LU-Zerlegung überträgt. Dazu gehören sogenannte *Bandmatrizen.*

Definition 2.25 *Eine Matrix* $A = (a_{ij}) \in \mathbb{K}^{n,n}$ *ist eine* (p,q)-**Bandmatrix**, *falls für alle* $i > j + p$ *und für alle* $j > i + q$ *gilt:* $a_{ij} = 0$. *Die* **Bandbreite** *von* A *ist dann* $p + q + 1$.

Die folgende Matrix ist ein Beispiel für eine $(2,1)$-Bandmatrix:

$$A = \begin{pmatrix} 3 & 2 & 0 & 0 & 0 \\ 4 & 1 & 1 & 0 & 0 \\ 1 & 3 & 1 & 5 & 0 \\ 0 & 4 & 3 & 1 & 2 \\ 0 & 0 & 4 & 0 & 1 \end{pmatrix}$$

Jede untere Dreiecksmatrix ist eine $(n-1,0)$-Bandmatrix, jede obere Dreiecksmatrix ist eine $(0,n-1)$-Bandmatrix.

Satz 2.26 *Sei $A = LU$ die LU-Zerlegung einer (p,q)-Bandmatrix A mit oberer Dreiecksmatrix U und normierter unterer Dreiecksmatrix L. Dann ist L eine $(p,0)$-Bandmatrix und U eine $(0,q)$-Bandmatrix.*

Beweis: Wir beweisen den Satz für feste p, q mittels vollständiger Induktion nach n. Für $n = 1$ ist nichts zu zeigen. Für den Induktionsschritt $n \to n+1$ nehmen wir also an, dass die Aussage für Matrizen der Dimension $n \times n$ richtig ist. Betrachte nun eine Matrix $A \in \mathbb{K}^{n+1,n+1}$ mit LU-Zerlegung $A = LU$. Wir partitionieren A wie folgt

$$A = \begin{pmatrix} \alpha & w^T \\ v & B \end{pmatrix},$$

wobei $\alpha \in \mathbb{K}$ und B eine (p,q)-Bandmatrix der Dimension $n \times n$ ist und die Vektoren $v, w \in \mathbb{K}^n$ erfüllen, dass $v_i = 0$ für alle $i > p$ und $w_j = 0$ für alle $j > q$. Sei weiterhin

$$L = \begin{pmatrix} 1 & 0 \\ l & L_1 \end{pmatrix}, \quad U = \begin{pmatrix} u_{11} & u^T \\ 0 & U_1 \end{pmatrix}.$$

Es gilt

$$LU = \begin{pmatrix} u_{11} & u^T \\ v_{11} l & l u^T + L_1 U_1 \end{pmatrix} = \begin{pmatrix} \alpha & w^T \\ v & B \end{pmatrix},$$

also ist $\alpha = u_{11}$, $w = u$, $l = \frac{1}{\alpha} v$. Betrachte

$$B - \frac{1}{\alpha} v w^T.$$

Aufgrund der Struktur von v und w ist B ebenfalls eine (p,q)-Bandmatrix mit LU-Zerlegung

$$B - \frac{1}{\alpha} v w^T = L_1 U_1.$$

Nach der Induktionsannahme ist also die untere Dreiecksmatrix L_1 eine normierte $(p,0)$-Bandmatrix und die obere Dreiecksmatrix U_1 eine $(0,q)$-Bandmatrix und es gilt

$$\begin{pmatrix} 1 & 0 \\ \frac{1}{\alpha} v & L_1 \end{pmatrix} \cdot \begin{pmatrix} \alpha & w^T \\ 0 & U_1 \end{pmatrix} = \begin{pmatrix} \alpha & w^T \\ v & B \end{pmatrix} = A,$$

eine LU-Zerlegung von A mit der geforderten Eigenschaft ist also gefunden.

QED

Mit folgendem Algorithmus kann man die LU-Zerlegung einer (p,q)-Bandmatrix bestimmen (falls sie existiert).

Algorithmus 5: LU-Zerlegung einer Bandmatrix

Input: (p,q)-Bandmatrix $A \in \mathbb{K}^{n,n}$, für die eine LU-Zerlegung existiert.

Schritt 1: For $k = 1$ to $n-1$, for $i = k+1$ to $\min\{k+p, n\}$ do

Schritt 1.1: $a_{ik} := \frac{a_{ik}}{a_{kk}}$

Schritt 1.2: For $j = k+1$ to $\min\{k+q, n\}$ do $a_{ij} := a_{ij} - a_{ik}a_{kj}$

Ergebnis: LU-Zerlegung von A wobei L und U gegeben sind durch

$$l_{ij} = \begin{cases} a_{ij} & \text{für } i > j \\ 1 & \text{für } i = j \\ 0 & \text{für } i < j \end{cases} \qquad u_{ij} = \begin{cases} a_{ij} & \text{für } i \leq j \\ 0 & \text{für } i > j \end{cases}$$

Natürlich sind auch Vorwärts- und Rückwärtselimination für Bandmatrizen einfacher. Abschließend betrachten wir noch den Spezialfall von Tridiagonalmatrizen. Dazu führen wir die folgende Notation ein.

Notation 2.27 *Für drei Vektoren $a, b, c \in \mathbb{K}^n$ mit $b_1 = c_n = 0$ ist die* **Tridiagonalmatrix** *bezüglich a, b, c gegeben durch*

$$\operatorname{tridiag}(b,a,c) = \begin{pmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \ddots & \ddots & & \\ & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & b_n & a_n \end{pmatrix}$$

Nach Satz 2.26 wissen wir, dass (falls sie existieren) die Matrizen L und U der LU-Zerlegung das folgende Aussehen haben

$$L = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & l_{n-1} & 1 & \\ & & & l_n & 1 \end{pmatrix} \quad U = \begin{pmatrix} u_1 & c_1 & & & \\ & u_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & u_{n-1} & c_{n-1} \\ & & & & u_n \end{pmatrix} \tag{2.15}$$

wobei durch einen ersten Koeffizientenvergleich schon ausgenutzt wurde, dass die Werte $c_1, \ldots, c_{n-1}$ der oberen Nebendiagonale von A in der oberen Nebendiagonalen von U erhalten bleiben. Es sind also die Unbekannten $u_1, \ldots, u_n$ und $l_1, \ldots, l_n$ zu bestimmen. Durch Multiplikation der Matrizen L und U und erneutem Koeffizientenvergleich mit A ergeben sich die folgenden Berechnungsvorschriften:

$$\begin{aligned} \text{Start:} \quad u_1 &:= a_1 \\ \text{Für } i = 2, \ldots, n: \quad l_i &:= \frac{b_i}{u_{i-1}} \\ u_i &:= a_i - l_i c_{i-1}. \end{aligned}$$

Man kommt also mit einer in n linearen Anzahl an Operationen aus. Bei n Unbekannten ist das das beste, was man erreichen kann. Allerdings lässt sich nicht für jede Tridiagonalmatrix eine LU-Zerlegung finden. Das folgende Lemma gibt eine hinreichende Bedingung für die Durchführbarkeit der LU-Zerlegung für Tridiagonalmatrizen.

Lemma 2.28 *Für $A = \text{tridiag}(b, a, c)$ mit $b_1 = c_n = 0$ sei für $j = 1, \ldots, n$ $|c_j| < |a_j|$ und $|b_j| + |c_j| \leq |a_j|$. Dann gibt es eine LU-Zerlegung von A mit Matrizen wie in (2.15).*

Aufgabe: *Beweisen Sie das Lemma durch Induktion!*

Kapitel 3

Störungsrechnung

3.1 Metrische und normierte Räume

Bevor wir uns mit der Fehleranalyse bei linearen Gleichungssystemen beschäftigen können, benötigen wir einige Begriffe aus der Funktionalanalysis. Dazu gehört insbesondere, dass wir messen können, um wieviel sich ein Vektor x von einem gestörten Vektor $\tilde{x}$ unterscheidet. Den Unterschied

$$\tilde{x} - x$$

als Vektor anzugeben, hilft uns nicht weiter, da wir zwei verschiedene gestörte Vektoren $\tilde{x}$ und x' mangels einer Ordnung im $\mathbb{K}^n$ nicht vergleichen können. Wir suchen also eine Funktion, die die Differenz zwischen zwei Vektoren durch eine reelle, positive Zahl ausdrückt. Solche Funktionen nennt man auch *Distanzfunktionen*. Mit beliebigen Distanzfunktionen geben wir uns aber nicht zufrieden, sondern betrachten *Metriken* als spezielle Distanzfunktionen.

Definition 3.1 *Sei $\mathcal{R}$ eine nichtleere Menge. Eine Abbildung $d : \mathcal{R} \times \mathcal{R} \to \mathbb{R}$ heißt* **Metrik** *auf $\mathcal{R}$ falls sie die folgenden Bedingungen erfüllt:*

(M1) $d(x, y) = 0 \Longleftrightarrow x = y$ *für alle* $x, y \in \mathcal{R}$

(M2) $d(x, y) = d(y, x)$ *für alle* $x, y \in \mathcal{R}$ *(Symmetrie)*

(M3) $d(x, y) \leq d(x, z) + d(z, y)$ *für alle* $x, y, z \in \mathcal{R}$ *(Dreiecksungleichung)*

$(\mathcal{R}, d)$ *heißt dann* **metrischer Raum**.

Man beachte, dass aus den Metrik-Eigenschaften sofort folgt, dass

$$d(x, y) \geq 0 \text{ für alle } x, y \in \mathcal{R},$$

denn

$$d(x,y) = \frac{1}{2}(d(x,y) + d(x,y)) = \frac{1}{2}(d(x,y) + d(y,x)) \geq \frac{1}{2}d(x,x) = 0.$$

Eine Metrik ist z.B. der sogenannte *Hamming-Abstand* d_H, der für $x, y \in \mathbb{K}^n$ gegeben ist durch

$$d_H(x,y) = \#\{i = \{1,\ldots,n\} : x_i \neq y_i\}.$$

Wir wiederholen zunächst einige Begriffe, die auf jedem metrischen Raum definiert sind.

Definition 3.2 *Sei $(\mathcal{R}, d)$ ein metrischer Raum.*

- *Eine Folge $(x_n) \subseteq \mathcal{R}$* **konvergiert bezüglich der Metrik** *d, falls es ein Element $\bar{x} \in \mathcal{R}$ gibt, das folgendes erfüllt: Zu jedem $\epsilon > 0$ existiert eine natürliche Zahl $N(\epsilon)$, so dass*

  $$d(\bar{x}, x_n) < \epsilon \text{ für alle } n \geq N(\epsilon).$$

  In diesem Fall nennt man $\bar{x}$ den **Grenzwert** *der Folge (x_n). Eine nichtkonvergente Folge heißt* **divergent**.

- *Eine Folge $(x_n) \subseteq \mathcal{R}$ heißt* **Cauchy-Folge** *falls es zu jedem $\epsilon > 0$ eine natürliche Zahl $N(\epsilon)$ gibt, so dass*

  $$d(x_n, x_m) < \epsilon \text{ für alle } n, m \geq N(\epsilon).$$

- *Ein metrischer Raum $(\mathcal{R}, d)$ heißt vollständig, falls jede Cauchy-Folge konvergiert. Einen vollständigen normierten Raum nennt man auch* **Banachraum**.

Lemma 3.3

- *Sei (x_n) eine konvergente Folge. Dann ist ihr Grenzwert eindeutig bestimmt.*
- *Jede konvergente Folge ist eine Cauchy-Folge.*
- *Es gibt metrische Räume, in denen nicht jede Cauchy-Folge konvergiert.*

Übung: *Beweisen Sie Lemma 3.3!*

Auf metrischen Räumen lassen sich weitere Strukturen erarbeiten. So reichen die Begriffe *Folge* und *Konvergenz einer Folge* insbesondere aus, um offene und abgeschlossene Mengen zu definieren. Das bedeutet, dass jeder metrische Raum auch ein topologischer Raum ist.

Die wichtigsten Beispiele für metrische Räume sind *normierte Räume*, für die wir allerdings als Grundmenge einen Vektorraum V voraussetzen.

Definition 3.4 *Sei V ein Vektorraum über einem Körper $\mathbb{K}$. Eine Abbildung $\|\cdot\| : V \to \mathbb{R}_0^+$ heißt* **Norm** *auf V falls sie die folgenden drei Bedingungen erfüllt:*

(N1) $\|x\| = 0 \Longleftrightarrow x = 0$ *für alle* $x \in V$.

(N2) $\|\alpha x\| = |\alpha|\|x\|$ *für alle* $\alpha \in \mathbb{K}, x \in V$. *(Skalierbarkeit)*

(N3) $\|x + y\| \leq \|x\| + \|y\|$ *für alle* $x, y \in V$. *(Dreiecksungleichung)*

Der Raum $(V, \|\cdot\|)$ heißt dann **normierter Raum**. *Weiterhin nennt man die Menge*

$$B_{\|\cdot\|} = \{x \in V : \|x\| \leq 1\}$$

den **Einheitskreis** *der Norm $\|\cdot\|$.*

Bemerkung: Ersetzt man im Fall $\mathbb{K} = \mathbb{R}$ die Bedingung (N2) durch $\|\alpha x\| = \alpha\|x\|$ für alle $\alpha \in \mathbb{R}^+, x \in V$, so erhält man ein reelles **Minkowski-Funktional** oder einen **Gauge**.

Für Normen gilt (ähnlich wie im Fall von Metriken) dass

$$\|x\| \geq 0 \text{ für alle } x \in V,$$

denn aus (N2) folgt für $\alpha = -1$ insbesondere $\|(-1)x\| = \|x\|$, und daraus

$$\|x\| = \frac{1}{2}(\|x\| + \|x\|) = \frac{1}{2}(\|x\| + \|-x\|) \geq \frac{1}{2}(\|x + (-x)\|) = \frac{1}{2}(\|0\|) = 0.$$

Wichtige Normen auf dem $\mathbb{K}^n$ sind die folgenden:

$$\begin{aligned}
\text{Manhattan-Norm:} \quad & \|x\|_1 = \sum_{i=1}^{n} |x_i| \\
\text{Maximum-Norm:} \quad & \|x\|_\infty = \max_{i=1}^{n} |x_i| \\
\text{Euklidische Norm:} \quad & \|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{x^T x} \\
p\text{–Norm:} \quad & \|x\|_p = \sqrt[p]{\sum_{i=1}^{n} |x|_i^p}, \text{ für } 1 \leq p \leq \infty,
\end{aligned}$$

wobei die p–Norm die drei erstgenannten Normen als Spezialfälle ($p = 1$, $p = \infty$, $p = 2$) enthält.

Um einzusehen, dass es sich bei diesen Abbildungen tatsächlich um Normen handelt, sind die Bedingungen (N1),(N2) und (N3) zu zeigen. Dabei sind (N1) und (N2) direkt klar. (N3) kann man für die Fälle $p = 1$ und $p = \infty$ leicht nachrechnen; in beiden Fällen folgt die Bedingung aus der Dreiecksungleichung für Beträge. Für $p = 2$ ergibt sich (N2) aus der Cauchy-Schwarzschen Ungleichung, für beliebiges $p \in (1, \infty)$ aus der Minkowski-Ungleichung, die im folgenden beschrieben ist.

Lemma 3.5 *Sei $x, y \in \mathbb{K}^n$. Dann gilt*

$$\sum_{i=1}^{n} x_i y_i \leq \sum_{i=1}^{n} |x_i||y_i| \leq \|x\|_p \cdot \|y\|_q$$

falls entweder $1 < p, q < \infty$ und $\frac{1}{p} + \frac{1}{q} = 1$ oder falls $p = 1, q = \infty$ oder $p = \infty, q = 1$.

Bemerkung: Der Fall $p = q = 2$ führt ausgeschrieben zur *Cauchy-Schwarz'schen Ungleichung*

$$\sum_{i=1}^{n} x_i y_i \leq \sum_{i=1}^{n} |x_i||y_i| \leq \sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}.$$

Normen haben verschiedene wichtige Eigenschaften. Dazu zählt insbesondere, dass man aus jeder Norm durch

$$d(x, y) = \|y - x\|$$

eine Metrik d definieren kann. Man nennt diese Metrik dann auch die *von der Norm $\|\cdot\|$ abgeleitete Metrik*. Die Metrik-Eigenschaften lassen sich leicht durch die Norm-Eigenschaften beweisen. Weiterhin folgt für alle Normen die folgende Abschätzung.

Lemma 3.6 *Sei $\|\cdot\|$ eine Norm auf V. Dann gilt für alle $x, y \in V$*

$$| \, \|x\| - \|y\| \, | \leq \|x - y\|.$$

Beweis: Für alle $x, y \in V$ gilt $\|x\| = \|x - y + y\| \leq \|x - y\| + \|y\|$. Daraus folgt

$$\|x\| - \|y\| \leq \|x - y\|.$$

Aus Symmetriegründen erhält man analog

$$\|y\| - \|x\| \leq \|x - y\|,$$

zusammen ergibt sich die Behauptung. QED

Wir haben erwähnt, wie man mit Hilfe einer Norm eine Metrik und damit Konvergenz bezüglich einer Norm definieren kann. Die Frage ist nun, in wie weit sich diese Konvergenz-Definitionen für verschiedene Normen unterscheiden. Dazu ist die folgende Definition hilfreich.

Definition 3.7 *Zwei Normen $\|\cdot\|_a$ und $\|\cdot\|_b$ auf einem Vektorraum V heißen* **äquivalent**, *wenn es positive reelle Zahlen c, C gibt, so dass für alle $x \in V$ gilt:*

$$c\|x\|_a \leq \|x\|_b \leq C\|x\|_a$$

Es lässt sich leicht zeigen, dass die in der Definition genannte Äquivalenz tatsächlich eine Äquivalenzrelation ist. Weiterhin gilt der folgende Satz:

Satz 3.8 *Zwei Normen $\|\cdot\|_a$ und $\|\cdot\|_b$ auf einem Vektorraum V sind genau dann äquivalent, wenn jede bezüglich der Norm $\|\cdot\|_a$ konvergente Folge aus V auch bezüglich der Norm $\|\cdot\|_b$ konvergiert.*

Beweis:

- Nehmen wir zunächst an, dass die beiden Normen $\|\cdot\|_a$ und $\|\cdot\|_b$ äquivalent sind. Da eine Folge (x_n) genau dann gegen $\bar{x}$ konvergiert, wenn $x_n - \bar{x}$ eine Nullfolge ist, reicht es, die Aussage für Nullfolgen zu zeigen.

  Sei dazu also $x_n \to 0$ eine Nullfolge bezüglich $\|\cdot\|_a$, d.h. zu jedem $\epsilon > 0$ existiert eine natürliche Zahl $N(\epsilon)$ so dass $\|x_n\|_a \leq \epsilon$ für alle $n \geq N(\epsilon)$. Wegen

  $$\|x_n\|_b \leq C \cdot \|x_n\|_a \leq C\epsilon \text{ für alle } n \geq N(\epsilon)$$

  folgt für jedes ϵ', dass $\|x_n\|_b \leq \epsilon'$ für alle $n \geq N(\frac{\epsilon'}{C})$, also ist x_n auch bezüglich $\|\cdot\|_b$ eine Nullfolge.

  Die Umkehrung gilt analog.

- Gelte nun die Äquivalenz der Konvergenz-Definitionen. Durch Widerspruch zeigen wir zunächst, dass es eine Zahl $C > 0$ gibt mit

  $$\|x\|_b \leq C \text{ für alle } x \in V \text{ mit } \|x\|_a = 1. \tag{3.1}$$

  Angenommen also, eine solche Zahl C existiert nicht. Dann existiert zu jedem $C = C(n) := n^2$ ein x_n mit $\|x_n\|_a = 1$ und $\|x_n\|_b > n^2$. Die Folge

  $$y_n := \frac{x_n}{n}$$

  erfüllt also

  $$\|y_n\|_a = \frac{1}{n} \text{ und } \|y_n\|_b > n.$$

  Das heißt, (y_n) konvergiert gegen Null bezüglich $\|\cdot\|_a$, aber divergiert bezüglich $\|\cdot\|_b$, ein Widerspruch.

  Somit gibt es ein $C > 0$, das (3.1) erfüllt. Damit ergibt sich für alle $x \in V$:

  $$\|x\|_b = \left\| \|x\|_a \frac{x}{\|x\|_a} \right\|_b = \|x\|_a \left\| \frac{x}{\|x\|_a} \right\|_b \leq C\|x\|_a,$$

  die erste Ungleichung für die Normäquivalenz ist also erfüllt. Die zweite Ungleichung ergibt sich durch Vertauschen der Normen. QED

Die obige Aussage gilt für alle Vektorräume V. Wir diskutieren nun den Fall eines endlich-dimensionalen Raums.

Satz 3.9 *Sei V ein endlich-dimensionaler Vektorraum. Dann sind alle Normen über V äquivalent.*

Beweis: Sei $v_1, \dots v_n$ eine Basis von V. Jedes Element $x \in V$ lässt sich also darstellen durch

$$x = \sum_{k=1}^{n} \alpha_k v_k.$$

Wir konstruieren nun eine Norm (die *Maximum-Norm auf V*) und zeigen anschließend, dass jede weitere Norm auf V zu dieser Norm äquivalent ist. Wegen der Transitivität der Normäquivalenz folgt daraus die Behauptung des Satzes.

Man rechnet schnell nach, dass

$$\|x\|_\infty := \max_{k=1,\dots,n} |\alpha_k|$$

eine Norm auf V definiert. Sei nun also $\|\cdot\|$ eine beliebige andere Norm auf V. Definiere

$$C := \sum_{k=1}^{n} \|v_k\|$$

als die Summe der Normen aller Basisvektoren. Dann folgt:

$$\begin{aligned}
\|x\| &= \left\| \sum_{k=1}^{n} \alpha_k v_k \right\| \\
&\leq \sum_{k=1}^{n} |\alpha_k| \|v_k\| \text{ wegen (N2) und (N3)} \\
&\leq \sum_{k=1}^{n} \|x\|_\infty \|v_k\| \text{ weil } |\alpha_k| \leq \|x\|_\infty \\
&= C \cdot \|x\|_\infty.
\end{aligned}$$

Für die andere Richtung definieren wir die gesuchte Konstante c durch

$$c := \inf\{\|x\| : x \in V \text{ und } \|x\|_\infty = 1\}.$$

Weil für alle $x \in V \setminus \{0\}$ gilt, dass

$$\left\| \frac{x}{\|x\|_\infty} \right\|_\infty = 1$$

folgt daraus, dass

$$\left\| \frac{x}{\|x\|_\infty} \right\| \geq c,$$

das heißt $\|x\| \geq c \cdot \|x\|_\infty$ für alle $x \neq 0$. Weil für $x = 0$ nichts zu zeigen ist, ergibt das also die Behauptung.

Allerdings bleibt noch zu zeigen, dass $c > 0$ gilt. Dazu führen wir einen Widerspruchsbeweis. Wir nehmen also an, dass $c = 0$. Dann gibt es eine Folge (y_m) mit $\|y_m\|_\infty = 1$ und $\|y_m\| \to 0$ für $m \to \infty$. Die Basisdarstellung in V liefert für jedes Folgenglied y_m

$$y_m = \sum_{k=1}^{n} \alpha_{km} v_k,$$

und damit n Folgen für die Koeffizienten $\alpha_{1m}, \alpha_{2m}, \ldots, \alpha_{nm}$ aus dem zugrunde liegende Körper. Weil $\|y_m\|_\infty = 1$ für alle m gelten die folgenden beiden Aussagen für die Koeffizienten der Folgen:

$$\text{Für alle } m : |\alpha_{km}| \leq 1 \text{ für alle } k = 1, \ldots, n. \tag{3.2}$$

$$\text{Für alle } m \text{ existiert ein } k \in \{1, \ldots, n\} \text{ so dass } |\alpha_{km}| = 1. \tag{3.3}$$

Wegen (3.3) erfüllt mindestens eine der Koeffizienten-Folgen $\bar{k}$, dass $\#\{m : |\alpha_{\bar{k}m}| = 1\} = \infty$, d.h. es kommen unendlich viele Einsen (oder unendlich viele $-$ Einsen) vor. Sei oBdA $\bar{k} = 1$, und $\#\{m : \alpha_{1m} = 1\} = \infty$. Wähle dann eine Teilfolge der $(y_m^{(1)}) \subseteq (y_m)$, in der die Koeffizienten-Teilfolge bezüglich der Koeffizienten α_{1m} des ersten Basisvektors nur aus Einsen besteht.
Weiterhin sind wegen (3.2) alle der n Koeffizienten-Folgen beschränkt. Nach dem Satz von Bolzano-Weierstrass wählen wir nun eine Teilfolge $(y_m^{(2)}) \subseteq (y_m^{(1)})$ für die die zweite Koeffizienten-Folge a_{2m} eine konvergente Teilfolge ist. Aus den Indizes dieser Folge wählen wir wiederum eine bezüglich der dritten Koeffizienten-Folge a_{3m} konvergente Teilfolge und so weiter, bis wir eine Teilfolge

$$y_m^{(n)} = \sum_{k=1}^{n} \alpha'_{km} v_k,$$

erhalten, für die alle Koeffizienten-Folgen konvergieren, d.h.

$$\begin{aligned} \alpha'_{1m} &\to \alpha_1 = 1 \\ \alpha'_{2m} &\to \alpha_2 \\ &\vdots \\ \alpha'_{nm} &\to \alpha_n. \end{aligned}$$

Nach Konstruktion wissen wir, dass (a'_{1m}) nur aus Einsen besteht und also gegen 1 konvergiert. Für

$$y = \sum_{k=1}^{n} \alpha_k v_k$$

gilt dann nach Teil 1 dieses Beweises

$$\|y_m^{(n)} - y\| \leq C\|y_m^{(n)} - y\|_\infty = \max_{k=1,\ldots,n} \{|\alpha'_{km} - \alpha_k|\} \to 0 \text{ für } m \to \infty.$$

Weil $\|y_m\| \to 0$ folgt daraus $y = 0$, ein Widerspruch zum Grenzwert $\alpha_1 = 1$ der ersten Koeffizienten-Folge. QED

Der gerade bewiesene Satz zeigt, dass es auf dem $\mathbb{K}^n$ nicht darauf ankommt, bezüglich welcher Norm man von Konvergenz redet. Genauso induzieren alle Normen auf dem $\mathbb{K}^n$ die gleiche Topologie: Begriffe wie Abgeschlossenheit, Beschränktheit und Kompaktheit hängen also nicht von der Wahl der Norm ab. Allerdings sollte man beachten, dass die Konstanten c, C nicht nur von den jeweiligen Normen, sondern auch von der Dimension des Raumes n abhängen. Weiterhin darf man nicht vergessen, dass der Satz nur für endlich-dimensionale Vektorräume gilt; auf Räume mit unendlicher Dimension (z.B. Funktionenräume) lässt er sich im allgemeinen nicht übertragen.

Als Beispiel wollen wir abschließend noch die p-Normen auf dem Raum der stetigen Funktionen $C[a, b]$ über einem Intervall $[a, b]$ angeben. Für eine stetige Funktion $f : [a, b] \to \mathbb{K}$ definiert man

$$\|f\|_{L_p[a,b]} := \begin{cases} \left(\int_a^b |f(x)|^p dx\right)^{\frac{1}{p}} & \text{falls } 1 \leq p < \infty \\ \max_{x\in[a,b]} |f(x)| & \text{falls } p = \infty \end{cases}$$

3.2 Normen für Abbildungen und Matrizen

Definition 3.10 *Es seien $(V, \|\cdot\|_V)$ und $(W, \|\cdot\|_W)$ zwei normierte Räume und $F : V \to W$ eine lineare Abbildung. Dann heißt F* **beschränkt**, *falls es eine Konstante $C > 0$ gibt, sodass für alle $v \in V$:*

$$\|F(v)\|_W \leq C\|v\|_V.$$

Wir untersuchen zunächst die Stetigkeit solcher linearen Abbildungen.

Lemma 3.11 *Sei $F : V \to W$ eine lineare Abbildung zwischen normierten Vektorräumen. Dann ist F genau dann beschränkt, wenn F stetig ist.*

Beweis:

- Ist F beschränkt, so folgt aus

  $$\|F(v) - F(w)\|_W = \|F(v - w)\|_W \leq C\|v - w\|_v$$

  direkt die Stetigkeit von F.

- Ist F stetig, so gibt es zu jedem $\epsilon > 0$ ein $\delta > 0$ so, dass $\|F(v) - Fw\|_W < \epsilon$ für alle v, w mit $\|v - w\|_V \leq \delta$. Für $\epsilon = 1$ und $w = 0$ erhält man wegen $F(0) = 0$ also ein $\delta > 0$ so, dass

$$\|F(v)\|_W < 1 \text{ für alle } \|v\|_V \leq \delta.$$

Für jedes $v \in V \setminus \{0\}$ gilt

$$\begin{aligned} & \left\| \delta \frac{v}{\|v\|_V} \right\|_V \leq \delta \\ \Longrightarrow \quad & \left\| F\left(\delta \frac{v}{\|v\|_V} \right) \right\|_W \leq 1 \\ \Longrightarrow \quad & \|F(v)\|_W = \frac{\|v\|_V}{\delta} \left\| F\left(\delta \frac{v}{\|v\|_V} \right) \right\|_W \leq \frac{1}{\delta} \|v\|_V, \end{aligned}$$

also folgt die Beschränktheit mit $C = \frac{1}{\delta}$. QED

Zwischen endlich-dimensionalen Räumen stellt sich die Situation noch einfacher dar.

Lemma 3.12 *Sei $F : V \to W$ eine lineare Abbildung zwischen zwei normierten endlich-dimensionalen Vektorräumen. Dann ist F beschränkt und stetig.*

Beweis: Sei $F : V \to W$ linear und sei $v_1, \ldots, v_n$ eine Basis von V. Dann gilt

$$\begin{aligned} & v = \sum_{k=1}^{n} \alpha_k v_k \\ \Longrightarrow \quad & F(v) = F\left(\sum_{k=1}^{n} \alpha_k v_k \right) = \sum_{k=1}^{n} \alpha_k F(v_k) \\ \Longrightarrow \quad & \|F(v)\|_W = \left\| \sum_{k=1}^{n} \alpha_k F(v_k) \right\|_W \leq \sum_{k=1}^{n} |\alpha_k| \|F(v_k)\|_W \\ & \qquad \leq \max_{k=1\ldots n} \|F(v_k)\|_W \sum_{k=1}^{n} |\alpha_k| = \max_{k=1\ldots n} \|F(v_k)\|_W \|v\|_1 \\ & \qquad \leq C \|v\|_V, \end{aligned}$$

wobei beim letzten Schritt ausgenutzt wurde, dass nach Satz 3.9 alle Normen auf V äquivalent sind. Damit ist F also beschränkt und nach Lemma 3.11 auch stetig. QED

Auf dem Raum der beschränkten linearen Abbildungen zwischen zwei normierten Vektorräumen definieren wir nun folgende Norm.

Definition 3.13 *Es seien $(V, \|\cdot\|_V)$ und $(W, \|\cdot\|_W)$ zwei normierte Räume. Für eine beschränkte lineare Abbildung $F : V \to W$ definiert man die zu $\|\cdot\|_V$ und $\|\cdot\|_W$* **zugeordnete Norm** *durch*

$$\|F\|_{V,W} := \sup_{v \in V \setminus \{0\}} \frac{\|F(v)\|_W}{\|v\|_V}.$$

Gilt $V = W$ und $\|\cdot\|_V = \|\cdot\|_W$ so schreiben wir auch $\|F\|_V$ statt $\|F\|_{V,W}$.

Weil F als beschränkt vorausgesetzt wurde gilt für alle $v \in V \setminus \{0\}$

$$\frac{\|F(v)\|_W}{\|v\|_V} \leq \frac{C\|v\|_V}{\|v\|_V} = C.$$

Wir erhalten also $\|F\|_{V,W} < \infty$. Die Norm der Abbildung F ist also die kleinstmögliche Konstante C, mit der man die Beschränktheit der Abbildung abschätzen kann.

Wir erwähnen noch, dass wir auch wirklich von Normen sprechen dürfen:

Satz 3.14 *Es seien $(V, \|\cdot\|_V)$ und $(W, \|\cdot\|_W)$ zwei normierte Räume. Dann ist $\|\cdot\|_{V,W}$ eine Norm auf dem Raum der beschränkten linearen Abbildungen von $V \to W$.*

Aufgabe: *Beweisen Sie Satz 3.14!*

Folgende Umformulierung erweist sich als nützlich.

Lemma 3.15 *Es seien $(V, \|\cdot\|_V)$ und $(W, \|\cdot\|_W)$ zwei normierte Räume und $F : V \to W$ eine beschränkte lineare Abbildung. Dann gilt*

$$\|F\|_{V,W} = \sup_{v \in V : \|v\|_V = 1} \|F(v)\|_W. \tag{3.4}$$

Beweis: Zunächst ist klar, dass

$$\sup_{v \in V : \|v\|_V = 1} \|F(v)\|_W \leq \|F\|_{V,W}.$$

Um $\sup_{v \in V : \|v\|_V = 1} \|F(v)\|_W \geq \|F\|_{V,W}$ zu zeigen, bemerken wir, dass wegen der Skalierbarkeit (Eigenschaft (N2)) der Norm $\|\cdot\|_W$ für alle $v \neq 0$, $v \in V$ gilt:

$$\frac{\|F(v)\|_W}{\|v\|_V} = \frac{1}{\|v\|_V}\|F(v)\|_W = \left\| F\left(\frac{v}{\|v\|_V}\right)\right\|_W.$$

Es gibt also zu jedem $v \neq 0$ ein u mit $\|u\|_V = 1$ so dass $\frac{\|F(v)\|_W}{\|v\|_V} = \|F(u)\|_W$. Entsprechend folgt

$$\sup_{v \neq 0} \frac{\|F(v)\|_W}{\|v\|_V} \leq \sup_{v : \|v\|_V = 1} \|F(v)\|_W$$

und zusammen ergibt sich die Behauptung. QED

Betrachte nun ein beliebiges $v \in V$. Dann gilt:

$$\frac{\|F(v)\|_W}{\|v\|_V} \leq \sup_{v' \in V \setminus \{0\}} \frac{\|F(v')\|_W}{\|v'\|_V} = \|F\|_{V,W},$$

woraus wir

$$\|F(v)\|_W \leq \|F\|_{V,W} \cdot \|v\|_V \tag{3.5}$$

folgern. Wir sagen auch, $\|\cdot\|_{V,W}$ ist *passend* zu den Normen $\|\cdot\|_V$ und $\|\cdot\|_W$. Diese Eigenschaft wird später noch wichtig werden. Eine Verallgemeinerung ist die folgende.

Definition 3.16 *Es seien $(V, \|\cdot\|_V)$ und $(W, \|\cdot\|_W)$ zwei normierte Räume. Eine Norm $\|\cdot\|$ auf dem Raum der beschränkten, linearen Abbildungen von V nach W heißt* **zu den Normen $\|\cdot\|_V$ und $\|\cdot\|_W$ passend**, *oder* **mit den Normen $\|\cdot\|_V$ und $\|\cdot\|_W$ verträglich**, *falls für alle $v \in V$ gilt:*

$$\|F(v)\|_W \leq \|F\| \cdot \|v\|_V.$$

Gleichung (3.5) zeigt, dass die Norm $\|\cdot\|_{V,W}$ immer zu ihren natürlichen oder zugeordneten Normen $\|\cdot\|_V$ und $\|\cdot\|_W$ passt.

Aufgabe: *Seien $(U, \|\cdot\|_U)$,$(V, \|\cdot\|_V)$ und $(W, \|\cdot\|_W)$ normierte endlich-dimensionale Vektorräume und seien $F : U \to V$ und $G : V \to W$ beschränkte lineare Abbildungen. Zeigen Sie, dass dann für $G \circ F : U \to W$ gilt:*

$$\|G \circ F\|_{UW} \leq \|G\|_{VW} \|F\|_{UV}.$$

Wir möchten nun den Fall linearer Abbildungen zwischen den endlich-dimensionalen Vektorräumen

$$A : \mathbb{K}^n \to \mathbb{K}^m$$

genauer untersuchen. Jede lineare Abbildung kann dann durch eine Matrix A repräsentiert werden, so dass wir die zugehörige Norm $\|A\|_{V,W}$ in diesem Fall auch *Matrixnorm* nennen.

Im folgenden entwickeln wir Formeln für einige Matrixnormen, die aus den wichtigsten Normen auf dem $\mathbb{K}^n, \mathbb{K}^m$ entstehen.

Satz 3.17 *Sei $A \in \mathbb{K}^{m,n}$ eine lineare Abbildung vom $\mathbb{K}^n$ in den $\mathbb{K}^m$.*

1. *Betrachte $(\mathbb{K}^n, \|\cdot\|_1)$ und $(K^m, \|\cdot\|_1)$ jeweils mit Manhattan-Norm. Dann heißt die die zugehörige Matrixnorm Spaltensummennorm und sie ist gegeben durch*

$$\|A\|_1 = \sup_{x \in \mathbb{K}^n : \|x\|_1 = 1} \|Ax\|_1 = \max_{k=1,\ldots,n} \sum_{i=1}^{m} |a_{ik}|.$$

2. *Betrachte* $(\mathbb{K}^n, \|\cdot\|_\infty)$ *und* $(K^m, \|\cdot\|_\infty)$ *jeweils mit Maximum-Norm. Dann heißt die die zugehörige Matrixnorm Zeilensummennorm und sie ist gegeben durch*

$$\|A\|_\infty = \sup_{x\in\mathbb{K}^n:\|x\|_\infty=1} \|Ax\|_\infty = \max_{i=1,\dots,m} \sum_{k=1}^{n} |a_{ik}|.$$

Beweis:

ad 1: Für alle $x \in \mathbb{K}^n$ gilt zunächst, dass

$$\begin{aligned}
\|Ax\|_1 &= \sum_{i=1}^{m} |(Ax)_i| = \sum_{i=1}^{m} \left| \sum_{k=1}^{n} a_{ik}x_k \right| \\
&\leq \sum_{k=1}^{n} |x_k| \sum_{i=1}^{m} |a_{ik}| \leq \left(\max_{k=1,\dots,n} \sum_{i=1}^{m} |a_{ik}| \right) \sum_{k=1}^{n} |x_k| \\
&= \max_{k=1,\dots,n} \sum_{i=1}^{n} |a_{ik}| \|x\|_1.
\end{aligned}$$

Damit gilt also

$$\|A\|_1 \leq \max_{k=1,\dots,n} \sum_{i=1}^{m} |a_{ik}|.$$

Um $\|A\|_1 \geq \max_{k=1,\dots,n} \sum_{i=1}^{m} |a_{ik}|$ zu zeigen, wählen wir j so dass

$$\sum_{i=1}^{m} |a_{ij}| = \max_{k=1,\dots,n} \sum_{i=1}^{m} |a_{ik}|.$$

Für den $j$ten Einheitsvektor e_j gilt dann

$$\|Ae_j\|_1 = \|A_j\|_1 = \sum_{i=1}^{m} |a_{ij}| = \max_{k=1,\dots,n} \sum_{i=1}^{m} |a_{ik}|.$$

Für die Norm von A folgt (mit (3.4)) daraus

$$\|A\|_1 = \sup_{x:\|x\|_1=1} \|Ax\|_1 \geq \|Ae_j\|_1 = \max_{k=1,\dots,n} \sum_{i=1}^{m} |a_{ik}|.$$

ad 2: Für die Maximums-Norm erhalten wir analog für $x \in \mathbb{K}^n$

$$\begin{aligned}
\|Ax\|_\infty &= \max_{i=1,\dots,m} |(Ax)_i| = \max_{i=1,\dots,m} \left| \sum_{k=1}^{n} a_{ik}x_k \right| \\
&\leq \max_{i=1,\dots,m} \sum_{k=1}^{n} |a_{ik}||x_k| \leq \max_{i=1,\dots,m} \sum_{k=1}^{n} |a_{ik}| \|x\|_\infty,
\end{aligned}$$

also

$$\|A\|_\infty \leq \max_{i=1,\ldots,m} \sum_{k=1}^{n} |a_{ik}|.$$

Für die "$\geq$" Richtung wählen wir hier den Index j als den der Zeile mit maximaler Summe, d.h. so dass

$$\sum_{k=1}^{n} |a_{jk}| = \max_{i=1,\ldots,m} \sum_{k=1}^{n} |a_{ik}|.$$

Weiterhin wählen wir einen Vektor $z \in \mathbb{K}^n$ passend zum Index j durch

$$z_k = \begin{cases} \frac{\bar{a}_{jk}}{|a_{jk}|} & \text{falls } a_{jk} \neq 0 \\ 1 & \text{falls } a_{jk} = 0 \end{cases}$$

Dann gilt

a) $\|z\|_\infty = 1$, und

b) $a_{jk} z_k = \frac{a_{jk}\bar{a_{jk}}}{|a_{jk}|} = |a_{jk}|$, insbesondere ist $a_{jk}z_k$ positiv und reell.

Für die Norm von Az erhalten wir daraus, dass

$$\begin{aligned} \|Az\|_\infty &= \max_{i=1,\ldots,m} |(Az)_i| = \max_{i=1,\ldots,m} \left| \sum_{k=1}^{n} a_{ik} z_k \right| \\ &\geq \left| \sum_{k=1}^{n} a_{jk} z_k \right| = \sum_{k=1}^{n} |a_{jk}| = \max_{i=1,\ldots,m} \sum_{k=1}^{n} |a_{ik}|. \end{aligned}$$

Wie für die Manhattan-Norm folgern wir daraus, dass

$$\|A\|_\infty \geq \max_{i=1,\ldots,m} \sum_{k=1}^{n} |a_{ik}|.$$

QED

Wir betrachten jetzt noch die Matrixnorm $\|A\|_2$. Dazu benötigen wir den auch in anderen Bereichen der Numerik wichtigen Begriff des *Spektralradius* einer Matrix.

Definition 3.18 *Sei $A \in \mathbb{K}^{n,n}$.*

- *$\lambda \in \mathbb{K}$ heißt* **Eigenwert** *von A falls es ein $v \in \mathbb{K}^n \setminus \{0\}$ gibt, so dass*

$$Av = \lambda v.$$

v heißt dann **Eigenvektor** *von A bezüglich des Eigenwertes λ*

- *Der* **Spektralradius** $\rho(A)$ *einer Matrix* A *ist der betragsmäßig größte Eigenwert von* A*, d.h.*

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

Wir müssen zunächst an die folgenden Begriffe aus der linearen Algebra erinnern:

Notation 3.19

- *Eine Matrix* $A \in \mathbb{R}^{n,n}$ *heißt orthogonal, falls* $A^T A = I$ *beziehungsweise* $A^{-1} = A^T$*.*
- *Eine Matrix* $A \in \mathbb{C}^{n,n}$ *heißt unitär, falls* $A^{-1} = \bar{A}^T$*.*

Bemerkung: Die Spalten $A_1, \ldots, A_n$ von A von orthogonalen oder unitären Matrizen bilden eine Orthonormalbasis des $\mathbb{K}^n$. Das sieht man, indem man das Produkt $B = \bar{A}^T A$ durch Produkte der Spalten von A beschreibt. Weil B die Einheitsmatrix ist, gilt für das Element

$$b_{ij} = \bar{A}_i^T A_j = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{sonst,} \end{cases}$$

entsprechend folgt die Behauptung.
Folgenden Satz werden wir verwenden.

Satz 3.20 (Hauptachsentransformation) *Sei* $A \in \mathbb{K}^{n,n}$ *eine symmetrische (bzw. hermitesche) Matrix. Dann gibt es eine reguläre orthogonale (bzw. unitäre) Matrix* $Q \in \mathbb{R}^{n,n}$ *(bzw.* $Q \in \mathbb{C}^{n,n}$*) und eine Diagonalmatrix* $D = diag(d_1, \ldots, d_n) \in \mathbb{R}^{n,n}$ *so dass*

$$A = QDQ^{-1}.$$

Dabei sind $d_1, \ldots, d_n$ *die Eigenwerte der Matrix* A*, und die Spalten von* Q *bilden eine Orthonormalbasis, die aus den zugehörigen Eigenvektoren besteht. Das heißt, es gilt*

$$AQ_j = d_{jj} Q_j \text{ für } j = 1, \ldots, n$$

Wir beweisen folgende Folgerung aus Satz 3.20.

Lemma 3.21 *Sei* $A \in \mathbb{K}^{n,n}$ *eine symmetrische (bzw. hermitesche) positiv semidefinite Matrix, und sei* $\lambda^{\min}$ *ihr betragsmäßig kleinster und* $\rho(A) = \lambda^{max}$ *ihr betragsmäßig größter Eigenwert. Dann gilt* $\lambda^{\min} \geq 0$ *und alle* $x \in \mathbb{K}^n$ *erfüllen die folgende Abschätzung:*

$$\lambda^{\min} \|x\|_2^2 \leq \bar{x}^T A x \leq \lambda^{\max} \|x\|_2^2.$$

Beweis: Weil A positiv semi-definit ist, sind die Eigenwerte $\lambda_1, \ldots, \lambda_n$ von A nicht-negativ (siehe Definition 2.18 auf Seite 37). Sei nach Satz 3.20 weiter $v_1, \ldots, v_n$ eine Orthonormalbasis des $\mathbb{K}^n$, die aus Eigenvektoren von A besteht. Wir schreiben $x = \sum_{i=1}^n \alpha_i v_i$ und rechnen wegen

$$Ax = A\sum_{i=1}^n \alpha_i v_i = \sum_{i=1}^n \alpha_i A(v_i) = \sum_{i=1}^n \alpha_i \lambda_i v_i$$

nach, dass

$$\bar{x}^T Ax = \sum_{j,k=1}^n \bar{\alpha}_j \alpha_k \lambda_k \bar{v}_j^T v_k = \sum_{j=1}^n |\alpha_j|^2 \lambda_j,$$

wobei letztere Gleichheit aus der Orthogonalität der v_i folgt. Weiterhin gilt $\|x\|_2^2 = x^T x = \sum_{i=1}^n |\alpha_i|^2$ und entsprechend folgt

$$\lambda^{\min} \sum_{j=1}^n |\alpha_j|^2 \leq \sum_{j=1}^n |\alpha_j|^2 \lambda_j \leq \lambda^{\max} \sum_{j=1}^n |\alpha_j|^2,$$

zusammen also

$$\lambda^{\min} \|x\|_2^2 \leq \bar{x}^T Ax \leq \lambda^{\max} \|x\|_2^2.$$

QED

Wir können nun endlich auch die Matrixnorm $\|A\|_2$ bezüglich der Euklidischen Norm berechnen.

Satz 3.22 *Für $A : \mathbb{K}^n \to \mathbb{K}^m$, also $A \in \mathbb{K}^{m,n}$ gilt*

$$\|A\|_2 = \sup_{x \in \mathbb{K}^n : x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \sqrt{\rho(\bar{A}^T A)}$$

Man nennt $\|A\|_2$ auch die **Spektralnorm** *von A.*

Beweis: Zunächst gilt, dass $\bar{A}^T A \in \mathbb{K}^{n,n}$ eine hermitesche und positiv semi-definite Matrix ist. Daher sind alle ihre Eigenwerte größer oder gleich Null. Sei $\rho(\bar{A}^T A) = \lambda^{\max}$ der größte Eigenwert von $\bar{A}^T A$. Es gilt

$$\|Ax\|_2^2 = \overline{(Ax)}^T (Ax) = \bar{x}^T \bar{A}^T Ax \leq \lambda^{\max} \|x\|_2^2,$$

wobei die letzte Ungleichung aus Lemma 3.21 folgt. Die Ungleichung ergibt also

$$\|A\|_2 \leq \sqrt{\rho(\bar{A}^T A)}.$$

Um Gleichheit zu zeigen, wählen wir z als Eigenvektor zu $\lambda^{\max}$ und erhalten

$$\|Az\|_2^2 = \bar{z}^T \bar{A}^T Az = \bar{z}^T \lambda^{\max} z = \lambda^{\max} \bar{z}^T z = \lambda^{\max} \|z\|_2^2.$$

Daraus ergibt sich analog zu dem Beweis von Satz 3.17, dass

$$\|A\|_2 = \sup_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2} \geq \frac{\|Az\|_2}{\|z\|_2} = \sqrt{\lambda^{\max}} = \sqrt{\rho(\bar{A}^T A)}.$$

QED

Leider ist die Spektralnorm für größere Matrizen aufwändig zu berechnen. Daher ersetzt man sie manchmal durch eine der folgenden Normen:

Definition 3.23

$$\textit{Gesamtnorm:} \qquad \|A\|_G := n \max\{|a_{ij}| : 1 \leq i \leq m, 1 \leq j \leq n\}$$

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

Beides sind wirklich Normen (da sie bis auf Vorfaktoren mit $\|\cdot\|_\infty$ beziehungsweise mit $\|\cdot\|_2$ auf dem $\mathbb{K}^{n\cdot m}$ übereinstimmen).

Lemma 3.24

1. *Die Norm $\|\cdot\|_G$ ist passend zu $\|\cdot\|_\infty$.*
2. *Die Norm $\|\cdot\|_F$ ist passend zu $\|\cdot\|_2$.*

Beweis: Nach Definition 3.16 ist für den ersten Teil zu zeigen, dass

$$\|Ax\|_\infty \leq \|A\|_G \cdot \|x\|_\infty.$$

Das rechnet man leicht nach durch

$$\begin{aligned} \|Ax\|_\infty &= \max_{i=1,\ldots,m} \left| \sum_{j=1}^{n} a_{ij} x_j \right| \leq \|x\|_\infty \max_{i=1\ldots m} \sum_{j=1}^{n} |a_{ij}| \\ &\leq \|x\|_\infty \, n \max\{|a_{ij}| : 1 \leq i \leq m, 1 \leq j \leq n\} = \|A\|_G \cdot \|x\|_\infty. \end{aligned}$$

Für den zweiten Teil müssen wir uns überzeugen, dass

$$\|Ax\|_2 \leq \|A\|_F \cdot \|x\|_2.$$

Wieder rechnen wir

$$\begin{aligned} \|Ax\|_2^2 &= \sum_{i=1,\ldots,m} \left| \sum_{j=1}^{n} a_{ij} x_j \right|^2 \leq \sum_{i=1}^{m} \left(\sum_{j=1}^{n} |a_{ij}|^2 \sum_{j=1}^{n} |x_j|^2 \right) \text{ siehe Lemma 3.5} \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2 \sum_{j=1}^{n} |x_j|^2 = \|A\|_F^2 \, \|x\|_2^2 \end{aligned}$$

und erhalten so das gewünschte Resultat. QED

3.3 Kondition

Zum Abschluss dieses Kapitels wollen wir die gewonnen Erkenntnisse anwenden, um die *Kondition* einer Matrix zu definieren. Diese wird uns helfen, die Übertragung von Fehlern abzuschätzen.

Betrachten wir dazu ein lineares Gleichungssystem $Ax = b$ mit folgenden Fehlern in den Eingangsdaten:

- ΔA sei der Fehler in der Matrix A,
- sowie Δb der Fehler im Ergebnisvektor b.

Lässt sich anhand dieser Daten der Fehler im Ergebnis abschätzen?

Um diese Frage zu beantworten, bemerken wir zunächst, dass

$$\Delta x = \tilde{x} - x$$

ist, wobei x als exakte Lösung des Gleichungssystems $Ax = b$ und $\tilde{x}$ durch die gewonnene Lösung

$$(A + \Delta A)\tilde{x} = b + \Delta b$$

definiert ist. Es gilt also

$$(A + \Delta A)(x + \Delta x) = b + \Delta b.$$

Multipliziert man diese Gleichung aus und verwendet $Ax = b$ so ergibt sich

$$(A + \Delta A)\Delta x = \Delta b - \Delta A x.$$

Nehmen wir nun zunächst an, dass die gestörte Matrix $A + \Delta A$ invertierbar wäre. Dann könnte man nach Δx auflösen und dadurch die Norm von x abschätzen, also

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

wobei wir eine multiplikative und zur Vektornorm passende Matrixnorm gewählt haben. Der relative Fehler ergibt sich entsprechend als

$$\begin{aligned} \frac{\|\Delta x\|}{\|x\|} &\leq \|(A + \Delta A)^{-1}\| \left(\frac{\|\Delta b\|}{\|x\|} + \|\Delta A\| \right) \\ &= \|(A + \Delta A)^{-1}\|\|A\| \left(\frac{\|\Delta b\|}{\|A\|\|x\|} + \frac{\|\Delta A\|}{\|A\|} \right) \\ &\leq \|(A + \Delta A)^{-1}\|\|A\| \left(\frac{\|\Delta b\|}{\|Ax\|} + \frac{\|\Delta A\|}{\|A\|} \right) \\ &\leq \underbrace{\|(A + \Delta A)^{-1}\|\|A\|}_{\text{Vergrößerungsfaktor}} \left(\underbrace{\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|}}_{\text{relative Fehler der Eingangsdaten}} \right) \end{aligned} \tag{3.6}$$

Bevor wir den Term des Vergrößerungsfaktors weiter abschätzen, beschäftigen wir uns mit der Frage, wann die Inverse von $A + \Delta A$ existiert.

Lemma 3.25 *Seien $A, \Delta A \in \mathbb{K}^{n,n}$, A regulär und $\|A^{-1}\|\|\Delta A\| < 1$, wobei $\|\cdot\|$ eine zu eine Vektornorm passende multiplikative Matrixnorm ist, die $\|I\| = 1$ erfüllt. Dann ist $A + \Delta A$ regulär, und es gilt*

$$\|(A + \Delta A)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\Delta A\|}.$$

Beweis: Schreibe

$$\begin{array}{rl} & x = A^{-1}(A + \Delta A)x - A^{-1}(\Delta A)x \\ \Longrightarrow & \|x\| \leq \|A^{-1}\|\|(A + \Delta A)x\| + \|A^{-1}\|\|(\Delta A)\|\|x\| \\ \Longrightarrow & \|x\| \underbrace{(1 - \|A^{-1}\|\|\Delta A\|)}_{>0 \text{ nach Vor.}} \leq \|A^{-1}\|\|(A + \Delta A)x\|. \end{array}$$

Also folgt aus $(A + \Delta A)x = 0$ dass $\|x\| = 0$ und entsprechend auch $x = 0$. Die Abbildung $A+\Delta A$ ist somit injektiv und damit auch surjektiv, also ist die Matrix $A + \Delta A$ invertierbar.

Wir können also $B := (A + \Delta A)^{-1}$ definieren. Um die im Lemma genannte Abschätzung zu erhalten, rechnen wir nach

$$\begin{array}{rcl} 1 & = & \|I\| = \|B(A + \Delta A)\| = \|BA + BAA^{-1}\Delta A\| \\ & \geq & \|BA\| - \|BA\|\|A^{-1}\|\|\Delta A\| \\ & = & \|BA\| \underbrace{(1 - \|A^{-1}\|\|\Delta A\|)}_{>0}. \end{array}$$

Daraus erhalten wir

$$\|BA\| \leq \frac{1}{1 - \|A^{-1}\|\|\Delta A\|}$$

und schließlich

$$\|(A + \Delta A)^{-1}\| = \|BAA^{-1}\| \leq \|BA\|\|A^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\Delta A\|}.$$

QED

Definition 3.26 *Für eine Matrix $A \in \mathbb{K}^{n,n}$ definieren wir*

$$\text{cond}(A) := \|A\| \, \|A^{-1}\|$$

als die **Kondition** *von A.*

Wozu man diese Definition verwenden kann, zeigt der folgende Satz und das anschließende Korollar.

Satz 3.27 *Sei $\|\cdot\|$ eine Matrixnorm wie in Lemma 3.25. Sei $\|b\| \neq 0$ und $\|A^{-1}\|\ \|\Delta A\| < 1$. Sei $Ax = b$. Dann gilt für jede gestörte Lösung $x + \Delta x$ des gestörten Systems*

$$(A + \Delta A)\tilde{x} = b + \Delta b$$

die folgende Abschätzung:

$$\frac{\|\Delta x\|}{\|x\|} \leq \mathrm{cond}(A)\frac{1}{1 - \mathrm{cond}(A)\frac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right)$$

Zunächst bemerken wir, dass der Ausdruck wegen

$$1 > 1 - \mathrm{cond}(A)\frac{\|\Delta A\|}{\|A\|} = 1 - \|A^{-1}\|\ \|\Delta A\| > 0$$

wohldefiniert ist. Man sieht hier auch schon, dass eine kleinere Kondition zu kleineren relativen Fehlern führen wird.

Beweis: Aus (3.6) und der Abschätzung aus Lemma 3.25 folgt, dass

$$\begin{aligned}\frac{\|\Delta x\|}{\|x\|} &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\ \|\Delta A\|} \cdot \frac{\|A\|}{\frac{\|A\|}{\|A\|}}\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right) \\ &= \frac{\mathrm{cond}(A)}{1 - \mathrm{cond}(A)\frac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right).\end{aligned}$$

QED

Eine einfache und oft betrachtete Anwendung dieses Ergebnisses ist das folgende Korollar, das man auch direkt aus (3.6) ohne die Voraussetzungen aus Lemma 3.25 herleiten kann.
Korollar: Hat man nur eine Störung in b (ist also $\Delta A = 0$), so übertragen sich die Fehler in b mit maximal der Kondition von A. Genauer:

$$\frac{\|\Delta x\|}{\|x\|} \leq \mathrm{cond}(A)\frac{\|\Delta b\|}{\|b\|}.$$

Diese Aussage ergibt sich direkt aus Satz 3.27, da die Voraussetzung $\|A^{-1}\|\ \|\Delta A\| = 0 < 1$ erfüllt ist.

Abschließend geben wir noch zwei nützliche Aussagen zur Bestimmung der Kondition einer Matrix an.

Lemma 3.28 *Für jede zu einer Vektornorm passend gewählte Matrixnorm und jede invertierbare Matrix A gilt: cond$(A) \geq 1$.*

Beweis: Für den Beweis verwenden wir die Definition für *passend* für A und A^{-1} in folgendem Sinn. Sei $x \neq 0$. Dann gilt $A^{-1}x \neq 0$ und entsprechend

$$\begin{aligned} \|A^{-1}x\| &\leq \|A^{-1}\| \, \|x\| \\ \|A(A^{-1}x)\| &\leq \|A\| \, \|(A^{-1}x)\| \end{aligned}$$

Zusammen ergibt sich

$$\|A\| \, \|A^{-1}\| \geq \|A\| \frac{\|A^{-1}x\|}{\|x\|} \geq \frac{\|A(A^{-1}x)\|}{\|A^{-1}x\|} \frac{\|A^{-1}x\|}{\|x\|} = \frac{\|x\|}{\|x\|} = 1.$$

QED

Für die der Euklidischen Norm zugeordnete Spektralnorm gelten die folgenden Aussagen.

Lemma 3.29 *Sei Q eine orthogonale (unitäre) Matrix. Dann gilt*

1. *cond$(Q) = 1$, und*
2. *cond(QA) = cond(A) = cond(AQ) für alle Matrizen A, das heißt die Multiplikation mit Q ändert die Kondition der Matrix A nicht.*

Beweis:

1.

$$\begin{aligned} \text{cond}(Q) &= \|Q\|_2 \|Q^{-1}\|_2 = \sqrt{\rho(\bar{Q}^T Q)} \sqrt{\rho(\bar{Q}^{-1^T} Q^{-1})} \\ &= \sqrt{\rho(I)} \sqrt{\rho(Q\bar{Q}^T)} = \sqrt{\rho(I)} \sqrt{\rho(I)} = 1 \end{aligned}$$

2.

$$\begin{aligned} \|A\|_2 &= \|Q^T Q A\|_2 \leq \|Q^T\|_2 \, \|QA\|_2 = \|QA\|_2 \\ &\leq \|Q\|_2 \, \|A\|_2 = \|A\|_2, \end{aligned}$$

also ist $\|A\|_2 = \|QA\|_2$. Analog ergibt sich $\|A\|_2 = \|AQ\|_2$, also erhält man $\text{cond}(QA) = \text{cond}(Q) = \text{cond}(AQ)$.

QED

Kapitel 4

Orthogonalisierungsverfahren

4.1 Die QR-Zerlegung

Bisherige Lösung von Gleichungssystemen:

$$A \to L \cdot A = \begin{pmatrix} \ddots & & * \\ & \ddots & \\ 0 & & \ddots \end{pmatrix}$$

Dabei galt für die Kondition von $(L \cdot A)$:

$$\begin{aligned} \operatorname{cond}(L \cdot A) &\leq \|L\| \cdot \|L^{-1}\| \cdot \|A\| \cdot \|A^{-1}\| \\ &= \operatorname{cond}(A) \cdot \operatorname{cond}(L), \end{aligned}$$

die Kondition vergrößert sich also um bis zu $\operatorname{cond}(L)$.

Idee: *Die Kondition lässt sich verbessern, indem man A durch Multiplikation mit orthogonalen bzw. unitären Matrizen auf eine obere Δs-Gestalt bringt, denn für orthogonale/unitäre Matrizen gilt nach Lemma 3.29*

$$\operatorname{cond}(QA) = \operatorname{cond}(A)$$

sowohl für die Euklidische Norm als auch für $\|\cdot\|_F$.

Lemma 4.1 *Sei Q orthogonal (bzw. unitär). Dann gilt $\|Qx\|_2 = \|x\|_2$.*

Beweis:

$$\|Qx\|_2^2 = (Qx)^*(Qx) = x^* \underbrace{Q^*Q}_{I} x = x^*x = \|x\|_2^2$$

QED

Definition 4.2 *Die Zerlegung einer Matrix $A \in \mathbb{K}^{m,n}$ der Form $A = QR$ mit einer unitären Matrix $Q \in \mathbb{K}^{m,m}$ und einer oberen Δs-Matrix $R \in \mathbb{K}^{m,n}$ heißt* ***QR-Zerlegung*** *von A. Dabei hat die Matrix R folgende Gestalt:*

$$R = \left(\begin{array}{ccc} r_{11} & & * \\ & \ddots & \\ 0 & & r_{kk} \\ \hline & 0 & \ddots \\ \hline & 0 & \end{array} \right) \begin{array}{l} \Big\} n \\ \\ \\ \\ \Big\} m \end{array}$$

wobei $k \leq \min\{n, m\}$ und $r_{11}, r_{22}, ..., r_{kk} \neq 0$.

Definition 4.3 *Sei $h \in \mathbb{K}^n$ normiert, d.h. $h^*h = \|h\|^2 = 1$. Dann heißt*

$$H = I - 2hh^*$$

Householder-Matrix.

Bemerkung:

$$h = \begin{pmatrix} h_1 \\ \vdots \\ h_n \end{pmatrix} \qquad h^* = (\overline{h}_1, \ldots, \overline{h_n}) \qquad hh^* = \begin{pmatrix} h_1\overline{h}_1 & h_1\overline{h}_2 & \cdots & h_1\overline{h}_n \\ h_2\overline{h}_1 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ h_n\overline{h}_1 & \cdots & \cdots & h_n\overline{h}_n \end{pmatrix}$$

Lemma 4.4 *Sei H eine Householder-Matrix. Dann gilt $HH^* = H^*H = I$ und $H = H^*$.*

Beweis:

$$\begin{aligned} H^* &= (I - 2hh^*)^* = I - 2hh^* = H \\ HH^* &= (I - 2hh^*)(I - 2hh^*) = I - 4hh^* + 4h \underbrace{h^*h}_{1} h^* = I \end{aligned}$$

QED

Beispiel: Sei

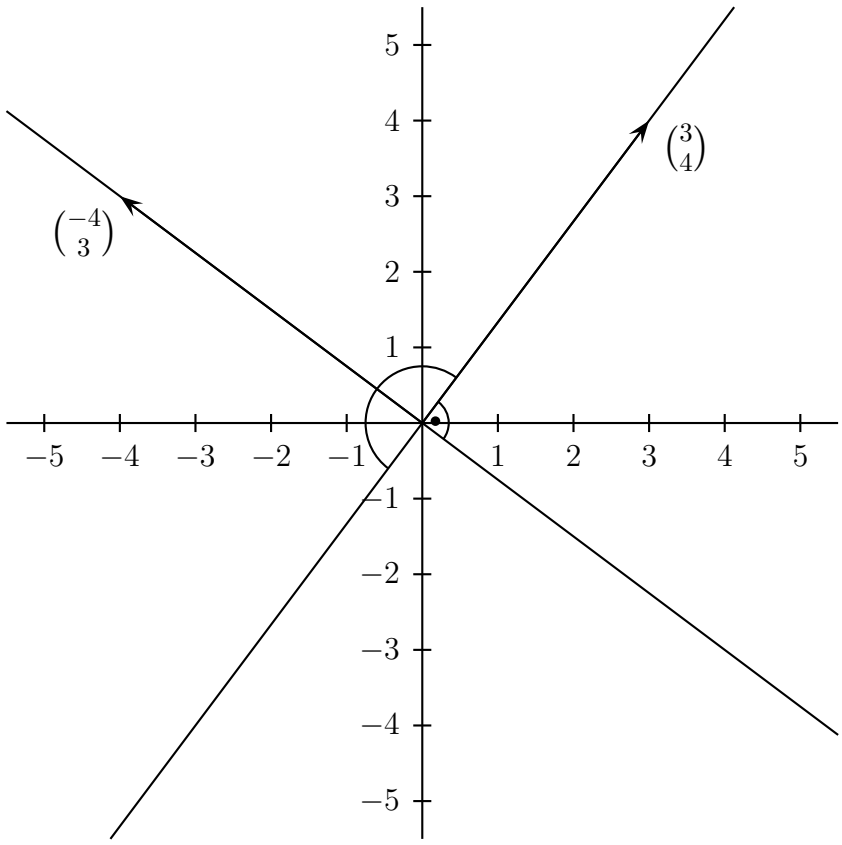
$$h = \frac{1}{5} \begin{pmatrix} 3 \\ 4 \end{pmatrix},$$

dann gilt $h^Th = 1$. Für die Householder-Matrix ergibt sich daraus:

$$H = I - 2hh^T = I - 2 \begin{pmatrix} \frac{3}{5} \\ \frac{4}{5} \end{pmatrix} \begin{pmatrix} \frac{3}{5} & \frac{4}{5} \end{pmatrix} = \frac{1}{25} \begin{pmatrix} 7 & -24 \\ -24 & -7 \end{pmatrix}$$

wobei $H = H^T$ und $H^2 = I$. Nun kann man beliebige Punkte durch Multiplikation mit der Householder-Matrix auf andere abbilden, zum Beispiel:

$$H\begin{pmatrix}3\\4\end{pmatrix} = \begin{pmatrix}-3\\-4\end{pmatrix} \qquad H\left(\lambda\begin{pmatrix}3\\4\end{pmatrix}\right) = -\lambda\begin{pmatrix}3\\4\end{pmatrix}$$
$$H\begin{pmatrix}4\\-3\end{pmatrix} = \begin{pmatrix}4\\3\end{pmatrix} \qquad H\left(\lambda\begin{pmatrix}4\\-3\end{pmatrix}\right) = \lambda\begin{pmatrix}4\\-3\end{pmatrix}$$



Aufgrund des Bildes scheint H also eine Spiegelung an der Geraden durch den Ursprung senkrecht zu h zu sein. Das wollen wir im folgenden begründen:

Lemma 4.5 *Die Abbildung $H : \mathbb{R}^n \to \mathbb{R}^n$ entspricht geometrisch einer Spiegelung an der zu h orthogonalen Ebene durch den Ursprung.*

Sei $x \in \mathbb{R}^n$. Wir zerlegen x in einen Anteil in Richtung h und einen Anteil orthogonal zu h.

$$x = \alpha h + \beta t \qquad \text{mit } t \perp h \quad (t^T h = 0)$$
$$\left[x = (hh^T)x + y \text{ ist geeignet, da } y \perp h\right]$$

Im Beispiel:

$$\begin{pmatrix}0\\4\end{pmatrix} = \frac{16}{5}\cdot\frac{1}{5}\begin{pmatrix}3\\4\end{pmatrix} + \frac{12}{25}\begin{pmatrix}-4\\3\end{pmatrix}$$

Dann gilt:

$$\begin{aligned} H(x) &= (I - 2hh^T)\cdot(\alpha h + \beta t) \\ &= \alpha\cdot I\cdot h + \beta\cdot I\cdot t - 2\alpha h\underbrace{h^T h}_{1} - 2\beta h\underbrace{h^T t}_{0} \\ &= -\alpha h + \beta t \end{aligned}$$

Also ist H tatsächlich eine Spiegelung an der Ebene durch den Ursprung senkrecht zu h.

Unser Ziel ist es jetzt, die Kondition bei der Lösung von Gleichungssystemen zu verbessern, indem man A durch Multiplikation mit orthogonalen bzw. unitären Matrizen auf obere Dreiecksgestalt bringt, also:

$$Q\cdot A = \begin{pmatrix} * & & \\ & \ddots & \\ 0 & & * \end{pmatrix}$$

wobei Q eine orthogonale bzw. unitäre Matrix darstellt. Dazu verwenden wir Householder-Matrizen H, für welche gilt:

$$H = I - 2hh^*, \text{ wobei } \|h^*\| = 1 \qquad H^*\cdot H = H\cdot H^* = I \qquad H^* = H$$

Unser Ziel ist es, eine Householdermatrix A so zu bestimmen, dass die Anwendung von H auf A zu einer Matrix führt, in der die erste Spalte ein Vielfaches des Einheitsvektors ist, also

$$H\cdot A_1 = \begin{pmatrix}*\\0\\\vdots\\0\end{pmatrix} = \lambda\cdot e_1 \qquad \text{und} \qquad HA = \left(\begin{array}{c|c} * & \\ 0 & \\ \vdots & * \\ 0 & \end{array}\right)$$

Das nächste Lemma zeigt, wie man so eine Matrix H wählen muss.

Lemma 4.6 *Sei $x \in \mathbb{K}^n \setminus \{0\}$. Für*

$$u := x + x_1\cdot\frac{1}{|x_1|}\cdot\|x\|_2\cdot e_1 \text{ und } H = I - 2\cdot\frac{uu^*}{u^*u} \text{ gilt:}$$

$$Hx = \underbrace{-x_1\cdot\frac{\|x\|_2}{|x_1|}}_{\in\mathbb{K}}\cdot e_1$$

Beweis: Wir suchen $u \in \mathbb{K}^n \setminus \{0\}$ mit $Hx = c \cdot e_1$, das heißt

$$Hx = x - 2 \cdot \frac{uu^*}{u^*u} = c \cdot e_1.$$

Das ist erfüllt, falls die beiden folgenden Bedingungen gelten:

$$\frac{2u^*x}{u^*u} = 1 \tag{4.1}$$

$$u = x - c \cdot e_1 \tag{4.2}$$

Aus (4.1) folgt:

$$\begin{array}{rrcll} & 2u^*x & = & u^*u \in \mathbb{R} & \\ \Rightarrow & u^*x \in \mathbb{R} & & & \\ \overset{(4.2)}{\Rightarrow} & (x - ce_1)^*x \in \mathbb{R} & & & \\ \Rightarrow & x^*x - \overline{c}x_1 \in \mathbb{R} & & & \\ \Rightarrow & \overline{c}x_1 \in \mathbb{R} & & & (*) \\ \Rightarrow & c & = & \alpha \cdot x_1 \qquad \alpha \in \mathbb{R} & (**) \end{array}$$

Weiterhin gilt:

$$\begin{aligned} 0 &= 2u^*x - u^*u = u^*(2x - u) \\ &\overset{(4.2)}{=} (x - ce_1)^*(x + ce_1) \\ &= x^*x + x^*ce_1 - \overline{c}e_1^Tx - \overline{c}ce_1^Te_1 \\ &= x^*x + c\overline{x}_1 - \underbrace{\overline{c}x_1}_{\substack{\in\mathbb{R}\text{, nach } (**) \\ \Rightarrow \overline{c}x_1 = c\overline{x_1}}} - |c|^2 \\ &= \|x\|_2^2 - |c|^2 \end{aligned}$$

Zusammen mit (**) gilt:

$$\|x\|_2 = |c| \overset{(**)}{=} |\alpha||x_1|$$

Nun folgt:

$$|\alpha| = \frac{\|x\|_2}{|x_1|} \quad \text{also} \quad \alpha = \pm\frac{\|x\|_2}{|x_1|} \in \mathbb{R}$$

Daher ergeben sich als Lösung:

$$c \overset{(**)}{=} \pm x_1 \cdot \frac{\|x\|_2}{|x|} \quad \text{und} \quad u = x \mp x_1 \cdot \|x\|_2 \frac{1}{|x_1|} \cdot e_1.$$

QED

Die numerisch stabilere der beiden $\pm$ Variante ist $u = x + x_1 \cdot \frac{1}{|x_1|} \cdot \|x_2\| \cdot e_1$, weil es dann in der ersten Koordinate von u zu keiner Auslöschung kommen kann. Dieses funktioniert sogar, falls $x = \alpha \cdot e_1$.

Beispiel: Sei $x = \binom{3i}{4}$ gegeben. Gesucht sind nun $u(H)$ und c, sodass folgende Bedingungen erfüllt sind:

$$Hx = x - \frac{2}{u^*u} uu^*x = \begin{pmatrix} c \\ 0 \end{pmatrix}$$
$$u = \begin{pmatrix} u_1 + \tilde{u}_1 i \\ u_2 + \tilde{u}_2 i \end{pmatrix}$$

Es gilt:

$$2u^*x = 6\tilde{u}_1 + 8u_2 + i \cdot (6u_1 - 8\tilde{u}_2) \qquad u^*u = u_1^2 + \tilde{u}_2^2 + u_2^2\tilde{u}_2^2$$

Wir machen nun eine Fallunterscheidung. Im ersten Fall ist u reell, im zweiten Fall komplex. Es ergibt sich:

1. $2u^*x = u^*u$:
   (a) $6u_1 - 8\tilde{u}_2 = 0$
   (b) $6\tilde{u}_1 + 8u_2 = u_1^2 + \tilde{u}_1^2 u_2^2 + \tilde{u}_2$
2. $\binom{3i}{4} - \binom{c_1 + \tilde{c}_1 i}{0} = \binom{u_1 + \tilde{u}_1 i}{u_2 + \tilde{u}_2 i}$
   (a) $-c_1 = u_1$
   (b) $3 - \tilde{c}_1 = \tilde{u}_1$
   (c) $4 = u_2$
   (d) $0 = \tilde{u}_2$

Aus 2(a) - 2(d) folgen folgende Werte:

$$\tilde{u}_2 = 0 \qquad u_2 = 4 \qquad \text{aus 2(a) folgt } u_1 = 0$$

Außerdem gilt:

$$6\tilde{u}_1^2 + 8 \cdot 4 = 0 + \tilde{u}_1^2 + 16 + 0 \quad \Rightarrow \tilde{u}_1^2 = \begin{cases} 8 \\ -2 \end{cases}$$

Daraus ergeben sich für u die Werte:

$$u = \begin{pmatrix} 8i \\ 4 \end{pmatrix} \quad \text{oder} \quad u = \begin{pmatrix} -2i \\ 4 \end{pmatrix}$$

Nach Lemma 4.6 gilt nun:

$$u = x \pm x_1 \cdot \frac{\|x\|_2}{|x_1|} \cdot e_1 = \begin{pmatrix} 3i \\ 4 \end{pmatrix} \pm 3i \cdot \frac{5}{3} = \begin{pmatrix} 8i \\ 4 \end{pmatrix} \text{ oder } \begin{pmatrix} -2i \\ 4 \end{pmatrix}$$

Im ersten Fall erhält man die folgende Householder-Matrix:

$$\begin{aligned} H = I - 2\frac{uu^*}{u^*u} &= I - \frac{2}{80} \cdot \begin{pmatrix} 64 & 32i \\ -32i & 16 \end{pmatrix} \\ &= \frac{1}{10} \cdot \left[\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} - \begin{pmatrix} 16 & 8i \\ -8i & 4 \end{pmatrix} \right] = \frac{1}{5} \cdot \begin{pmatrix} -3 & 4i \\ -4i & 3 \end{pmatrix} \end{aligned}$$

Außerdem gilt:

$$Hx = \begin{pmatrix} -5i \\ 0 \end{pmatrix}$$

Satz 4.7 *Für eine Matrix $A \in \mathbb{K}^{m,n}$ mit dem Rang n (also $m \geq n$) existiert eine QR-Zerlegung.*

Beweis: Die Idee ist, in Lemma 4.6 $x = A_1$ zu wählen und anschließend eine Householder-Matrix $H^{(1)}$ so zu bestimmen, dass die Gleichung $H^{(1)}x = \alpha \cdot e_1$ erfüllt ist. Es muss also gelten:

$$H^{(1)}A = \left(\begin{array}{c|c} \alpha_1 & \\ \hline 0 & \\ \vdots & * \\ 0 & \end{array} \right).$$

In dieser Darstellung ist die erste Spalte schon korrekt, der Rest interessiert uns noch nicht. Durch weitere Iteration ergeben sich dann die restlichen Spalten der Matrix. Seien bereits nach k Schritten die unitären Matrizen $H^{(1)}, \ldots H^{(1)}$ so bestimmt, dass für $A^{(k)}$ gilt ($A \in \mathbb{K}^{m,n}$):

$$A^{(k)} = H^{(k)} \cdot \ldots \cdot H^{(1)} \cdot A = \left(\begin{array}{cc|c} * & & \\ & \ddots & B^{(k)} \\ & * & \\ \hline 0 & & C^{(k)} \end{array} \right) \Bigg\} k$$

wobei $B^{(k)} \in \mathbb{K}^{k,m-k}$ und $C^{(k)} \in \mathbb{K}^{n-k,m-k}$ und $a_{ij}^{(k)} = 0$ für alle $j \leq k$ und $i > j$ gilt. Falls $k = n$ ist, so ist $A^{(n)}$ mit $A^{(n)} = H^{(n)} \cdot \ldots \cdot H^{(1)} \cdot A$ eine obere Dreiecksmatrix und $Q = H^{(n)} \cdot \ldots \cdot H^{(1)}$ unitär. Ansonsten sei nun $\tilde{x}^{(k+1)} = C_1 \in \mathbb{K}^{m-k}$. Zunächst gilt $\tilde{x}^{(k+1)} \neq 0$, denn wären die ersten $k+1$ Spalten linear abhängig, dann wäre der Rang von $A^{(k)}$ nicht n. Aufgrund der Regularität von

$H^{(i)}$ wäre in diesem Fall auch der Rang von A ungleich n. Nun benötigen wir einige Definitionen:

$$\tilde{u}^{(k+1)} := \tilde{x}^{(k+1)} + \frac{\tilde{x}_1^{(k+1)}}{|\tilde{x}_1^{(k+1)}|} \left\|\tilde{x}^{(k+1)}\right\|_2 e_1$$
$$\tilde{H}^{(k+1)} := I_{m-k} - 2\frac{\tilde{u}^{(k+1)}(\tilde{u}^{(k+1)})^*}{(\tilde{u}^{(k+1)})^*\tilde{u}^{(k+1)}} \qquad \text{wie in Lemma 4.6}$$

Durch diese Definitionen folgt:

$$\tilde{H}^{(k+1)}C^{(k)} = \left(\begin{array}{c|c} \alpha & \\ 0 & * \\ \vdots & \\ 0 & \end{array}\right) \quad \text{mit } \alpha = \frac{-\tilde{x}_1^{(k+1)}}{|\tilde{x}_1^{(k+1)}|}\|\tilde{x}^{(k+1)}\|_2$$

Um aber die Transformation auf ganz $A^{(k)}$ statt nur auf $C^{(k)}$ anwenden zu können, definieren wir nun

$$u^{(k+1)} := \left(\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline \tilde{u}^{(k+1)} \end{array}\right) \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} k \\ \left.\begin{array}{l} \end{array}\right\} m-k \end{array}$$

$$H^{(k+1)} := I_m - 2\frac{u^{(k+1)}(u^{(k+1)})^*}{(u^{(k+1)})^*u^{(k+1)}} = \left(\begin{array}{c|c} I_k & 0 \\ \hline 0 & \tilde{H}^{(k+1)} \end{array}\right)$$

wobei $\tilde{H}^{(k+1)}$ eine $m-k \times n-k$-Matrix ist. Mit diesen Definitionen erhält man jetzt für die Matrix $A^{(k+1)}$:

$$A^{(k+1)} = H^{(k)} \cdot A^{(k)} = \left(\begin{array}{ccc|c} * & & & \\ & \ddots & & B^{(k)} \\ & & * & \\ \hline 0 & & & \tilde{H}^{(k+1)}C^{(k)} \end{array}\right)$$

mit der geforderten Eigenschaft, dass $a_{ij}^{(k+1)} = 0$ für alle $j \leq k+1$ und $i > j$. QE D

Algorithmus 6: QR-Verfahren (Matrixversion)

Input: $A \in \mathbb{K}^{m,n}$ `mit` $\text{Rang}(A) = n$

Schritt 1: $A^{(0)} := A$

Schritt 2: **For** $k = 1, \ldots, n$ **do**

$$\begin{aligned} d &:= a_{kk}^{(k-1)} + a_{kk}^{(k-1)} \frac{1}{|a_{kk}^{(k-1)}|} \sqrt{\sum_{i=k}^{m} \left| a_{ik}^{(k-1)} \right|^2} \\ u^{(k)} &:= (0, \ldots 0, d, a_{k+1,k}^{(k-1)}, \ldots, a_{mk}^{(k-1)})^T \\ H^{(k)} &:= I_m - 2 \frac{u^{(k)} (u^{(k)})^*}{(u^{(k)})^* u^{(k)}} \\ A^{(k)} &:= H^{(k)} A^{(k-1)} \end{aligned}$$

Ergebnis: QR-`Zerlegung mit`

$$\begin{aligned} A &:= QR \texttt{ mit} \\ R &:= A^{(n)} \texttt{ ist obere Dreiecksmatrix und} \\ Q &:= H^{(1)} \cdot \ldots \cdot \ldots \cdot H^{(n)} \texttt{ ist unitäre Matrix} \end{aligned}$$

Diese Berechnungen sind aufwändig, insbesondere die Anwendung von von $H^{(k)}$ auf alle Spalten von $A^{(k)}$. Wir suchen nun eine bessere Rechenvorschrift für die Berechnung von dem Produkt Hv der Householder-Matrix H und einem beliebigen Vektor $v \in \mathbb{K}^m$. Nun gelten:

$$H = I - \frac{2}{u^* u} uu^* = I - \beta uu^* \text{ mit } u = x + \frac{x_1}{|x_1|} \|x\|_2 e_1 \text{ und}$$

$$\beta = \frac{2}{u^* u} = \frac{1}{\|x\|_2 (\|x\|_2 + |x_1|)} \tag{4.3}$$

Dann folgt:

$$\begin{aligned} Hv = (I - \beta uu^*)v &= v - \beta uu^* v \\ &= v - \beta u^* v u = v - su \qquad (4.4) \\ \text{mit } s &= \beta u^* v \in \mathbb{K} \qquad (4.5) \end{aligned}$$

Damit kann man HA_k für die Spalten A_k von A also effizient berechnen. Bei betragsmäßig kleinem β ergeben sich allerdings numerische Probleme. Sie kann man mit Hilfe des folgenden Lemmas vermeiden.

Lemma 4.8 *Sei H eine Householder-Matrix aus Lemma 4.6 zu $x \neq 0$ und H' die Householder-Matrix aus Lemma 4.6 zu $y = \alpha x$ mit $\alpha \in \mathbb{R}^+$ und $x, y \in \mathbb{K}^n$. Dann gilt $H = H'$.*

Beweis: Zunächst gelten folgende Gleichungen:

$$u = x + \frac{x_1}{|x_1|}\|x\|_2 e_1 \qquad H = I - \frac{2}{u^* u} u u^*$$

$$u' = y + \frac{y_1}{|y_1|}\|y\|_2 e_1 = \alpha x + \frac{\alpha x_1}{|\alpha x_1|}\|\alpha x_1\|_2 = \alpha u$$

Nun folgt daraus

$$H' = I - \frac{2}{(u')^* u'} u'(u')^* = I - \frac{2}{\alpha^2 u^* u}\alpha^2 u u^* = H$$

QED

Im folgenden Algorithmus wird statt x also $\frac{x}{\|x\|_\infty}$ verwendet, um die numerische Stabilität von β aus (4.3) zu gewährleisten.

Algorithmus 7: QR-Verfahren (Implementations-Variante)

Input: $A \in \mathbb{K}^{m,n}$ `mit` $\mathrm{Rang}(A) = n$

Schritt 1: $u_{ik} := 0$ `für alle` $i = 1, \ldots, m$ `und` $k = 1, \ldots, n$

Schritt 2: **For** $k = 1, \ldots, n$ **do**

Schritt 2.1: ${A_k}^{\max} := \max_{i=k,\ldots,m} |a_{ik}|$

Schritt 2.2: $\alpha := 0$

Schritt 2.3: `For` $i = k, \ldots, m$ `do`

Schritt 2.3.1. $u_{ik} := \frac{a_{ik}}{{A_k}^{max}}$ `(Normierung der k-ten Restspalte)`

Schritt 2.3.2. $\alpha := \alpha + |u_{ik}|^2$ `(Norm`^{2} `der k-ten Restspalte)`

Schritt 2.4: $\alpha := \sqrt{\alpha}$

Schritt 2.5: $\beta_k := \frac{1}{\alpha(\alpha + |u_{kk}|)}$ (β `aus 4.3)`

Schritt 2.6: $u_{kk} = u_{kk} + \frac{a_{kk}}{|a_{kk}|} \cdot \alpha$ `(1. Komponente von` u_k `nach Lemma 4.6)`

Schritt 2.7: $a_{kk} := -\frac{a_{kk}}{|a_{kk}|} \cdot {A_k}^{\max}$

Schritt 2.8: `For` $i = k+1, \ldots m$ `do` $a_{ik} := 0$ `(erste Spalte von` $H{A_k}^{\max}$`)`

Schritt 2.9: `For` $j = k+1, \ldots, n$ `do`

Schritt 2.9.1. $s := \beta_k \sum_{i=k}^{m} \overline{u_{ik}} a_{ij}$ `(s aus 4.5)`

Schritt 2.9.2. `For` $i = k, \ldots, m$ `do` $a_{ij} = a_{ij} - s \cdot u_{ik}$ `(Berechnung von H` $A_j^{(k)}$ `nach 4.4)`

Ergebnis: QR-Zerlegung der Originalmatrix A mit

$$\begin{aligned} R &:= A \\ Q &:= H^{(1)} \cdot \ldots \cdot H^{(n)} \text{ mit } H^{(k)} = I - \frac{2}{u_k^* u_k} u_k u_k^* \text{ und} \\ & \qquad u_k = \begin{pmatrix} u_{1k} \\ \vdots \\ u_{nk} \end{pmatrix} \text{ für } k = 1, \ldots, n \end{aligned}$$

Bemerkung:

- oft benötigt man Q nicht explizit und kann sich die Berechnung sparen

- R enthält viele Nullen, die man zum Speichern (eines Teils) der u_{ik} verwenden kann, genauer s. Beweis von Satz 4.7, und man hat nur die Diagonale extra zu speichern

$$\left(\begin{array}{c|c|ccc} * & * & * & * & * \\ \hline 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{array}\right)$$

- U ist untere Dreiecksmatrix

Aufwand: Die QR-Zerlegung einer Matrix $A \in \mathbb{K}^{m,n}$ mit $\text{Rang}(A) = n$ erfordert

$$n^2(m - \frac{1}{3}n) + O(mn) (= \mathrm{O}(n^2 m))$$

wesentliche Operationen.

Der teuerste Schritt ist 9.2 mit

$$\begin{aligned}
\left(\sum_{k=1}^{n}\sum_{j=k+1}^{n} 2(m-k)\right) + O(mn) &= \sum_{k=1}^{n} 2(n-k)(m-k) + O(mn) \\
&= \left(\sum_{k=1}^{n} 2nm - 2nk - 2km + 2k^2\right) + O(mn) \\
&= 2n^2m + 2\sum_{k=1}^{n} k^2 - k(n+m) + O(mn) \\
&= 2n^2m + 2\underbrace{\frac{n(n+1)(2n+1)}{6}}_{\sum k^2} - 2(n+m)\underbrace{\frac{n(n+1)}{2}}_{\sum k} + O(mn) \\
&= 2n^2m + 2\frac{2n^3}{6} - n^2(n+m) + \underbrace{O(mn) + O(n^2)}_{=O(mn)} \\
&= n^2\left(m - \frac{1}{3}n\right) + O(mn)
\end{aligned}$$

Für $m = n$ ergibt sich also $\frac{2}{3}n^3 + O(n^2)$, das ist etwas höher als der Aufwand von $\frac{1}{3}n^3$ bei dem Gauss-Verfahren. Für schlecht konditionierte Matrizen lohnt es sich aber, diesen Aufwand in Kauf zu nehmen.

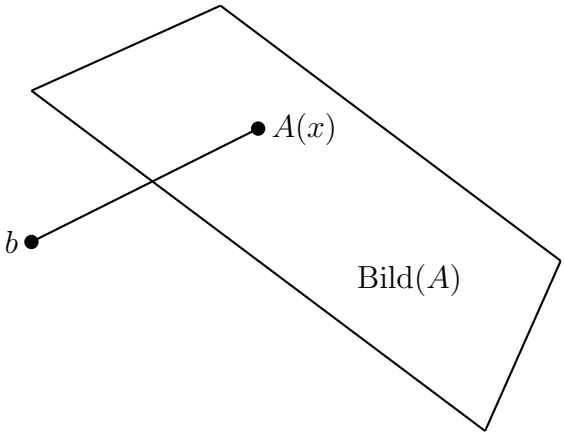
Bemerkung: Im Gegensatz zur LU-Zerlegung ist bei der QR-Zerlegung (bei $\mathrm{Rang}(A) = n$) keine Pivotisierung nötig. Das gilt allerdings nicht im Fall $\mathrm{Rang}(A) < n$. In diesem Fall tauscht man in jedem Schritt k vor der Berechnung der u_k und der β_k die Restspalte mit größter Euklidischer Norm an die k-te Position.

4.2 Lineare Ausgleichsprobleme

Beim Lösen linearer Gleichungssysteme bestand die Aufgabe darin, ein x zu finden, so dass $Ax = b$ gilt. Was passiert aber nun, wenn $Ax = b$ nicht lösbar ist? In diesem Fall versucht man, ein x zu finden, so dass der Ausdruck Ax die rechte Seite b möglichst gut annähert. Verwendet man zur Bewertung der Qualität der Annäherung die Euklidische Norm, führt das zu dem Minimierungsproblem

$$\min_{x \in \mathbb{K}^n} \|Ax - b\|_2,$$

in dem man unter allen Vektoren $x \in \mathbb{K}^n$ den sucht, der die Euklidische Norm von $Ax - b$ minimiert.



Da es äquivalent ist, statt $\|Ax-b\|_2$ die quadratische Funktion $\|Ax-b\|_2^2$, zu minimieren, definieren wir das **lineare Ausgleichsproblem** wie folgt:

(AuP) $\min_{x\in\mathbb{K}^n} F(x)$ mit $F(x)=\|Ax-b\|_2^2$, $A\in\mathbb{K}^{m,n}$, $b\in\mathbb{K}^m$

Beispiel:

$$A=\begin{pmatrix}2 & 1\\ 1 & -1\\ 1 & 1\end{pmatrix} \qquad b=\begin{pmatrix}1\\0\\1\end{pmatrix}$$

Zwei mögliche Lösungen für x werden im folgenden untersucht:

$$x=\begin{pmatrix}\frac{1}{3}\\ \frac{1}{3}\end{pmatrix} \text{ Lösung bzgl. } \begin{pmatrix}2 & 1\\ 1 & -1\end{pmatrix}=b \Rightarrow F(x)=\left\|\begin{pmatrix}1 & 0 & \frac{2}{3}\end{pmatrix}^T-b\right\|_2=(\tfrac{1}{3})^2$$

$$x=\begin{pmatrix}\frac{1}{2}\\ \frac{1}{2}\end{pmatrix} \text{ Lösung bzgl. } \begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}=b \Rightarrow F(x)=\left\|\begin{pmatrix}\frac{3}{2} & 0 & 1\end{pmatrix}^T-b\right\|_2=(\tfrac{1}{2})^2$$

Im folgenden wollen wir untersuchen, wie man die beste Lösung für solche Ausgleichsprobleme findet.

Satz 4.9 *Sei $A\in\mathbb{K}^{m,n}$, $b\in\mathbb{K}^m$. Dann gilt*

1. *(AuP) ist lösbar*
2. *$x\in\mathbb{K}^n$ ist Lösung von (AuP) genau dann, wenn*

$$A^*Ax=A^*b \tag{N}$$

   Man sagt ***„x löst die Normalengleichung (N) bezüglich A und b.“***

3. *(AuP) ist eindeutig lösbar genau dann wenn* $\mathrm{Rang}(A)=n$

Beweis:

1. Sei (x_k) eine so genannte Minimalfolge, d.h.

$$\|Ax_k - b\|_2 \to \alpha := \inf_{x \in \mathbb{K}^n} \|Ax - b\|_2$$

$\alpha > 0$ (Falls $\alpha = 0$ wäre $Ax = b$ und das Gleichungssystem wäre lösbar.)

Ist k groß genug, so gilt $\|Ax_k - b\|_2 < 2\alpha$, d.h.

$$\|Ax_k\|_2 = \|Ax_k - b + b\|_2 \leq \|Ax_k - b\|_2 + \|b\|_2 < 2\alpha + \|b\|_2.$$

Also ist die Folge $(Ax_k) \subseteq \text{Bild}(A)$ beschränkt.

Aufgrund der Stetigkeit von A ist $\text{Bild}(A)$ abgeschlossen. Vom Satz von Bolzano-Weierstrass wissen wir daher, dass es eine konvergente Teilfolge von (Ax_k) gibt, die gegen $\tilde{y} \in \text{Bild}(A)$ konvergiert. Also gibt es $\tilde{x}$ mit $A\tilde{x} = \tilde{y}$ und daher gilt

$$\|A\tilde{x} - b\|_2 = \|\tilde{y} - b\|_2 = \inf_{x \in \mathbb{K}^n} \|Ax - b\|_2,$$

also ist $\tilde{x}$ Lösung des Ausgleichsproblems.

2. Zunächst erinnern wir daran, dass

$$\begin{aligned} \text{Bild}(A^*) &= \{A^*z : z \in \mathbb{K}^m\} \\ \text{Bild}(A^*A) &= \{A^*Ax : x \in \mathbb{K}^n\}. \end{aligned}$$

Aus der linearen Algebra wissen wir $\text{Bild}(A^*) = \text{Bild}(A^*A)$, woraus folgt $A^*b \in \text{Bild}(A^*) \Rightarrow A^*b \in \text{Bild}(A^*A)$. Daher existiert

$$x_0 \in \mathbb{K}^n \text{ mit } A^*Ax_0 = A^*b. \tag{4.6}$$

Für jede Lösung x_0 von (N) und für jedes $x \in \mathbb{K}^n$ gilt

$$\begin{aligned} F(x) - F(x_0) &= \|Ax - b\|_2^2 - \|Ax_0 - b\|_2^2 \\ &= (Ax - b)^*(Ax - b) - (Ax_0 - b)^*(Ax_0 - b) \\ &= x^*A^*Ax - x^*A^*b - b^*Ax + b^*b \\ &\quad - x_0^*A^*Ax_0 + x_0^*A^*b + b^*Ax_0 - b^*b \end{aligned}$$

Weil x_0 (N) erfüllt ist, gilt $A^*b = A^*Ax_0$ bzw. $b^*A = x_0^*A^*A$. Unter Verwendung dieser Gleichungen erhält man weiter

$$\begin{aligned} &= x^*A^*Ax - x^*A^*Ax_0 - x_0^*A^*Ax \\ &\quad - x_0^*A^*Ax_0 + x_0^*A^*Ax_0 + x_0^*A^*Ax_0 \\ &= x^*A^*Ax - x^*A^*Ax_0 - x_0^*A^*Ax + x_0^*A^*Ax_0 \\ &= (x - x_0)^*A^*A(x - x_0) \\ &= \|A(x - x_0)\|_2^2 \geq 0 \end{aligned} \tag{4.7}$$

"$\Rightarrow$„ Sei x_0 eine Lösung von (N). Dann folgt $F(x) \geq F(x_0)$, $\forall x \in \mathbb{K}^n$, also ist x_0 eine Lösung von (AuP).

"$\Leftarrow$„ Sei andererseits x Lösung von (AuP). Nach (4.6) können wir x_0 als Lösung von (N) wählen, d.h. $A^*Ax_0 = A^*b$. Es folgt $F(x)-F(x_0) \geq 0$ wegen (4.7). Andererseits gilt $F(x) \leq F(x_0)$, weil x eine Lösung von (AuP) ist. Zusammen folgt $F(x) = F(x_0)$ und daraus $\|A(x-x_0)\|_2^2 = 0 = \|A(x-x_0)\|$ wegen (4.7). Nach dem ersten Normaxiom haben wir dann $A(x - x_0) = 0$ und entsprechend $A^*Ax = A^*Ax_0 = A^*b$, also löst x auch (N).

3. Falls $\text{Rang}(A) = n$ ist $A^*A \in \mathbb{K}^{n,n}$ regulär. Also ist $A^*Ax = A^*b$ eindeutig lösbar. Ist $\text{Rang}(A) < n$, so existiert wegen (4.6) eine Lösung x_0 von (AuP) sowie ein $z \in \text{Kern}(A)$ mit $z \neq 0$. Damit gilt:

$$F(x_0 + z) = \|A(x_0 + z) - b\|_2^2 = \|Ax_0 + Az - b\|_2^2 = \|Ax_0 - b\|_2^2 = F(x_0)$$

und $x_0 + z \neq x_0$, also gibt es zwei verschiedene Lösungen von (AuP)

Bemerkung: Ist die Lösung von (AuP) nicht eindeutig, so lässt sich aber aus

$$\text{Opt}^* = \{x \in \mathbb{K}^n : x \text{ ist Lösung von (AuP)}\}$$

ein eindeutiges $\tilde{x}$ mit minimaler Euklidischer Norm $\|\tilde{x}\|_2$ wählen. D.h.

$$\min_{x \in \text{Opt}^*} \|x\|_2$$

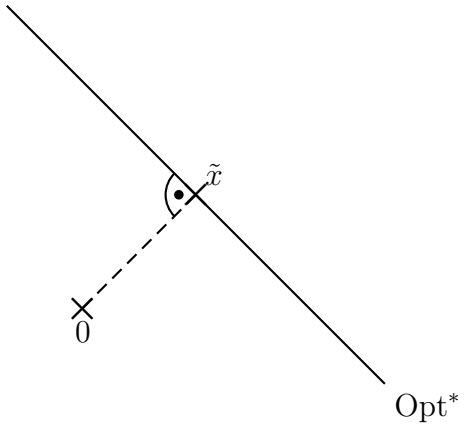
ist eindeutig lösbar.

Beweis: $\text{Opt}^* = \{x \in \mathbb{K}^n : A^*Ax = A^*b\}$ ist ein affin linearer Teilraum. Dieser enthält genau ein Element mit minimaler Euklidischer Länge, nämlich die orthogonale Projektion von 0 auf Opt^* QED

Aufgabe: Sei $L \in \mathbb{R}^n$ ein affin linearer Teilraum und $a \in \mathbb{R}^n$. Zeigen Sie, dass das Minimierungsproblem

$$\min_{x \in L} \|a - x\|$$

eindeutig lösbar ist, und zwar von der orthogonalen Projektion von a auf L.



Lösung des Ausgleichproblems

Idee 1 Nutze Kriterium 2 aus Satz 4.9 und löse das Gleichungssystem $A^*Ax = A^*b$ durch Cholesky. Das ist schnell, aber oft ungenau.

Idee 2 Führe QR-Zerlegung von A durch. Man erhält

$$A = QR = Q\left(\frac{\hat{R}}{0}\right), \qquad \hat{R} \text{ obere Dreiecksmatrix}$$

Ist $\text{Rang}(A) = n$, so ist $\hat{R}$ regulär. Es gilt

$$\begin{aligned} \|Ax - b\|_2^2 = \|QRx - b\|_2^2 &= \|Q^*(QRx - b)\| \text{ nach (Lemma 3.29)} \\ &= \|Rx - Q^*b\|_2^2 \\ &= \| \left(\frac{\hat{R}x}{0}\right) - \left(\frac{c}{d}\right) \|_2^2 = \|\hat{R}x - c\|_2^2 + \|d\|_2^2 \end{aligned} \tag{4.8}$$

wobei $\left(\frac{c}{d}\right) = Q^*b$ eine Zerlegung des Vektors $Q^*b \in \mathbb{K}^m$ in $c \in \mathbb{K}^n$, $d \in \mathbb{K}^{m-n}$ ist.

Lemma 4.10 *Sei $\|Ax - b\|_2^2 \to \min$ ein lineares Ausgleichsproblem mit $A \in \mathbb{K}^{m,n}, m \geq n$ und $\text{Rang}(A) = n$, und $A = QR$ eine QR-Zerlegung von A,*

$$R = \left(\frac{\hat{R}}{0}\right) \text{ und } Q^*b = \left(\frac{c}{d}\right)$$

Dann ist

$$x = \hat{R}^{-1}c$$

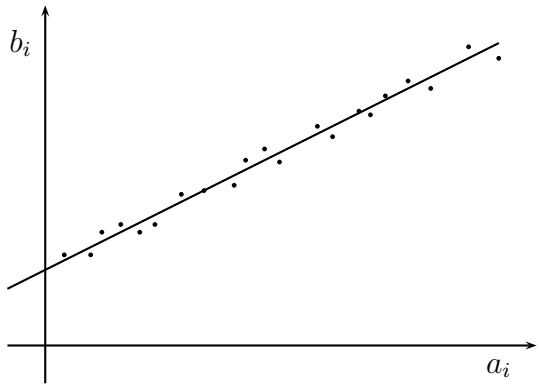
die eindeutige Lösung von (AuP) und $\|d\|_2^2$ der zugehörige Zielfunktionswert.

Beweis: Nach (4.8) wissen wir, dass $\|Ax-b\|_2^2 = \|\hat{R}x-c\|_2^2+\|d\|_2^2$. Dieser Ausdruck wird minimal, falls $c = \hat{R}x$. Da $\hat{R}$ regulär, existiert so ein x, nämlich $\hat{R}^{-1}c$. Die Zielfunktion ergibt sich als

$$\|Ax - b\|_2^2 = 0 + \|d\|_2^2$$

also $\|Ax - b\|^2 = \|d\|_2$. QED

Anwendungsbeispiel (Statistik): Es seien Messdaten (a_i, b_i) mit $i = 1, \ldots, m$ gegeben, bei denen ein (unbekannter) linearer Zusammenhang besteht.



Gesucht sind die Parameter α, β, die diesen linearen Zusammenhang beschreiben. Dabei soll b_i durch die Gleichung $\alpha a_i + \beta$ in Abhängigkeit von a_i möglichst gut geschätzt werden können, d.h. $\alpha a_i + \beta$ soll möglichst nahe an b_i sein. Wir wollen die Qualität dieser Schätzung maximieren und versuchen dazu, die Summe aller quadrierten Schätzfehler zu minimieren. Das führt auf das folgende Problem:

$$\min_{\alpha,\beta} \sum_{i=1}^{m} |\alpha a_i + \beta b_i|^2.$$

Mit

$$A = \begin{pmatrix} a_1 & 1 \\ \vdots & \\ a_n & 1 \end{pmatrix} \qquad x = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \qquad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

erhält man

$$\|Ax - b\|_2^2 = \| \begin{pmatrix} a_1\alpha + \beta - b_1 \\ \vdots \\ a_n\alpha + \beta - b_n \end{pmatrix} \|_2^2 = \sum_{i=1}^{n} |\alpha a_i + \beta - b_i|^2,$$

also ein lineares Ausgleichsproblem. Man nennt dies "Methode der kleinsten Quadrate".

4.3 Singulärwertzerlegung

In diesem Abschnitt beschäftigen wir uns mit Orthogonalisierungsverfahren für nicht quadratische Matrizen. Sei $A = (a_{ij}) \in \mathbb{K}^{m,n}$ eine solche Matrix. Wir bezeichnen A als **Diagonalmatrix** falls $a_{ij} = 0$ für alle $i \neq j$ mit $i \in \{1, \ldots, m\}$ und $j \in \{1, \ldots, n\}$. Mit dieser Bezeichnung führen wir den Begriff der Singulärwertzerlegung ein.

Definition 4.11 *Sei $A \in \mathbb{K}^{m,n}$. Eine Zerlegung der Form $A = U\Sigma V^*$ mit unitären Matrizen $U \in \mathbb{K}^{m,m}$ und $V \in \mathbb{K}^{n,n}$ und einer Diagonalmatrix $\Sigma \in \mathbb{K}^{m,n}$ heißt eine* Singulärwertzerlegung *von A.*

Wir benutzen die Dimensionsformel: Für $A \in \mathbb{K}^{m,n}$

$$n = \mathrm{Rang}(A) + \dim(\mathrm{Kern}(A)) \text{ und } \mathrm{Kern}(A) = \mathrm{Kern}(A^*A)$$

sowie die folgende Aussage aus der linearen Algebra.

Lemma 4.12 *Sei $A \in \mathbb{K}^{m,n}$. Dann gelten*

$$\begin{aligned} \mathrm{Kern}(A) &= \mathrm{Kern}(A^*A) \\ \mathrm{Rang}(A) &= \mathrm{Rang}(A^*) = \mathrm{Rang}(A^*A) = \mathrm{Rang}(AA^*) \end{aligned}$$

Satz 4.13 *Jede Matrix $A \in \mathbb{K}^{m,n}$ besitzt eine Singulärwertzerlegung.*

Beweis: Seien $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ die Eigenwerte von A^*A mit zugehörigen Eigenvektoren $v_1, \ldots, v_n$, so dass

$$A^*Av_j = \lambda_j v_j \text{ und } v_j^* v_k = \begin{cases} 1 & \text{falls } j = k \\ 0 & \text{falls } j \neq k \end{cases}$$

Sei $\mathrm{Rang}(A^*A) = r$. Dann sind genau r der Eigenwerte positiv und die restlichen Null. Weil ebenso $r = \mathrm{Rang}(AA^*)$, hat also auch AA^* genau r positive Eigenwerte. Definiere

$$\sigma_j = \sqrt{\lambda_j} \quad \text{und} \quad u_j = \frac{1}{\sigma_j} Av_j \qquad 1 \leq j \leq r \tag{4.9}$$

Dann gilt

$$AA^*u_j = \frac{1}{\sigma_j} A \underbrace{A^*Av_j}_{\lambda_j v_j} = \frac{1}{\sigma_j} \lambda_j Av_j = \lambda_j u_j \qquad 1 \leq j \leq r$$

$$u_j^* u_k = \frac{1}{\overline{\sigma}_j \sigma_k} v_j^* \underbrace{A^*Av_k}_{\lambda_k v_k} = \frac{\lambda_k}{\sigma_j \sigma_k} v_j^* v_k = \begin{cases} 1 & \text{falls } j = k \\ 0 & \text{sonst} \end{cases}$$

Also sind $u_1, \ldots, u_r$ ein Orthonormalsystem von Eigenvektoren zu den Eigenwerten $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$ der Matrix AA^*. Ergänze $\{u_1, \ldots, u_r\}$ zu einer Orthonormalbasis $\{u_1, \ldots, u_m\}$ aus Eigenvektoren und setze

$$V := (v_1, \ldots, v_n) \in \mathbb{K}^{n,n} \quad \text{und} \quad U := (u_1, \ldots, u_m) \in \mathbb{K}^{m,m}$$

Dann gilt

$$Av_j = \begin{cases} \sigma_j u_j & \text{für} \quad 1 \leq j \leq r \text{ wegen (4.9)} \\ 0 & \text{für} \quad r+1 \leq j \leq n \text{ weil } v_j \in \operatorname{Kern}(A^*A) = \operatorname{Kern}(A) \end{cases}$$

Also erhält man

$$AV = U\Sigma \text{ mit } \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_r, \underbrace{0, \ldots, 0}_{\min\{n,m\}-r}) \in \mathbb{K}^{m,n},$$

wobei die Matrizen V, U unitär sind, weil ihre Spalten jeweils also orthonormal zueinander konstruiert wurden. Es folgt $A = U\Sigma V^*$. QED

Bemerkung:

- Die Einträge von Σ sind eindeutig, wenn man Positivität verlangt.
- Ist A selber quadratisch und hermitesch, dann gilt $\sigma_j = |\mu_j|$ wenn μ_j Eigenwert von A ist.

Definition 4.14 *Die positiven Werte $\sigma_j > 0$, die in der Singulärwertzerlegung der Matrix A aus Satz 4.13 auftreten, heißen* Singulärwerte *von A.*

Eine effiziente Berechnung der Singulärwertzerlegung wird im Kapitel über Eigenwerte und Eigenvektoren besprochen.

4.4 Anwendung der Singulärwertzerlegung auf lineare Ausgleichsprobleme

Bisher hatten wir zwei Methoden kennen gelernt, um lineare Ausgleichsprobleme zu lösen. In diesem Anschnitt kommt eine weitere — nämlich durch Anwendung der Singulärwertzerlegung — dazu.

(AuP) $\min \|Ax - b\|_2^2$

Methode 1 Löse die Normalengleichung $A^*Ax = A^*b$ durch das Cholesky-Verfahren.

Methode 2 Bestimme eine QR-Zerlegung von A und löse

$$\|Ax - b\|_2^2 = \|\hat{R}x - c\|_2^2 + \|d\|_2^2$$

Methode 3 Die dritte Methode beruht auf der Singulärwertzerlegung und wird im folgenden erläutert.

Sei $A = U\Sigma V^*$ eine Singulärwertzerlegung von A. Setze $y := V^*x \in \mathbb{K}^n$, $c := U^*b \in \mathbb{K}^m$. Es folgt

$$\begin{aligned}
\|Ax-b\|_2^2 &= \|U\Sigma V^*x - UU^*b\|_2^2 \\
&= \|U(\Sigma y - c)\|_2^2 \\
&= \|\Sigma y - c\|_2^2 \qquad \text{nach Lemma 4.1 weil } U \text{ unitär} \\
&= \|(\sigma_1 y_1, \sigma_2 y_2, \ldots, \sigma_r y_r, 0, \ldots 0)^T - c\|_2^2 \\
&= \sum_{j=1}^{r} (\sigma_j y_j - c_j)^2 + \sum_{j=r+1}^{m} c_j^2
\end{aligned}$$

Satz 4.15 *Eine Lösung des linearen Ausgleichsproblems (AuP) ist gegeben durch*

$$x = \sum_{j=1}^{r} \frac{c_j}{\sigma_j} V_j + \sum_{j=r+1}^{m} \alpha_j V_j,$$

wobei $A = U\Sigma V^*$, σ_j die Singulärwerte, V_j die Spalten von V, und $c = U^*b$ sind. Die α_j können beliebig gewählt werden. Für $\alpha_j = 0$ erhält man die Lösung von (AuP) mit minimaler Euklidischer Norm.

Beweis: Um $\|Ax-b\|_2^2$ zu minimieren, minimieren wir

$$\sum_{j=1}^{r} (\sigma_j \underbrace{y_j}_{\text{variabel}} - c_j)^2 + \sum_{j=r+1}^{m} c_j^2$$

Also wähle für beliebiges α_j, $j = r+1, \ldots, n$

$$y_j = \begin{cases} \frac{c_j}{\sigma_j} & \text{für } j = 1, \ldots, r \\ \alpha_j & \text{für } j = r+1, \ldots, n \end{cases}$$

x ergibt sich dann aus

$$x = Vy = \sum_{j=1}^{n} V_j y_j = \sum_{j=1}^{r} \frac{c_j}{\sigma_j} V_j + \sum_{j=r+1}^{m} \alpha_j V_j.$$

Die Norm von x berechnet man durch

$$\|x\|_2^2 = x^*x = \cdots = \left\| \sum_{j=1}^{r} \frac{c_j}{\sigma_j} V_j \right\|_2^2 + \sum_{j=r+1}^{m} \alpha_j^2$$

und dieser Ausdruck ist minimal für $\alpha_j = 0$, $j = r+1, \ldots, m$. QED

Zum Abschluss vergleichen wir die drei besprochenen Methoden. Sei eine Matrix A gegeben mit $\text{Rang}(A) = n$ und Singulärwerten $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$. Wir untersuchen die Kondition der drei möglichen Verfahren für das lineare Ausgleichsproblem.

Cholesky Löse $A^*Ax = A^*b$.

$$\begin{aligned}\text{cond}(A^*A) &= \|A^*A\|_2 \cdot \|(A^*A)^{-1}\|_2 \\ \|A^*A\|_2 &= \sqrt{\rho((A^*A)^*(A^*A))} = \sqrt{\rho(A^*AA^*A)} \\ &= \sqrt{\rho((A^*A)^2)} = \sqrt{\lambda_1^2} = \sigma_1^2,\end{aligned}$$

denn für eine beliebige Matrix B folgt aus $Bx = \lambda x$ dass $B^2x = \lambda^2 x$. Weiter gilt:

$$\begin{aligned}\|(A^*A)^{-1}\| &= \sqrt{\rho(((A^*A)^{-1}(A^*A)^{-1}))} = \sqrt{\rho(((A^*A)^2)^{-1})} \\ &= \sqrt{\frac{1}{\lambda_n^2}} = \frac{1}{\sigma_n^2},\end{aligned}$$

denn aus $Bx = \lambda x$ folgt $B^{-1}x = \frac{1}{\lambda}x$ und außerdem gilt $(B^2)^{-1} = (B^{-1})^2$. Zusammen erhalten wir

$$\text{cond}(A^*A) = \frac{\sigma_1^2}{\sigma_n^2} = \left(\frac{\sigma_1}{\sigma_n}\right)^2$$

Singulärwertzerlegung Löse $\Sigma y = c$ und $x = Vy$ (mit orthogonaler Matrix V)

$$\text{cond}(\Sigma) = \|\Sigma\|_2\|\Sigma^{-1}\|_2 = \frac{\sigma_1}{\sigma_n}$$

QR-Zerlegung Löse $\hat{R}x = Q^*b$

$$\text{cond}(R) = \|R\|_2\|R^{-1}\|_2$$

Weil $A^*A = (QR)^*(QR) = R^*Q^*QR = R^*R$ ist der größte (bzw. kleinste) Eigenwert von A^*A auch der größte (bzw. kleinste) Eigenwert von R^*R, also folgen

$$\begin{aligned}\|R\|_2 &= \sqrt{\lambda_1} = \sigma_1 \\ \|R^{-1}\|_2 &= \frac{1}{\sqrt{\lambda_n}} = \frac{1}{\sigma_n},\end{aligned}$$

weil $(R^{-1})^*R^{-1} = (RR^*)^{-1}$ Inverses von RR^* mit Eigenwerten $\lambda_1, \ldots, \lambda_n$. Also

$$\text{cond}(R) = \frac{\sigma_1}{\sigma_n}$$

Die Kondition der Cholesky-Zerlegung ist also das Quadrat der Kondition aus QR-Verfahren oder Singulärwertzerlegung.

Kapitel 5

Iterationsverfahren

5.1 Das Verfahren der sukzessiven Approximation

In diesem Kapitel betrachten wir nach den Eliminationsverfahren und den Orthogonalisierungsverfahren noch eine dritte Klasse von Verfahren, die man zur Lösung von linearen (und nichtlinearen) Gleichungssystemen verwenden kann, so genannte *iterative Verfahren*. Wir betrachten dazu gleich relativ allgemein Funktionen $f_1, \ldots, f_m$ mit

$$f_i : \mathbb{R}^n \to \mathbb{R},\ i = 1, \ldots, m$$

und bezeichnen das System

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0 \\ f_2(x_1, \ldots, x_n) &= 0 \\ \vdots &\quad \vdots \\ f_m(x_1, \ldots, x_n) &= 0 \end{aligned} \tag{5.1}$$

als **nichtlineares Gleichungssystem** mit den Variablen $x_1, \ldots, x_n$. Definiert man

$$F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

so kann man das Gleichungssystem in Kurzform auch als

$$F(x) = 0$$

schreiben.

Gilt für $x \in \mathbb{R}^n$, dass $F(x) = 0$, so nennt man x eine Lösung des Gleichungssystems. Dass wir in dem Gleichungssystem die rechte Seite zu Null gesetzt

haben, ist keine Einschränkung, weil man ein Gleichungssystem $F(x) = b$ mit $b = (b_1, \ldots, b_m)^T \in \mathbb{R}^m$ jederzeit zu $G(x) = F(x) - b = 0$ umformen kann.

Nichtlineare Gleichungssysteme lassen sich im Allgemeinen nicht durch algebraische Manipulationen exakt auflösen. Wir betrachten in diesem Kapitel daher *iterative Verfahren* bzw. *Iterationsverfahren*, die eine gegebene Lösung in jedem Schritt verbessern, bis eine vorgegebene Genauigkeit erreicht ist. Dazu betrachten wir Gleichungssysteme $F(x) = 0$, die als **Fixpunktgleichung** vorliegen.

Definition 5.1 *Sei $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ eine Funktion. Die Gleichung*

$$\Phi(x) = x$$

wird als **Fixpunktgleichung** *betrachtet. Jedes $x \in \mathbb{R}^n$, für das $\Phi(x) = x$ gilt, wird als* **Fixpunkt** *von Φ bezeichnet.*

Der Zusammenhang zwischen Fixpunktgleichungen und linearen Gleichungssystemen wird im folgenden Lemma beschrieben.

Lemma 5.2

1. *Sei $m \leq n$ und das Gleichungssystem $F(x) = 0$ wie in (5.1) gegeben. Sei $M : \mathbb{R}^m \to \mathbb{R}^n$ eine lineare, injektive Abbildung. Definiere*

$$\Phi(x) = M(F(x)) + x. \tag{5.2}$$

   Dann ist x ein Fixpunkt von Φ genau dann, wenn x das Gleichungssystem löst. Die Fixpunktgleichung $\Phi(x) = x$ ist also äquivalent zu dem Gleichungssystem $F(x) = 0$.

2. *Sei andererseits die Abbildung $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ gegeben. Definiere*

$$F(x) = \Phi(x) - x.$$

   Dann ist das Gleichungssystem $F(x) = 0$ äquivalent zu der Fixpunktgleichung $\Phi(x) = x$.

Beweis:

ad 1: Es gilt $\Phi(x) = x \iff M(F(x)) = 0$. Wegen der Injektivität der linearen Abbildung M ist das genau dann der Fall, wenn $F(x) = 0$ ist.

ad 2: Es gilt $F(x) = 0 \iff \Phi(x) = x$. QED

Gleichungssystem $F(x) = 0$ mit $m \leq n$ können wir also lösen, wenn wir Fixpunkte bestimmen können. Damit werden wir uns im folgenden beschäftigen. (Ausgleichsprobleme mit $m > n$ behandeln wir später.)

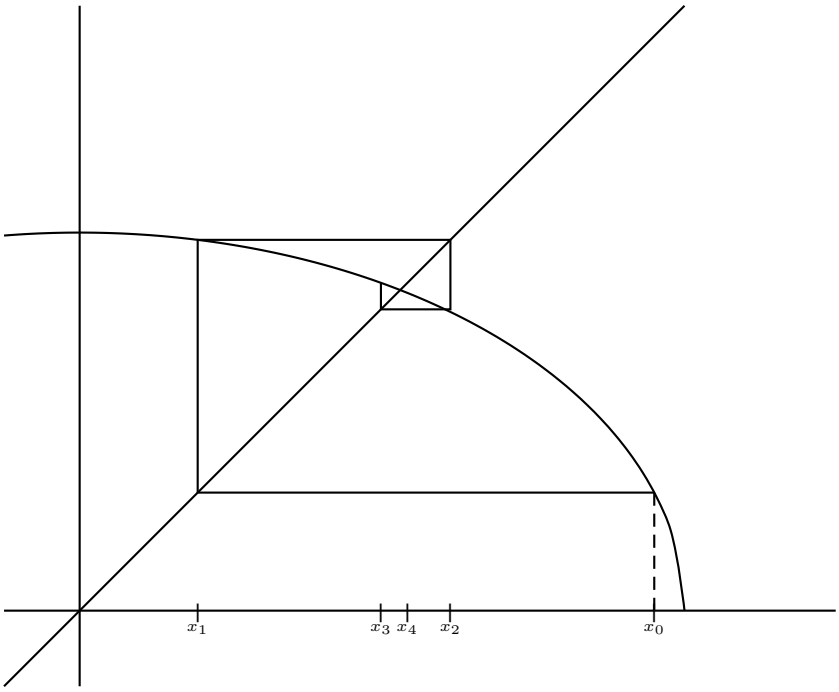
Die Idee der sukzessiven Approximation ist nun die folgende. Man betrachtet für ein gegebenes $x^{(0)}$ die Folge

$$x^{(k+1)} = \Phi(x^{(k)}),\ k = 0, 1, 2, \ldots.$$

Angenommen, Φ ist stetig und die Folge der $x^{(k)}$ konvergiert. Dann gibt es einen Grenzwert

$$y = \lim_{k \to \infty} x^{(k)},$$

für den gilt $y = \Phi(y)$, y ist also ein Fixpunkt von Φ.



Definition 5.3 *Die Iterationsvorschrift*

$$x^{(k+1)} = \Phi(x^{(k)})$$

nennt man **Verfahren der sukzessiven Approximation**.

Als Beispiel betrachten wir die Gleichung

$$f(x) = 2x - \tan(x) = 0.$$

Wir schreiben die Gleichung als Fixpunktgleichung um.

- In der ersten Variante schreiben wir

$$\phi(x) = f(x) + x = 3x - \tan(x)$$

und suchen einen Fixpunkt von ϕ mittels der Folge

$$x^{(k+1)} = 3x^{(k)} - \tan(x^{(k)})$$

- In einer zweiten Variante schreiben wir

$$x = \frac{\tan(x)}{2}$$

und erhalten die Folge

$$x^{(k+1)} = \frac{1}{2}\tan(x^{(k)}).$$

- Als drittes verwenden wir

$$x = \arctan(2x),$$

was zu der Folge

$$x^{(k+1)} = \arctan(2x^{(k)})$$

führt.

Implementiert man in allen drei Fällen das Verfahren der sukzessiven Iteration so ergeben sich unterschiedliche Verhalten der drei Formeln: Zum Beispiel für den Startwert 1.2 geht Iterationsvorschrift 1 gegen unendlich, Iterationsvorschrift 2 gegen Null und Iterationsvorschrift 3 gegen 1.1656...

Im folgenden wollen wir untersuchen, wann solche Iterationsvorschriften konvergieren. Zunächst beweisen wir den folgenden Satz für *skalare* Funktionen $f : \mathbb{R}^n \to \mathbb{R}^n$.

Satz 5.4 *Sei $I \subseteq \mathbb{R}$ ein abgeschlossenes Intervall, $q \in [0, 1)$ und $\phi : I \to I$ eine Funktion, die für alle $x, y \in I$*

$$|\phi(x) - \phi(y)| \leq q|x - y| \tag{5.3}$$

erfüllt. Besitzt ϕ einen Fixpunkt $x^* \in I$, so konvergiert die Folge

$$x^{(k+1)} = \phi(x^{(k)}), k = 0, 1, \ldots$$

für jeden Startwert $x^{(0)}$ gegen x^* und es gilt

$$|x^{(k)} - x^*| \leq q^k|x^{(0)} - x^*| \text{ für } k = 0, 1, 2, \ldots$$

Beweis: Zunächst ist die Iterationsformel für $x^{(k)}$ ist wohldefiniert, weil $x^k \in I$ für alle k. Die Aussage lässt sich dann für alle $k \in \mathbb{N}_0$ durch Induktion zeigen. Der Induktionsanfang für $k = 0$ ist klar. Für den Induktionsschritt $k \to k+1$ rechnet man

$$\begin{aligned} |x^{(k+1)} - x^*| &= |\phi(x^{(k)}) - \phi(x^*)| \leq q|x^{(k)} - x^*| \text{ wegen (5.3)} \\ &\leq qq^k|x^{(0)} - x^*| \text{ wegen der Induktionsannahme} \\ &= q^{(k+1)}|x^{(0)} - x^*| \end{aligned}$$

QED

Mit Hilfe dieses Satzes können wir erklären, warum die dritte Iterationsformel $x^{(k+1)} = \arctan(2x^{(k)})$ in unserem Beispiel für jeden Startwert $x^{(0)} \in I = [1, \infty)$ konvergiert:

- Dazu überlegt man zunächst, dass $\phi(x) \in [1, \infty) = I$ für alle $x \in I$, denn $\phi(1) > 1$ und ϕ ist monoton wachsend.
- Weiterhin besitzt ϕ einen Fixpunkt, denn die Funktion $f(x) = x - \phi(x)$ erfüllt $f(1) < 0$ und $f(x) \to \infty$ für $x \to \infty$. Also hat f nach dem Zwischenwertsatz eine Nullstelle in I und entsprechend hat ϕ in dem Intervall einen Fixpunkt.
- Jetzt muss noch die Kontraktions-Voraussetzung (5.3) nachgewiesen werden. Dazu verwenden wir den Mittelwertsatz, von dem wir wissen, dass für jedes $x, y \in I$ eine Zwischenstelle $\epsilon \in (x, y)$ existiert, so dass

  $$\phi(x) - \phi(y) = \phi'(\epsilon)(x - y).$$

  Kann man nun zeigen, dass $\phi'(\epsilon) \leq q < 1$ für alle $\epsilon \in I$ so ist die Kontraktionsbedingung (5.3) erfüllt. In unserem Beispiel rechnet man nach, dass

  $$\phi'(x) = \frac{2}{1 + 4x^2} \leq \phi'(1) = \frac{2}{5} < 1,$$

  weil ϕ' monoton fallend ist.

Also sind die Voraussetzungen von Satz 5.4 erfüllt und die Konvergenz der Iterationsformel ist bewiesen.

5.2 Der Banach'sche Fixpunktsatz

In diesem Abschnitt werden wir die Konvergenzeigenschaften der sukzessiven Approximation weiter untersuchen. Unser Ziel ist eine Verallgemeinerung von Satz 5.4 aus dem letzten Abschnitt, bei der wir die Existenz eines Fixpunktes

nicht voraussetzen müssen. Außerdem gelingt es, den neuen Satz nicht nur für skalare Funktionen ϕ sondern für Operatoren Φ in beliebigen Banach-Räume X zu zeigen - d.h. die Unbekannte $x \in X$ kann nicht nur ein Vektor, sondern sogar eine Funktion sein.

Wir erinnern zunächst daran, dass jeder vollständige und normierte Raum ein **Banach-Raum** ist, d.h. also dass in einem Banach-Raum jede Cauchy-Folge konvergiert. Weiterhin übertragen wir (5.3) aus dem letzten Abschnitt auf normierte Räume.

Definition 5.5 *Sei X ein Banach-Raum mit Norm $\|\cdot\|$ und $U \subseteq X$ eine abgeschlossene Teilmenge von X. Eine Abbildung $\Phi : U \to X$ heißt* **kontrahierend**, *falls es einen reellen Kontraktionsfaktor $q < 1$ gibt, so dass*

$$\|\Phi(x) - \Phi(y)\| \leq q\|x - y\| \text{ für alle } x, y \in U.$$

Wir können nun den Banach'schen Fixpunktsatz formulieren und beweisen.

Satz 5.6 (Banach'scher Fixpunktsatz) *Sei X ein Banach-Raum mit Norm $\|\cdot\|$ und $U \subseteq X$ eine abgeschlossene Teilmenge von X. Sei weiterhin $\Phi : U \to U$ eine kontrahierende Abbildung mit Kontraktionsfaktor $q < 1$. Dann gilt:*

1. *Φ besitzt einen eindeutig bestimmten Fixpunkt x^*.*
2. *Die Iterationsvorschrift der sukzessiven Approximation $x^{(k+1)} = \Phi(x^{(k)})$, $k = 0, 1, \ldots$ konvergiert gegen x^* für jeden Startwert $x^{(0)} \in U$.*
3. *Es gilt die* **a priori Fehlerschranke**

$$\|x^{(k)} - x^*\| \leq \frac{q^k}{1-q}\|x^{(1)} - x^{(0)}\| \quad \text{für alle } k = 1, 2 \ldots \tag{5.4}$$

4. *Es gilt die* **a posteriori Fehlerschranke**

$$\|x^{(k)} - x^*\| \leq \frac{q}{1-q}\|x^{(k)} - x^{(k-1)}\| \quad \text{für alle } k = 1, 2 \ldots \tag{5.5}$$

Beweis: Zunächst ist die Folge $x^{(k)}$ wohldefiniert weil $x^{(k)} \in U$ für alle $k \in \mathbb{N}_0$. Für den Beweis nutzen wir aus, dass in einem Banach-Raum alle Cauchy-Folgen konvergieren und zeigen daher als erstes, dass $x^{(k)}$ eine Cauchy-Folge ist.

Schritt 1: $x^{(k)}$ ist eine Cauchy-Folge: Es gilt

$$\begin{aligned} \|x^{(k)} - x^{(k-1)}\| &= \|\Phi(x^{(k-1)}) - \Phi(x^{(k-2)})\| \\ &\leq q\|x^{(k-1)} - x^{(k-2)}\| \leq \ \ldots \ \leq \\ &\leq q^j\|x^{(k-j)} - x^{(k-j-1)}\| \end{aligned} \tag{5.6}$$

für alle natürlichen Zahlen j mit $0 \leq j \leq k-1$. Mit Hilfe dieser Ungleichung rechnet man nun nach, dass

$$\begin{aligned}
\|x^{(l)} - x^{(k)}\| &\leq \|x^{(l)} - x^{(l-1)}\| + \|x^{(l-1)} - x^{(l-2)}\| + \ldots + \|x^{(k+1)} - x^{(k)}\| \\
&\leq q^{l-k}\|x^{(k)} - x^{(k-1)}\| + q^{l-k-1}\|x^{(k)} - x^{(k-1)}\| + \ldots \\
&\qquad \ldots \quad + q\|x^{(k)} - x^{(k-1)}\| \\
&= \|x^{(k)} - x^{(k-1)}\| \sum_{j=1}^{l-k} q^j \\
&\leq \|x^{(k)} - x^{(k-1)}\| \sum_{j=1}^{\infty} q^j \\
&= \|x^{(k)} - x^{(k-1)}\| \frac{q}{1-q} \qquad (5.7) \\
&\leq q^{k-1}\|x^{(1)} - x^{(0)}\| \frac{q}{1-q} = \frac{q^k}{1-q}\|x^{(1)} - x^{(0)}\|. \qquad (5.8)
\end{aligned}$$

Weil $\frac{q^k}{1-q} \to 0$ für $k \to \infty$ ist $x^{(k)}$ also eine Cauchy-Folge.

Schritt 2: Existenz des Fixpunktes. Weil $x^{(k)}$ eine Cauchy-Folge ist, gibt es $x^* = \lim_{k\to\infty} x^{(k)}$. Für x^* gilt dann

$$\|\Phi(x^*) - \Phi(x^{(k)})\| \leq q\|x^* - x^{(k)}\| \to 0 \text{ für } k \to \infty,$$

entsprechend haben wir

$$\Phi(x^*) = \lim_{k\to\infty} \Phi(x^{(k)}) = \lim_{k\to\infty} x^{(k+1)} = x^*$$

Schritt 3: Eindeutigkeit des Fixpunktes. Angenommen, $\tilde{x}$ sei ein weiterer Fixpunkt von Φ. Dann gilt

$$\|x^* - \tilde{x}\| = \|\Phi(x^*) - \Phi(\tilde{x})\| \leq q\|x^* - \tilde{x}\|.$$

Weil $q < 1$ folgt daraus, dass $\|x^* - \tilde{x}\| = 0$, also $x^* = \tilde{x}$.

Schritt 4: Fehlerschranken. Wir nutzen die in Schritt 1 aufgestellte Ungleichungskette für

$$\begin{aligned}
\|x^* - x^{(k)}\| &= \lim_{l\to\infty} \|x^{(l)} - x^{(k)}\| \\
&\leq \|x^{(k)} - x^{(k-1)}\| \frac{q}{1-q} \text{ wegen (5.7)} \\
&\leq \frac{q^k}{1-q}\|x^{(1)} - x^{(0)}\| \text{ wegen (5.8).}
\end{aligned}$$

Damit ist der Satz gezeigt. QED

Zum Nachweis der Kontraktion verallgemeinern wir noch das bereits für skalare Funktionen verwendete Kriterium auf den $\mathbb{R}^n$. Dabei bezeichnen wir für eine Funktion $F : \mathbb{R}^n \to \mathbb{R}^m$ die Jacobi-Matrix von F an der Stelle $x \in \mathbb{R}^n$ mit $DF(x)$, das heißt

$$DF(x) = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \cdots & \frac{\partial F_m}{\partial x_n} \end{pmatrix}.$$

Für $F : \mathbb{R}^n \to \mathbb{R}$ bezeichnen wir den Tangentialvektor $DF(x)$ auch einfach mit $f'(x)$.

Lemma 5.7 *Sei $U \subseteq \mathbb{R}^n$ eine konvexe Menge und $\Phi : U \to \mathbb{R}^n$ stetig differenzierbar mit $\|D\Phi(x)\| \leq q < 1$ für alle $x \in U$ (wobei die Matrixnorm $\|\cdot\|$ die der Vektornorm zugeordnete Norm sein soll). Dann ist Φ kontrahierend mit Kontraktionsfaktor q.*

Beweis: Seien $x, y \in U$. Wir definieren eine Abbildung $f : \mathbb{R} \to \mathbb{R}^n$ durch

$$f(t) = \Phi(x + t(y - x)) \quad \text{für } t \in [0, 1].$$

Dann gilt

$$\begin{aligned} \|\Phi(y) - \Phi(x)\| &= \|f(1) - f(0)\| = \| \int_0^1 f'(t)dt\| \\ &\qquad \text{nach dem Hauptsatz der Differential- und Integralrechnung} \\ &= \| \int_0^1 D\Phi(x + t(y - x))(y - x)dt\| \\ &\qquad \text{nach der multivariaten Kettenregel} \\ &\leq \int_0^1 \|D\Phi(x + t(y - x))\| \, \|y - x\|dt \\ &\leq \|y - x\| \int_0^1 q \; dt = q\|x - y\|. \end{aligned}$$

QED

Abschließend untersuchen wir noch, wie wir bei der Approximation des Fixpunktes eine Genauigkeit von ε garantieren können. Wir wollen also erreichen, dass

$$\|x^{(k)} - x^*\| \leq \varepsilon$$

wenn k die Iteration ist, bei der wir abbrechen. Dazu können wir sowohl die a-priori als auch die a-posteriori Schranke aus Satz 5.6 nutzen.
Die a-priori Schranke sagt, dass

$$\|x^{(k)} - x^*\| \leq \frac{q^k}{1 - q}\|x^{(1)} - x^{(0)}\| \quad \text{für alle } k = 1, 2 \ldots.$$

$\|x^{(k)} - x^*\| \leq \varepsilon$ ist also gewährleistet, falls

$$\frac{q^k}{1-q}\|x^{(1)} - x^{(0)}\| \leq \varepsilon,$$

und das lässt sich auflösen zu

$$k \geq \frac{\ln\left(\frac{(1-q)\varepsilon}{\|x^{(1)}-x^{(0)}\|}\right)}{\ln(q)}.$$

Ist der Kontraktionsfaktor q also klein, werden weniger Iterationsschritte benötigt als für einen großen Kontraktionsfaktor q.

Um während des Verfahrens ein Abbruchkriterium zu haben, nutzt man dagegen oft die (schärfere) a posteriori Fehlerschranke aus Satz 5.6, die besagt, dass

$$\|x^{(k)} - x^*\| \leq \frac{q}{1-q}\|x^{(k)} - x^{(k-1)}\| \quad \text{für alle } k = 1, 2 \ldots$$

und zu dem Abbruchkriterium

$$\frac{q}{1-q}\|x^{(k)} - x^{(k-1)}\| \leq \varepsilon$$

führt. Leider ist der Kontraktionsfaktor q oft nicht bekannt. In diesen Fällen behilft man sich mit folgender Abschätzung von q durch $\hat{q}_k$:

$$\hat{q}_k := \frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k-1)} - x^{(k-2)}\|}.$$

Es gilt $\hat{q}_k \leq q$, denn

$$\hat{q}_k = \frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k-1)} - x^{(k-2)}\|} = \frac{\|\Phi(x^{(k-1)}) - \Phi(x^{(k-2)})\|}{\|x^{(k-1)} - x^{(k-2)}\|} \leq q.$$

Algorithmus 8: Sukzessive Approximation mit heuristischem Abbruchkriterium

Input: `abgeschlossene Menge` $U \subseteq \mathbb{R}^n$, `Kontraktion` $\Phi : U \to U$, `Startwert` $x^{(0)} \in U$, `Toleranzwert` ε.

Schritt 1: $x^{(1)} := \Phi(x^{(0)})$

Schritt 2: $k := 1$

Schritt 3: `Repeat`

Schritt 3.1: $k := k + 1$

Schritt 3.2: $x^{(k)} := \Phi\left(x^{(k-1)}\right)$

Schritt 3.3: $q_k := \frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k-1)} - x^{(k-2)}\|}$

Schritt 3.4: `If` $q_k \geq 1$ `STOP: Φ ist keine Kontraktion.`

Until $\frac{q_k}{1-q_k}\|x^{(k)} - x^{(k-1)}\| \leq \varepsilon$

Ergebnis: `approximierter Fixpunkt` $x^* = x^{(k)}$

Bemerkung: Die Fehlerschranke $\|x^{(k)} - x^*\| \leq \varepsilon$ kann nicht garantiert werden, weil wir nur wissen, dass

$$\frac{q_k}{1 - q_k} \leq \frac{q}{1 - q}.$$

Meistens konvergiert q_k aber gegen q, so dass das Abbruchkriterium in der Regel ausreichend gut funktioniert.

5.3 Iterative Verfahren für lineare Gleichungssysteme

Wir wenden nun den Banach'schen Fixpunktsatz auf lineare Operatoren an. Zunächst halten wir uns weiter in Banach-Räumen auf, kommen dann aber zur Lösung linearer Gleichungssysteme (also zum endlichdimensionalen Fall) zurück.

Satz 5.8 *Sei $B : X \to X$ ein linearer beschränkter Operator in einem Banach-Raum $(X, \|\cdot\|)$ mit $\|B\| < 1$ in der der Norm des Banach-Raumes zugeordneten Matrixnorm. Dann gilt*

1. *Der Operator $I - B$ ist invertierbar, das heißt das System $x - Bx = b$ hat genau eine Lösung x^* für jedes $b \in X$.*

2. *Der inverse Operator* $(I-B)^{-1}$ *ist beschränkt mit*

$$\|(I-B)^{-1}\| \leq \frac{1}{1-\|B\|}.$$

3. *Die Iterationsvorschrift der sukzessiven Approximation* $x^{(k+1)} = Bx^{(k)} + b,\ k = 0, 1, \ldots$ *konvergiert gegen* x^* *für jeden Startwert* $x^{(0)} \in X$.

4. *Es gilt die* **a priori Fehlerschranke**

$$\|x^{(k)} - x^*\| \leq \frac{\|B\|^k}{1-\|B\|}\|x^{(1)} - x^{(0)}\| \quad \textit{für alle } k = 1, 2 \ldots$$

5. *Es gilt die* **a posteriori Fehlerschranke**

$$\|x^{(k)} - x^*\| \leq \frac{\|B\|}{1-\|B\|}\|x^{(k)} - x^{(k-1)}\| \quad \textit{für alle } k = 1, 2 \ldots$$

Beweis: Sei $b \in X$ beliebig aber fest. Definiere den linearen Operator Φ punktweise durch

$$\Phi x := Bx + b \text{ für alle } x \in X$$

Wegen $\|\Phi x - \Phi\tilde{x}\| = \|B(x-\tilde{x})\| \leq \|B\|\|x-\tilde{x}\|$ ist Φ kontrahierend mit $q := \|B\| < 1$. Satz 5.6 ergibt damit direkt die folgenden Aussagen:

ad 1. Es existiert ein eindeutiger Fixpunkt x^*, der $\Phi x^* = x^*$ erfüllt. Weil

$$\Phi x = x \Longleftrightarrow Bx + b = x \Longleftrightarrow (I-B)x = b$$

gibt es also eine eindeutige Lösung von $x - Bx = b$ und $(I-B)$ ist invertierbar.

ad 3. $x^{(k+1)} = \Phi x^{(k)} = Bx^{(k)} + b$ konvergiert gegen x^* für jeden Startwert $x^{(0)}$.

ad 4. und 5. Hier folgt die Behauptung direkt mit $q := \|B\|$.

Als letzter Punkt bleibt noch die zweite Aussage zu zeigen, also die Beschränktheit der linearen Abbildung $(I-B)^{-1}$. Dazu definieren wir die Folge $x^{(k)}$ der sukzessiven Approximation mit Startwert $x^{(0)} := b$. Es ergibt sich

$$\begin{aligned}
x^{(0)} &= b \\
x^{(1)} &= Bx^{(0)} + b = Bb + b \\
x^{(2)} &= Bx^{(1)} + b = B^2b + Bb + b \\
&\vdots \\
x^{(k)} &= \sum_{j=0}^{k} B^j b.
\end{aligned}$$

Deswegen gilt

$$\|x^{(k)}\| \leq \sum_{j=0}^{k} \|B^j b\| \leq \|b\| \sum_{j=0}^{k} \|B\|^j \leq \frac{\|b\|}{1-\|B\|}$$

Weiter wissen wir von Aussage 3 und 1, dass $x^{(k)} \to x^* = (I-B)^{-1}b$. Daraus folgt, dass

$$\|(I-B)^{-1}b\| \leq \frac{\|b\|}{1-\|B\|}.$$

Weil b beliebig war, gilt diese Aussage für alle $b \in X$. Somit erhält man

$$\|(I-B)^{-1}\| = \sup_{b \in X} \frac{\|(I-B)^{-1}b\|}{\|b\|} \leq \frac{1}{1-\|B\|}.$$

QED

Im endlich-dimensionalen Fall sind alle Normen äquivalent, so dass aus der Konvergenz bezüglich einer Norm die Konvergenz in allen anderen Normen folgt. Da es unhandlich sein kann, das Kriterium für verschiedene Normen zu testen, wollen wir im folgenden ein notwendiges und hinreichendes Kriterium für die Konvergenz der sukzessiven Approximation herleiten. Dieses Kriterium wird über den Spektralradius $\rho(B)$ der obigen Matrix B formuliert werden. Um das Kriterium herleiten zu können, benötigen wir das folgende Resultat aus der Linearen Algebra.

Satz 5.9 (Lemma von Schur) *Sei $A \in \mathbb{K}^{n,n}$ eine Matrix. Dann gibt es eine unitäre Matrix Q so, dass*

$$Q^*AQ = R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ 0 & & & r_{nn} \end{pmatrix}$$

Mit Hilfe des Lemmas von Schur zeigen wir nun erst die folgende Aussage.

Lemma 5.10 *Sei $A \in \mathbb{K}^{n,n}$. Dann gilt $\rho(A) \leq \|A\|$. Andererseits gibt es zu jedem $\varepsilon > 0$ eine Norm $\|\cdot\|_\varepsilon$ auf $\mathbb{K}^n$ so dass*

$$\|A\|_\varepsilon \leq \rho(A) + \varepsilon.$$

Beweis: Zum Beweis des ersten Teils der Aussage wählen wir einen Eigenwert λ von A mit zugehörigem normierten Eigenvektor $u \in \mathbb{K}^n$. Dann gilt, dass

$$\|A\| = \sup_{x:\|x\|=1} \|Ax\| \geq \|Au\| = \|\lambda u\| = |\lambda|.$$

Das gilt für alle Eigenwerte λ, also auch für den betragsmäßig größten.

Sei nun $\varepsilon > 0$ gegeben. Wir können ohne Beschränkung der Allgemeinheit annehmen, dass A nicht die Nullmatrix ist. Wir werden nun die gesuchte Norm $\|\cdot\|_\varepsilon$ konstruieren.
Nach dem Lemma von Schur (Satz 5.9) finden wir eine unitäre Matrix Q, so dass

$$R := Q^*AQ = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ 0 & & & r_{nn} \end{pmatrix}$$

eine obere Dreiecksmatrix (nicht die Nullmatrix) ist. Zunächst beobachten wir, dass

$$\begin{aligned} \det(\lambda I - A) &= \det(Q^*)\det(\lambda I - A)\det(Q) = \det(Q^*(\lambda I - A)Q) \\ &= \det(\lambda I - Q^*AQ) = \det(\lambda I - R) \\ &= (\lambda - r_{11})(\lambda - r_{22})\dots(\lambda - r_{nn}), \end{aligned}$$

also sind die Eigenwerte von A als Nullstellen des charakteristischen Polynoms genau die Diagonalelemente von R. Man definiert nun

$$\begin{aligned} r &:= \max_{i,j} |r_{ij}| > 0 \\ \delta &:= \min\left\{1, \frac{\varepsilon}{(n-1)r}\right\}, \text{ und} \\ D &:= \operatorname{diag}(1, \delta, \delta^2, \dots, \delta^{n-1}) \end{aligned}$$

Weil $\delta > 0$ ist D invertierbar und $D^{-1} = \operatorname{diag}(1, \delta^{-1}, \delta^{-2}, \dots, \delta^{-(n-1)})$. Wir berechnen nun

$$\begin{aligned} C :&= D^{-1}RD \\ &= D^{-1}\begin{pmatrix} r_{11} & \delta r_{12} & \delta^2 r_{13} & \dots & \delta^{n-1} r_{1n} \\ & \delta r_{22} & \delta^2 r_{23} & \dots & \delta^{n-1} r_{2n} \\ & & \delta^2 r_{33} & \dots & \delta^{n-1} r_{3n} \\ & & & \ddots & \vdots \\ 0 & & & & \delta^{n-1} r_{nn} \end{pmatrix} = \begin{pmatrix} r_{11} & \delta r_{12} & \delta^2 r_{13} & \dots & \delta^{n-1} r_{1n} \\ & r_{22} & \delta r_{23} & \dots & \delta^{n-2} r_{2n} \\ & & r_{33} & \dots & \delta^{n-3} r_{3n} \\ & & & \ddots & \vdots \\ 0 & & & & r_{nn} \end{pmatrix}. \end{aligned}$$

Satz 3.17 (siehe Seite 55) liefert, dass

$$\begin{aligned} \|C\|_\infty &= \max_{i=1,\dots,n} \sum_{j=1}^{n} c_{ij} \\ &\le \max_{i=1,\dots,n} r_{ii} + \delta r(n-1) \text{ weil } \delta^j \le \delta \\ &\le \rho(A) + \frac{\varepsilon}{(n-1)r} r(n-1) = \rho(A) + \varepsilon \end{aligned}$$

Setze

$$V := QD$$

und definiere damit

$$\|x\|_\varepsilon := \|V^{-1}x\|_\infty.$$

Das ist eine Norm, da V regulär. Um zu zeigen, dass diese Norm die gewünschte Eigenschaft hat, bemerken wir zunächst, dass

$$V^{-1}AV = D^{-1}Q^*AQD = D^{-1}RD = C$$

gilt. Damit erhält man schließlich

$$\begin{aligned} \|Ax\|_\varepsilon &= \|V^{-1}Ax\|_\infty \\ &= \|CV^{-1}x\|_\infty \\ &\leq \|C\|_\infty \|V^{-1}x\|_\infty \\ &= \|C\|_\infty \|x\|_\varepsilon, \end{aligned}$$

also ist

$$\|A\|_\varepsilon = \sup_{x\in\mathbb{K}^n} \frac{\|Ax\|_\varepsilon}{\|x\|_\varepsilon} \leq \|C\|_\infty = \rho(A) + \varepsilon.$$

QED

Nun können wir (endlich!) das angekündigte Kriterium für die Konvergenz der sukzessiven Approximation formulieren.

Satz 5.11 *Sei $B \in \mathbb{K}^{n,n}$. Die Folge $x^{(k+1)} = Bx^{(k)} + b$ mit $k = 0, 1, 2, \ldots$ konvergiert für jedes $b \in \mathbb{K}^n$ und jeden Startwert $x^{(0)} \in \mathbb{K}^n$ genau dann wenn $\rho(B) < 1$.*

Beweis:

„$\Longleftarrow$“: Sei $\rho(B) < 1$. Nach Lemma 5.10 existiert eine Norm $\|\cdot\|_\varepsilon$ so dass $\|B\|_\varepsilon \leq \rho(B) + \varepsilon$ für jedes $\varepsilon > 0$. Wähle ε nun so, dass

$$\rho(B) + \varepsilon < 1,$$

dann konvergiert $x^{(k)}$ bezüglich der Norm $\|\cdot\|_\varepsilon$. Da in $\mathbb{K}^n$ alle Normen äquivalent (Satz 3.9) sind, folgt die Konvergenz in jeder Norm.

„$\Longrightarrow$“: Angenommen, $\rho(B) \geq 1$. Dann gibt es einen Eigenwert $\lambda \geq 1$ und einen zugehörigen Eigenvektor $v \neq 0$. Starte das Verfahren der sukzessiven Approximation für $b = v$ mit dem Startvektor $x^{(0)} = v$. Man erhält

$$\begin{aligned} x^{(0)} &= v \\ x^{(1)} &= Bv + v = \lambda v + v \\ x^{(2)} &= B(\lambda v + v) + v = \lambda^2 v + \lambda v + v \\ &\vdots \\ x^{(k)} &= \left(\sum_{j=0}^{k} \lambda^j\right) v \to \infty \quad \text{weil } \lambda \geq 1. \end{aligned}$$

Also konvergiert die Folge $x^{(k)}$ in diesem Fall nicht. QED

Wir möchten die Ergebnisse nun konkret auf die Lösung linearer Gleichungssysteme anwenden. Sei also ein lineares Gleichungssystem

$$Ax = b$$

mit $A \in \mathbb{K}^{n,n}, b \in \mathbb{K}^n$ gegeben. Wir bringen das Gleichungssystem mit Hilfe einer regulären Matrix M in Fixpunktform und erhalten die äquivalente Fixpunktgleichung

$$x + M^{-1}(b - Ax) = x,$$

die zur sukzessiven Approximation

$$x^{(k+1)} = x^{(k)} + M^{-1}(b - Ax^{(k)}) \text{ beziehungsweise } M(x^{(k+1)} - x^{(k)}) = b - Ax^{(k)}$$

führt. Numerisch kann man $x^{(k+1)}$ in jeder Iteration durch das sukzessive Lösen der beiden Systeme

$$Mw^{(k+1)} = b - Ax^{(k)} \text{ und } x^{(k+1)} = x^{(k)} + w^{(k+1)} \tag{5.9}$$

ermitteln. Allerdings macht das nur Sinn, wenn man eine Matrix M wählt, die gewährleistet, dass das System (5.9) effizient lösbar ist.

Wie soll man also M wählen? Nach Satz 5.11 konvergiert die Folge $x^{(k+1)} = Bx^{(k)} + b$ genau dann, wenn $\rho(B) < 1$. Weil in unserem Fall

$$x^{(k+1)} = (I - M^{-1}A)x^{(k)} + M^{-1}b$$

muss also $\rho(I - M^{-1}A) < 1$ gelten. Schreibt man $M = N + A$ (mit $N = M - A$) so ergibt sich, dass

$$I - M^{-1}A = I - M^{-1}(M - N) = I - M^{-1}M + M^{-1}N = M^{-1}N,$$

also die Bedingung, dass $\rho(M^{-1}N) < 1$ gelten soll.

Für die folgenden Verfahren zerlegen wir die gegebenen Matrix A in

$$A = A_D + A_L + A_R,$$

wobei $A_D = \text{diag}(a_{11}, a_{22}, \ldots, a_{nn})$ und

$$A_L = \begin{pmatrix} 0 & \ldots & 0 \\ & \ddots & \vdots \\ a_{ij} & & 0 \end{pmatrix}, \text{ und } A_R = \begin{pmatrix} 0 & & a_{ij} \\ \vdots & \ddots & \\ 0 & \ldots & 0 \end{pmatrix}$$

den Anteil des unteren und oberen Dreiecks aus A beinhalten. Weiterhin setzen wir voraus, dass (eventuell nach Pivotisierungs-Schritten) die Inverse

$$A_D^{-1} \text{ existiert,} \tag{5.10}$$

d.h. dass alle Elemente der Hauptdiagonalen von A nicht Null sind. (Wie wir von Gauss-Verfahren wissen, lässt sich das bei regulären Matrizen immer erreichen.) Wir betrachten nun zunächst zwei vom Konzept her sehr ähnliche Verfahren, das *Gesamtschritt-Verfahren* und das *Einzelschritt-Verfahren.*

Gesamtschritt - oder Jacobi-Verfahren

Im so genannten Gesamtschritt-Verfahren (GSV) wählt man die nach unserer Voraussetzung (5.10) reguläre Matrix $M = A_D$. Als Fixpunktgleichung erhält man

$$\begin{aligned} x &= x + A_D^{-1}(b - Ax) = x - A_D^{-1}Ax + A_D^{-1}b \\ &= -A_D^{-1}(A - A_D)x + A_D^{-1}b \\ &= -A_D^{-1}(A_L + A_R)x + A_D^{-1}b \end{aligned} \tag{5.11}$$

Das Verfahren der sukzessiven Approximation ergibt sich folglich zu

$$x^{(k+1)} = -A_D^{-1}(A_L + A_R)x^{(k)} + A_D^{-1}b,\ k = 0, 1, 2, \ldots$$

mit der Iterationsmatrix

$$B = I - A_D^{-1}A = -A_D^{-1}(A_L + A_R) \tag{5.12}$$

Komponentenweise kann man schreiben

$$x_i^{(k+1)} = -\sum_{j \in \{1,\ldots,n\} \setminus \{i\}} \frac{a_{ij}}{a_{ii}} x_j^k + \frac{b_i}{a_{ii}},\ i = 1, \ldots, n.$$

Das Konvergenzverhalten analysiert der folgende Satz.

Satz 5.12 *Die Matrix $A = (a_{ij}) \in \mathbb{K}^n$ genüge einer der drei folgenden Bedingungen:*

Zeilensummenkriterium: $q_\infty = \max_{i=1,\ldots,n} \sum_{j \in \{1,\ldots,n\} \setminus \{i\}} \left| \frac{a_{ij}}{a_{ii}} \right| < 1$

Spaltensummenkriterium: $q_1 = \max_{j=1,\ldots,n} \sum_{i \in \{1,\ldots,n\} \setminus \{j\}} \left| \frac{a_{ij}}{a_{jj}} \right| < 1$

Quadratsummenkriterium: $q_2 = \sqrt{\sum_{i,j \in \{1,\ldots,n\}, i \neq j} \left| \frac{a_{ij}}{a_{ii}} \right|^2} < 1$

Dann konvergiert das Jacobi-Verfahren bezüglich jeder Norm im $\mathbb{K}^n$ für jede rechte Seite $b \in \mathbb{K}^n$ und für jeden Startwert $x^{(0)} \in \mathbb{K}^n$, und zwar gegen die eindeutig bestimmte Lösung x^* von $Ax^* = b$. Weiterhin gilt für $p \in \{1, 2, \infty\}$:

- *A priori Fehlerschranke:* $\|x^{(k)} - x^*\|_p \leq \frac{q_p^k}{1-q_p} \|x^{(1)} - x^{(0)}\|_p$
- *A posteriori-Fehlerschranke:* $\|x^{(k)} - x^*\|_p \leq \frac{q_p}{1-q_p} \|x^{(k)} - x^{(k-1)}\|_p$

Beweis: Wir untersuchen die Norm der Iterationsmatrix

$$B = -A_D^{-1}(A_L + A_R).$$

Nach Satz 3.17 gilt, dass

$$\| - A_D^{-1}(A_L + A_R)\|_p = q_p < 1 \text{ für } p \in \{\infty, 1\},$$

und aus Lemma 3.24 folgt, dass

$$\| - A_D^{-1}(A_L + A_R)\|_2 \leq \| - A_D^{-1}(A_L + A_R)\|_F = q_2 < 1.$$

Wir können also Satz 5.8 anwenden, aus dem sich der Rest der Behauptungen direkt ergibt. QED

Bemerkung: Die drei Konvergenzkriterien sind nicht äquivalent!

Der Algorithmus ergibt sich in kanonischer Weise:

Algorithmus 9: Jacobi-Verfahren

Input: `Reguläre Matrix` $A \in \mathbb{K}^{n,n}$ `mit` $a_{ii} \neq 0$ `für` $i = 1, \ldots, n$, $b \in \mathbb{K}^n$, $x^{(0)} \in \mathbb{K}^n$.

Schritt 1: $k := 0$

Schritt 2: `Repeat`

Schritt 2.1: `For` $i = 1, \ldots, n$ `do:` $x_i^{(k+1)} := \frac{1}{a_{ii}} \left(- \sum_{j \in \{1,\ldots,n\} \setminus \{i\}} a_{ij} x_j^{(k)} + b_j \right)$

Schritt 2.2: $k := k + 1$

Until `Abbruchkriterium`

Ergebnis: `Approximierte Lösung` x^* `von` $Ax^* = b$.

Ein Abbruchtest kann wie bei Algorithmus 8 besprochen mit $q = q_p$ durchgeführt werden.

Einzelschritt - oder Gauß-Seidel-Verfahren

Im jetzt zu besprechenden Einzelschritt-Verfahren (ESV) wählt man $M = A_D + A_L$. Nach der Voraussetzung (5.10) ist M regulär. Als Fixpunktgleichung erhält man

$$\begin{aligned} x &= x + (A_D + A_L)^{-1}(b - Ax) \\ &= -(A_D + A_L)^{-1}(-(A_D + A_L) + A)x + (A_D + AL)^{-1}b \\ &= -(A_D + A_L)^{-1}A_R x + (A_D + A_L)^{-1}b \end{aligned} \tag{5.13}$$

Das Verfahren der sukzessiven Approximation ergibt sich folglich zu

$$x^{(k+1)} = -(A_D + A_L)^{-1}A_R x^{(k)} + (A_D + A_L)^{-1}b,\ k = 0, 1, 2, \ldots$$

Die Iterationsmatrix ist entsprechend

$$C = I - (A_D + A_L)^{-1}A = -(A_D + A_L)^{-1}A_R.$$

Rechnerisch nutzt man die Umformulierung zu

$$(A_D + A_L)x^{(k+1)} = -A_R x^{(k)} + b,\ k = 0, 1, 2, \ldots \tag{5.14}$$

Um das komponentenweise zu schreiben löst man dieses System mittels Vorwärtselimination auf, um die Unbekannten $x_i^{(k+1)}$ für $i = 1, \ldots, n$ zu bestimmen. Man erhält

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

Die Formel stimmt fast mit der entsprechenden komponentenweisen Iterationsformel des Jacobi-Verfahrens überein. Der Unterschied besteht lediglich darin, dass beim vorliegenden Gauß-Seidel-Verfahren zur Berechnung von $x_i^{(k+1)}$ die neuen (und hoffentlich besseren) Werte $x_j^{(k+1)}$ für $j = 1, \ldots, i-1$ herangezogen werden anstatt der Werte $x_j^{(k)}$ wie im Jacobi-Verfahren. Das ist der Grund, warum das Gauß-Seidel-Verfahren in den meisten Fällen schneller konvergiert als das Jacobi-Verfahren.
Über das Konvergenzverhalten gibt der folgende Satz Auskunft.

Satz 5.13 *Die Matrix $A = (a_{ij}) \in \mathbb{K}^{n,n}$ genüge dem Kriterium nach Sassenfeld:*

$$p := \max_{i=1,\ldots,n} p_i < 1$$

mit den Werten

$$\begin{aligned} p_1 &:= \sum_{j=2}^{n} \left| \frac{a_{1j}}{a_{11}} \right| \\ p_i &:= \sum_{j=1}^{i-1} \left| \frac{a_{ij}}{a_{ii}} \right| p_j + \sum_{j=i+1}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| \quad \textit{für } i = 2, \ldots, n \end{aligned}$$

Dann konvergiert das Gauß-Seidel-Verfahren für jede rechte Seite $b \in \mathbb{K}^n$ und bei beliebigem $x^{(0)} \in \mathbb{K}^n$ gegen die eindeutig bestimmte Lösung x^* von $Ax^* = b$. Weiterhin gilt:

- *A priori-Fehlerschranke:* $\|x^{(k)} - x^*\|_\infty \leq \frac{p^k}{1-p}\|x^{(1)} - x^{(0)}\|_\infty$
- *A posteriori Fehlerschranke:* $\|x^{(k)} - x^*\|_\infty \leq \frac{p}{1-p}\|x^{(k)} - x^{(k-1)}\|_\infty$

Beweis: Wir wollen die Zeilensummennorm von $(A_D + A_L)^{-1}A_R$ abschätzen. Sei hierzu

$$(A_D + A_L)x = -A_R z \quad , \|z\|_\infty = 1,$$

das heißt,

$$\|x\|_\infty = \| - (A_D + A_L)^{-1}A_R z\|_\infty.$$

Vorwärtselimination ergibt

$$x_i = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} z_j \text{ für } i = 1, \ldots, n$$

Wir zeigen zunächst, dass $|x_i| \leq p_i$ für $i = 1, \ldots, n$:

Induktionsanfang: $i = 1$. Weil $|z_i| \leq 1$ für alle i erhält man:

$$|x_1| = \left| \sum_{j=2}^{n} \frac{a_{1j}}{a_{11}} z_j \right| \leq \sum_{j=2}^{n} \left| \frac{a_{1j}}{a_{11}} \right| |z_j| \leq \sum_{j=2}^{n} \left| \frac{a_{1j}}{a_{11}} \right| = p_1$$

Induktionsschritt: $i - 1 \to i$.

$$\begin{aligned} |x_i| &= \left| -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} z_j \right| \\ &\leq \sum_{j=1}^{i-1} \left| \frac{a_{ij}}{a_{ii}} \right| \underbrace{|x_j|}_{\leq p_j} + \sum_{j=i+1}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| \underbrace{|z_j|}_{\leq 1} \\ &\leq \sum_{j=1}^{i-1} \left| \frac{a_{ij}}{a_{ii}} \right| p_j + \sum_{j=i+1}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| = p_i \end{aligned}$$

Also gilt $\|x\|_\infty \leq p$ und entsprechend für $x(z) := (A_D + A_L)^{-1}A_R z$, dass

$$\|(A_D+A_L)^{-1}A_R\|_\infty = \sup_{z \in \mathbb{K}^n : \|z\|_\infty = 1} \|(A_D+A_L)^{-1}A_R z\|_\infty = \sup_{z \in \mathbb{K}^n : \|z\|_\infty = 1} \|x(z)\|_\infty \leq p.$$

Weil $p < 1$ vorausgesetzt war, folgt die Behauptung nach Satz 5.8. QED

Bemerkung: Erfüllt eine Matrix das Zeilensummenkriterium, so auch das Sassenfeldkriterium. Das heißt, das Zeilensummenkriterium ist ebenfalls hinreichend für die Konvergenz des Einzelschritt-Verfahrens.

Andererseits erfüllt nicht jede Matrix, die dem Sassenfeld-Kriterium genügt, auch das Zeilensummenkriterium, wie die folgende Matrix A zeigt:

$$A = \begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix}$$

Bemerkung: In Satz 5.17 werden wir zeigen, dass das Gauss-Seidel-Verfahren bei Gleichungssystemen mit hermitescher und positiv definiter Koeffizientenmatrix konvergiert.

Der Vollständigkeit halber sei der Algorithmus des Einzelschrittverfahrens ebenfalls skizziert.

Algorithmus 10: Gauß-Seidel-Verfahren

Input: `Reguläre Matrix` $A \in \mathbb{K}^{n,n}$ `mit` $a_{ii} \neq 0$ `für` $i = 1, \ldots, n$, $b \in \mathbb{K}^n$, $x^{(0)} \in \mathbb{K}^n$.

Schritt 1: $k := 0$

Schritt 2: `Repeat`

Schritt 2.1: `For` $i = 1, \ldots, n$ `do:`
$x_i^{(k+1)} := \frac{1}{a_{ii}} \left(- \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} + b_j \right)$

Schritt 2.2: $k := k + 1$

Until `Abbruchkriterium`

Ergebnis: `Approximierte Lösung` x^* `von` $Ax^* = b$.

Relaxations-Verfahren

Die Idee der Relaxations-Verfahren besteht darin, die Konvergenz des Gesamtschrittverfahrens beziehungsweise des Einzelschrittverfahrens zu verbessern, indem man durch Einführen eines so genannten Relaxations-Parameters den Spektralradius der Iterationsmatrix verkleinert.

Wir betrachten zunächst das Gesamtschritt-Verfahren. Die Iterationsvorschrift ergibt sich nach (5.11) auf Seite 101:

$$x^{(k+1)} = x^{(k)} + A_D^{-1}(b - Ax^{(k)}).$$

In jedem Iterationsschritt wird also $x^{(k)}$ durch das A_D^{-1}-fache des **Residuums** $z^{(k)} = b - Ax^{(k)}$ korrigiert. Dabei ist oft zu beobachten, dass die Korrektur um einen festen Faktor zu klein ist. Deshalb kann es sinnvoll sein, den Wert um $\omega z^{(k)}$ statt um $z^{(k)}$ zu ändern, wobei ω ein beliebiger positiver Parameter sein darf. Das resultierende Verfahren ist das relaxierte Gesamtschrittverfahren.

Definition 5.14 *Das Iterationsverfahren*

$$x^{(k+1)} = x^{(k)} + \omega A_D^{-1}(b - Ax^{(k)})$$

heißt **Gesamtschritt-Relaxationsverfahren**.

Komponentenweise berechnen sich die Werte $x_i^{(k+1)}$ durch

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j \in \{1,\ldots,n\} \setminus \{i\}} a_{ij} x_j^{(k)} \right), i = 1, \ldots, n$$

Es gilt

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \omega A_D^{-1}(b - Ax^{(k)}) \\ &= (I - \omega A_D^{-1} A)x^{(k)} + \omega A_D^{-1} b \\ &= \left[I - \omega I + \omega(-A_D^{-1}A + I) \right] x^{(k)} + \omega A_D^{-1} b \\ &= \left[(1-\omega)I + \omega B \right] x^{(k)} + \omega A_D^{-1} b, \end{aligned} \tag{5.15}$$

wobei im letzten Schritt $B = I - A_D^{-1}A$ die Iterationsmatrix des Gesamtschrittverfahrens aus (5.12) bezeichnet (siehe Seite 101). Die Iterationsmatrix des Gesamtschritt-Relaxationsverfahrens mit Relaxationsparameter ω bezeichnen wir im folgenden mit

$$B_\omega = (I - \omega A_D^{-1} A) = (1 - \omega)I + \omega B.$$

Wir bemerken, dass die Matrix des Gesamtschrittverfahrens gerade $B = B_1$ ist.

Satz 5.11 legt nahe, den Relaxationsparameter ω so zu wählen, dass der Spektralradius der Iterationsmatrix B_ω möglichst klein wird. Der folgende Satz gibt Auskunft darüber, wie dieses Ziel erreicht werden kann.

Satz 5.15 *Die zum Gesamtschrittverfahren gehörende Iterationsmatrix $B = I - A_D^{-1}A$ habe nur reelle Eigenwerte und einen Spektralradius $\rho(B) < 1$. Sei weiterhin $-1 < \lambda_{\min}$ der kleinste Eigenwert von B und $\lambda_{\max} < 1$ der größte. Für die Iterationsmatrix des Gesamtschritt-Relaxationsverfahrens*

$$B_\omega = (1-\omega)I + \omega B$$

gilt dann:

$$\rho(B_\omega) \text{ wird minimal für } \omega^* = \frac{2}{2 - \lambda_{\min} - \lambda_{\max}}.$$

Speziell erhält man $\rho(B_\omega) < \rho(B)$ falls $\lambda_{min} \neq -\lambda_{\max}$.

Beweis: Zunächst bemerken wir dass für $\omega \neq 0$

$$Bu = \lambda u \Longleftrightarrow [(1-\omega)I + \omega B]\, u = [(1-\omega) + \omega\lambda]\, u$$

gilt. Das heißt, λ ist Eigenwert von B genau dann wenn $(1-\omega) + \omega\lambda$ Eigenwert von B_ω ist. Weil $\omega > 0$ erhält man insbesondere, dass

$$\begin{array}{ll} (1-\omega) + \omega\lambda_{\min} & \text{der kleinste Eigenwert von } B_\omega \text{ ist, und} \\ (1-\omega) + \omega\lambda_{\max} & \text{der größte.} \end{array}$$

Bei gegebenem ω ist der Spektralradius der Matrix B_ω folglich

$$\rho(B_\omega) = \max\{-(1-\omega) - \omega\lambda_{\min}, (1-\omega) + \omega\lambda_{\max}\}$$

Jetzt möchten wir ω so bestimmen, dass dieser Ausdruck möglichst klein wird. Dazu überlegt man sich, dass die beiden Funktionen

$$\begin{aligned} -(1-\omega) - \omega\lambda_{\min} &= -1 + \omega(1 - \lambda_{\min}) \\ (1-\omega) + \omega\lambda_{\max} &= 1 + \omega(\lambda_{\max} - 1) \end{aligned}$$

Geraden sind. Das Maximum von zwei Geraden ist eine konvexe Funktion, die aus zwei linearen Abschnitten besteht und ihr eindeutiges Minimum genau am Schnittpunkt der beiden Geraden annimmt, falls dieser existiert. In unserem Fall ist das gegeben, da die beiden Steigungen der Geraden aufgrund der Bedingung $\lambda_{\min} < \lambda_{\max} < 1$

$$1 - \lambda_{\min} \neq \lambda_{\max} - 1$$

erfüllen; die Geraden sind also nicht parallel. Ihr eindeutiger Schnittpunkt errechnet sich durch Auflösen der Gleichung

$$-(1-\omega) - \omega\lambda_{\min} = (1-\omega) + \omega\lambda_{\max}$$

und liegt entsprechend bei

$$w^* = \frac{2}{2 - \lambda_{\min} - \lambda_{\max}}.$$

Da für $\lambda_{\min} \neq -\lambda_{\max}$ gilt, dass $\omega^* \neq 1$ ist, und das Minimum ω^* eindeutig ist, folgt, dass

$$\rho(B_{\omega^*}) < \rho(B),$$

der Spektralradius von der Iterationsmatrix des Gesamtschritt-Relaxationsverfahrens B_{ω^*} ist in diesem Fall also echt kleiner als der Spektralradius der Matrix B des (unrelaxierten) Gesamtschrittverfahrens. QED

Der optimale Relaxationskoeffizient ω^* liegt also im Bereich $(0, \infty)$.

- Ist $\omega^* < 1$ so spricht man von *Unterrelaxation*. Sie tritt auf, falls $-\lambda_{\min} > \lambda_{\max}$.
- Für $\omega^* = 1$ (also wenn $-\lambda_{\min} = \lambda_{\max}$) erhält man das normale Gesamtschrittverfahren.
- Ist $\omega^* > 1$ so spricht man von *Überrelaxation*. Sie tritt auf, falls $-\lambda_{\min} < \lambda_{\max}$.

Um ω^* zu berechnen, sind scharfe Schranken für die Eigenwerte der Matrix A (inklusive Vorzeichen) nötig.

Das Gesamtschritt-Relaxationsverfahren hat noch eine andere Interpretation: Bezeichnet man die (unrelaxierte) Iterierte aus dem Gesamtschrittverfahren mit

$$z^{(k+1)} = Bx^{(k)} + A_D^{-1}b,$$

dann gilt nach (5.15), dass

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega z^{(k+1)}, \tag{5.16}$$

der neue Wert $x^{(k+1)}$ entsteht also, indem man zwischen dem letzten Wert $x^{(k)}$ und dem Wert $z^{(k+1)}$ aus dem Gesamtschrittverfahren linear interpoliert.

Wir untersuchen nun, wie man ein Relaxationsverfahren bezüglich des Einzelschrittverfahrens definieren kann. Im Einzelschrittverfahren (siehe (5.13)) hatten wir die Fixpunktgleichung

$$x = x + (A_D + A_L)^{-1}(b - Ax),$$

aus der sich die Iterationsmatrix

$$C = I - (A_D + A_L)^{-1}A = -(A_D + A_L)^{-1}A_R$$

ergibt. Zur numerischen Berechnung wurde die Umformulierung

$$(A_D + A_L)x^{(k+1)} = -A_R x^{(k)} + b, \;\; k = 0, 1, 2 \ldots$$

angegeben, die wir jetzt weiter zu

$$A_D x^{(k+1)} = b - A_L x^{(k+1)} - A_R x^{(k)}$$

umformulieren. Wir definieren das Relaxationsverfahren jetzt ähnlich wie für das Gesamtschrittverfahren, indem wir auch hier die auf der linken Seite im Einzelschrittverfahren auftretenden $x^{(k+1)}$ zu $z^{(k+1)}$ umbenennen. Das heißt, wir schreiben obige Gleichung als

$$A_D z^{(k+1)} = b - A_L x^{(k+1)} - A_R x^{(k)}. \tag{5.17}$$

Wie in (5.16) wählen wir den relaxierten Wert für $x^{(k+1)}$ als

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega z^{(k+1)}.$$

Diese Definition möchten wir nun in (5.17) einsetzen. Dazu multiplizieren wir (5.17) mit ω und substituieren $\omega z^{(k+1)}$ durch $x^{(k+1)} - (1-\omega)x^{(k)}$ wie in (5.16) gefordert. Man erhält

$$A_D x^{(k+1)} = (1-\omega)A_D x^{(k)} + \omega b - \omega A_L x^{(k+1)} - \omega A_R x^{(k)}, \tag{5.18}$$

was sich zu

$$(A_D + \omega A_L)x^{(k+1)} = \left[(1-\omega)A_D - \omega A_R\right] x^{(k)} + \omega b$$

und schließlich zu

$$x^{(k+1)} = (A_D + \omega A_L)^{-1} \left[(1-\omega)A_D - \omega A_R\right] x^{(k)} + \omega (A_D + \omega A_L)^{-1} b$$

umformulieren lässt. Daraus ergibt sich die Iterationsmatrix für das Einzelschritt-Relaxationsverfahren in Abhängigkeit von ω zu

$$C_w = (A_D + \omega A_L)^{-1} \left[(1-\omega)A_D - \omega A_R\right]. \tag{5.19}$$

Man sieht, dass auch hier $C_1 = C$ gilt, d.h. für den Relaxationsparameter $\omega = 1$ erhält man die Iterationsmatrix aus dem normalen Einzelschrittverfahren.

Um die Werte $x_i^{(k+1)}$ komponentenweise zu bestimmen, multipliziert man (5.18) mit von links A_D^{-1} und formuliert die entstehende Gleichung dann folgendermaßen um:

$$\begin{aligned} x^{(k+1)} &= (1-\omega)x^{(k)} + \omega A_D^{-1} b - \omega A_D^{-1} A_L x^{(k+1)} - \omega A_D^{-1} A_R x^{(k)} \\ &= x^{(k)} + \omega A_D^{-1}(b - A_L x^{(k+1)} - A_D x^{(k)} - A_R x^{(k)}) \\ &= x^{(k)} + \omega A_D^{-1}(b - A_L x^{(k+1)} - (A_D + A_R)x^{(k)}) \end{aligned}$$

Man bestimmt dann $x^{(k+1)}$ via

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij} x_j^{(k)} \right) \quad i = 1, \ldots, n.$$

Das Verfahren nennt man auch *Successive overrelaxation*, abgekürzt als SOR-Verfahren (obwohl man streng genommen nur für $\omega > 1$ von einer Überrelaxation sprechen sollte.)

Als nächstes untersuchen wir, für welche Relaxationsparameter w, wir Konvergenz erwarten können. Zunächst geben wir ein negatives Ergebnis.

Satz 5.16 *Sei $A \in \mathbb{K}^{n,n}$ mit $a_{ii} \neq 0$ für $i = 1, \ldots, n$. Dann gilt*

$$\rho(C_w) \geq |\omega - 1|.$$

Insbesondere ist $\rho(C_w) \geq 1$ falls $\omega \notin (0, 2)$, d.h. das SOR-Verfahren konvergiert in diesen Fällen im allgemeinen nicht.

Beweis: Wir schreiben die Iterationsmatrix C_ω um zu

$$\begin{aligned} C_\omega &= (A_D + \omega A_L)^{-1} A_D A_D^{-1} \left[(1-\omega)A_D - \omega A_R\right] \\ &= \left[A_D^{-1}(A_D + \omega A_L)\right]^{-1} \left[(1-\omega)I - \omega A_D^{-1} A_R\right] \\ &= \left[(I + \omega A_D^{-1} A_L)\right]^{-1} \left[(1-\omega)I - \omega A_D^{-1} A_R\right], \end{aligned}$$

also dem Produkt von

- einer nach Satz 2.9 normierten unteren Dreiecksmatrix $(I + \omega A_D^{-1} A_L)^{-1}$, und
- einer oberen Dreiecksmatrix $(1-\omega)I - \omega A_D^{-1} A_R$ mit Diagonalelementen $(1-\omega)$.

Es gilt also

$$\det(C_\omega) = \det(I + \omega A_D^{-1} A_L)^{-1} \det\left[(1-\omega)I - \omega A_D^{-1} A_R\right] = (1-\omega)^n.$$

Weil die Determinante einer Matrix gleich dem Produkt ihrer Eigenwerte ist, gilt insbesondere $|\det(C_\omega)| \leq (\rho(C_\omega))^n$, also

$$|1-\omega|^n \leq (\rho(C_\omega))^n$$

und damit folgt die Behauptung. QED

Abschließend zeigen wir, dass die Rückrichtung der obigen Aussage zumindest für hermitesche und positiv definite Matrizen richtig ist: Für alle Werte $\omega \in (0, 2)$ des Relaxationsparameter konvergiert das Verfahren.

Satz 5.17 *Sei $A \in \mathbb{K}^{n,n}$ hermitesch und positiv definit. Dann konvergiert das Einzelschritt-Relaxationsverfahren (SOR-Verfahren) für jeden Relaxationsparameter $\omega \in (0, 2)$.*

Beweis: Wir berechnen den Spektralradius der Iterationsmatrix C_ω, und zeigen, dass $\rho(C_\omega) < 1$ gilt. Dazu sei also λ ein Eigenwert von C_ω mit zugehörigem Eigenvektor x. Unser Ziel ist, $|\lambda| < 1$ nachzuweisen. Nach (5.19) ist $C_\omega x = \lambda x$ gleichbedeutend mit

$$[(1-\omega)A_D - \omega A_R]\, x = \lambda(A_D + \omega A_L)x. \tag{5.20}$$

Wir nutzen nun folgende beide Aussagen, die sich direkt aus $A = A_L + A_D + A_R$ ergeben:

1. $(2-\omega)A_D - \omega A - \omega(A_R - A_L) = 2(1-\omega)A_D - 2\omega A_R$
2. $(2-\omega)A_D + \omega A - \omega(A_R - A_L) = 2A_D + 2\omega A_L$

Damit folgt aus (5.20), dass

$$[(2-\omega)A_D - \omega A - \omega(A_R - A_L)]\, x = \lambda\,[(2-\omega)A_D + \omega A - \omega(A_R - A_L)]\, x$$

Um diese Gleichung nach λ aufzulösen, bilden wir das Skalarprodukt durch die Multiplikation beider Seiten von links mit x^*. Um abzukürzen, führen die Bezeichnungen

$$\begin{aligned} d &:= x^* A_D x \\ a &:= x^* A x \end{aligned}$$

ein, und bemerken, dass $a > 0$ und $d > 0$ gilt, weil A positiv definit ist. Die Multiplikation von links mit x^* ergibt nun

$$(2-\omega)d - \omega a - \omega x^*(A_R - A_L)x = \lambda\,[(2-\omega)d + \omega a - \omega x^*(A_R - A_L)x] \tag{5.21}$$

Zunächst machen wir uns klar, dass

$$\begin{aligned} (ix^*(A_R - A_L)x)^* &= x^*(A_R^* - A_L^*)x\bar{i} \\ &= x^*(A_L - A_R)x(-i) \quad \text{weil } A = A^* \\ &= ix^*(A_R - A_L)x \end{aligned}$$

gilt, und daher $s := ix^*(A_R - A_L)x \in \mathbb{R}$ ist und wir (5.21) weiter umformulieren können zu

$$(2-\omega)d - \omega a + i\ \omega s = \lambda\,[(2-\omega)d + \omega a + i\ ws]\,,$$

in der bis auf λ alle auftretenden Werte $\omega, d, a, s \in \mathbb{R}$ sind. Mit

$$\begin{aligned} \alpha &:= (2-\omega)d - \omega a \in \mathbb{R} \\ \tilde{\alpha} &:= (2-\omega)d + \omega a \in \mathbb{R} \\ \beta &:= \omega s \in \mathbb{R} \end{aligned}$$

erhalten wir endlich

$$\alpha + i\beta = \lambda(\tilde{\alpha} + i\beta).$$

Dann gilt auch für die Beträge, dass

$$|\alpha + i\beta| = |\lambda||\tilde{\alpha} + i\beta|.$$

Wir nutzen noch aus, dass $\tilde{\alpha} > \alpha$ (weil $\omega \in (0, 2)$) und erhalten

$$\alpha^2 + \beta^2 = |\lambda|(\tilde{\alpha}^2 + \beta^2) > |\lambda|(\alpha^2 + \beta^2),$$

also $|\lambda| < 1$. QED

Folge: Da das Einzelschrittverfahren ein Spezialfall des SOR-Verfahrens, nämlich mit $\omega = 1$ ist, haben wir mit dem vorliegenden Satz bewiesen, dass das Einzelschrittverfahren ESV für hermitesche und positiv definite Matrizen immer konvergiert.

5.4 Iterative Verfahren für nichtlineare Gleichungssysteme

In diesem Abschnitt betrachten wir nun endlich nichtlineare Gleichungssysteme

$$F(x) = 0$$

mit einer reellen Funktion $F : \mathbb{R}^n \to \mathbb{R}^n$,

$$F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix}$$

Wir nehmen zunächst an, dass unser Gleichungssystem bereits in Fixpunktform $G(x) = x$ vorliegt mit einer Funktion

$$G(x) = \begin{pmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_n(x) \end{pmatrix}.$$

Eine Fixpunktgleichung kann man z.B. durch die Funktion G mit

$$G(x) = x + M^{-1}(x)(F(x))$$

erzeugen. Dabei ist M ein linearer Operator, der von x abhängen darf. Wie wichtig es ist, die Funktion G sinnvoll zu wählen, zeigt das folgende Beispiel.

Wir betrachten die Funktion $f(x) = x - \cos x$ im Intervall $[0, 1]$.

- Wähle $g(x) = \cos x$. Dann gilt $f(x) = 0$ genau dann wenn $g(x) = x$. Weiterhin gilt $g : [0,1] \to [0,1]$. Jetzt wollen wir noch zeigen, dass g eine Kontraktion ist. Wir wenden Lemma 5.7 an und erhalten

$$q = \sup_{0 \le x \le 1} |g'(x)| = \sup_{0 \le x \le 1} \sin x = \sin 1 < 1,$$

  also ist g eine Kontraktion. Nach Satz 5.6 konvergiert also das Verfahren $x^{(k+1)} = \cos x^{(k)}$. Allerdings ist die Konvergenzgeschwindigkeit unbefriedigend.

- Betrachten wir nun

$$g(x) = x - \frac{x - \cos x}{1 + \sin x}$$

  Auch dann gilt $f(x) = 0$ genau dann wenn $g(x) = x$, und man sieht nach kurzer Rechnung, dass $g : [0,1] \to [0,1]$, und dass

$$\begin{aligned} g'(x) &= 1 - \frac{(1+\sin x)^2 - (x - \cos x)\cos x}{(1+\sin x)^2} \\ &= 1 - 1 + \frac{(x - \cos x)\cos x}{(1+\sin x)^2} \\ &= 1 \quad \text{für } x = 0. \end{aligned}$$

  Also ist g keine Kontraktion. Das Verfahren der sukzessiven Approximation konvergiert dennoch. Die Konvergenz mit Hilfe dieser Fixpunktgleichung ist sogar sehr schnell!

  Um die schnelle Konvergenz zu erklären, schreibt man die Ableitung um zu

$$g'(x) = \frac{f(x)\cos x}{(1+\sin x)^2}.$$

  Weil f eine Nullstelle x^* in $[0,1]$ hat, gilt $g'(x^*) = 0$, also gibt es eine Umgebung um x^*, in der g eine Kontraktion ist. Der Kontraktionsfaktor in dieser Umgebung ist nahe bei Null (also sehr klein), und das Konvergenzverhalten daher gut.

Bevor wir diese Beobachtung im Newton-Verfahren ausnutzen, verallgemeinern wir Satz 5.12 auf nichtlineare Funktionen und beweisen damit, dass das Verfahren der sukzessiven Approximation unter ähnlichen Bedingungen wir im Satz 5.12 auch im nichtlinearen Fall konvergiert.

Satz 5.18 *Sei $U \subseteq \mathbb{R}^n$ eine konvexe Menge und $G : U \to U$ eine stetig differenzierbare Abbildung (d.h. jedes der Elemente der Jacobi-Matrix DG ist stetig in U). Weiterhin gelte eine der folgenden Bedingungen:*

Zeilensummenkriterium: $q_\infty = \sup_{x \in U} \max_{i=1,\ldots,n} \sum_{j=1}^{n} \left| \frac{\partial g_i}{\partial x_j} \right| < 1$

Spaltensummenkriterium: $q_1 = \sup_{x\in U} \max_{j=1,\ldots,n} \sum_{i=1}^{n} \left|\frac{\partial g_i}{\partial x_j}\right| < 1$

Quadratsummenkriterium: $q_2 = \sup_{x\in U} \sqrt{\sum_{i,j=1}^{n} \left|\frac{\partial g_i}{\partial x_j}\right|^2} < 1$

Dann konvergiert das Verfahren der sukzessiven Approximation bezüglich jeder Norm im $\mathbb{R}^n$ für jeden Startwert $x^{(0)} \in \mathbb{R}^n$, und zwar gegen die eindeutig bestimmte Lösung x^* des nichtlinearen Gleichungssystems $G(x^*) = x^*$. Ist $q_p < 1$ für $p \in \{1, 2, \infty\}$ so gelten für dieses p außerdem die folgenden Schranken.

- *A priori Fehlerschranke:* $\|x^{(k)} - x^*\|_p \leq \frac{q_p^k}{1-q_p}\|x^{(1)} - x^{(0)}\|_p$
- *A posteriori-Fehlerschranke:* $\|x^{(k)} - x^*\|_p \leq \frac{q_p}{1-q_p}\|x^{(k)} - x^{(k-1)}\|_p$

Beweis: Nach Satz 3.17 und Lemma 3.24 zeigen, dass

$$\begin{aligned} \sup_{x\in U} \|DG(x)\|_p &= q_p \quad \text{für } p \in \{\infty, 1\} \\ \sup_{x\in U} \|DG(x)\|_2 &\leq q_2 \end{aligned}$$

Daher ist nach Lemma 5.7 die Abbildung $G : U \to U$ kontrahierend, falls $q_p < 1$ für ein $p \in \{\infty, 1, 2\}$. Satz 5.6 ergibt die Behauptung. QED

Unter den Voraussetzungen des letzten Satzes konvergiert also das Verfahren der sukzessiven Approximation auch im nichtlinearen Fall. Das Verfahren

$$x_i^{(k+1)} = g_i(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}) \; i = 1, \ldots, n, \; k = 0, 1, 2, \ldots$$

nennt man auch **nichtlineares Gesamtschrittverfahren**, während man das Verfahren

$$\begin{aligned} x_1^{(k+1)} &= g_1(x_1^{(k)}, \ldots, x_n^{(k)}) \\ x_i^{(k+1)} &= g_i(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_i^{(k)}, \ldots, x_n^{(k)}) \; i = 2, \ldots, n, \end{aligned}$$

als **nichtlineares Einzelschrittverfahren** bezeichnet. Die in Abschnitt 5.3 besprochenen Verfahren GSV und ESV sind Spezialfälle dieser Verfahren.

Das Newton-Verfahren für skalare Funktionen

Wir kommen nun wieder zurück zu dem originalen nichtlinearen Gleichungssystem

$$F(x) = 0$$

und entwickeln mit dem nun zu besprechenden Newton-Verfahren eine Fixpunktform, die — wenn sie konvergiert — zu einem schnelleren Konvergenzverhalten führt. Wir beginnen unsere Überlegung für eine reelle Funktion $f : \mathbb{R} \to \mathbb{R}$.

Gesucht ist eine Nullstelle x^* der Funktion f. Haben wir schon eine Schätzung der Nullstelle $x^{(0)}$ und ist f stetig differenzierbar, so besteht die Idee des Newton-Verfahrens darin, f durch seine Tangente durch den Punkt $x^{(0)}$

$$f \approx f(x^{(0)}) + f'(x^{(0)})(x - x^{(0)})$$

(also der Taylorreihe bis zum linearen Glied) zu ersetzen. Man sucht also die Nullstelle der Näherung anstatt der Nullstelle von f. Eine Nullstelle der Näherung $f(x^{(0)}) + f'(x^{(0)})(x - x^{(0)})$ existiert, falls $f(x^{(0)}) \neq 0$ ist und ist in diesem Fall gegeben durch

$$x = x^{(0)} - \frac{f(x^{(0)})}{f'(x^{(0)})}$$

Wiederholt man das Vorgehen mit $x^{(1)} := x$ so erhält man das Newton-Verfahren. Erfüllt die Ableitung $f(x) \neq 0$ so erhält man die Fixpunktgleichung

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Kommen wir kurz zu dem Beispiel $f(x) = x - \cos x$ von Seite 112 zurück: Hier wurde mit

$$g(x) = x - \frac{1}{1 + \sin x}(x - \cos x) = x - \frac{f(x)}{f'(x)}$$

im zweiten Versuch genau die Fixpunktform des Newton-Verfahrens verwendet. Leider ist die Funktion g im allgemeinen keine Kontraktion auf dem gesamten zu betrachtenden Intervall, so dass Satz 5.18 nicht anwendbar ist. Dennoch gilt die folgende lokale Konvergenzaussage.

Satz 5.19 *Sei x^* eine einfache Nullstelle der $f : \mathbb{R} \to \mathbb{R}$. Sei weiterhin f in einer Umgebung von x^* zwei mal stetig differenzierbar. Dann konvergiert das Newton-Verfahren für jeden Startwert $x^{(0)}$, der hinreichend dicht bei x^* liegt.*

Beweis: Weil x^* einfache Nullstelle ist, gilt $f'(x^*) \neq 0$ und entsprechend gibt es eine Umgebung $U := U(x^*)$ so dass $f'(x) \neq 0$ für alle $x \in U$. Die Verfahrensvorschrift $g(x) = x - \frac{f(x)}{f'(x)}$ ist damit für alle $x \in U$ definiert. Die Ableitung von g ist

$$g'(x) = 1 - \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2} = \frac{f''(x)}{[f'(x)]^2} f(x),$$

also $g'(x^*) = 0$. Wegen der Stetigkeit von g' gibt es Zahlen $\delta > 0, q < 1$ sodass für alle $x \in U' = [x^* - \delta, x^* + \delta] \cap U$ gilt $|g'(x)| \leq q < 1$. Daraus folgt

$$|g(x) - x^*| = |g(x) - g(x^*)| \leq q|x - x^*| \leq \delta \quad \text{für alle } x \in U',$$

das heißt, $g : U' \to U'$ ist eine Kontraktion. Satz 5.6 liefert die Behauptung.

QED

Das Newton-Verfahren für mehrdimensionale Funktionen

Wir formulieren zunächst das Newton-Verfahren auch im mehrdimensionalen Fall.

Definition 5.20 *Sei $U \subseteq \mathbb{R}^n$ offen und $F : U \to \mathbb{R}^n$ eine stetig differenzierbare Funktion mit einer für alle $x \in U$ regulären Jacobimatrix $DF(x)$. Dann heißt das Verfahren*

$$x^{(k+1)} = x^{(k)} - [DF(x^{(k)})]^{-1} F(x^{(k)}) \quad k = 0, 1, 2, \ldots$$

mit Startwert $x^{(0)} \in U$ **Newton-Verfahren**.

Die Motivation für das Newton-Verfahren ist die gleiche wie für skalare Funktionen: Anstatt die Nullstelle $F(x) = 0$ zu suchen, ersetzt man

$$F \approx F(x^{(0)}) + (DF(x^{(0)}))(x - x^{(0)}).$$

Existiert $[DF(x^{(0)})]^{-1}$, so kann man diese Gleichung nach x auflösen und erhält

$$x = x^{(0)} - [DF(x^{(0)})]^{-1} F(x^{(0)})$$

als nächste Iterierte. Das entspricht der Fixpunktgleichung (5.2)

$$G(x) = x + MF(x)$$

mit der regulären Matrix $M = [DF(x)]^{-1}$.

Um das Newton-Verfahren numerisch zu realisieren wird zur Bestimmung von

$$x^{(k+1)} = x^{(k)} - [DF(x^{(k)})]^{-1} F(x^{(k)}) \quad k = 0, 1, 2, \ldots$$

in jedem Schritt das lineare Gleichungssystem

$$DF(x^{(k)})(x^{(k+1)} - x^{(k)}) = -F(x^{(k)})$$

gelöst. Das geschieht durch das Lösen des Systems

$$DF(x^{(k)}) w^{(k)} = -F(x^{(k)})$$

und anschließendes Berechnen von

$$x^{(k+1)} = x^{(k)} + w^{(k)}.$$

Bevor wir auf die Konvergenzeigenschaften näher eingehen, formulieren wir das Verfahren.

Algorithmus 11: Newton-Verfahren

Input: Offene Menge $U \subseteq \mathbb{R}^n$, Differenzierbare Abbildung $F : U \to \mathbb{R}^n$ mit Jacobi-Matrix $DF : U \to \mathbb{R}^{n,n}$. Startwert $x^{(0)} \in U$, Toleranzwert $\varepsilon > 0$.

Schritt 1: $k := 0$

Schritt 2: Repeat

Schritt 2.1: Finde $w^{(k)}$ als Lösung des Gleichungssystems

$$DF(x^{(k)})w^{(k)} = -F(x^{(k)}).$$

Schritt 2.2: $x^{(k+1)} := x^{(k)} + w^{(k)}$

Schritt 2.3: $q_k := \frac{\|w^{(k)}\|}{\|w^{(k-1)}\|}$

Schritt 2.4: If $q_k \geq 1$ oder $x^{(k+1)} \notin U$ STOP: Das Verfahren scheint nicht zu konvergieren.

Schritt 2.5: $k := k + 1$

Until $\frac{q_k}{1-q_k}\|w^{(k)}\| \leq \varepsilon$

Ergebnis: Approximierte Nullstelle $x^{(k)}$ von F.

Leider ist die Berechnung von $DF(x)$ für große n aufwändig, so dass man die Jacobi-Matrix in der Praxis nicht in jedem Schritt neu berechnet, sondern häufig die folgenden Varianten verwendet:

- Frozen Newton: Es wird nur einmal die Jacobi-Matrix berechnet, für die dann mittels LU-Zerlegung alle in den Iterationen auftretende Gleichungssysteme effizient lösbar sind.
- Quasi-Newton: Die Jacobi-Matrix wird in jedem Schritt (approximativ) angepasst.

Um den Konvergenzbereich des Verfahrens zu vergrößern verwendet man auch das so genannte *gedämpfte* Newton-Verfahren, in dem man die Iterationsvorschrift

$$x^{(k+1)} = x^{(k)} + \lambda_k w^{(k)}, \lambda_k \in [0, 1]$$

verwendet.

Das Konvergenzverhalten des mehrdimensionalen Newton-Verfahrens lässt sich nicht ganz so einfach analysieren wie im eindimensionalen Fall. Daher ist die Verallgemeinerung von Satz 5.19 etwas schwieriger zu zeigen. Wir beweisen im folgenden Satz aber mehr, nämlich dass das Newton-Verfahren sogar **quadratisch konvergiert**.

Definition 5.21 *Sei $x^{(k)} \to x^*$ eine Folge im $\mathbb{K}^n$ mit $x^{(k)} \neq x^*$ für alle k. Wenn es eine Konstante q und eine Zahl M gibt, so dass*

$$\|x^{(k+1)} - x^*\| \leq q\|x^{(k)} - x^*\|^p \quad \text{für alle } k \geq M$$

so liegt eine Konvergenz der **Konvergenzordnung** *p gegen x^* vor. Den Fall $p = 1$ bezeichnet man als* **lineare Konvergenz**, *den Fall $p = 2$ als* **quadratische Konvergenz**.

Man beachte, dass sich die Anzahl der korrekt gefundenen Stellen einer Zahl bei quadratischer Konvergenz in jedem Schritt etwa verdoppelt. Wir formulieren jetzt den Satz zur Konvergenz des Newton-Verfahrens.

Satz 5.22 *Sei $U \subseteq \mathbb{R}^n$ offen und konvex und sei $F : U \to \mathbb{R}^n$ stetig differenzierbar. Für $x^{(0)} \in U$ erfülle F außerdem die folgenden vier Bedingungen in einer (beliebigen) Norm $\|\cdot\|$ auf dem $\mathbb{R}^n$:*

B1: Es existiert eine Nullstelle $x^* \in U$ der Funktion F.

B2: $DF(x)$ ist regulär für alle $x \in U$.

B3: Es gibt $\omega > 0$ so dass für alle $x, y \in U$ die folgenden beiden Bedingungen gelten:

- *(a) $\|[DF(x)]^{-1}(DF(y) - DF(x))\| \leq \omega\|x - y\|$.*
- *(b) Für $\rho := \|x^* - x^{(0)}\|$ gilt $\frac{\omega}{2}\rho < 1$.*

B4: Die Kugel $B_\rho(x^*) := \{x \in \mathbb{R}^n : \|x - x^*\| < \rho\}$ mit Radius ρ um die Nullstelle x^* ist in U enthalten.

Für die im Newton-Verfahren definierte Folge

$$x^{(k+1)} := x^{(k)} - [DF(x^{(k)})]^{-1}F(x^{(k)})$$

gilt dann:

1. *$x^{(k)} \in B_\rho(x^*)$ für alle $k = 1, 2, \ldots$.*
2. *$x^{(k)}$ konvergiert gegen x^*.*
3. *Für $k = 0, 1, 2, \ldots$ gilt die folgende a priori Fehlerschranke*

$$\|x^{(k)} - x^*\| \leq \rho \left(\frac{\omega\rho}{2}\right)^{2^k - 1} \tag{5.22}$$

4. *Für $k = 0, 1, 2, \ldots$ gilt die folgende a posteriori Fehlerschranke:*

$$\|x^{(k+1)} - x^*\| \leq \frac{\omega}{2}\|x^{(k)} - x^*\|^2 \tag{5.23}$$

Beweis:

Teil 1:

Wir zeigen zunächst, dass aus $x^{(k)} \in U$ die a-posteriori Fehlerschranke (5.23) für k folgt, danach beweisen wir die Wohldefiniertheit für alle k per Induktion.

Dazu benötigen wir zunächst eine Funktion $g : [0,1] \to \mathbb{R}^n$, die wir als

$$g(t) = F(x^{(k)} + t(x^* - x^{(k)}))$$

definieren. Durch die multivariate Kettenregel erhalten wir

$$g'(t) = DF(x^{(k)} + t(x^* - x^{(k)}))(x^* - x^{(k)}),$$

woraus nach dem Hauptsatz der Differential- und Integralrechnung wie in Lemma 5.7 folgt, dass

$$F(x^*) - F(x^{(k)}) = g(1) - g(0) = \int_0^1 g'(t)dt = \int_0^1 DF(x^{(k)} + t(x^* - x^{(k)}))(x^* - x^{(k)})dt.$$

Jetzt setzen wir wie oben beschrieben voraus, dass $x^{(k)} \in U$. Nach der Definition der Newton-Iteration gilt dann, dass

$$\begin{aligned}
A &:= x^{(k+1)} - x^* \\
&= x^{(k)} - [DF(x^{(k)})]^{-1} F(x^{(k)}) - x^* \\
&= x^{(k)} - x^* - [DF(x^{(k)})]^{-1}(F(x^{(k)}) - F(x^*)) \\
&= [DF(x^{(k)})]^{-1} \left(F(x^*) - F(x^{(k)}) - DF(x^{(k)})(x^* - x^{(k)})\right) \\
&= [DF(x^{(k)})]^{-1} \left(g(1) - g(0) - DF(x^{(k)})(x^* - x^{(k)})\right) \\
&= [DF(x^{(k)})]^{-1} \left(\int_0^1 DF(x^{(k)} + t(x^* - x^{(k)}))(x^* - x^{(k)})dt - DF(x^{(k)})(x^* - x^{(k)})\right) \\
&= \int_0^1 [DF(x^{(k)})]^{-1} \left\{DF(x^{(k)} + t(x^* - x^{(k)})) - DF(x^{(k)})\right\} (x^* - x^{(k)})dt
\end{aligned}$$

Gehen wir zur Norm davon über, so erhalten wir

$$\begin{aligned}
\|A\| &= \|x^{(k+1)} - x^*\| \\
&\leq \int_0^1 \|[DF(x^{(k)})]^{-1} \left\{DF(x^{(k)} + t(x^* - x^{(k)})) - DF(x^{(k)})\right\}\| \, \|x^* - x^{(k)}\|dt
\end{aligned}$$

Nach der ersten Voraussetzung [B3] gilt aber, dass

$$\begin{aligned}
& [DF(x^{(k)})]^{-1} \left(DF(x^{(k)} + t(x^* - x^{(k)})) - DF(x^{(k)})\right) \\
\leq\ & \omega \|x^{(k)} - (x^{(k)} + t(x^* - x^{(k)}))\| \\
=\ & \omega t \|x^{(k)} - x^*\|.
\end{aligned}$$

Setzen wir dieses Ergebnis in die obige Ungleichung ein, so ergibt sich

$$\begin{aligned}
\|A\| &= \|x^{(k+1)} - x^*\| \\
&\leq \int_0^1 \|[DF(x^{(k)})]^{-1} \left(DF(x^{(k)} + t(x^* - x^{(k)})) - DF(x^{(k)})\right)\| \, \|x^* - x^{(k)}\| dt \\
&\leq \int_0^1 \omega t \|x^{(k)} - x^*\| \, \|x^{(k)} - x^*\| dt = \int \omega t \|x^{(k)} - x^*\|^2 dt \\
&= \frac{\omega}{2} \|x^{(k)} - x^*\|^2.
\end{aligned}$$

Damit ist also gezeigt, dass (5.23) gilt, falls $x^{(k)} \in U$.

Teil 2:
Mit Hilfe der Aussage aus Teil 1 können wir nun Satz 5.22 per Induktion beweisen.

Induktionsanfang Für $k = 0$ sind nur (5.22) und (5.23) zu zeigen.

- (5.22): Für $k = 0$ erhält man $\left(\frac{\omega\rho}{2}\right)^{2^k - 1} = 1$. Daher gilt $\|x^{(0)} - x^*\| = \rho$ nach der Definition von ρ.
- (5.23): Weil $x^{(0)} \in U$ können wir Teil 1 des Beweises verwenden. Wir erhalten:
$$\|x^{(1)} - x^*\| \leq \frac{\omega}{2} \|x^{(0)} - x^*\|^2.$$

Induktionsschritt: $k \to k+1$. Sei also der Satz richtig für k. Dann ist $x^{(k)} \in U$ nach der Induktionsannahme. Wir rechnen

$$\begin{aligned}
\|x^{(k+1)} - x^*\| &\leq \frac{\omega}{2} \|x^{(k)} - x^*\|^2 \quad \text{wegen Teil 1} \\
&\leq \frac{\omega}{2} \left(\frac{\omega\rho}{2}\right)^{2(2^k - 1)} \rho^2 \\
&\qquad \text{denn es gilt (5.22) nach Induktionsannahme} \\
&= \left(\frac{\omega\rho}{2}\right)^{2^{k+1} - 1} \rho < \rho \text{ wegen [B3], Teil (b).}
\end{aligned}$$

Aus der letzten Zeile folgt die a-priori Fehlerschranke (5.22) für $k+1$, sowie die Aussage $x^{(k+1)} \in B_\rho(x^*)$. Entsprechend ist die Folge wohldefiniert, und nach Teil 1 gilt auch (5.23) für $k + 1$.

Wegen (5.22) und [B3], Teil (b) erhält man außerdem direkt die Konvergenz gegen x^*.

QED

Zum Abschluss geben wir noch einen Satz an, der zeigt, dass eine lokale quadratische Konvergenz in der Nähe eine Nullstelle meistens erreicht werden kann.

Satz 5.23 *Sei $U \subseteq \mathbb{R}^n$ offen und $F : U \to \mathbb{R}^n$ eine zweimal stetig differenzierbare Funktion. Sei $x^* \in U$ mit $F(x^*) = 0$ und $\det(DF(x^*)) \neq 0$. Dann existiert ein $\rho > 0$ so dass das Newton-Verfahren für alle Startwerte $x^{(0)} \in U := B_\rho(x^*)$ quadratisch konvergiert.*

Beweis: Im Beweis untersuchen wir die Voraussetzungen von Satz 5.22. Zunächst gilt [B1] nach Voraussetzung. Weil die Funktion

$$h(x) = [DF(x)]^{-1}$$

als Matrixinversion stetig ist, gibt es $\rho > 0$ so dass

$$\|h(x)\| - \|h(x^*)\| \leq \|h(x) - h(x^*)\| \leq \varepsilon \text{ für alle } x \in B_\rho(x^*).$$

Mit $\varepsilon := \|[DF(x^*)]^{-1}\|$ ergibt das

$$\|[DF(x)]^{-1}\| - \|[DF(x^*)]^{-1}\| \leq \|[DF(x^*)]^{-1}\| \text{ für alle } x \in B_\rho(x^*),$$

oder, äquivalent,

$$\|[DF(x)]^{-1}\| \leq 2\|[DF(x^*)]^{-1}\| \text{ für alle } x \in B_\rho(x^*).$$

Daraus folgt [B2] und mit der Definition $U := B_\rho(x^*)$ trivialerweise [B4].

Als letztes muss also noch [B3] gezeigt werden. Hierfür nutzt man aus, dass DF nach Voraussetzung für $x \in U$ differenzierbar ist. Also gibt es eine Lipschitzkonstante $L > 0$ so dass

$$\|[DF(x)] - [DF(y)]\| \leq L\|x - y\| \text{ für alle } x, y \in U.$$

Wählt man $\omega := 2L\|[DF(x^*)]^{-1}\|$ so gilt

$$\begin{aligned} \|[DF(x)]^{-1}([DF(y)] - [DF(x)])\| &\leq \underbrace{\|[DF(x)]^{-1}\|}\ \|\underbrace{([DF(y)] - [DF(x)])}\| \\ &\leq 2\|[DF(x^*)]^{-1}\|\ \ L\|x - y\| \\ &= \omega\|x - y\|, \end{aligned}$$

also gilt [B3], Teil (a). Da man immer $\rho < \frac{2}{\omega}$ wählen kann, folgt wegen $\frac{\omega}{2}\rho < 1$ auch Teil (b) von [B3], und damit die quadratische Konvergenz nach Satz 5.22. QED

Kapitel 6

Interpolation

In diesem Kapitel beschäftigen wir uns mit der Interpolation von Funktionen. Dazu wollen wir aus einer gegebenen Klasse von Funktionen $\mathcal{M}$ eine auswählen, die an vorgegebenen Punkten $x_0, x_1, \ldots, x_n$ ihres Definitionsbereichs gewissen Bedingungen genügt. Im einfachsten Fall fordert man z.B.

$$f(x_i) = y_i \text{ für } i = 0, \ldots, n, \quad x_i, y_i \text{ gegeben},$$

man kann aber auch Bedingungen an die Ableitungen in den Punkten stellen. Ist $\mathcal{M}$ die Klasse der Polynome vom Grad $\leq n$, so spricht man von **Polynom-Interpolation**, bei trigonometrischen Funktionen von **trigonometrischer Interpolation** und ist M die Klasse der stückweise polynomialen Funktionen, so nennt man das Problem **Spline-Interpolation**.

6.1 Polynomiale Interpolation

Definition 6.1 *Ein* ***Polynom*** p *ist eine Funktion von der Form* $p(x) = a_n x^n + \ldots + a_1 x + a_0$, $x \in \mathbb{K}$ *und Koeffizienten* $a_0, \ldots, a_n \in \mathbb{K}$. *Ist* $a_n \neq 0$, *so heißt* n *der* ***Grad*** *des Polynoms. Per Definition ist der Grad von* $p \equiv 0$ *als* -1 *festgesetzt.* Π_n *sei die Menge aller Polynome vom Grad* $\leq n$.

Aus der linearen Algebra ist bekannt, dass Π_n ein Vektorraum mit komponentenweiser Addition und Skalarmultiplikation ist. Weiterhin wiederholen wir

Satz 6.2 *(Hauptsatz der Algebra) Ist* $p(x) = a_n x^n + \ldots + a_1 x + a_0$ *ein komplexes Polynom vom Grad* n, *so gibt es eindeutig bestimmte Zahlen* $b_1, \ldots, b_n \in \mathbb{C}$ *so, dass* $p(x) = a_n(x - b_1) \cdot \ldots \cdot (x - b_n)$. *Die Zahlen* b_j *sind die* Nullstellen *von* p.

Kommt der Faktor $x - b_j$ in $p(x)$ genau k-mal vor, sagt man, die Nullstelle b_j hat die *Vielfachheit* k.

Bemerkung: Sei a eine Nullstelle von p. Dann hat a die Vielfachheit k genau dann, wenn $p^{(j)}(a) = 0$ für $j = 0, 1, \ldots, k-1$.

Satz 6.3 *Sei $p(x) = a_n x^n + \ldots + a_1 x + a_0$ ein Polynom $\in \Pi_n$. Hat p mehr als n Nullstellen, so verschwindet p identisch, das heißt $p \equiv 0$. Insbesondere gilt dann $a_j = 0$ für alle $j = 0, \ldots, n$.*

Aus diesem Satz lässt sich direkt ableiten, dass die Monome

$$M_k(x) := x^k \in \Pi_k, k = 0, 1, \ldots, n$$

als Funktionen $M_k : [a, b] \to \mathbb{R}$, $[a, b] \subseteq \mathbb{R}$, linear unabhängig sind. Da man durch Linearkombination der M_k jedes Polynom erzeugen kann, ist

$$\{M_0, M_1, \ldots, M_n\}$$

also eine Basis des Π_n.
Die lineare Unabhängigkeit der M_j sieht man wie folgt:

> Sei $\sum_{k=0}^{n} \alpha_k M_k(x) = 0$ für alle $x \in [a, b]$. Dann hat $\sum_{k=0}^{n} \alpha_k M_k(x)$ mehr als n Nullstellen, also sind nach Satz 6.3 alle Koeffizienten $\alpha_0 = \alpha_1 = \ldots = \alpha_n = 0$.

Um Polynome auszuwerten, das heißt Werte $p(x)$ eines Polynoms p zu berechnen, verwendet man das Horner-Schema. Dazu klammert man das Polynom $p(x)$ geschickt und erhält:

$$p(x) = (\ldots((a_n x + a_{n-1})x + a_{n-2})x + \ldots + a_1)x + a_0$$

Das führt zu folgendem Verfahren:

Algorithmus 12: Horner-Schema zur Auswertung von Polynomen

Input: `Koeffizienten` $a_0, a_1, \ldots, a_n$ `eines Polynoms` $p = a_n x^n + \ldots a_1 x_1 + a_0 \in \Pi_n$ `und feste Zahl` x.

Schritt 1: $y := a_n$

Schritt 2: **For** $k = n - 1$ `to` 0 **do** $y := y \cdot x + a_k$

Ergebnis: $p(x) := y$

Die Berechnung mittels des Horner-Schemas ist effizienter als die „normale“ Auswertung.

Beispiel: Sei ein Polynom

$$p(x) = 2x^4 - 4x^3 - 5x^2 + 7x + 11$$

gegeben. Gesucht ist der Wert an der Stelle $x = 2$. In tabellarischer Schreibweise erhält man:

Koeffizienten:	2		-4		-5		7		11
Zahl: $x = 2$		$\cdot 2 \nearrow$	4	$\cdot 2 \nearrow$	0	$\cdot 2 \nearrow$	-10	$\cdot 2 \nearrow$	-6
Summe:	2		0		-5		-3		$\underline{5}$

Der gesuchte Wert ist also $p(2) = 5$.

Wir definieren nun das Problem, mit dem wir uns in diesem Abschnitt beschäftigen:

Lagrange Interpolationsaufgabe: Gegeben seien $n + 1$ Stützstellen x_i, $i = 0, \ldots, n$ und Stützwerte y_i, $i = 0, \ldots, n$. Gesucht ist ein Polynom $p \in \Pi_n$, so dass

$$p(x_i) = y_i \text{ für } i = 0, \ldots, n. \tag{L-Int}$$

Als erste Idee verwendet man die Monome als Basis des Π_n und stellt das gesuchte Polynom dar durch

$$p(x) = \sum_{k=0}^{n} \alpha_k M_k(x).$$

Die Bedingungen L-Int führen zu folgendem Gleichungssystem mit Unbekannten $\alpha_0, \ldots, \alpha_n$:

$$\sum_{k=0}^{n} \alpha_k M_k(x_j) = y_j \text{ für } j = 0, \ldots, n$$

oder, ausgeschrieben,

$$\sum_{k=0}^{n} \alpha_k {x_j}^k = y_j \text{ für } j = 0, \ldots, n.$$

Die Koeffizienten-Matrix ist die *Vandermonde-Matrix*

$$A = \begin{pmatrix} 1 & x_0 & {x_0}^2 & \cdots & {x_0}^n \\ 1 & x_1 & {x_1}^2 & \cdots & {x_1}^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & {x_n}^2 & \cdots & {x_n}^n \end{pmatrix} \in \mathbb{K}^{n+1,n+1}.$$

Das Problem ist im Allgemeinen schlecht konditioniert, aber dennoch von theoretischem Interesse:

Satz 6.4 *Die Lagrange Interpolationsaufgabe ist für $n+1$ paarweise verschiedene Stützstellen $x_0, \ldots, x_n$ eindeutig lösbar, die Lösung ist gegeben durch*

$$L_n(x) = \sum_{k=0}^{n} y_k l_k(x)$$

wobei

$$l_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j} \text{ für } k = 0, \ldots, n$$

die so genannten ***Lagrange-Polynome*** *sind.*

Beweis: Per Konstruktion ist $L_n \in \Pi_n$ und es gilt wegen $l_k(x_j) = \begin{cases} 1 & \text{falls } k = j \\ 0 & \text{falls } k \neq j \end{cases}$:

$$L_n(x_j) = \sum_{k=0}^{n} y_k l_k(x_j) = \sum_{k=0}^{n} y_k \delta_{k_j} = y_j \text{ für } j = 0, \ldots, n$$

Es bleibt noch die Eindeutigkeit zu zeigen. Seien dazu p_1, $p_2 \in \Pi_n$ beides Polynome, die (L-Int) erfüllen. Dann gilt für ihre Differenz $p := p_1 - p_2$, dass

$$p(x_j) = p_1(x_j) - p_2(x_j) = y_j - y_j = 0 \text{ für } j = 0, \ldots, n$$

Also hat das Polynom p (mindestens) $n+1$ Nullstellen. Da $p \in \Pi_n$ folgt daraus nach Satz 6.3, dass $p \equiv 0$, also $p_1 \equiv p_2$. QED

Beispiel: Seien drei Stützstellen $x_0 = 0$, $x_1 = 1$ und $x_2 = 3$ mit Stützwerten $y_0 = 1$, $y_1 = 3$ und $y_2 = 2$ gegeben. Gesucht ist der Wert $L_2(x)$ des interpolierenden Lagrange-Polynoms an der Stelle 2. Es gilt:

$$L_2(x) = \sum_{k=0}^{2} y_k l_k(x).$$

Für $x = 2$ gilt:

$$\begin{aligned} l_0(x) &= \frac{(x-1)(x-3)}{(0-1)(0-3)} = \frac{1}{3} \cdot (x-1) \cdot (x-3) \\ l_1(x) &= \frac{(x-0)(x-3)}{(1-0)(1-3)} = -\frac{1}{2} \cdot x \cdot (x-3) \\ l_2(x) &= \frac{(x-0)(x-1)}{(3-0)(3-1)} = \frac{1}{6} \cdot x \cdot (x-1) \end{aligned}$$

$$\begin{aligned} \Rightarrow L_2(2) &= 1 \cdot l_0(2) + 3 \cdot l_1(2) + 2 \cdot l_2(2) \\ &= -\tfrac{1}{3} + 3 + \tfrac{2}{3} = \tfrac{10}{3} \end{aligned}$$

Praktisch hat die Lagrange-Formel allerdings wenig Relevanz, da die Hinzunahme einer weiteren Stützstelle eine komplette Neuberechnung erfordert, und das Problem nicht gut konditioniert ist.
Möchte man das interpolierende Polynom an nur wenigen Stellen auswerten, bietet sich das folgende Verfahren an.

Interpolation von Neville & Aitken

Definition 6.5 *Für gegebene, paarweise verschiedene Stützstellen x_i mit $i = 0, \ldots, n$ und Stützwerte y_i mit $i = 0, \ldots, n$ sei $P_i^k \in \Pi_k$ das Polynom mit der Eigenschaft:*

$$P_i^k(x_j) = y_j \quad \forall i \leq j \leq i+k$$

Insbesondere ist P_0^n das Interpolationspolynom zu allen Daten.

Wir bemerken, dass P_i^k wegen Satz 6.4 eindeutig bestimmt ist. Die Idee des nun zu entwickelnden Verfahrens beruht auf dem folgenden Satz.

Satz 6.6 *Es gilt:*

$$\begin{aligned} P_i^0(x) &= y_i \in \Pi_0 && i = 0, \ldots, n \\ P_i^{k+1}(x) &= \frac{(x - x_i)P_{i+1}^k(x) - (x - x_{i+k+1})P_i^k(x)}{x_{i+k+1} - x_i} && 0 \leq i \leq n-k-1 \end{aligned}$$

Beweis: Wir führen Induktion nach k durch.
Für $k = 0$ erfüllt $P_i^0(x) = y_i$ gerade $P_i^0(x_i) = y_i$ für $i = 0, \ldots, n$. Nehmen wir nun an, die Aussage stimmt für k. Bezeichne mit $h(x)$ die rechte Seite der Rekursionsformel für $P_i^{k+1}(x)$, das heißt

$$h(x_j) = \frac{(x_j - x_i)P_{i+1}^k(x_j) - (x_j - x_{i+k+1})P_i^k(x_j)}{x_{i+k+1} - x_i}$$

Fall 1: $i < j \leq i + k$:
Nach Induktionsannahme gilt

$$P_{i+1}^k(x_j) = y_j \text{ und } P_i^k(x_j) = y_j.$$

Also folgt:

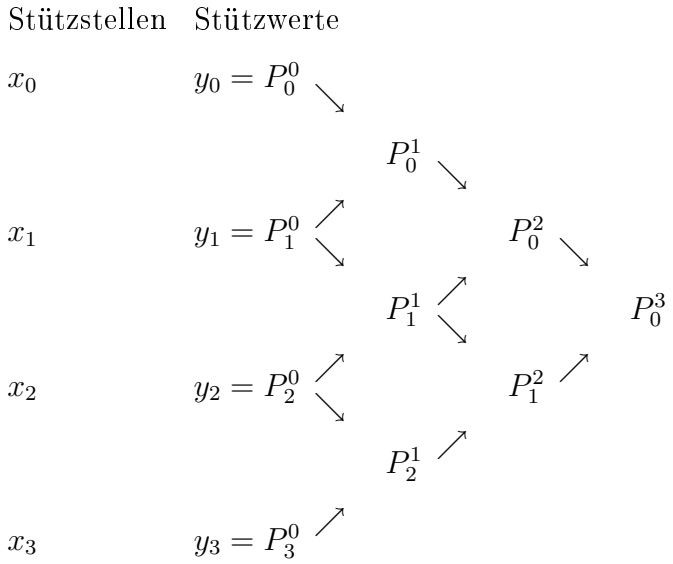
$$h(x_j) = \frac{(x_j - x_i)y_j - (x_j - x_{i+k+1})y_j}{x_{i+k+1} - x_i} = y_j$$

Fall 2: $j = i$:
In diesem Fall gilt $P_i^k(x_j) = y_j = y_i$, woraus wir aber schon folgern, dass

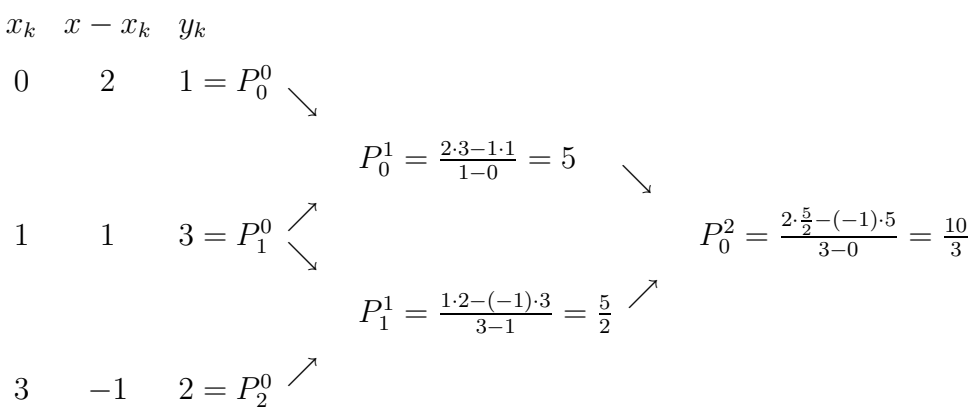
$$h(x_i) = \frac{(x_i - x_i)P_{i+1}^k(x_i) - (x_i - x_{i+k+1})y_i}{x_{i+k+1} - x_i} = y_i$$

Fall 3: $j = k + i + 1$:
Analog zu Fall 2. QED

Wir zeigen am Beispiel $n = 3$, wie sich die Polynome P_i^k effizient auswerten lassen.



An unserem alten Beispiel mit $\{(x_i, y_i), i = 0, 1, 2\} = \{(0, 1), (1, 3), (3, 2)\}$ sieht das folgendermaßen aus: Gesucht ist wieder der Wert des interpolierenden Polynoms an $x = 2$. Dazu rechnet man:



Als Algorithmus lässt sich das wie folgt beschreiben:

Algorithmus 13: Neville-Aitken-Verfahren

Input: $x_0, x_1, \ldots, x_n$ `paarweise verschieden` $y_1, \ldots, y_n$, `Punkt` x, `an dem das interpolierende Polynom ausgewertet werden soll.`

Schritt 1: **For** $j = 0, \ldots, n$ **do**

Schritt 1.1: $p_j := y_j$

Schritt 1.2: $t_j := x - x_j$

Schritt 2: **For** $k = 0, \ldots, n-1$ **do**

For $j = 0, \ldots, n-k-1$ **do**

$$p_j := \frac{t_j p_{j+1} - t_{j+k+1} p_j}{t_j - t_{j+k+1}}$$

Ergebnis: $p(x) := p_0$, `wobei` p `das Interpolationspolynom ist.`

Dieses Verfahren ist sinnvoll, falls man das Interpolationspolynom an nur wenigen Stellen auswerten möchte.

Newtonsche Interpolationsformel

In der Newtonschen Interpolationsformel verwendet man eine weitere Basis des Π_n, nämlich

$$h_k(x) = \prod_{i=0}^{k-1} (x - x_i)$$

wobei $x_0, x_1, \ldots, x_n$ wieder die Stützstellen sind.

Lemma 6.7 *Seien $x_0, x_1, \ldots, x_{n-1}$ paarweise verschieden. Die Newton-Polynome $h_k(x) = \prod_{i=0}^{k-1}(x - x_i)$ mit $k = 0, 1, \ldots, n$ bilden eine Basis des Π_n.*

Die Newton-Polynome haben das folgende Aussehen:

$$\begin{aligned} h_0(x) &= 1 \\ h_1(x) &= (x - x_0) \\ h_2(x) &= (x - x_1)(x - x_0) \\ &\vdots \end{aligned}$$

und es gilt:

$$h_k(x_j) = 0 \text{ für alle } k > j \quad \text{und} \quad h_k(x_j) \neq 0 \text{ für alle } k \leq j$$

Beweis: (von Lemma 6.7) Sei $\sum_{k=0}^{n} \alpha_k h_k(x) = 0$, das heißt

$$p(x) = \sum_{k=0}^{n} \alpha_k \prod_{i=0}^{k-1} (x - x_i) = 0$$

Insbesondere gilt

$$0 = p(x_0) = \alpha_0 h_0(x) = \alpha_0.$$

Daraus folgern wir weiter

$$0 = p(x_1) = \alpha_0 + \alpha_1(x - x_0) = \alpha_1 \underbrace{(x - x_0)}_{\neq 0},$$

also $\alpha_1 = 0$. Induktiv erhält man, dass alle Koeffizienten $\alpha_0, \ldots, \alpha_n$ Null sind.

QED

Um das Lagrange-Interpolationsproblem zu lösen, betrachtet man also das folgende Gleichungssystem mit den Variablen $\alpha_0, \ldots, \alpha_n$:

$$\sum_{k=0}^{n} \alpha_k h_k(x_j) = y_j \qquad \text{mit } j = 0, \ldots, n. \tag{6.1}$$

Weil $\sum_{k=0}^{n} \alpha_k h_k(x_j) = \sum_{k=0}^{j} \alpha_k h_k(x_j)$ erhält man die folgende Koeffizienten-Matrix:

$$A = \begin{pmatrix} h_0(x_0) & 0 & \ldots\ldots\ldots\ldots & 0 \\ h_0(x_1) & h_1(x_1) & 0 \quad \ldots\ldots & 0 \\ h_0(x_2) & h_1(x_2) & h_2(x_2) \quad \ldots\ldots & \vdots \\ \vdots & & \ddots & 0 \\ h_0(x_n) & h_1(x_n) & \ldots\ldots\ldots\ldots & h_n(x_n) \end{pmatrix}$$

A ist eine untere Dreiecks-Matrix, die wegen $h_k(x_k) \neq 0$ für alle $k = 0, \ldots, n$ regulär ist. Man kann die gesuchten Koeffizienten also durch Vorwärtselimination bestimmen. Das ergibt:

$$\begin{aligned} \alpha_0 &= \frac{y_0}{h_0(x_0)} = \frac{y_0}{1} = y_0 \\ \alpha_1 &= \frac{1}{h_1(x_1)}(y_1 - \alpha_0 h_0(x_1)) = \frac{1}{x - x_0}(y_1 - y_0) \\ \alpha_2 &= \frac{1}{h_2(x_2)}(y_2 - \alpha_1 h_1(x_2) - \alpha_0 h_0(x_2)) \\ &= \frac{1}{(x_2 - x_0)(x_2 - x_1)} \left(y_2 - \frac{1}{x - x_0}(y_1 - y_0) - y_0 \right) \\ &\vdots \end{aligned}$$

Da diese Formeln recht mühsam werden, gehen wir einen anderen Weg. Dazu definieren wir zunächst „Abschnittspolynome“ für die Lösung $\alpha_0, \alpha_1, \ldots, \alpha_n$ von (6.1):

$$Q^k(x) = \sum_{j=0}^{k} \alpha_j h_j(x) = \alpha_0 + \alpha_1(x - x_0) + \ldots + \alpha_k(x - x_0) \cdot \ldots \cdot \ldots \cdot (x - x_{k-1})$$

das heißt

$$\begin{aligned} Q^0(x) &= \alpha_0 \\ Q^1(x) &= \alpha_0 + \alpha_1(x - x_0) \\ Q^2(x) &= \alpha_0 + \alpha_1(x - x_0) + \alpha_2(x - x_0)(x - x_1) \\ &\vdots \end{aligned}$$

Es gelten die folgenden Eigenschaften:

Lemma 6.8

1. $Q^k(x) = P_0^k \in \Pi_k$
2. $P_0^{k+1}(x) = P_0^k(x) + \alpha_{k+1} h_{k+1}(x)$
3. *α_k ist der Koeffizient von x^k im Polynom $P_0^k(x)$.*

Beweis: Eigenschaft 1 folgt, weil aufgrund der Wahl der α_i gilt $Q^k(x_j) = y_j$ für $j = 0, \ldots, k$ und die Polynom-Interpolation eindeutig ist (Satz 6.4).
2. und 3. ergeben sich für Q^k aus der Konstruktion und gelten nach 1. also auch für P_0^k. QED

Wir definieren:

Definition 6.9 *Seien x_i, y_i mit $i = 0, \ldots, n$ mit paarweise verschiedenen x_i gegeben. Dann definiert man die **dividierten Differenzen** rekursiv durch*

$$\begin{aligned} D_i^0 &:= y_i && \text{mit } i = 0, \ldots, n \\ D_i^k &:= \frac{D_{i+1}^{k-1} - D_i^{k-1}}{x_{i+k} - x_i} && \text{mit } i = 0, 1, \ldots, n-k \\ & && \text{und } k = 1, 2, \ldots, n \end{aligned}$$

Unser Ziel ist es nun, zu beweisen, dass

$$\alpha_k = D_0^k$$

gilt. Wir werden das zuerst untersuchen und danach ein Schema angeben, mit dem man D_i^k effizient berechnen kann.

Satz 6.10 *Es gilt*

$$P_i^k(x) = D_i^0 + D_i^1(x - x_i) + \ldots + D_i^k(x - x_i) \cdot \ldots \cdot (x - x_{i+k-1})$$

das heißt $P_i^k(x_j) = y_j$ *für* $j = i, \ldots, k + i$. *Insbesondere gilt für* $i = 0$ *und* $k = n$*:*

$$\begin{aligned} P_0^n(x) &= D_0^0 + D_0^1(x - x_0) + \ldots + D_0^n(x - x_0) \cdot \ldots \cdot (x - x_{n-1}) \\ &= \sum_{j=0}^{n} D_0^j h_j(x) \end{aligned}$$

ist die Lösung des Lagrange-Interpolationsproblemes.

Beweis: Wir induzieren über k. Sei $k = 0$, dann ist $P_i^0(x) = D_i^0 = y_i$ richtig. Sei die Aussage richtig für $k - 1$ und $k \geq 1$. Wir verwenden Lemma 6.8, Teil (2), indem wir zunächst

$$\begin{aligned} \tilde{x}_0 &:= x_i \\ \tilde{x}_1 &:= x_{i+1} \\ &\vdots \\ \tilde{x}_{k-1} &:= x_{i+k-1} \\ \tilde{x}_k &:= x_{i+k} \end{aligned}$$

definieren. Bezüglich der $\tilde{x}$ definieren wir nun die Polynome $\tilde{P}_0^{k-1}, \tilde{P}_0^k$, das Newton-Polynom $\tilde{h}_k(x) = (x - \tilde{x}_0) \cdots (x - \tilde{x}_{k-1})$ und die Koeffizienten $\tilde{\alpha}_j$ mit $j = 0, \ldots, k$. Dann gilt

$$\begin{aligned} P_i^k = \tilde{P}_0^k(x) &= \tilde{P}_0^{k-1}(x) + \tilde{\alpha}_k \tilde{h}_k(x) \text{ nach Lemma 6.8 Teil (2)} \\ &= P_i^{k-1}(x) + a(x - x_i) \cdots (x - x_{i+k-1}) \end{aligned}$$

Wobei $a := \tilde{\alpha}_k$ der (noch unbekannte) Koeffizient von x^k im Polynom $\tilde{P}_0^k = P_i^k$ ist, nach 6.8 Teil (3). Nach der Induktionsannahme ist

$$P_i^{k-1}(x) = D_i^0 + D_i^1(x - x_i) + \ldots + D_i^{k-1}(x - x_i) \ldots (x - x_{i+k-2})$$

und entsprechend

$$P_i^k(x) = D_i^0 + D_i^1(x - x_i) + \ldots + D_i^{k-1}(x - x_i) \cdots (x - x_{i+k-2}) + a(x - x_i) \cdots (x - x_{i+k-1})$$

Es ist also $a = D_i^k$ zu zeigen. Nach Induktionsannahme gilt

- der höchste Koeffizient von P_i^{k-1} ist D_i^{k-1}
- der höchste Koeffizient von P_{i+1}^{k-1} ist D_{i+1}^{k-1}

Also ist

$$P_i^{k-1}(x) = \tilde{p}(x) + D_i^{k-1}x^{k-1}$$
$$P_{i+1}^{k-1}(x) = \tilde{p}'(x) + D_{i+1}^{k-1}x^{k-1} \text{ mit } \tilde{p}, \tilde{p}' \in \Pi_{k-2}$$

Wir verwenden nun die Nevillsche Interpolationsformel aus Satz 6.6 und erhalten

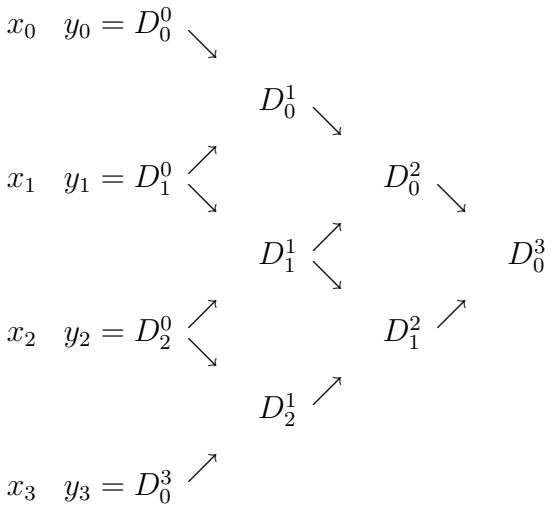
$$P_i^k(x) = \frac{(x - x_i)P_{i+1}^{k-1}(x) - (x - x_{i+k})P_i^{k-1}(x)}{x_{i+k} - x_i} = p' + \frac{x^k D_{i+1}^{k-1} - x^k D_i^{k-1}}{x_{i+k} - x_i}$$

mit $p' \in \Pi_{k-1}$. Der höchste Koeffizient auf der rechten Seite ist also

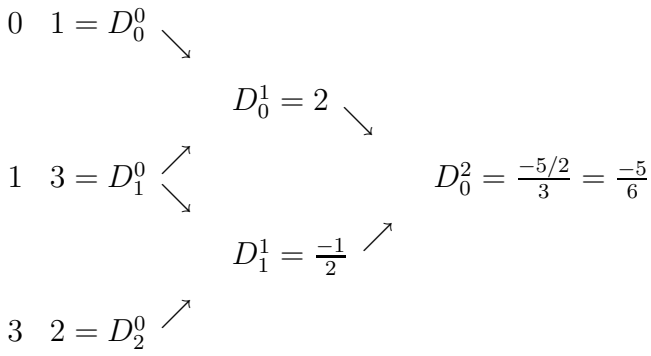
$$a = \frac{D_{i+1}^{k-1} - D_i^{k-1}}{x_{i+k} - x_i} = D_i^k$$

QED

Um die D_i^k zu berechnen, benötigt man mittels des folgenden Schemas einen Aufwand von $O(n^2)$:



Verwenden wir auch hier das Beispiel mit den Paaren $(0, 1)$, $(1, 3)$, $(3, 2)$, so erhalten wir



und das interpolierende Polynom ergibt sich zu

$$\begin{aligned} P_0^2(x) &= 1 + 2 \cdot (x - x_0) - \tfrac{5}{6}(x - x_0)(x - x_1) \\ &= 1 + 2x - \tfrac{5}{6}x(x - 1). \end{aligned}$$

An $x = 2$ erhalten wir wiederum $P_0^2(2) = \frac{10}{3}$. Abschließend geben wir noch eine andere analytische Darstellung der D_i^k.

Lemma 6.11

$$D_i^k = \sum_{j=i}^{i+k} y_j \left(\prod_{\substack{r=i \\ r \neq j}}^{i+k} \frac{1}{x_j - x_r} \right) \qquad \begin{matrix} i = 0, \ldots, n-k \\ k = 0, \ldots, n \end{matrix}$$

Beweis: Kann durch vollständige Induktion geführt werden.

Statt nur Funktionswerte von Punkten vorzugeben, kann man auch Bedingungen an die Ableitungen stellen. Man erhält das folgende Problem.

Hermite-Interpolationsproblem: Seien $x_0 \leq \cdots \leq x_n \in [a, b]$ und $y_0, \ldots, y_n \in \mathbb{R}$ gegeben, wobei die x_i nicht paarweise disjunkt sein müssen. Gesucht ist ein Polynom p, das folgende Bedingungen erfüllt: Ist $x_{i-1} < x_i = x_{i+1} = \cdots = x_{i+r} < x_{i+r+1}$, so soll gelten

$$\begin{aligned} p(x_i) &= y_i \\ p^{(1)}(x_i) &= y_{i+1} \\ &\vdots \\ p^{(r)}(x_i) &= y_{i+r} \end{aligned} \tag{6.2}$$

Meistens wird dieses Problem gestellt, um eine (unbekannte) Funktion f zu interpolieren. Sind die $x_0, \ldots, x_n$ gegeben, so bedeutet die Bedingung (6.2) das folgende: Fallen r der $x_0, \ldots, x_n$ in einem Punkt $z \in [a, b]$ zusammen, so interpoliert p die Funktion f an der Stelle z bis zur $(r-1)$-ten Ableitung. Natürlich setzt man dabei implizit voraus, dass f auch hinreichend oft stetig differenzierbar ist. Die meisten Ergebnisse für das Lagrange-Interpolationsproblem kann man auf das Hermite-Interpolationsproblem verallgemeinern.

Satz 6.12 *Es gibt genau ein Polynom $p \in \Pi_n$, welches das Hermite-Interpolationsproblem löst.*

Beweis: Wir betrachten die folgende lineare Abbildung

$$\begin{aligned} T : C^n[a, b] &\to \mathbb{R}^{n+1} \\ f &\to (f^{(0)}(x_0), f^{(r_1)}(x_1), \ldots, f^{(r_n)}(x_n)) \end{aligned}$$

wobei $r_j = r$, falls $x_{i-1} < x_i = x_{i+1} = \cdots = x_{i+r}$ mit $j = i + r$. Wenn man T auf Π_n anwendet, ergibt sich

$$T : \Pi_n \to \mathbb{R}^{n+1}$$

mit $\dim(\Pi_n) = n + 1$. Ist T injektiv, dann auch surjektiv und jedes Hermite-Interpolationsproblem hat genau eine Lösung. Wir müssen also die Injektivität von T nachweisen. Sei dazu $T_{p_1} = T_{p_2}$, $p_1, p_2 \in \Pi_n$. Dann ist

$$T(p_1 - p_2) = T_{p_1} - T_{p_2} = 0 \in \mathbb{R}^{n+1}$$

Also hat das Polynom $p_1 - p_2 \in \Pi_n$ mehr als n Nullstellen (mit Vielfachheiten gezählt) und ist nach Satz 6.3 identisch Null, d.h. $p_1 \equiv p_2$. QED

Um das Hermite-Interpolationsproblem zu lösen, betrachtet man zunächst den Spezialfall

$$x_0 = x_1 = \cdots = x_k$$

Hier ist also ein Polynom $p \in \Pi_k$ gesucht, das vorgegebene Bedingungen an den Funktionswert y_0 und an die Werte seiner ersten k Ableitungen

$$p^{(i)}(x_0) = y_i$$

erfüllt. Stellt man sich vor, dass die Werte $y_0, \ldots, y_k$ von einer (unbekannten) k mal stetig differenzierbaren Funktion f kommen, d.h.

$$f^{(i)}(x_0) = y_i$$

gilt, so sieht man, dass das interpolierende Polynom genau das Taylorpolynom

$$p(x) = \sum_{i=0}^{k} \frac{(x - x_0)^i}{i!} f^{(i)}(x_0)$$

ist. Diese Beobachtung verwendet man zur Berechnung der dividierten Differenzen D_i^k wie folgt.

Definition 6.13 *Die dividierten Differenzen werden für das Hermite-Interpolationsproblem wie folgt definiert*

$$D_i^0 = y_i \quad mit \quad j = \min\{l : x_l = x_i\} \qquad i = 0, \ldots, n$$

$$D_i^k = \begin{cases} \frac{y_{i+k}}{k!} & falls\ x_i = x_{i+1} = \cdots = x_k \\ \frac{D_{i+1}^{k-1} - D_i^{k-1}}{x_{i+k} - x_i} & falls\ x_i \neq x_{i+k} \end{cases}$$

Mit dieser Definition kann man zeigen, dass Satz 6.10 richtig bleibt. Bei seiner Anwendung ist allerdings zu beachten, dass manche der auftretenden Faktoren in den Newton-Polynomen $h_k(x)$ identisch sind. Wir wollen Satz 6.10 für das Hermite-Interpolationsproblem hier nicht beweisen, aber seine Anwendung an einem Beispiel demonstrieren.

Beispiel:

$$\begin{array}{llll} x_0 = -2 & x_1 = 0 & x_2 = 0 & x_3 = 1 \\ y_0 = 6 & y_1 = 2 & y_2 = 4 & y_3 = 8 \end{array}$$

Berechnung der D_i^k

$$\begin{array}{llllll} -2 & 6 = D_0^0 & \searrow & & & \\ & & & D_0^1 = -2 & \searrow & \\ 0 & 2 = D_1^0 & \nearrow & & & D_0^2 = 3 \quad \searrow \\ & & \searrow & & & \\ & & & D_1^1 = \quad 4 & \nearrow & \qquad\qquad D_0^3 = -\frac{1}{3} \\ & & & & \searrow & \\ 0 & 2 = D_2^0 & \nearrow & & & D_1^2 = 2 \quad \nearrow \\ & & \searrow & & & \\ & & & D_2^1 = \quad 6 & \nearrow & \\ 0 & 8 = D_3^0 & \nearrow & & & \end{array}$$

Das interpolierende Polynom ergibt sich entsprechend zu

$$\begin{aligned} p(x) &= 6 - 2\underbrace{(x+2)}_{h_1(x)} + 3\underbrace{(x+2)x}_{h_2(x)} - \frac{1}{3}\underbrace{(x+2)x \cdot x}_{h_3(x)} \\ &= 2 + 4x + \tfrac{7}{3}x^2 - \tfrac{1}{3}x^3 \\ p'(x) &= 4 + \tfrac{14}{3}x - x^2 \end{aligned}$$

und $p(-2) = 6$, $p(0) = 2$, $p'(0) = 4$, $p(1) = 8$.

6.2 Abschätzung des Interpolationsfehlers und Konvergenzanalyse

Sei f die "komplizierte", stetige Funktion, die wir durch das "einfachere" Interpolationspolynom $L_n f$ ersetzen. Dabei seien die Stützstellen $x_0, \ldots, x_n$ gegeben, an denen $L_n f$ und f übereinstimmen, d.h. die Interpolationsbedingung

$$(L_n f)(x_i) = f(x_i) \qquad i = 0, \ldots, n$$

ist erfüllt. Den Operator

$$L_n : C[a, b] \to \Pi_n$$

nennt man auch **Lagrange-Interpolationsoperator**. Wir interessieren uns für den Interpolationsfehler

$$f - L_n f.$$

Wir möchten die größtmögliche Differenz zwischen f und $L_n f$ untersuchen. Dazu bezeichnen wir für eine Funktion $g : [a, b] \to \mathbb{R}$

$$\|g\|_\infty := \max_{x \in [a,b]} |g(x)|.$$

Eine Folge von Funktionen $g_1, g_2, g_3, \ldots$ **konvergiert gleichmäßig** gegen eine Funktion g, falls

$$\|g - g_n\|_\infty \to 0 \text{ für } n \to \infty$$

gilt. Wir werden das nun auf den Interpolationsfehler anwenden, und zunächst $\|f - L_n f\|_\infty$ untersuchen. Danach wollen wir diesen Fehler für eine steigende Anzahl an Stützstellen abschätzen.

Satz 6.14 *Sei $f : [a, b] \to \mathbb{R}$ eine $(n + 1)$-mal stetig differenzierbare Funktion. Dann hat das Restglied*

$$R_n f := f - L_n f$$

bei der Polynom-Interpolation an den $n + 1$ paarweise verschiedenen Stützstellen $x_0, \ldots, x_n \in [a, b]$ die Darstellung

$$(R_n f)(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (x - x_k).$$

Das gilt für alle $x \in [a, b]$, wobei $\xi = \xi(x)$ eine von x abhängige Zwischenstelle aus $[a, b]$ ist.

Beweis: Ist $x = x_j$ für ein $j \in \{0, 1, \ldots, n\}$, so gilt $(R_n f)(x_j) = 0$ und die Aussage ist richtig. Sei nun $h_{n+1}(x) = \prod_{k=0}^{n}(x - x_k)$. Für festes (aber beliebiges) $x \in [a, b], x \neq x_k$ für alle $k = 0, \ldots, n$ definiert man die Funktion $g : [a, b] \to \mathbb{R}$ durch

$$g(y) = f(y) - (L_n f)(y) - h_{n+1}(y) \frac{f(x) - (L_n f)(x)}{h_{n+1}(x)}.$$

Für g gilt:

- g ist $(n + 1)$-mal stetig differenzierbar.
- g hat $x, x_0, \ldots, x_n$ als Nullstellen.

Der Satz von Rolle besagt nun, dass es zu je zwei Nullstellen x_a, x_b von g eine Zwischenstelle $\xi \in (x_a, x_b)$ gibt mit $g^{(1)}(\xi) = 0$. Also hat die Ableitung $g^{(1)}$ mindestens $n + 1$ paarweise verschiedene Nullstellen auf $[a, b]$. Sukzessive Wiederholung dieses Arguments ergibt die Aussage, dass $g^{(r)}$ mindestens $n + 2 - r$

Nullstellen hat, für alle $r = 0, 1, \ldots, n+1$, also hat $g^{(n+1)}$ eine Nullstelle auf $[a, b]$. Wir bezeichnen diese Nullstelle mit ξ. Dann gilt:

$$0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - (n+1)! \frac{(R_n f)(x)}{h_{n+1}(x)},$$

denn $L_n f^{(n+1)} = 0$, weil $L_n f \in \Pi_n$ gilt. Der Term $(n+1)!$ ergibt sich, weil $h_{n+1} \in \Pi_{n+1}$ und als höchsten Koeffizienten Eins hat. Also ist

$$(R_n f)(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} h_{n+1}(x)$$

QED

Bemerkung: Satz 6.14 gilt auch, falls einige der x_j gleich sind und statt dem Lagrange-Problem das Hermite-Problem betrachtet wird. Im Spezialfall $x_0 = x_1 = \cdots = x_n$ ist das interpolierende Polynom das Taylor-Polynom und die Fehlerabschätzung genau das entsprechende Restglied der Taylorformel.

Aus der Darstellung des Restglieds ergibt sich folgende Abschätzung

Korollar 6.15 *Sei $f : [a, b] \to \mathbb{R}$ mindestens $(n+1)$-mal stetig differenzierbar und $x_0, x_1, \ldots, x_n$ paarweise verschiedene Stützstellen. Dann gilt*

$$\|R_n f\|_\infty = \|f - L_n f\|_\infty \leq \frac{1}{(n+1)!} \|h_{n+1}\|_\infty \|f^{n+1}\|_\infty.$$

Da man $\|h_{n+1}\| \leq (b-a)^{n+1}$ abschätzen kann, folgt

$$\|f - L_n f\|_\infty \leq \frac{(b-a)^{n+1}}{(n+1)!} \|f^{(n+1)}\|_\infty.$$

Finite-Elemente-Methoden verbessern den Fehler durch Zerlegung von $[a, b]$ in kleinere Stücke. Hat man bei den Stützstellen Wahlmöglichkeit, so sollte man diese so festlegen, dass

$$\max_{x \in [a,b]} \left| \prod_{k=0}^{n} (x - x_k) \right|$$

möglichst klein wird. Wir betrachten nun noch die Konvergenz von Interpolationspolynomen bei wachsenden Stützstellen.

Satz 6.16 *Sei $f \in C^\infty[a, b]$ und $\|f^{(n)}\|_\infty \leq M$ für alle $n = 0, 1, \ldots$. Dann konvergiert der Interpolationsfehler $\|R_n f\|_\infty$ für $n \to \infty$ gleichmäßig auf $[a, b]$ gegen Null.*

Beweis: Nach Korollar 6.15 gilt

$$\|R_n f\|_\infty \leq \frac{M}{(n+1)!}\|b-a\|^{n+1} \to 0 \text{ für } n \to \infty.$$

Leider sind die Voraussetzungen von Satz 6.14 normalerweise nicht erfüllt. Bei nur stetigen Funktionen gilt die Aussage des Satzes nicht.

Beispiel: Sei $k \in 0, \ldots, n$ und

$$f(x) = \begin{cases} x \sin \frac{\pi}{x} & x \in (0,1] \\ 0 & x = 0 \end{cases}.$$

Mit $x_k = \frac{1}{k+1}$ ist wegen $f(x_k) = 0$ das Interpolationspolynom $L_n f \equiv 0$ für alle $n \in \mathbb{N}$. Die Folge der Interpolationspolynome konvergiert also ausschließlich an den Stützstellen x_k gegen die Funktion f; der Fehler

$$\|R_n f\|_\infty = \max_{x \in [0,1]} |f(x)|$$

bleibt konstant.

Allerdings kann man zeigen, dass es zu jeder stetigen Funktion eine Folge von Stützstellen gibt, so dass $L_n f$ gleichmäßig auf $[a,b]$ gegen f konvergiert. Dennoch ist für beliebige Stützstellen die Interpolation mit Polynomen hohen Grades im Allgemeinen nicht sinnvoll.

6.3 Spline Interpolation

Wir hatten anhand des Beispiels im letzten Abschnitt gesehen, dass die Interpolationspolynome nicht unbedingt gegen die zu interpolierende Funktion f konvergieren. Einen Ausweg bietet die stückweise polynomiale Interpolation durch Splines. Anwendungen hat dieses Gebiet auch in der numerischen Integration und bei der Diskretisierung von Differentialgleichungen.

Definition 6.17 *Sei $a = x_0 < x_1 < \ldots < x_n = b$ eine Unterteilung des Intervalls $[a,b]$. Dann heißt eine Funktion*

$$s : [a,b] \to \mathbb{R}$$

Spline m-ten Grades, *falls die folgenden beiden Bedingungen erfüllt sind:*

(i) $s \in C^{m-1}[a,b]$, *d.h. die Funktion und ihre ersten $m-1$ Ableitungen sind stetig differenzierbar.*

(ii) $s|_{[x_{j-1},x_j]} \in \Pi_m[x_{j-1},x_j]$ *für* $j = 1, \ldots, n$.

Die Menge aller Splines m-ten Grades zu der Unterteilung $a = x_0 < x_1 < \ldots < x_n = b$ mit $n+1$ Stützstellen wird mit $S_n^m[a,b]$ bezeichnet. Für $m = 1$ bezeichnet man die Splines als linear, für $m = 2$ als quadratisch, für $m = 3$ als kubisch.

Der einfachste Fall liegt für lineare Splines ($m = 1$) vor: $S_n^1[a,b]$ enthält alle Polygonzüge auf $[a,b]$ mit maximal $n-1$ Knickpunkten an den Stützstellen $x_1, \ldots, x_{n-1}$, und linearen Teilstücken, die jeweils benachbarte Punkte $(x_j, s(x_j))^T$ und $(x_{j+1}, s(x_{j+1}))^T$ miteinander verbinden.

Das Spline-Interpolationsproblem lässt sich wie folgt beschreiben:

Spline-Interpolationsproblem: Seien $a = x_0 < x_1 < \cdots < x_n = b$ und $y_0, \ldots, y_n \in \mathbb{R}$ gegeben. Gesucht ist ein Spline $s \in S_n^m[a,b]$, der

$$s(x_j) = y_j \text{ für alle } j = 0, \ldots, n$$

erfüllt.

Man kann leicht zeigen, dass zu gegebenen Stützstellen $x_0 < x_1 < \ldots < x_n$ und Stützwerten $y_0, y_1, \ldots, y_n$ der Spline $s \in S_n^1[x_0, x_n]$, der die Interpolationsbedingungen $s(x_j) = y_j$, $j = 1, \ldots, n$ erfüllt, eindeutig bestimmt ist (nämlich gerade der Polygonzug durch die Punkte $(x_j, y_j)^T$, $j = 1, \ldots, n$). Weiterhin gilt für lineare Splines die folgende Aussage.

Lemma 6.18 *Sei $f \in C^2[a,b]$, $a = x_0 < x_1 < \ldots < x_n = b$ eine Unterteilung des Intervalls $[a,b]$ und gelte*

$$h := \max_{j=1,\ldots,n} |x_j - x_{j-1}| \to 0 \quad \textit{für } n \to \infty.$$

Dann konvergiert der Spline $s \in S_n^1[a,b]$, der die Interpolationsbedingungen

$$s(x_j) = f(x_j), j = 1, \ldots, n$$

erfüllt, gleichmäßig gegen f.

Beweis: Betrachte für festes $j \in \{1, \ldots, n\}$ das Teilintervall $[x_{j-1}, x_j]$. Dann ist

$$s|_{[x_{j-1}, x_j]} \in \Pi_1[x_{j-1}, x_j].$$

Um das Restglied auf dem Intervall $[x_{j-1}, x_j]$ abzuschätzen, notieren wir zunächst

$$\|f^{(2)}(x)\|_\infty = \sup_{x: x_{j-1} \le x \le x_j} f^{(2)}(x) =: M < \infty,$$

da $f^{(2)}$ nach Voraussetzung eine stetige Funktion auf dem kompakten Intervall $[x_j, x_{j-1}]$ ist, ihr Supremum also annimmt. Mit der Definition $h_j := x_j - x_{j-1}$

ergibt sich aus Korollar 6.15 entsprechend für das Restglied auf dem Intervall $[x_{j-1}, x_j]$:

$$\|R_m f\| = \|(s-f)\|_\infty \leq \frac{1}{2!} h_j^2 \|f^{(2)}\|_\infty \leq \frac{1}{2} h^2 M.$$

Für $n \to \infty$ geht nach Voraussetzung $h \to 0$, entsprechend gilt $|R_m f| \to 0$. QED

Meistens verwendet man Splines, wenn man eine Funktion f durch eine möglichst "glatte" Funktion interpolieren möchte. Dabei wird der gesuchte Spline umso glatter, je höher die Spline-Ordnung m gewählt wird. Andererseits steigt mit der Spline-Ordnung m der Rechenaufwand zur Bestimmung eines interpolierenden Splines. Kubische Splines haben sich dabei als guter Kompromiss zwischen Glattheit und Rechenaufwand herausgestellt. Um Splines höherer Ordnung zu bestimmen, suchen wir zunächst eine Basis des Spline-Raumes. Dazu definiert man

$$x_+^m := \begin{cases} x^m & \text{falls } x \geq 0 \\ 0 & \text{falls } x < 0. \end{cases}$$

Das System der **Kardinal-Splines** ist dann die Menge der folgenden $m+n$ Funktionen

$$\begin{aligned} \Phi_k(x) &:= (x-x_0)^k \quad \text{für } k = 0, \ldots, m \\ \Psi_j(x) &:= (x-x_j)_+^m \quad \text{für } j = 1, \ldots, n-1. \end{aligned} \tag{6.3}$$

Unser Ziel ist es nun, zu zeigen, dass die Kardinal-Splines eine Basis des Spline-Raumes $S_n^m[a,b]$ bilden. Wir zeigen zunächst die lineare Unabhängigkeit der Kardinal-Splines (Lemma 6.19) und weisen dann im Beweis zu Satz 6.20 nach, dass sie außerdem ein Erzeugendensystem für alle Splines mit Knickpunkten in $x_0, x_1, \ldots, x_n$ bilden.

Lemma 6.19 *Die $n+m$ Kardinal-Splines sind linear unabhängig.*

Beweis: Sei

$$\zeta(x) = \sum_{k=0}^{m} a_k (x-x_0)^k + \sum_{j=1}^{n-1} b_j (x-x_j)_+^m = 0 \text{ für alle } x \in [a,b].$$

Um die lineare Unabhängigkeit zu zeigen, müssen wir nachweisen, dass alle Koeffizienten a_k und b_j Null sind. Wir gehen iterativ vor:

- Für $x < x_1$ ist $\sum_{j=1}^{n-1} b_j (x-x_j)_+^m = 0$, also ist

  $$\zeta(x) = \sum_{k=0}^{m} a_k (x-x_0)^k = 0 \text{ für alle } x \in [a,b].$$

  Genau wie für die Monome lässt sich zeigen, dass

  $$1, (x-x_0), (x-x_0)^2, \ldots, (x-x_0)^m$$

  eine Basis des Π_m bilden. Daraus folgt, dass $a_k = 0$ für $k = 0, \ldots, n$.

- Jetzt berechnen wir für $x \in (x_1, x_2]$, dass $\zeta(x) = b_1(x - x_1)_+^m = 0$ gilt. Weil $x > x_1$ folgt $(x - x_1)_+^m \neq 0$, daher $b_1 = 0$.
- Analog ergibt sich für $x \in (x_i, x_{i+1}]$, dass $b_i = 0$ für $i = 2, 3, \ldots, n-1$.

QED

Der folgende Satz beweist, dass die Kardinal-Splines ein Erzeugendensystem von $S_n^m[a, b]$ bilden, und daher eine Basis sind.

Satz 6.20 *Der Raum $S_n^m[a, b]$ ist ein linearer Raum der Dimension $n + m$. Insbesondere sind die Kardinal-Splines eine Basis des $S_n^m[a, b]$.*

Beweis: Die Linearität des Raumes ist klar. Wir zeigen, dass die Kardinal-Splines den Raum $S_n^m[a, b]$ erzeugen, dann sind sie (wegen Lemma 6.19) eine Basis des Raumes und die Dimension des Raumes ist $n + m$.

Um zu beweisen, dass die Kardinal-Splines ein Erzeugendensystem sind, müssen wir zeigen, dass man jedes $s \in S_n^m[a, b]$ darstellen kann durch

$$s(x) = \sum_{k=0}^{m} a_k(x - x_0)^k + \sum_{j=1}^{n-1} b_j(x - x_j)_+^m \text{ für alle } x \in [a, b].$$

Wir zeigen das mittels Induktion über die Anzahl der Teilintervalle n. Der Induktionsanfang $n = 1$ ergibt eine Unterteilung $a = x_o < x_1 = b$, die aus einem einzigen Intervall besteht. Entsprechend ist

$$S_1^m[a, b] = \Pi_m[a, b]$$

und jedes $s \in \Pi_m$ lässt sich durch $s(x) = \sum_{k=0}^m a_k(x - x_0)^k$ darstellen.

Betrachten wir nun den Übergang von n zu $n+1$. Sei $s \in S_{n+1}^m[a, b]$ ein beliebiger Spline. Wir betrachten diesen Spline s auf dem Intervall $[a, x_n]$ und nennen ihn dort $\tilde{s}$, das heißt $\tilde{s}(x) = s|_{[a,x_n]}$. Für $\tilde{s}$ gilt die Induktionsannahme, also gibt es $a_0, \ldots, a_m$ und $b_1, \ldots, b_{n-1}$ so dass

$$\tilde{s}(x) = \sum_{k=0}^{m} a_k(x - x_0)^k + \sum_{j=1}^{n-1} b_j(x - x_j)_+^m \text{ für alle } x \in [a, x_n].$$

Für die Differenz von s uns $\tilde{s}$,

$$d(x) = s(x) - \tilde{s}(x)$$

gilt:

- $d(x) = 0$ für alle $x \in [a, x_n]$,

- $d|_{[x_n,x_{n+1}]} \in \Pi_m[x_n, x_{n+1}]$.

Aufgrund der Eigenschaften des Splines ist s auf $[x_0, x_{n+1}]$ eine $(m-1)$ mal stetig differenzierbare Funktion. Weil auf $[x_0, x_n]$ die Splines s und $\tilde{s}$ identisch sind, müssen auch ihre (linksseitigen) Ableitungen in x_n übereinstimmen. Wegen der Stetigkeit von $s^{(j)}$ gilt dann

$$d^{(j)}(x_n) = s^{(j)}(x_n) - \tilde{s}^{(j)}(x_n) = 0 \text{ für } j = 0, \ldots, m-1.$$

Zusammenfassend ist die Differenzfunktion d auf dem Intervall $[x_n, x_{n+1}]$ also ein Polynom m-ten Grades mit einer m-fachen Nullstelle an x_n. Das heißt, d muss die folgende Form

$$d(x) = \beta(x - x_n)^m$$

auf $[x_n, x_{n+1}]$ haben, wobei β eine (unbekannte) Konstante ist. Weil $d(x) = 0$ für $x \in [a, x_n]$ können wir diese Vorschrift für d auf ganz $[a, b]$ fortsetzen, zu

$$d(x) = \beta(x - x_n)_+^m.$$

Mit $b_n := \beta$ erhält man entsprechend

$$s(x) = \tilde{s}(x) + d(x) = \sum_{k=0}^{m} a_k(x-x_0)^k + \sum_{j=1}^{n-1} b_j(x-x_j)_+^m + b_n(x-x_n)_+^m \text{ für alle } x \in [a, b].$$

QED

Kommen wir nun auf das Spline-Interpolationsproblem zurück. Bei $n+1$ Stützstellen sind $n+1$ Bedingungen vorgegeben. Da $\dim(S_n^m[a, b]) = n + m$ nach Satz 6.20 werden also $n + m - (n + 1) = m - 1$ Freiheitsgrade nicht genutzt. Einzig im Fall linearer Splines ($m = 1$) liegen keine Freiheitsgrade mehr vor. Für $m > 1$ kann man also zusätzliche Bedingungen stellen, die wir aufgrund der Symmetrie hier nur für ungerade $m \geq 3$ betrachten werden.

Wir nehmen zur Beschreibung der Randbedingungen an, dass wir eine hinreichend oft stetig differenzierbare Funktion f durch einen Spline $s \in S_n^m[a, b]$ interpolieren wollen. Die Interpolationsbedingungen sind also

$$s(x_j) = f(x_j) =: y_j.$$

Weiterhin sei m eine ungerade Zahl, die wir durch $m = 2l-1$ mit $l \geq 2$ darstellen. Folgende Randbedingungen können betrachtet werden:

Hermite-Randbedingungen: Es werden jeweils die ersten $l-1$ Ableitungen am Rand des Interpolationsintervalls festgelegt:

$$\begin{aligned} s^{(j)}(a) &= f^{(j)}(a) \text{ für } j = 1, \ldots, l-1 \\ s^{(j)}(b) &= f^{(j)}(b) \text{ für } j = 1, \ldots, l-1 \end{aligned} \tag{6.4}$$

Natürliche Randbedingungen: Die Ableitungen höherer Ordnung (von $l, l+1, \ldots m-1$) werden an beiden Rändern auf Null gesetzt:

$$s^{(l+j)}(a) \quad = 0 = \quad f^{(l+j)}(b) \text{ für } j = 0, \ldots, l-2 \tag{6.5}$$

Periodizitätsbedingungen: Ist die zu interpolierende Funktion periodisch mit Periode $b-a$, gilt also insbesondere $f^{(j)}(a) = f^{(j)}(b)$, so bietet es sich an, zu verlangen, dass

$$s^{(j)}(a) = f^{(j)}(a) = f^{(j)}(b) = s^{(j)}(b) \text{ für } j = 1, \ldots, l-1. \tag{6.6}$$

Das ist ein Spezialfall der Hermite-Randbedingungen.

Für den Fall kubischer Splines sind also jeweils zwei Bedingungen festzulegen. Diese sind die folgenden:

Für (6.4): $s^{(1)}(a) = f^{(1)}(a)$ und $s^{(1)}(b) = f^{(1)}(b)$.
Für (6.5): $s^{(2)}(a) = 0$ und $s^{(2)}(b) = 0$.
Für (6.6): $s^{(1)}(a) = f^{(1)}(a) = f^{(1)}(b) = s^{(1)}(b)$.

Als nächstes wollen wir beweisen, dass mit jeder dieser Randbedingungen eine eindeutige Lösung des Spline-Interpolationsproblemes existiert. Dazu brauchen wir folgende Vorarbeit.

Lemma 6.21 *Sei $f \in C^l[a,b]$ für $l \in \mathbb{N}$, $l \geq 2$ und sei $s \in S_n^m[a,b]$ der interpolierende Spline bezüglich der Unterteilung $a = x_0 < x_1 < \cdots < x_n = b$. Ferner gelte eine der Randbedingungen (6.4), (6.5) oder (6.6). Dann gilt*

$$\int_a^b [f^{(l)}(x) - s^{(l)}(x)]^2 dx = \int_a^b [f^{(l)}(x)]^2 dx - \int_a^b [s^{(l)}(x)]^2 dx.$$

Beweis: Ausmultiplizieren des Quadrates im ersten Integral ergibt

$$\begin{aligned}
& \int_a^b [f^{(l)}(x) - s^{(l)}(x)]^2 dx \\
= & \int_a^b [f^{(l)}(x)]^2 dx - \int_a^b [s^{(l)}(x)]^2 dx + 2\int_a^b [s^{(l)}(x)]^2 dx - \int_a^b 2 f^{(l)}(x) s^{(l)}(x) dx \\
= & \int_a^b [f^{(l)}(x)]^2 dx - \int_a^b [s^{(l)}(x)]^2 dx - 2 \underbrace{\int_a^b s^{(l)}(x)[f^{(l)}(x) - s^{(l)}(x)] dx}_{=:S}.
\end{aligned}$$

Durch partielles Integrieren unter Berücksichtigung von (6.4), (6.5) oder (6.6) kann man zeigen, dass

$$S = (-1)^{l-1} \int_a^b [f^{(1)}(x) - s^{(1)}(x)] s^{(m)}(x) dx.$$

Bevor wir noch einmal partiell integrieren, zerlegen wir das Integral in einzelne Integrale auf jedem Teilintervall, und erhalten

$$\begin{aligned} S &= (-1)^{l-1} \sum_{j=1}^{n} \int_{x_{j-1}}^{x_j} [f^{(1)}(x) - s^{(1)}(x)] s^{(m)}(x) dx \\ &= (-1)^{l-1} \sum_{j=1}^{n} \left(\underbrace{[f(x) - s(x)]}_{=0} s^{(m)}(x)|_{x_{j-1}}^{x_j} - \int_{x_{j-1}}^{x_j} [f(x) - s(x)] \underbrace{s^{(m+1)}(x)}_{=0} dx \right) \\ &= 0, \end{aligned}$$

wobei die Stammfunktion Null ergibt, weil sie ausschließlich an Stützstellen $x_j, j = 0, \ldots, n$ ausgewertet wird. QED

Aus dem Lemma leitet man ab, dass

$$\int_a^b [s^{(l)}(x)]^2 dx \leq \int_a^b [f^{(l)}(x)]^2 dx$$

gilt. Diese Ungleichung kann man verwenden, um zu zeigen, dass die Splines genau die interpolierenden Funktionen mit minimaler Krümmung sind.

Wir nutzen die Aussage des Lemmas, um den folgenden Satz zu beweisen.

Satz 6.22 *Das Spline-Interpolationsproblem mit einer der Randbedingungen (6.4), (6.5) oder (6.6) ist eindeutig lösbar für alle Funktionen $f \in C^l[a, b]$.*

Beweis: Im Beweis verwenden wir die gleiche Idee wie im Beweis von Satz 6.12, und definieren eine Funktion

$$\alpha : C^l[a, b] \to \mathbb{R}^{n+m},$$

die jede Funktion f auf die Stützwerte an den Interpolationsstellen und die Ableitungswerte zu den jeweiligen Randbedingungen abbildet. Für (6.4) erhält man zum Beispiel

$$\alpha(f) = (f(x_0), f(x_1), \ldots, f(x_n), f^{(1)}(a), \ldots, f^{(l-1)}(a), f^{(1)}(b), \ldots, f^{(l-1)}(b))^T.$$

Wir wenden nun α an auf $S_n^m[a, b]$ und erhalten entsprechend eine Abbildung

$$\alpha : S_n^m[a, b] \to \mathbb{R}^{n+m},$$

zwischen zwei endlich-dimensionalen Vektorräumen gleicher Dimension. Bijektivität von α ist weiterhin äquivalent dazu, dass das Spline-Interpolationsproblem mit Randbedingungen für jede Funktion $f \in C^2[a, b]$ eindeutig lösbar ist. Wir zeigen also, dass α injektiv (und damit bijektiv) ist:

Dazu nehmen wir an, dass $\alpha(s) = 0$ gilt und müssen daraus folgern, dass $s \equiv 0$. Dazu betrachten wir die Nullfunktion $f \equiv 0$. Weil $\int_a^b [f^{(l)}(x)]^2 dx = 0$ gilt nach Lemma 6.21, dass

$$0 \leq \int_a^b [f^{(l)}(x) - s^{(l)}(x)]^2 dx = \int_a^b [f^{(l)}(x)]^2 dx - \int_a^b [s^{(l)}(x)]^2 dx = -\int_a^b [s^{(l)}(x)]^2 dx \leq 0,$$

entsprechend ist

$$\int_a^b [s^{(l)}(x)]^2 dx = 0.$$

Daraus folgert man, dass auf $[a, b]$ $s^{(l)} \equiv 0$ gilt. Weil $s \in C^{m-1}[a, b]$ gilt also $s \in \Pi_{l-1}[a, b]$. Aus den Randbedingungen ergeben sich mindestens l Bedingungen an (das Polynom) s, so dass wegen der Eindeutigkeit des Hermite-Interpolationsproblemes folgt, dass $s \equiv 0$. QED

Bevor wir ein Verfahren angeben, mit dem man Splines berechnen kann, wollen wir eine Verallgemeinerung der für Polynome und Splines gezeigten Sätze andeuten. Dazu benötigen wir den folgenden Begriff.

Definition 6.23 *Ein m-dimensionaler Unterraum $U \subseteq C[a, b]$ heißt* **unisolvent** *bezüglich der m paarweise verschiedenen Stützstellen $x_1, \ldots, x_m \in [a, b]$, wenn jede Funktion $u \in U$ mit Nullstellen $u(x_i) = 0$ für $i = 1, \ldots, m$ identisch verschwindet.*

Wir haben schon zwei Beispiele von unisolventen Räumen kennen gelernt:

- Die Menge der Polynome mit maximalem Grad n ist unisolvent bezüglich jeder Teilmenge $X \subseteq \mathbb{R}$ mit $|X| \geq n + 1$.
- Die Menge $\{s \in S_n^m : s \text{ erfüllt } (6.4)\}$ ist unisolvent bezüglich jeder Menge $X \subset \mathbb{R}$ mit $|X| \geq n+1$. Statt (6.4) kann man auch (6.5) oder (6.6) fordern.

Wir betrachten nun die folgende Interpolationsaufgabe:

Sei $U \subseteq C[a, b]$ ein $(n+1)$-dimensionaler Unterraum, der bezüglich der paarweise verschiedenen Stützstellen $x_0, x_1, \ldots, x_n$ unisolvent ist. Weiterhin seien Stützwerte $y_0, y_1, \ldots, y_n$ gegeben. Gesucht ist eine Funktion $u \in U$, so dass

$$u(x_i) = y_i \text{ für } i = 0, \ldots, n. \qquad \text{(U-Int)}$$

Die Lösbarkeit und Eindeutigkeit von (U-Int) beschreibt der folgende Satz.

Satz 6.24 *Es sei $U \subseteq C[a,b]$ ein $(n+1)$-dimensionaler Unterraum, der bezüglich der paarweise verschiedenen Stützstellen $x_0, x_1, \ldots, x_n$ unisolvent ist. Weiterhin seien Stützwerte $y_0, y_1, \ldots, y_n$ gegeben. Dann gibt es genau ein $u \in U$, das die Interpolationsaufgabe (U-Int) löst.*

Beweis: Betrachte $\alpha : U \to \mathbb{R}^{n+1}$ mit $\alpha(u) = (u(x_0), u(x_1), \ldots, u(x_n))^T \in \mathbb{R}^{n+1}$. Die Interpolationsaufgabe (U-Int) ist eindeutig lösbar, genau dann wenn die Abbildung α bijektiv ist. Wegen der Voraussetzung, dass $\dim(U) = n+1 = \dim(\mathbb{R}^{n+1})$ reicht es, die Injektivität von α nachzuweisen. Dazu sei $\alpha(u) = 0$. Es ist zu zeigen, dass dann $u \equiv 0$ gilt. Dies ist erfüllt, weil U unisolvent bezüglich der Stützstellen $x_0, x_1, \ldots x_n$ ist.

QED

Der Satz enthält die Eindeutigkeit der Lagrange-Polynomaufgabe und der Spline-Polynomaufgabe als Spezialfälle.

Wir kommen nun auf die Spline-Interpolationsaufgabe zurück und wollen uns im folgenden mit der Berechnung von Interpolations-Splines beschäftigen. Ein nahe liegender Ansatz ist, die Basis-Darstellung durch die Kardinal-Splines (6.3),

$$s(x) = \sum_{k=0}^{m} a_k (x - x_0)^k + \sum_{j=1}^{n-1} b_j (x - x_j)_+^m \text{ für alle } x \in [a,b]$$

zu nutzen. Die Koeffizienten a_k, b_j kann man dann durch Lösen des Gleichungssystems bestimmen, das aus den Interpolationsforderungen $s(x_j) = y_j$ und einer der Randbedingungen (6.4), (6.5) oder (6.6) entsteht. Die Koeffizientenmatrix bezüglich der Interpolationsbedingungen hat das folgende Aussehen:

$$\left(\begin{array}{ccccc|cccc}
1 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
1 & (x_1-x_0) & (x_1-x_0)^2 & \ldots & (x_1-x_0)^m & 0 & 0 & \ldots & 0 \\
1 & (x_2-x_0) & (x_2-x_0)^2 & \ldots & (x_2-x_0)^m & (x_2-x_1)^m & 0 & \ldots & 0 \\
1 & (x_3-x_0) & (x_3-x_0)^2 & \ldots & (x_3-x_0)^m & (x_3-x_1)^m & (x_3-x_2)^m & \ldots & 0 \\
 & \vdots & & & \vdots & & \vdots & & \\
1 & (x_n-x_0) & (x_n-x_0)^2 & \ldots & (x_n-x_0)^m & (x_n-x_1)^m & (x_n-x_2)^m & \ldots & (x_n-x_{n-1})^m
\end{array}\right)$$

Wie man sieht, ist die Koeffizientenmatrix stark besetzt, außerdem ist sie schlecht konditioniert. Das liegt daran, dass ein Teil der Kardinal-Splines das gesamte Intervall $[a,b]$ als Träger hat. Man versucht daher, Basisfunktionen mit einem kleinen Träger zu finden, d.h. Basisfunktionen, die nur auf einem kleinen Teilintervall von $[a,b]$ von Null verschieden sind. Das gelingt mit den so genannten B-Splines, die wir für den vereinfachten Fall äquidistanter Stützstellen beschreiben wollen.

Wir betrachten dazu die folgende Unterteilung $a = x_0 < x_1 < \ldots < x_n = b$ mit

$$x_j = a + jh \quad \text{und} \quad h = \frac{b-a}{n}.$$

Notation 6.25 *Die* **B-Splines** *sind reelle Funktionen, die folgendermaßen definiert werden. Ausgehend von*

$$B_0(x) := \begin{cases} 1 & \textit{falls } |x| \leq \frac{1}{2} \\ 0 & \textit{falls } |x| > \frac{1}{2} \end{cases}$$

definiert man für m=0,1,2,... rekursiv

$$B_{m+1}(x) := \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} B_m(t)dt.$$

Als Beispiel berechnen wir B_1 durch

$$\begin{aligned} B_1(x) &= \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} B_0(t)dt \\ &= \begin{cases} \int_{-\frac{1}{2}}^{x+\frac{1}{2}} 1dt & \text{falls } -1 < x \leq 0 \\ \int_{x-\frac{1}{2}}^{\frac{1}{2}} 1dt & \text{falls } 0 < x \leq 1 \\ 0 & \text{sonst} \end{cases} \\ &= \begin{cases} 1-|x| & \text{falls } |x| \leq 1 \\ 0 & \text{falls } |x| > 1 \end{cases}. \end{aligned}$$

Diese Funktion wird aufgrund ihrer Form auch *Hutfunktion* (*hat function*) genannt. Durch weiteres Integrieren der stückweise definierten Funktion erhält man für B_2 und schließlich für B_3 die folgenden Formeln.

$$\begin{aligned} B_2(x) &:= \frac{1}{2} \begin{cases} 2-(|x|-\frac{1}{2})^2-(|x|+\frac{1}{2})^2 & \text{falls } |x| \leq \frac{1}{2} \\ (|x|-\frac{3}{2})^2 & \text{falls } \frac{1}{2} < |x| \leq \frac{3}{2} \\ 0 & \text{falls } |x| > \frac{3}{2} \end{cases} \\ B_3(x) &:= \frac{1}{6} \begin{cases} (2-|x|)^3-4(1-|x|)^3 & \text{falls } |x| \leq 1 \\ (2-|x|)^3 & \text{falls } 1 < |x| \leq 2 \\ 0 & \text{falls } |x| > 2 \end{cases} \end{aligned}$$

Durch vollständige Induktion lassen sich die folgenden Eigenschaften der B-Splines für alle $m \in \mathbb{N}$ nachrechnen:

1. $B_m \in C^{m-1}(\mathbb{R})$.
2. $B_m(x) \geq 0$ für alle $x \in \mathbb{R}$.
3. $B_m(x) = 0$ für alle $x \notin [-\frac{m}{2} - \frac{1}{2}, \frac{m}{2} + \frac{1}{2}]$, d.h. der Träger von B_m ist $(-\frac{m}{2} - \frac{1}{2}, \frac{m}{2} + \frac{1}{2})$.
4. Ist m ungerade, so ist $B_m|_{[i,i+1]} \in \Pi_m$, ist m gerade, so ist $B_m|_{[i-\frac{1}{2},i+\frac{1}{2}]} \in \Pi_m$, für alle ganze Zahlen i.

Eine Basis aus den B-Splines muss, wie die Basis aus den Kardinalsplines, die Stützstellen

$$x_k = a + kh \text{ für } k = 0, 1, \ldots, n, \quad (\text{mit } h = \tfrac{b-a}{n})$$

berücksichtigen. Dazu definiert man für ganze Zahlen k die Funktionen

$$B_{m,k}(x) := B_m\left(\frac{x-a}{h} - k\right).$$

Für $k \in \{0, 1, \ldots, n\}$ gilt dann

$$\frac{x-a}{h} - k = \frac{x-a-hk}{h} = \frac{x-x_k}{h},$$

also sind die $B_{m,k}$ für $k = 0, \ldots, n$ über die Stützstellen der gegebenen Unterteilung definiert.

Durch Ausnutzen der oben gesammelten Eigenschaften der B-Splines und einiges an Technik kann man das folgende Ergebnis beweisen.

Satz 6.26 *Sei $a = x_0 < x_1 < \ldots < x_n = b$ mit $x_j = a + jh$ und $h = \frac{b-a}{n}$ eine äquidistante Unterteilung des Intervalls $[a, b]$. Sei weiterhin $m = 2l - 1$ mit $l \in \mathbb{N}$. Dann ist*

$$\{B_{m,k} : k = -l + 1, \ldots, 0, \ldots, n + l - 1\}$$

eine Basis des $S_n^m([a, b])$.

Die Anzahl der Funktionen $B_{m,k}$ in der Basis beträgt $(l-1) + 1 + (n + l - 1) = 2l+n-1 = m+n$ und stimmt also mit der uns aus Satz 6.20 bekannten Dimension des Raumes $S_m^n[a, b]$ überein.

Die Anwendung von B-Splines soll am Fall kubischer Splines demonstriert werden. Sei also $m = 3$ (entsprechend $l = 2$). Seien weiterhin Stützstellen

$$x_j = a + hj, \ j = 0, \ldots, n$$

mit $x_n = a+hn = b$ beziehungsweise $h = \frac{b-a}{n}$ gegeben. Gesucht wird der kubische Spline

$$s(x) = \sum_{j=-1}^{n+1} c_j B_3\left(\frac{x - x_j}{h}\right)$$

mit den Interpolationsbedingungen $s(x_j) = y_j := f(x_j)$ für $j = 0, 1, \ldots, n$ sowie den Hermite-Randbedingungen (6.4)

$$s'(a) = a_1 \text{ und } s'(b) = b_1.$$

Wegen

$$\frac{x_j - x_k}{h} = \frac{a + jh - a - kh}{h} = j - k$$

kann man $s(x_j)$ berechnen zu

$$\begin{aligned} s(x_j) &= \sum_{k=-l+1}^{n+l-1} c_k B_3\left(\frac{x_j - x_k}{h}\right) \\ &= \sum_{k=1}^{n+1} c_k B_3(j-k) \\ &= \sum_{k=j-1}^{j+1} c_k B_3(j-k) \text{ weil } B(x) = 0 \text{ für alle } |x| \geq 2 \\ &= c_{j-1}B(1) + c_j B(0) + c_{j+1}B(-1). \end{aligned}$$

Also berechnen wir

$$\begin{aligned} B_3(0) &= \frac{2}{3} \\ B_3(1) &= \frac{1}{6} \\ B_3(-1) &= \frac{1}{6} \end{aligned}$$

und erhalten

$$s(x_j) = \frac{1}{6}(c_{j-1} + 4c_j + c_{j+1}) \text{ für } j = 0, \ldots, n.$$

Wegen

$$\begin{aligned} B_3'(0) &= 0 \\ B_3'(1) &= -\frac{1}{2} \\ B_3'(-1) &= \frac{1}{2} \end{aligned}$$

ergibt sich weiter

$$\begin{aligned} s'(a) = s'(x_0) &= \sum_{j=-1}^{1} c_j \frac{1}{h} B_3'(0-j) \\ &= \frac{1}{h}\left(c_{-1}B_3'(1) + c_0 B_3'(0) + c_1 B_3'(-1)\right) \\ &= \frac{1}{2h}(c_1 - c_{-1}) \\ s'(b) = s'(x_n) &= \sum_{j=n-1}^{n+1} c_j \frac{1}{h} B_3'(n-j) \\ &= \frac{1}{h}\left(c_{n-1}B_3'(1) + c_n B_3'(0) + c_{n+1} B_3'(-1)\right) \\ &= \frac{1}{2h}(c_{n+1} - c_{n-1}) \end{aligned}$$

Die Interpolationsbedingungen $s(x_j) = y_j$ und die beiden Hermite-Bedingungen $s'(a) = a_1$, $s'(b) = b_1$ ergeben schließlich das folgende Gleichungssystem

$$AC = F$$

mit

$$F = \begin{pmatrix} a_1 \\ y_0 \\ y_1 \\ \vdots \\ y_n \\ b_1 \end{pmatrix}, C = \begin{pmatrix} c_{-1} \\ c_0 \\ c_1 \\ \vdots \\ c_n \\ c_{n+1} \end{pmatrix}$$

und der Koeffizienten-Matrix

$$A = \frac{1}{6}\begin{pmatrix} -\frac{3}{h} & 0 & \frac{3}{h} & & & & \\ 1 & 4 & 1 & & & 0 & \\ & 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 & \\ & 0 & & & 1 & 4 & 1 \\ & & & & -\frac{3}{h} & 0 & \frac{3}{h} \end{pmatrix}.$$

Die Matrix A ist eine reguläre Bandmatrix, so dass das Gleichungssystem mit den Methoden aus Abschnitt 2.4 oder mit Iterationsverfahren gelöst werden kann.

6.4 Trigonometrische Interpolation

Weiß man, dass die zu interpolierende Funktion periodisch ist, so möchte man gerne auch eine periodische Interpolante konstruieren. Dazu bietet sich die in diesem Abschnitt beschriebene trigonometrische Interpolation an. Zunächst wiederholen wir die Definition von *periodisch*.

Notation 6.27 *Eine Funktion $f : \mathbb{K} \to \mathbb{K}$, heißt* **periodisch** *mit Periode $T > 0$, falls für alle $t \in \mathbb{K}$ gilt:*

$$f(t+T) = f(t).$$

Im folgenden beschäftigen wir uns mit der Periode $T = 2\pi$. Trigonometrische Polynome sind dann als Linearkombinationen der trigonometrischen sin und cos Funktionen definiert:

Notation 6.28 *Die reelle Funktion $q : \mathbb{R} \to \mathbb{R}$ ist ein* **trigonometrisches Polynom**, *falls es reelle Koeffizienten $a_0, a_1, \ldots a_n$ und $b_1, \ldots, b_n$ gibt, so dass*

$$q(x) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos kt + b_k \sin kt). \tag{6.7}$$

Das trigonometrische Polynom q hat **Grad** *n falls $|a_n| + |b_n| \neq 0$ gilt. Der Raum der trigonometrischen Polynome mit maximalem Grad n wird mit T_n bezeichnet.*

Aufgrund der Additionstheoreme erhält man, dass für $p_1 \in T_{n_1}$ und für $p_2 \in T_{n_2}$ gilt: $p_1 \cdot p_2 \in T_{n_1+n_2}$. Weiterhin kann man die Darstellung $e^{it} = \cos t + i \sin t$ ausnutzen und entwickelt daraus die folgende äquivalente Darstellung

$$q(t) = \sum_{k=-n}^{n} c_k e^{ikt}, \tag{6.8}$$

wobei man für $k = 0, \ldots, n$ die Koeffizienten über

$$\begin{aligned} c_k &= \frac{1}{2}(a_k - ib_k) \\ c_{-k} &= \frac{1}{2}(a_k + ib_k) \end{aligned}$$

(mit $b_0 := 0$) erhält.

Wir wollen das folgende Interpolationsproblem lösen:

Sei T_n der Raum der trigonometrischen Polynome mit maximalem Grad n und seien die paarweise verschiedenen Stützstellen $x_0, x_1, \ldots, x_N$ sowie Stützwerte $y_0, y_1, \ldots, y_N$ gegeben. Gesucht ist eine Funktion $q \in T_n$, so dass

$$q(x_i) = y_i \text{ für } i = 0, \ldots, N. \tag{T-Int}$$

Für das Interpolationsproblem ist es nach Satz 6.24 nützlich, sich zunächst Gedanken über die Unisolvenz des Raumes T_n zu machen.

Lemma 6.29

- *Sei $q \in T_n$ und habe q mehr als $2n$ paarweise verschiedene Nullstellen im Periodizitätsintervall $[0, 2\pi)$. Dann verschwindet q identisch.*
- *Die Funktionen $\cos(kx)$, $k = 0, \ldots, n$ und $\sin(kx)$, $k = 1, \ldots, n$ sind linear unabhängig auf dem Raum $C([0, 2\pi))$.*

Beweis: (Idee) Man verwendet die Darstellung (6.8) und die Eindeutigkeit des entsprechenden algebraischen Polynoms $p \in \Pi_{2n}$ auf dem Einheitskreis in $\mathbb{C}$.

QED

Die folgenden beiden Sätze folgen direkt aus dem Lemma.

Satz 6.30

$$\dim(T_n) = 2n + 1$$

Satz 6.31 *T_n ist unisolvent bezüglich jeder Menge $X \subseteq [0, 2\pi)$ mit $|X| \geq 2n+1$.*

Aus dem letzten Satz folgt zusammen mit Satz 6.24 sofort die folgende Aussage.

Satz 6.32 *Das Interpolationsproblem T-Int ist für $N = 2n + 1$ paarweise verschiedene Stützstellen $x_0, x_1, \ldots, x_{2n} \in [0, 2\pi)$ eindeutig lösbar.*

Ähnlich wie bei den Lagrange-Polynomen rechnet man nach, dass eine Lösung des Interpolationsproblemes (T-Int) gegeben ist durch

$$p_n(x) = \sum_{k=0}^{2n} y_k l_k(x)$$

mit den Lagrange-Polynomen

$$l_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{\sin(\frac{1}{2}(x - x_j))}{\sin(\frac{1}{2}(x_k - x_j))} \quad \text{für } k = 0, \ldots, 2n$$

Wie auch im Fall der Interpolation mit algebraischen Polynomen ist die Lagrange-Basis allerdings aus numerischer Sicht wenig geeignet. Man geht daher von dem Ansatz (6.7) oder (6.8) aus und versucht, die Koeffizienten a_k, b_k oder c_k effizient zu berechnen. Für den Fall von äquidistanten Stützstellen führt die Lösung der entsprechenden Gleichungssysteme zu folgendem Ergebnis.

Satz 6.33 (n ungerade) *Es existiert genau ein trigonometrisches Polynom*

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

mit der Interpolationseigenschaft

$$p_n\left(j\frac{2\pi}{2n+1}\right) = y_j,\ j = 0, \ldots, 2n.$$

Dabei sind die Koeffizienten bestimmt durch

$$\begin{aligned} a_k &= \frac{2}{2n+1} \sum_{j=0}^{2n} y_j \cos\left(\frac{2\pi}{2n+1} jk\right),\ k = 0, \ldots, n \\ b_k &= \frac{2}{2n+1} \sum_{j=0}^{2n} y_j \sin\left(\frac{2\pi}{2n+1} jk\right),\ k = 1, \ldots, n \end{aligned}$$

Im Fall, dass n gerade ist, gibt es zu den $2n+1$ Freiheitsgraden eines trigonometrischen Polynoms aus dem Raum T_n zunächst nur $2n$ Interpolationsbedingungen an den äquidistanten Stützstellen $x_j = \frac{j\pi}{n}$ für $j = 0, 1, \ldots, 2n-1$. Man kann jedoch die Basisfunktion $\sin nx$ weglassen, das sie an allen Stützstellen eine Nullstelle hat. Mit relativ wenig Aufwand erhält man das folgende Ergebnis.

Satz 6.34 (n gerade) *Es existiert genau ein trigonometrisches Polynom*

$$P_n(x) = \frac{a_0}{2} + \sum_{k=1}^{n-1} (a_k \cos kx + b_k \sin kx) + \frac{1}{2} a_n \cos nx.$$

mit der Interpolationseigenschaft

$$p_n\left(j\frac{\pi}{n}\right) = y_j,\ j = 0, \ldots, 2n-1.$$

Dabei sind die Koeffizienten bestimmt durch

$$\begin{aligned} a_k &= \frac{1}{n} \sum_{j=0}^{2n-1} y_j \cos\left(\frac{\pi}{n} jk\right),\ k = 0, \ldots, n \\ b_k &= \frac{1}{n} \sum_{j=0}^{2n-1} y_j \sin\left(\frac{\pi}{n} jk\right),\ k = 1, \ldots, n-1 \end{aligned}$$

Es gibt folgenden Zusammenhang zu den so genannten *Fourier-Koeffizienten*. Zunächst definieren wir mittels

$$(f,g)_{L^2} := \int_0^{2\pi} f(t)\overline{g(t)}dt$$

ein Skalarprodukt auf der Menge der quadrat-integrierbaren Funktionen. Die entsprechende Norm ergibt sich zu $\|f\|_{L^2} := \sqrt{(f,f)_{L^2}}$. Dann definiert man zu einer quadrat-integrierbaren Funktion ihre Fourierkoeffizienten durch

$$\hat{a}_k = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos(kx)dx, \quad \hat{b}_k = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin(kx)dx, \ k = 0,1,\ldots,n.$$

Es gilt dann der folgende Satz.

Satz 6.35 *Sei für* $n \in \mathbb{N}$

$$P_n f(x) = \frac{\hat{a}_0}{2} + \sum_{k=1}^{n}(\hat{a}_k \cos kx + \hat{b}_k \sin kx)$$

Dann konvergiert P_n *gegen* f *in* $\|\cdot\|_{L^2}$, *d.h. es gilt* $\lim_{n\to\infty}\|f - P_n\|_{L^2} = 0$.

Beweis: (Idee) Man kann diese Formeln zeigen, wenn man in den Formeln für a_k, b_k aus Satz 6.33 formal den Grenzübergang $n \to \infty$ ausführt. QED

Man spricht im Fall von äquidistanten Stützstellen von der *diskreten Fourier-Transformation*. Die Fourier-Koeffizienten können mittels des Horner-Schemas berechnet werden. Bei großen Werten für n lässt sich der Aufwand durch die *schnelle Fourier-Transformation* weiter reduzieren. Die Idee besteht darin, die komplexen Einheitswurzeln von $n = 2^s$ geschickt zu berechnen.

Numerik II
Sommersemester 2007

Anita Schöbel

12. November 2007

Inhaltsverzeichnis

1 Numerische Integration **2**

1.1 Interpolationsquadraturen 4
1.2 Zusammengesetzte Newton-Côtes-Formeln 9
1.3 Gauß'sche Integrationsformeln 10
1.4 Fehleranalyse 17
1.5 Romberg-Verfahren 24
1.6 Zusammenfassung 31

2 Approximationstheorie **34**

2.1 Approximationssätze von Weierstraß 34
2.2 Existenzsätze 40
2.3 Tschebyscheff-Approximation in $C[a, b]$ 44
2.4 Zusammenfassung 55

3 Numerik gewöhnlicher Differentialgleichungen **58**

3.1 Einführung und Notation 58
3.2 Existenz und Eindeutigkeit 64
3.3 Einschritt-Verfahren 77
3.3.1 Grundlagen 77
3.3.2 Beispiele 78
3.3.3 Konsistenz und Eindeutigkeit 82
3.3.4 Explizite Runge-Kutta-Verfahren 87
3.3.5 Implizite Runge-Kutta-Verfahren 96
3.4 Zusammenfassung 102

4 Optimierung **104**

4.1 Begriffe und Überblick 104
4.2 Iterative Optimierungsverfahren 108
4.2.1 Differenzierbare, nicht-restringierte Probleme 109
4.2.2 Restringierte Probleme 112

5 Eigenwertaufgaben **116**
5.1 Motivation 116
5.2 Eigenwerte 117
5.3 Lokalisierungssatz 118
5.4 Verfahren von Mises 120

Stichwortverzeichnis **126**

Kapitel 1

Numerische Integration

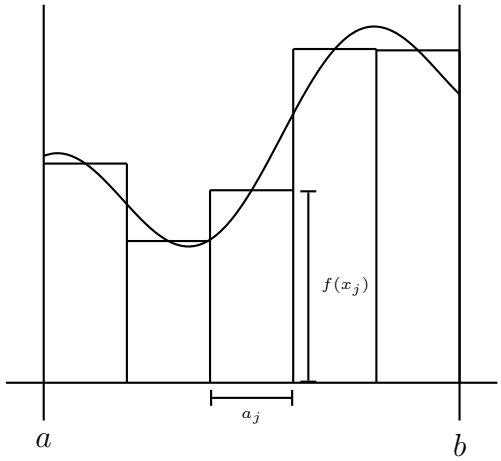
Unser Ziel ist es, eine einfache Formel zur Berechnung von

$$\int_a^b f(x)dx$$

zu finden. Eine Möglichkeit ist die Annäherung durch Rechtecke:

$$\int_a^b f(x)dx \approx \sum_{j=1}^{n} a_j f(x_j).$$

Hierbei ist a_j die Breite des jeweiligen Rechtecks und $f(x_j)$ die Höhe. Diese Summenformel ist „einfach“ zu berechnen.



Notation 1.1 *Sei $x_0, \ldots, x_n \in [a, b]$. Eine Abbildung $Q : \mathbb{R}^{[a,b]} \to \mathbb{R}$ heißt **Quadraturformel** bzgl. $x_0, \ldots, x_n$ falls gilt:*

$$Q(f) = \sum_{j=0}^{n} a_j f(x_j) \text{ für } a_0, \ldots, a_n \in \mathbb{R}.$$

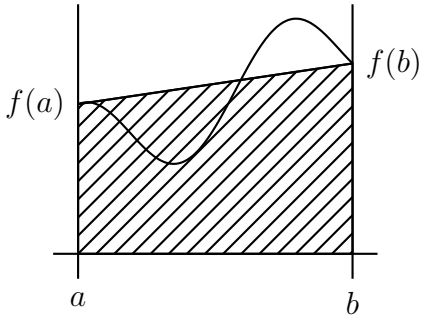
Bemerkung: Q ist eine lineare Abbildung.

Wir versuchen im Folgenden, Quadraturformeln Q zu finden, die Integrale $I(f) := \int_a^b f(x)dx$ annähern, d.h. Q mit $Q(f) \approx I(f)$.

Beispiel: Die Trapezregel ist eine Vorschrift, $\int_a^b f(x)dx$ durch die Fläche eines Trapezes mit den Ecken $(a,0)$, $(a,f(a))$, $(b,f(b))$, $(b,0)$ zu approximieren:

$$\int_a^b f(x)dx \approx \frac{(b-a)}{2}(f(a)+f(b)).$$

Die Trapezregel ist also eine Quadraturformel mit $n=1$, $x_0=a$, $x_1=b$ und $a_0=a_1=\frac{1}{2}(b-a)$. Sie integriert alle affin-linearen Funktionen exakt.



Notation 1.2 *Eine Quadraturformel Q heißt **exakt** für $\mathfrak{F} \subseteq \mathbb{R}^{[a,b]}$, falls $Q(f) = I(f)$ für alle $f \in \mathfrak{F}$ gilt.*

Satz 1.3 *Sei $\mathfrak{F} \subseteq \mathbb{R}^{[a,b]}$ ein endlich-dimensionaler Unterraum von $\mathbb{R}^{[a,b]}$ und $f_0, \ldots, f_N$ eine Basis von $\mathfrak{F}$. Gilt dann $Q(f_i) = I(f_i)$ für alle $i = 0, \ldots, N$ für eine Quadraturformel Q, dann ist Q exakt für $\mathfrak{F}$.*

Beweis: Sei $f \in \mathfrak{F}$. Dann kann man f bezüglich der Basis $f_0, \ldots, f_N$ darstellen als

$$f = \sum_{i=0}^{N} a_i f_i.$$

Es gilt nun

$$\begin{aligned} Q(f) &= Q\left(\sum_{i=0}^{N} a_i f_i\right) = \sum_{i=0}^{N} a_i Q(f_i), \text{ da } Q \text{ linear} \\ &= \sum_{i=0}^{N} a_i I(f_i), \text{ da } Q \text{ exakt für } f_i \text{ und} \\ &= I\left(\sum_{i=0}^{N} a_i f_i\right) = I(f), \text{ da Integrale linear sind.} \end{aligned}$$

QED

Im Folgenden betrachten wir

- Interpolationsquadraturen nach Newton-Côtes,
- Gauß'sche Quadraturformeln und
- die Rombergquadratur.

1.1 Interpolationsquadraturen

Seien $x_0, \ldots, x_n \in [a,b]$ gegeben. Eine Idee ist es, das Integral von f durch das Integral des eindeutig bestimmten Interpolationspolynoms $(L_n f)(x)$ bezüglich der Stützstellen $(x_j, f(x_j))$, $j = 0, \ldots, n$ zu approximieren.

Definition 1.4 *Eine Quadraturformel $Q_n(f) = \sum_{j=0}^{n} a_j f(x_j)$ heißt **Interpolationsquadratur der Ordnung** n, falls für alle $f \in \mathcal{C}[a,b]$ gilt:*

$$Q_n(f) = \sum_{j=0}^{n} a_j f(x_j) = \int_a^b (L_n f)(x)dx = I(L_n f).$$

Dabei ist $L_n f$ das eindeutig bestimmte Interpolationspolynom zu f bezüglich der Stützstellen $x_0, \ldots, x_n$.

Wir erinnern uns an Numerik I, wo wir im Kapitel 6.1 verschiedene Darstellungen für $L_n f$ hergeleitet hatten. Eine war die Lagrange-Darstellung:

$$(L_n f)(x) = \sum_{j=0}^{n} f(x_j) l_j(x) \text{ mit } l_j(x) = \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}.$$

Wir werden im Folgenden die Koeffizienten a_j von Interpolationsquadraturen der Ordnung n herleiten. Kennt man diese, so kann man alle Polynome $p \in \Pi_n$ exakt integrieren. Erstaunlicherweise gilt auch die Umkehrung dieser Aussage:

Satz 1.5 *Eine Quadraturformel Q_n ist genau dann eine Interpolationsquadratur vom Grad n, wenn alle Polynome $p \in \Pi_n[a,b]$ exakt integriert werden.*

Beweis: „$\Rightarrow$“: Sei $Q_n(f) = \int_a^b (L_n f)(x)dx$ eine Interpolationsquadratur der Ordnung n und sei $f \in \Pi_n[a,b]$. Nach dem Satz 6.4 aus Numerik I gilt dann $f = L_n f$, also ist $Q(f) = \int_a^b f(x)dx = I(f)$ und Q_n ist exakt für alle $f \in \Pi_n[a,b]$.
„$\Leftarrow$“: Sei umgekehrt $Q(f) = \sum_{j=0}^{n} a_i f(x_j)$ eine Quadraturformel die $I(p) = Q(p)$ für alle $p \in \Pi_n[a,b]$ erfüllt. Sei $f \in \mathcal{C}[a,b]$. Dann ist $L_n f \in \Pi_n[a,b]$ und es gilt

$$\begin{aligned} I(L_n f) = Q(L_n f) &= \sum_{j=0}^{n} a_j (L_n f)(x_j) \\ &= \sum_{j=0}^{n} a_j f(x_j) = Q(f), \end{aligned}$$

also ist $Q(f)$ eine Interpolationsquadratur der Ordnung n. QED

Satz 1.6 *Sei $h_{n+1}(x) = \prod_{j=0}^{n}(x - x_j)$. Seien $x_0, \ldots, x_n$ paarweise verschieden aus $[a, b]$. Dann existiert genau eine Interpolationsquadratur der Ordnung n zu $x_0, \ldots, x_n$, die durch die Gewichte*

$$a_j = \frac{1}{h'_{n+1}(x_j)} \int_a^b \frac{h_{n+1}(x)}{x - x_j} dx \text{ mit } j = 0, \ldots, n$$

gegeben ist.

Beweis: Wir zeigen zunächst die Eindeutigkeit. Seien

$$Q_A(f) = \sum_{j=0}^{n} a_j f(x_j) \text{ und } Q_B(f) = \sum_{j=0}^{n} b_j f(x_j)$$

zwei Interpolationsquadraturen der Ordnung n. Dann gilt:

$$Q_A(f) = \int_a^b (L_n f)(x) dx = Q_B(f) \text{ für alle } f \in \mathcal{C}[a, b].$$

Wir wählen nun zu jedem j ein f_j mit $f_j(x_j) \neq 0$ und $f_j(x_k) = 0$ für alle $k \neq j$ – z.B. $f_j = l_j$, die Lagrangepolynome. Dann gilt $Q_A(f_j) = a_j = b_j = Q_B(f_j)$, d.h. die Interpolationsquadraturen sind gleich.
Zum Existenzbeweis: $L_n f$ ist stetig und deshalb integrierbar. Wir gehen über die Lagrange-Darstellung:

$$\int_a^b (L_n f)(x) dx = \int_a^b \sum_{j=0}^{n} f(x_j) l_j(x) dx = \sum_{j=0}^{n} f(x_j) \int_a^b l_j(x) dx,$$

d.h. $\int_a^b (L_n f)(x) dx = \sum_{j=0}^{n} f(x_j) a_j$ mit $a_j = \int_a^b l_j(x) dx$ ist tatsächlich eine Interpolationsquadratur. Weiterhin gilt:

$$a_j = \int_a^b \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k} dx = \frac{1}{h'_{n+1}(x_j)} \int_a^b \frac{h_{n+1}(x)}{x - x_j} dx,$$

wobei

$$h'_{n+1}(x) = \sum_{k=0}^{n} \prod_{\substack{i=0 \\ i \neq k}}^{n} (x - x_i)$$

gilt und insbesondere

$$h'_{n+1}(x_j) = \prod_{\substack{i=0 \\ i \neq j}}^{n} (x_j - x_i).$$

QED

Zur Vereinfachung der Formeln betrachten wir den Fall äquidistanter Stützstellen

$$x_j = a + j \cdot h \text{ mit } j = 0, \ldots, n.$$

Wir bemerken:

$$x_n = a + n \cdot h = b, \text{ also } h = \frac{b-a}{n}.$$

Definition 1.7 *Die Interpolationsquadratur der Ordnung n zu den Stützstellen $x_j = a + j \cdot h$ mit $j = 0, \ldots, n$ mit Schrittweite $h = \frac{b-a}{n}$ heißt* ***Newton-Côtes-Formel*** *der Ordnung n.*

Lemma 1.8 *Die Gewichte der Newton-Côtes-Formel der Ordnung n ergeben sich aus*

$$a_j = h \cdot A_j$$
$$\text{mit } A_j = A_{n-j} = \frac{(-1)^{n-j}}{j!(n-j)!} \int_0^n \prod_{\substack{k=0 \\ k \neq j}}^{n} (z-k)dz \qquad \text{für } j = 0, \ldots, n.$$

Beweis: Übung.

Bemerkung: Die Werte A_j hängen ausschließlich von der Anzahl n der Stützstellen ab, nicht aber von den Werten x_j der Stützstellen und auch nicht von a, b oder h!

Einfacher als die Berechnung der A_j nach Lemma 1.8 ist ihre Ermittlung über die Lösung eines linearen Gleichungssystems. Dazu fordert man speziell für die Monome $p(x) = x^k$ für $k = 0, \ldots, n$, dass

$$\sum_{i=0}^{n} a_i p(x_i) = \int_a^b p(x)dx.$$

Nach Satz 1.3 folgt daraus, dass $\sum_{i=0}^{n} a_i p(x_i) = I(p)$ für alle $p \in \Pi_n$ gilt.
Auf diese Weise berechnen wir nun die Koeffizienten für die Fälle $n = 1$ und $n = 2$.

$\boldsymbol{n = 1}$: Nach der Bemerkung nach Lemma 1.8 können wir o.B.d.A. $a = -1$ und $b = 1$ setzen. Somit gilt $h = 2$ und wir erhalten:

- $p(x) = x^0$:

$$\int_{-1}^{1} x^0 dx = [x]_{-1}^{1} = 2 \text{ und } \sum_{j=0}^{1} a_j p(x_j) = a_0 + a_1,$$

  also $a_0 + a_1 = 2$ als erste Bedingung.

- $p(x) = x$:

$$\int_{-1}^{1} x^1 dx = \left[\tfrac{1}{2}x^2\right]_{-1}^{1} = 0 \text{ und } \sum_{j=0}^{1} a_j p(x_j) = a_0 \cdot (-1) + a_1 \cdot 1,$$

  also $-a_0 + a_1 = 0$ als zweite Bedingung.

Die Lösung des Systems

$$\begin{aligned} a_0 + a_1 &= 2 \\ -a_0 + a_1 &= 0 \end{aligned}$$

ist $a_0 = a_1 = 1$ (bzw. $A_0 = A_1 = \frac{1}{2}$) und man erhält daraus

$$\int_{-1}^{1} f(x)dx \approx 1 \cdot f(-1) + 1 \cdot f(1).$$

Für beliebige Integrationsgrenzen a, b ändern sich A_0 und A_1 nicht, sodass wir die auf Seite 3 schon beschriebene **Trapez-Regel**

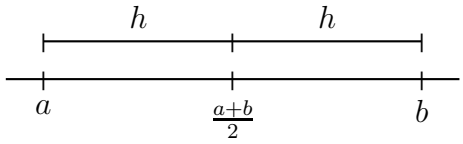
$$\int_a^b f(x)dx \approx \frac{b-a}{2}(f(a) + f(b))$$

erhalten.
$\boldsymbol{n = 2}$: Wie schon im vorherigen Fall wählen wir $a = -1$ und $b = 1$. Daraus ergeben sich nun $h = 1$, $x_0 = -1$, $x_1 = 0$, $x_2 = 1$ und entsprechend die Gleichungen

$$\begin{aligned} \int_{-1}^{1} x^0 dx &= 2 = a_0 + a_1 + a_2, \\ \int_{-1}^{1} x^1 dx &= 0 = -a_0 + a_2, \\ \int_{-1}^{1} x^2 dx &= \tfrac{2}{3} = a_0 + a_2, \end{aligned}$$

woraus man $a_0 = A_0 = \frac{1}{3}$, $a_1 = A_1 = \frac{4}{3}$ und $a_2 = A_2 = \frac{1}{3}$ als eindeutige Lösung errechnet. Daraus erhält man die **Simpson-Regel**

$$\begin{aligned} \int_a^b f(x)dx &\approx \frac{h}{3}(f(x_0) + 4f(x_1) + f(x_2)) \\ &= \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right). \end{aligned}$$



Die folgende Tabelle gibt die Gewichte der ersten fünf Newton-Côtes Formeln an:

n	A_0	A_1	A_2	A_3	A_4	A_5	Bezeichnung
1	$\frac{1}{2}$	$\frac{1}{2}$					Trapez-Regel
2	$\frac{1}{3}$	$\frac{4}{3}$	$\frac{1}{3}$				Simpson-Regel
3	$\frac{3}{8}$	$\frac{9}{8}$	$\frac{9}{8}$	$\frac{3}{8}$			Newton-$\frac{3}{8}$-Regel
4	$\frac{14}{45}$	$\frac{64}{45}$	$\frac{24}{45}$	$\frac{64}{45}$	$\frac{14}{45}$		1. Milne-Regel
5	$\frac{95}{288}$	$\frac{375}{288}$	$\frac{250}{288}$	$\frac{250}{288}$	$\frac{375}{288}$	$\frac{95}{288}$	2. Milne-Regel

Leider tauchen ab $n \geq 8$ auch negative Gewichte auf, die unerwünschte Nebeneffekte haben:

- Auslöschung ist möglich und führt zu numerischer Instabilität und
- es lassen sich positive Funktionen $f \geq 0$ konstruieren, sodass $Q(f) < 0$ gilt.

Wir betrachten nun folgendes Beispiel für die Simpson-Regel:

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

Dann gilt

$$\int_a^b f(x)dx = 0,$$

denn f ist punktsymmetrisch zu $(\frac{a+b}{2}, 0)$:

$$-f(\tfrac{a+b}{2} + x) = -x^3 = (-x)^3 = f(\tfrac{a+b}{2} - x).$$

Wendet man die Simpson-Regel auf f an, so erhält man

$$\begin{aligned} Q_2(f) &= \tfrac{b-a}{6}(f(a) + 4f(\tfrac{a+b}{2}) + f(b)) \\ &= \tfrac{b-a}{6}((\tfrac{a-b}{2})^3 + 4(0)^3 + (\tfrac{b-a}{2})^3) = 0 \end{aligned}$$

also ist die Simpson-Regel für dieses kubische Polynom exakt. Das gilt sogar für alle kubischen Polynome!

Lemma 1.9 *Die Simpson-Regel Q_2 ist exakt für alle $p \in \Pi_3[a, b]$.*

Beweis: Nach Satz 1.3 ist eine Quadraturformel auf Π_3 exakt, wenn sie auf einer Basis von Π_n exakt ist. Wir wählen als Basis

$$(x - \tfrac{a+b}{2})^3,\ (x - \tfrac{a+b}{2})^2,\ x - \tfrac{a+b}{2},\ 1.$$

Im vorangehenden Beispiel haben wir bereits $Q_2((x - \frac{a+b}{2})^3) = I((x - \frac{a+b}{2})^3)$ gezeigt und für die anderen Basisvektoren folgt die Exaktheit aus Satz 1.5, denn alle Polynome aus $\Pi_2[a,b]$ werden von einer Interpolationsquadratur der Ordnung 2 exakt integriert. QED

Man kann diese Aussage weiter verallgemeinern:

Satz 1.10 *Sei $Q_n(f) = \sum_{j=0}^{n} a_j f(x_j)$ eine Newton-Côtes Formel mit geradem n. Dann gilt*

$$Q_n(p) = I(p) \text{ für alle } p \in \Pi_{n+1}[a,b].$$

Beweis: Übung.

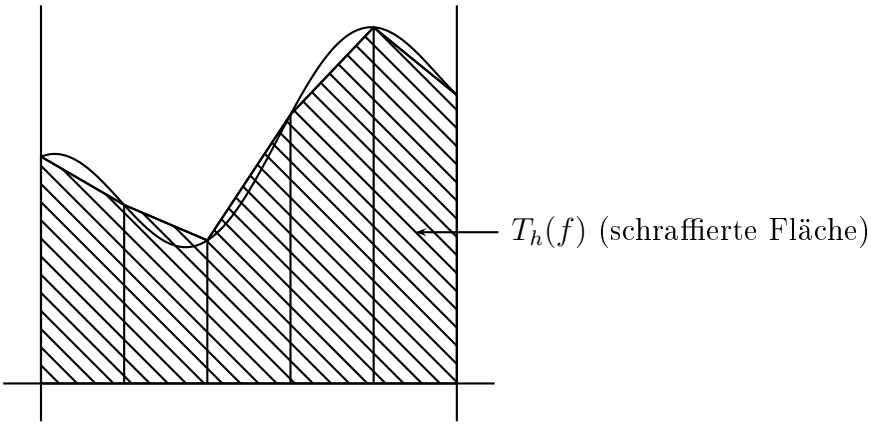
1.2 Zusammengesetzte Newton-Côtes-Formeln

Analog zur Spline-Interpolation zerlegt man bei den zusammengesetzten Newton-Côtes-Formeln das Integrationsintervall in m Teilintervalle und wendet auf je n zusammenhängenden Teilintervallen eine Quadraturformel niederer Ordnung an. Dafür wählen wir m so, dass n Teiler von m ist.

Wir erhalten auf diese Weise die **zusammengesetzte Trapezregel**:

$$\int_a^b f(x)dx \approx T_h(f) := h\left(\tfrac{1}{2}f(x_0) + \sum_{i=1}^{m-1} f(x_i) + \tfrac{1}{2}f(x_m)\right),$$

mit $x_i = a_i + ih$ für $i = 0, \ldots, m$, also $x_0 = a$ und $x_m = b$. Dabei ist $h = \frac{b-a}{m}$.



Analog ergibt sich die **zusammengesetzte Simpsonregel**: Sei dazu $x_0 = a < x_1 < \cdots < x_m = b$ eine äquidistante Zerlegung in eine gerade Anzahl an Teilintervallen. Wir wenden die Simpson-Regel jeweils auf zwei aufeinander folgende Intervalle an und erhalten:

$$\int_a^b f(x)dx \approx S_h(f) = \frac{h}{3} \sum_{j=0}^{\frac{m}{2}-1} \tfrac{1}{3} f(x_{2j}) + \tfrac{4}{3} f(x_{2j+1}) + \tfrac{1}{3} f(x_{2j+2})$$

1.3 Gauß'sche Integrationsformeln

Für die Gauß'schen Integrationsformeln sollen neben den Gewichten $a_0, a_1, \ldots, a_n$ auch die Stützstellen $x_0, \ldots, x_n$ gewählt werden. Man hat also $2(n+1)$ Freiheitsgrade. Entsprechend darf man $2n+2$ Bedingungen stellen, z. B.

$$\sum_{i=0}^{n} a_i p(x_i) = \int_a^b p(x)dx \text{ für } p(x) \in \{1, x, x^2, \ldots, x^{2n+1}\}.$$

Sind diese Bedingungen erfüllt, kann man alle Polynome aus Π_{2n+1} exakt integrieren (Satz 1.3). Wir wollen gerne ohne dieses nichtlineare Gleichungssystem auskommen.
Dabei ist es in verschiedenen Anwendungen günstig, den allgemeineren Fall von Quadraturformeln zu betrachten, nämlich **gemischte Integrale**

$$I_w(f) := \int_a^b \omega(x) f(x) dx$$

mit einer auf (a, b) stetigen und positiven Gewichtsfunktion ω. Weiterhin fordert man, dass

$$\int_a^b \omega(x) x^k dx$$

für alle $k \in \mathbb{N}_0$ existiert.

Typische Beispiele für solche Gewichtsfunktionen sind

- Gauß-Legendre: $\omega(x) = 1$ auf $[a, b]$
- Gauß-Tschebyscheff 1. Art: $\omega(x) = \frac{1}{\sqrt{1-x^2}}$ auf $x \in [-1, 1]$.
- Gauß-Tschebyscheff 2. Art: $\omega(x) = \sqrt{1-x^2}$ auf $x \in [-1, 1]$.
- Gauß-Laguerre: $\omega(x) = e^{-x}$ auf $[0, \infty)$.
- Gauß-Hermite: $\omega(x) = e^{-x^2}$ auf $(-\infty, \infty)$

Definition 1.11 *Eine Quadraturformel* $Q_n(f) = \sum_{i=0}^{n} a_i f(x_i)$ *nennt man* ***Gauß'sche Quadraturformel der Ordnung*** n*, wenn sie alle Polynome* $p \in \Pi_{2n+1}[a,b]$ *exakt integriert, d.h. wenn*

$$Q_n(p) = \int_a^b \omega(x)p(x)dx \text{ für alle } p \in \Pi_{2n+1}[a,b].$$

Lemma 1.12 *Seien* $x_0, \ldots, x_n \in [a,b]$*. Sei* $L_n f$ *das Interpolationspolynom bzgl.* $x_0, \ldots, x_n$ *an die Funktion* f*. Sei weiter* ω *eine zulässige Gewichtsfunktion. Dann ist*

$$\int_a^b \omega(x)(L_n f)(x)dx$$

eine Quadraturformel bzgl. $x_0, \ldots, x_n$*. Genauer gilt:*

$$\int_a^b \omega(x)(L_n f)(x)dx = \sum_{j=0}^{n} a_j f(x_j)$$

mit $a_j = \int_a^b \omega(x) l_j(x)dx$*.*

Beweis: Den Fall $\omega \equiv 1$ haben wir in Satz 1.6 behandelt. Für andere ω verläuft der Beweis analog. QED

Wann ist $Q_n(f)$ eine Gauß'sche Quadraturformel?

Satz 1.13 *Sei* ω *eine zulässige Gewichtsfunktion und seien* $x_0, \ldots, x_n \in [a,b]$ *paarweise verschieden. Sei* $L_n f$ *die Interpolation von* f *bzgl. der Stützstellen* $x_0, \ldots, x_n$*. Sei weiterhin* $h_{n+1}(x) = \prod_{j=0}^{n}(x - x_j)$*. Dann sind die folgenden beiden Aussagen äquivalent:*

1. $Q_n(f) := \int_a^b \omega(x)(L_n f)(x)dx$ *ist eine Gauß'sche Quadraturformel der Ordnung* n*, d.h.* $Q_n(p) = I_w(p)$ *für alle* $p \in \Pi_{2n+1}$*.*

2. $\int_a^b \omega(x) h_{n+1}(x) p(x)dx = 0$ *für alle* $p \in \Pi_n$*.*

Beweis: „1 $\Rightarrow$ 2“: Sei $Q_n(f) = I_w(f)$ für alle $f \in \Pi_{2n+1}$. Sei $p \in \Pi_n$. Dann ist $h_{n+1} \cdot p \in \Pi_{2n+1}$, also gilt nach Lemma 1.12:

$$\int_a^b \omega(x) h_{n+1}(x) p(x)dx = Q_n(h_{n+1} \cdot p) = \sum_{j=0}^{n} a_j \underbrace{h_{n+1}(x_j)}_{=0} p(x_j) = 0,$$

also gilt 2.

„2 $\Rightarrow$ 1“: Sei $p \in \Pi_{2n+1}$. Betrachte $L_n p \in \Pi_n$ bzgl. $x_0, \ldots, x_n$. Dann hat $p - L_n p$ die Nullstellen $x_0, \ldots, x_n$, also gibt es nach dem Hauptsatz der Algebra ein Polynom $q \in \Pi_n$, so dass $p - L_n p = h_{n+1} \cdot q$. Damit gilt

$$\int_a^b \omega(x)p(x)dx = \int_a^b \omega(x)L_n p(x)dx + \underbrace{\int_a^b \omega(x)h_{n+1}(x)q(x)dx}_{=0 \text{ nach 2. weil } q \in \Pi_n}$$

$$= Q_n(p),$$

also ist Q_n Gauß'sche Quadraturformel. QED

Notation 1.14 *Für $f, g \in C([a,b])$ definieren wir*

$$(f,g)_\omega := \int_a^b \omega(x)f(x)g(x)dx.$$

Gilt $(f,g)_\omega = 0$, so bezeichnet man f und g als ω-orthogonal.

Bemerkung: Der Ausdruck $(f,g)_\omega$ existiert für alle Polynome f und g, wenn ω eine zulässige Gewichtsfunktion ist. Es lässt sich sogar zeigen, dass $(f,g)_\omega$ ein Skalarprodukt ist (d.h. bilinear, symmetrisch und positiv für $f = g$, sowie streng positiv für $f = g \neq 0$).

Satz 1.13 lässt sich jetzt folgendermaßen formulieren:

$\int_a^b \omega(x)(L_n f)(x)dx$ ist genau dann eine Gauß'sche Quadraturformel der Ordnung n, wenn $(h_{n+1}, p)_\omega = 0$ für alle $p \in \Pi_n$.

Die gesuchten Stützstellen der Quadraturformel müssen also die Nullstellen eines Polynoms $q(x) = \alpha h_{n+1}(x) \in \Pi_{n+1}$ sein, das ω-orthogonal zu allen $q \in \Pi_n$ ist. Solche Polynome wollen wir im Folgenden konstruieren.

Satz 1.15 *Sei ω eine zulässige Gewichtsfunktion. Dann gilt*

1. *Es existieren Polynome $p_n \in \Pi_n[a,b]$ für alle $n \in \mathbb{N}_0$ mit*

$$(p_n, p_m) = \delta_{n,m} = \begin{cases} 1 & \text{falls } n = m \\ 0 & \text{falls } n \neq m \end{cases} \quad \text{für alle } n, m \in \mathbb{N}_0.$$

2. *Für alle $n \in \mathbb{N}_0$ gilt: Die Nullstellen von p_n sind alle reell und liegen in (a,b).*

Beweis: **ad 1.** Die Folge p_i der gesuchten Polynome lässt sich durch Anwenden des Schmidt'schen Orthonormalisierungsverfahrens auf die Monome konstruieren. Man erhält entsprechend eine Orthonormalbasis. Die Monombasis ist $\{x^0, \ldots, x^n\}$. Man setzt nun

$$p_0(x) = \frac{x^0}{\sqrt{(x^0, x^0)_\omega}} = \frac{1}{\sqrt{\int_a^b \omega(x)dx}}$$

und erhält $(p_0, p_0)_\omega = 1$.
Zur Konstruktion von p_n nehmen wir an, dass $p_0, \ldots, p_{n-1}$ bereits konstruiert sind und dass sie $(p_i, p_j) = \delta_{ij}$ erfüllen. Dann ergibt sich $p_n \in \Pi[a, b]$ aus

$$p_n(x) = \gamma_n \left(x^n - \sum_{i=0}^{n-1} (x^n, p_i)_\omega p_i(x) \right),$$

wobei die $(x^n, p_i(x))_\omega$ die Koeffizienten nach Schmidt sind. Das ist die Lösung, denn

- für $m = 0, \ldots, n-1$ gilt:

$$\begin{aligned}(p_n, p_m) &= \gamma_n \left((x^n, p_m)_\omega - \sum_{i=0}^{n-1} (x^n, p_i)_\omega (p_i, p_m) \right) \\ &= \gamma_n \left((x^n, p_m)_\omega - (x^n, p_m)_\omega \cdot 1 \right) = 0\end{aligned}$$

- und γ_n wird so gewählt, dass $(p_n, p_m)_\omega = 1$.

ad 2. Seien $x_1, \ldots, x_m$ die reellen Nullstellen von p_n in (a, b) mit ungerader Vielfachheit, d.h. genau die Nullstellen mit Vorzeichenwechsel von p_n. Sei

$$q_m(x) = \prod_{i=1}^{m} (x - x_i) \text{ mit } q_0(x) := 1.$$

Wir wollen nun zeigen, dass $m = n$ gilt. Angenommen, es gelte $m < n$. Dann gilt $q_m \in \Pi_m[a, b] \subseteq \Pi_{n-1}[a, b]$. Weiter ist $p_n q_m(x) \geq 0$ für alle $x \in (a, b)$, weil es nur Nullstellen mit gerader Vielfachheit hat. Weil $p_n q_m \neq 0$ folgt

$$(p_n, q_m)_\omega \neq 0.$$

Weil die in Teil 1 konstruierten Polynome $p_0, \ldots, p_m$ eine Basis von Π_m bilden, gilt andererseits

$$q_m = \sum_{i=0}^{m} \lambda_i p_i \text{ mit reellen Koeffizienten } \lambda_i$$

und nach Konstruktion der p_i ist

$$(p_n, q_m)_\omega = \sum_{i=0}^{m} \lambda_i (p_n, p_i)_\omega = 0,$$

denn $(p_n, p_i)_\omega = 0$, weil $m < n$. Das ist ein Widerspruch, also muss $m = n$ gelten und die Vielfachheit jeder Nullstelle ist entsprechend 1. QED

Wir fassen die Ergebnisse in folgendem Existenzsatz zusammen:

Satz 1.16 *Sei $n \in \mathbb{N}$ und ω eine zulässige Gewichtsfunktion. Dann existiert eine Gauß'sche Quadraturformel*

$$Q_n(f) = \sum_{j=0}^{n} a_j f(x_j),$$

wobei $x_0 < x_1 < \cdots < x_n \in (a, b)$ die Nullstellen des in Satz 1.15 konstruierten bzgl. aller $p \in \Pi_n$ ω-orthogonalen Polynoms p_{n+1} sind und

$$a_j = \int_a^b \omega(x) l_j(x) dx$$

gilt.

Beweis: Weil $p_{n+1} \in \Pi_{n+1}$ ist, gilt $p_{n+1} = \alpha h_{n+1}$ mit $\alpha \neq 0$. Nach Lemma 1.12 ist

$$Q_n(f) = \int_a^b \omega(x)(L_n f)(x) dx,$$

welches nach Satz 1.13 eine Gauß'sche Quadraturformel ist, wenn $(h_{n+1}, p) = 0$ für alle $p \in \Pi_n$. Das gilt, weil

$$\begin{aligned}(h_{n+1}, p) &= \frac{1}{\alpha}(p_{n+1}, p) = \frac{1}{\alpha}\left(p_{n+1}, \sum_{i=0}^{n} \lambda_i p_i\right) \\ &= \frac{1}{\alpha} \sum_{i=0}^{n} \lambda_i (p_{n+1}, p_i) = 0.\end{aligned}$$

QED

Es gilt also $Q_n(p) = I_\omega(p)$ für alle $p \in \Pi_{2n+1}$.

Im Gegensatz zu den Newton-Côtes-Formeln gilt für die Gauß'schen Quadraturformeln die folgende numerisch wertvolle Eigenschaft:

Lemma 1.17 *Die Gewichte a_i der Gauß'schen Quadraturformeln sind positiv.*

Beweis: Seien $x_0, \ldots, x_n$ die Stützstellen der Quadraturformel Q_n. Nach Konstruktion sind sie die Nullstellen von p_{n+1} bzgl. ω. Wir definieren

$$h_{n+1}(x) := \prod_{j=0}^{n}(x - x_j) \text{ und } f_i(x) = \left(\frac{h_{n+1}(x)}{x - x_i}\right)^2, \qquad i = 0, \ldots, n.$$

Es ist also $f_i \in \Pi_{2n}[a, b]$ und nach Satz 1.16 gilt

$$0 < \int_a^b \omega(x) f_i(x) dx = Q_n(f_i) = \sum_{j=0}^{n} a_j f_j(x_j) = a_i f_i(x_i).$$

Weil $f_i(x_i) > 0$ folgt auch, dass $a_i > 0$. QED

Übungsaufgabe: Beweisen Sie, dass es keine Quadraturformel der Form $\sum_{j=0}^{n} a_j f_j(x_j)$ geben kann, die auf Π_{2n+2} exakt ist.

Zum Abschluss folgen einige Beispiele für Gauß-Quadraturen.

1. Sei $I = [-1, 1]$ und $\omega \equiv 1$. Die orthogonalen Polynome $p_0, p_1, \ldots$ sind die sogenannten *Legendre-Polynome*

$$L_n(x) := \frac{1}{2^n n!} \frac{d^n}{dx^n}(x^2 - 1)^n,$$

genauer:

$$\begin{aligned} L_0 &= 1 & L_3 &= x^3 - \tfrac{3}{5}x \\ L_1 &= x & L_4 &= x^4 - \tfrac{6}{7}x^2 + \tfrac{3}{35} \\ L_2 &= x^2 - \tfrac{1}{3} & L_5 &= \ldots \end{aligned}$$

L_i ist orthogonal auf Π_{i-1}, d.h. $\int_{-1}^{1} L_i(x) p(x) dx = 0$ für alle $p \in \Pi_{i-1}[-1, 1]$.

Beispiel: Es gilt

$$\int_{-1}^{1} L_2(x) p(x) = 0, \text{ für alle } p \in \Pi_1[-1, 1].$$

Das kann man nachrechnen:

$$\begin{aligned} \int_{-1}^{1} L_2(x) p(x) dx &= \int_{-1}^{1} (x^2 - \tfrac{1}{3})(ax + b) dx \\ &= \int_{-1}^{1} ax^3 + bx^2 - \tfrac{1}{3}ax - \tfrac{1}{3}b dx \\ &= \left[\frac{a}{4}x^4 + \frac{b}{3}x^3 - \frac{a}{6}x^2 - \frac{b}{3}x\right]_{-1}^{1} \\ &= \frac{a}{4} + \frac{b}{3} - \frac{a}{6} - \frac{b}{3} - \left(\frac{a}{4} - \frac{b}{3} - \frac{a}{6} + \frac{b}{3}\right) = 0. \end{aligned}$$

Wie sehen die zugehörigen Quadraturen aus?

$n = 0$

- Die Stützstellen sind die Nullstellen des orthogonalen Polynoms aus Π_1 (Satz 1.16). Die einzige Nullstelle von L_1 ist $x_0 = 0$. Daraus folgt:
$$Q_0(f) = a_0 f(x_0) = a_0 f(0).$$
- Das Gewicht a_0 ergibt sich dadurch, dass z.B. $1 \in \Pi_1 = \Pi_{2n+1}$ exakt integriert wird, also
$$2 = \int_{-1}^{1} 1 dx = a_0 f(0) = a_0.$$

Wir erhalten:
$$Q_0(f) = 2f(0)$$
integriert alle linearen Polynome auf $[-1, 1]$ exakt.

$n = 1$

- Stützstellen sind Nullstellen von L_2, also $\pm\sqrt{\frac{1}{3}}$. Es folgt:
$$Q_1(f) = a_0 f\left(-\sqrt{\tfrac{1}{3}}\right) + a_1 f\left(\sqrt{\tfrac{1}{3}}\right).$$
- Die Gewichte folgen aus den Exaktheitsbedingungen z.B. für $1, x \in \Pi_1 \subseteq \Pi_{2n+1}$:
$$\left.\begin{aligned} 2 &= \int_{-1}^{1} 1 dx = a_0 + a_1 \\ 0 &= \int_{-1}^{1} x dx = -a_0\sqrt{\tfrac{1}{3}} + a_1\sqrt{\tfrac{1}{3}} \end{aligned}\right\} \Rightarrow a_0 = a_1 = 1.$$

Es gilt also:
$$Q_1(f) = f\left(-\sqrt{\tfrac{1}{3}}\right) + f\left(\sqrt{\tfrac{1}{3}}\right)$$
integriert alle Polynome vom Grad bis 3 exakt!

2. $I = [-1, 1]$ und $\omega(x) = \frac{1}{\sqrt{1-x^2}}$. Die orthogonalen Polynome sind die sogenannten *Tschebyscheff-Polynome*
$$T_n(x) := \cos(n \arccos(x)).$$

Mithilfe der Additionstheoreme erhält man die folgende Darstellung:
$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{n+1}(x) + T_{n-1}(x) &= 2xT_n(x). \end{aligned}$$

Daraus folgt $T_n \in \Pi_n$.

Lemma 1.18 *Es gilt:*

$$\int_{-1}^{1} \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}}dx = \begin{cases} \pi & n = m = 0 \\ \frac{\pi}{2} & n = m > 0 \\ 0 & n \neq m \end{cases}$$

und die Nullstellen von T_n sind:

$$x_i = \cos\left(\frac{2i+1}{2n}\pi\right) \qquad i = 0, \ldots, n-1.$$

Mit diesen Nullstellen erhalten wir die Gauß-Tschebyscheff Quadratur:

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}dx \approx Q_{n-1}(f) = \sum_{i=0}^{n-1} a_i f\left(\cos\frac{2i+1}{2n}\pi\right).$$

Die Gewichte ergeben sich aus den Exaktheitsbedingungen für T_m, $m = 0, \ldots, n-1$ und Lemma 1.18 zu $a_i = \frac{\pi}{n}$, $i = 0, \ldots, n-1$.

Übungsaufgabe: Zeigen Sie, dass $a_i = \frac{\pi}{n}$, $i = 0, \ldots, n-1$ die Gewichte der Gauß-Tschebyscheff-Quadraturformel Q_n sind!

Damit erhält man abschließend

$$Q_{n-1}(f) = \frac{\pi}{n}\sum_{i=0}^{n-1} f\left(\cos\frac{2i+1}{2n}\pi\right), \qquad n \in \mathbb{N}.$$

Bemerkung: Analog zu Kapitel 1.2 kann man auch zusammengesetzte Gauß-Quadraturen (vorzugsweise niedriger Ordnung) betrachten.

1.4 Fehleranalyse

Was für ein Fehler entsteht, wenn man das exakte Integral durch eine Quadraturformel annähert?
Seien zunächst für $n \in \mathbb{N}$ die Stützstellen

$$x_0^{(n)} < x_1^{(n)} < \cdots < x_n^{(n)}$$

mit den zugehörigen Gewichten $a_0^{(n)}, \ldots, a_n^{(n)}$ gegeben und sei

$$Q_n(f) = \sum_{j=0}^{n} a_j^{(n)} f(x_j^{(n)}).$$

Der zugehörige Fehler ist dann:

$$R_n(f) := I_\omega(f) - Q_n(f) = \int_a^b \omega(x) f(x) dx - \sum_{j=0}^{n} a_j^{(n)} f(x_j^{(n)}).$$

Es werfen sich einige Fragen auf:

- Wie groß ist $R_n(f)$ für festes n?
- Konvergiert $R_n(f) \to 0$ für $n \to \infty$?

Satz 1.19 *Sei Q_n eine Folge von Quadraturformeln über dem endlichen Intervall $[a, b]$. Gilt*

1. *$Q_n(p) \to I_\omega(p)$ für $n \to \infty$ für alle Polynome p und*
2. *$\sum_{j=0}^{n} |a_j^{(n)}| \leq C$ für alle $n \in \mathbb{N}$ mit einer Konstante $C > 0$,*

so konvergiert die Folge $Q_n(f)$ gegen $I_\omega(f)$ für jedes $f \in C[a, b]$.

Beweis: Wir verwenden einen Satz aus der Approximationstheorie, nämlich den Satz von Weierstrass: Jede stetige Funktion auf einem kompakten Intervall lässt sich beliebig gut durch Polynome annähern. Genauer gibt es zu jedem $\varepsilon > 0$ ein Polynom p mit $\|f - p\|_\infty := \max_{x \in [a,b]} |f(x) - p(x)| < \varepsilon$.
Sei $\varepsilon > 0$. Wähle p so, dass $\|f - p\|_\infty < \varepsilon$ und N so, dass $|I_\omega(p) - Q_n(p)| < \varepsilon$ für alle $n \geq N$. Dann gilt für alle $n \geq N$:

$$|I_\omega(f - p)| \leq \int_a^b |\omega(x)| \underbrace{|f(x) - p(x)|}_{<\varepsilon} dx < \varepsilon \int_a^b \omega(x) dx$$

$$|Q_n(f - p)| \leq \sum_{j=0}^{n} |a_j^{(n)}| \underbrace{|f(x_j^{(n)}) - p(x_j^{(n)})|}_{<\varepsilon} < \varepsilon \sum_{j=0}^{n} |a_j^{(n)}| \leq \varepsilon C$$

Es ergibt sich:

$$\begin{aligned}
|R_n(f)| &= |R_n((f - p) + p)| \\
&= |R_n(f - p) + R_n(p)| \\
&\leq |R_n(f - p)| + |R_n(p)| \\
&\leq |I_\omega(f - p)| + |Q_n(f - p)| + |I_\omega(p) - Q_n(p)| \\
&\leq \left(\int_a^b \omega(x) dx + C + 1 \right) \cdot \varepsilon
\end{aligned}$$

für alle $n \geq N$, also konvergiert $R_n(f)$ gegen Null. QED

Bedingung (1) des Satzes ist erfüllt, wenn alle Q_n interpolatorische Quadraturformeln sind. Sind weiter alle Gewichte nicht negativ, dann gilt:

$$\sum_{j=0}^{n} |a_j^{(n)}| = \sum_{j=0}^{n} a_j^{(n)} \cdot 1 = \int_a^b \omega(x) \cdot 1 dx = C.$$

Satz 1.20 *Für jeden stetigen Integranden auf dem Intervall $[a, b]$ konvergiert jede Folge von Gauß-Quadraturen Q_n gegen das Integral.*

Beweis: Nach Lemma 1.17 sind die Gewichte der Gauß-Quadraturen positiv, daher ist

$$\sum_{j=0}^{n} |a_j^{(n)}| \leq C$$

erfüllt. Bedingung (1) von Satz 1.19 gilt nach Satz 1.16, also folgt die Behauptung wegen Satz 1.19. QED

Leider ist die Aussage von Satz 1.20 für die Newton-Côtes-Formeln im Allgemeinen falsch und es lassen sich Gegenbeispiele konstruieren. (Der Grund dafür liegt in den negativen a_i, die in den Newton-Côtes-Formeln ab Grad 8 vorkommen.) Wir entwickeln nun Fehlerabschätzungen für die Quadraturformeln auf festen Intervallen $[a, b]$:

$$R_n(f) = I_\omega(f) - Q_n(f) = I_\omega(f) - I_\omega(L_n f) = I_\omega(f - L_n f),$$

wobei $L_n f$ das Interpolationspolynom zu f an den n Stützstellen von Q_n ist. Nun kann man die Fehlerabschätzungen für $f - L_n f$ (Numerik I, Korollar 6.15) heranziehen. Bessere Ergebnisse liefert aber der folgende (allgemeinere) Ansatz, bei dem man ausnutzt, dass für alle $p \in \Pi_m$ gilt:

$$R_n(p) = 0,$$

wobei man m im Falle von Newton-Côtes $\leq n$ wählen muss, falls n ungerade ist und $m = n+1$, falls n gerade ist. Bei der Gauß-Quadratur hingegen ist $m \leq 2n+1$ möglich.

Notation 1.21 *Sei $t \in \mathbb{R}$. Dann bezeichne*

$$z_{t,m}^+(x) = (x-t)_+^m = \begin{cases} (x-t)^m & \text{falls } x \geq t \\ 0 & \text{sonst} \end{cases}$$

und

$$K_m(t) := \frac{1}{m!} R_n(z_{t,m}^+) \text{ mit } t \in [a, b]$$

den Peano-Kern *bzgl. m und R_n.*

Bemerkung: Wir können die $z_{t,m}^+(x)$ sowohl als Funktion in x als auch in t auffassen.

Wir verwenden $z_{t,m}^+$ für folgende Umformulierung:

$$\int_a^x \frac{(x-t)^m}{m!} f^{(m+1)}(t)dt = \int_a^b \frac{(x-t)_+^m}{m!} f^{(m+1)}(t)dt,$$

das heißt, um bei den im Folgenden auftretenden Integralen die obere Grenze von x unabhängig zu machen. Wir erhalten:

Satz 1.22 *Sei Q_n eine auf $\Pi_m(\mathbb{R})$ exakte Quadraturformel. Dann gilt für jedes $f \in C^{m+1}[a,b]$:*

$$R_n(f) = \int_a^b K_m(t) f^{(m+1)}(t)dt.$$

Wechselt K_m auf $[a,b]$ das Vorzeichen nicht, so existiert ein $\xi \in [a,b]$ so, dass

$$\begin{aligned} R_n(f) &= f^{(m+1)}(\xi) \int_a^b K_m(t)dt \\ &= \frac{f^{(m+1)}(\xi)}{(m+1)!} R_n(x^{m+1}). \end{aligned}$$

Beweis: Wir verwenden die Taylor-Entwicklung von f mit Integral-Restglied zum Entwicklungspunkt a:

$$\begin{aligned} f(x) &= \sum_{j=0}^m \frac{f^{(j)}(a)}{j!}(x-a)^j + \int_a^x \frac{(x-t)^m}{m!} f^{(m+1)}(t)dt \\ &= \sum_{j=0}^m \frac{f^{(j)}(a)}{j!}(x-a)^j + \int_a^b \frac{(x-t)_+^m}{m!} f^{(m+1)}(t)dt. \end{aligned}$$

Daraus folgt:

$$\begin{aligned} R_n(f) &= R_n\left(\sum_{j=0}^m \frac{f^{(j)}(a)}{j!}(x-a)^j\right) + R_n\left(\int_a^b \frac{z_{t,m}^+}{m!} f^{(m+1)}(t)dt\right) \\ &= 0 + \int_a^b \frac{1}{m!} R_n(z_{t,m}^+) f^{(m+1)}(t)dt \\ &= \int_a^b K_m(t) f^{(m+1)}(t)dt, \end{aligned}$$

wobei im zweiten Schritt verwendet wurde, dass R_n auf Π_m verschwindet und dass man das Integral mit R_n nach Fubini vertauschen darf und dass Q_n nur aus

Punktauswertungen besteht. Für den Beweis der zweiten Aussage benutzen wir den ersten Mittelwertsatz der Integralrechnung und erhalten

$$R_n(f) = \int_a^b K_m(t) f^{(m+1)}(t) dt = f^{(m+1)}(\xi) \cdot \int_a^b K_m(t) dt \tag{1.1}$$

mit $\xi \in (a, b)$, da $f^{(m+1)}$ und K_m stetig sind und $K_m(t) \neq 0$ für alle $t \in (a, b)$ gilt. Setzt man in (1.1) nun $f(x) = x^{m+1}$ ein, so ergibt sich:

$$R_n(x^{m+1}) = (m+1)! \int_a^b K_m(t) dt,$$

also

$$\frac{R_n(x^{m+1})}{(m+1)!} \cdot f^{(m+1)}(\xi) = f^{(m+1)}(\xi) \cdot \int_a^b K_m(t) dt.$$

QED

Wir betrachten im Folgenden einige Anwendungen von Satz 1.22: Wir leiten Fehlerschranken her für die Trapez-Regel, die Simpson-Regel, die zusammengesetzte Trapez-Regel, die zusammengesetzte Simpson-Regel und für die Gauß-Quadratur. Wir beginnen mit der Trapez-Regel.

Fehlerabschätzung für die Trapez-Regel.

Trapez-Regel: Es ist $n = 1$ und sie ist exakt für $m = 1$, für den Peano-Kern ergibt sich:

$$\begin{aligned}
K_1(t) &= \frac{1}{1!} R_1(z_{t,1}^+) \\
&= \int_t^b (x-t)^1 dx - \frac{b-a}{2} [\underbrace{z_{t,1}(a)}_{=0} + \underbrace{z_{t,1}(b)}_{=(b-t)}] \\
&= \left[\frac{1}{2}(x-t)^2 \right]_t^b - \frac{b-a}{2}(b-t) \\
&= \frac{1}{2} \left[(b-t)^2 - (b-a)(b-t) \right] \\
&= \frac{1}{2}(b-t)[b-t-b+a] \\
&= \frac{1}{2}(b-t)(a-t).
\end{aligned}$$

Da $K_1(t) \leq 0$ für alle $t \in [a,b]$ gilt, können wir den zweiten Teil von Satz 1.22 zum Abschätzen verwenden. Wir erhalten:

$$\begin{aligned} R_1(x^2) &= \int_a^b x^2 dx - \frac{b-a}{2}(a^2+b^2) \\ &= \frac{1}{3}b^3 - \frac{1}{3}a^3 - \frac{1}{2}b^3 + \frac{1}{2}a^3 - \frac{a^2b}{2} + \frac{ab^2}{2} \\ &= \frac{1}{6}a^3 - \frac{1}{6}b^3 - \frac{a^2b}{2} + \frac{ab^2}{2} \\ &= \frac{1}{6}(a-b)^3. \end{aligned}$$

Also existiert zu jedem $f \in C^2[a,b]$ ein $\xi \in [a,b]$ mit

$$R_1(f) = \frac{f''(\xi)}{2} \cdot \frac{1}{6}(a-b)^3 = -\frac{h^3}{12} f''(\xi) \text{ mit } h = \frac{b-a}{1}. \tag{1.2}$$

Fehlerabschätzung für die Simpson-Regel.

Es sind $n = 2$, $m = 3$, dann gilt nach einiger Rechnerei:

$$K_3(t) = \begin{cases} -\frac{(t-a)^3}{72}(a+2b-3t) & \text{für } a \leq t \leq \frac{a+b}{2} \\ -\frac{(b-t)^3}{72}(3t-2a-b) & \text{für } \frac{a+b}{2} \leq t \leq b \end{cases}$$

und $K_3(t) \leq 0$ für alle $t \in [a,b]$. Außerdem gilt:

$$R_2(x^4) = -\frac{(b-a)^5}{120}.$$

Daraus folgt:

$$R_2(f) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi) = -\frac{h^5}{90} f^{(4)}(\xi) \text{ mit } h = \frac{b-a}{2}.$$

Fehlerabschätzung für die zusammengesetzte Trapez-Regel.

Seien nun $x_0, \ldots, x_n$ gegeben. Sei

$$T_h = h \cdot \left(\frac{f(a)}{2} + \sum_{j=1}^{n-1} f(x_j) + \frac{f(b)}{2} \right)$$

die zusammengesetzte Trapez-Regel.

Satz 1.23 *Ist $f \in C^2[a,b]$ und $\Pi_n(f)$ die Näherung an $\int_a^b f(x)dx$ aus der zusammengesetzten Trapezregel (siehe Abschnitt 1.2), so gilt für den Fehler*

$$R(f) = I(f) - T_h(f) = -\frac{h^2(b-a)}{12} f''(\xi)$$

mit einem ξ aus $[a,b]$.

Beweis: Wir benutzen (1.2) auf jedem Teilintervall $[x_j, x_{j+1}]$, das heißt es existiert ein $\xi_j \in [x_j, x_{j+1}]$ mit

$$\begin{aligned}\int_a^b f(x)dx &= \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(x)dx \\ &= \sum_{j=0}^{n-1} \left(\frac{h}{2}(f(x_j) + f(x_{j+1})) - \frac{h^3}{12} f''(\xi_j) \right) \\ &= T_h(f) - \frac{h^3}{12} \cdot \sum_{j=0}^{n-1} f''(\xi_j) \\ &= T_h(f) - \frac{h^2(b-a)}{12} \cdot \frac{1}{n} \sum_{j=0}^{n-1} f''(\xi_j)\end{aligned}$$

Sei $\frac{1}{n}\sum_{j=0}^{n-1} f''(\xi_j) =: c$. Weil

$$\min_{j=0,\ldots,n-1} f''(\xi_j) \leq c \leq \max_{j=0,\ldots,n-1} f''(\xi_j)$$

gilt und f'' stetig ist, gibt es nach dem Zwischenwertsatz ein $\xi \in [a, b]$ mit $f(\xi) = c$. Also folgt

$$\int_a^b f(x)dx = T_h(f) - \frac{h^2(b-a)}{12} \cdot f''(\xi)$$

und daraus schließlich

$$R(f) = I(f) - T_h(f) = -\frac{h^2(b-a)}{12} \cdot f''(\xi).$$

QED

Bemerkung: Im Fall der Trapez-Regel liegt also sogar quadratische Konvergenz vor!

Fehlerabschätzung für die zusammengesetzte Simpson-Regel.

Für die zusammengesetzte Simpson-Regel ergibt sich

$$R(f) = I(f) - S_h(f) = -\frac{h^4(b-a)}{180} f^{(4)}(\xi) \text{ mit } \xi \in [a, b].$$

Fehlerabschätzung für die Gauß-Quadratur.

Lemma 1.24 *Sei Q_n die Gauß-Quadratur in $n+1$ Punkten aus $[a,b]$ mit zulässiger Gewichtsfunktion ω. Dann hat der zugehörige Peano-Kern K_m für $0 \leq m \leq 2n+1$ genau $2n+1-m$ Nullstellen in $[a,b]$. Insbesondere wechselt K_{2n+1} auf $[a,b]$ das Vorzeichen nicht.*

Satz 1.25 *Sei Q_n die Gauß-Quadratur in $n+1$ Punkten aus $[a,b]$ mit zulässiger Gewichtsfunktion ω. Zu $f \in C^{2n+2}[a,b]$ gibt es ein $\xi \in [a,b]$ so, dass*

$$R_n(f) = I_\omega(f) - Q_n(f) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \int_a^b \omega(x)(h_{n+1}(x))^2 dx,$$

wobei wie üblich $h_{n+1}(x) = \prod_{j=0}^{n}(x - x_j)$.

Beweis: Wegen Lemma 1.24 darf man den 2. Teil von Satz 1.22 anwenden. Man erhält

$$R_n(f) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} R_n(x^{2n+2})$$

und rechnet nach, dass

$$R_n(x^{2n+2}) = \int_a^b \omega(x)(h_{n+1}(x))^2$$

gilt. QED

1.5 Romberg-Verfahren

Wir benötigen folgende Begriffe.

Definition 1.26 *Die* Bernoulli-Polynome $B_k \in \Pi_k$ *für* $k = 0, 1, \ldots$ *sind rekursiv definiert durch:*

$$B_0(t) := 1$$
$$\text{und } B_k'(t) = B_{k-1}(t) \text{ und } \int_0^1 B_k(t)dt = 0 \quad k = 1, 2, \ldots.$$

Die Zahlen $b_k := k!B_k(0)$ heißen Bernoulli-Zahlen.

Das $(k+1)$-te Bernoulli-Polynom entsteht also durch Integration aus dem k-ten, wobei die zweite Bedingung die Integrationskonstanten festlegt. Es gilt:

$$B_1'(t) = 1 \Rightarrow B_1(t) = t + C.$$

Wegen

$$\int_0^1 t + C = \left[\frac{1}{2}t^2 + Ct\right]_0^1 = \frac{1}{2} + C \stackrel{!}{=} 0$$

folgt $C = \frac{1}{2}$, also

$$B_1(t) = t - \frac{1}{2}$$

Ähnlich ergibt sich

$$B_2(t) = \frac{1}{2}t^2 - \frac{1}{2}t + \frac{1}{12}.$$

Lemma 1.27 *Es gilt:*

1. $B_k \in \Pi_k$ *für* $k = 1, 2, \ldots$.
2. $B_k(t) = (-1)^k B_k(1-t)$ *für* $k = 0, 1, 2, \ldots$.
3. $B_k(0) = B_k(1)$ *für* $k = 2, 3, \ldots$.
4. *Für* $m = 1, 2, \ldots$ *besitzt das Polynom* $B_{2m} - B_{2m}(0)$ *genau die Nullstellen* 0 *und* 1 *im Intervall* $[0, 1]$ *und das Polynom* B_{2m+1} *genau die Nullstellen* 0, $\frac{1}{2}$ *und* 1.

Mit Hilfe der Bernoulli-Zahlen untersuchen wir nochmals den bei der zusammengesetzten Trapez-Regel entstehenden Fehler in Abhängigkeit der Intervalllänge h. Sei dazu wie bisher

$$T_h(f) = h\left(\tfrac{1}{2}f(a) + \sum_{j=1}^{m-1} f(x_j) + \tfrac{1}{2}f(b)\right)$$

mit $h = \frac{b-a}{n}$ und $x_0 = a$, $x_j = a + jh$ und $x_m = b$ der Wert der zusammengesetzten Trapezregel.

Satz 1.28 (Euler-McLaurinsche Summenformel) *Sei* $l \in \mathbb{N}$ *und* $f \in C^{2l}([a, b])$. *Dann gilt*

$$T_h(f) \;=\; I(f) \;+\; \sum_{j=1}^{l-1} \frac{b_{2j} h^{2j}}{(2j)!}\left[f^{(2j-1)}(b) \;-\; f^{(2j-1)}(a)\right] \;+\; \frac{(b-a) b_{2l} h^{2l}}{(2l)!} f^{(2l)}(\xi)$$

für ein $\xi = \xi(h) \in (a, b)$.

Beweisidee: Partielle Integration von $\int f(t)dt = \int B_0(\frac{t-a}{h}) f(t)$ und Mittelwertsatz der Integralrechnung.

Korollar 1.29 *Ist f periodisch auf $[a,b]$ und genügt den Voraussetzungen aus Satz 1.28, so gibt es ein $\xi \in (a,b)$, so dass*

$$T_h(f) = I(f) + \frac{(b-a)b_{2l}h^{2l}}{(2l)!} f^{(2l)}(\xi).$$

Beweis: Da f periodisch auf $[a,b]$ ist, gilt

$$f^{(i)}(b) = f^{(i)}(a) \text{ für alle } i = 0, 1, \ldots, 2l.$$

QED

Im Wesentlichen besagt die Euler-McLaurinsche Formel also, dass man den Fehler bei der zusammengesetzten Trapezregel schreiben kann als

$$T_h(f) - I(f) = a_2 h^2 + a_4 h^4 + \cdots + a_{2l-2} h^{2l-2} + a_{2l}(h) h^{2l}$$

mit Koeffizienten $a_2, a_4, \ldots, a_{2l-2} \in \mathbb{R}$ und einer Funktion $a_{2l} : \mathbb{R} \to \mathbb{R}$. Genauer gilt:

$$a_{2j} = \frac{b_{2j}}{(2j)!}\left(f^{(2j-1)}(b) - f^{(2j-1)}(a)\right)$$

$$\text{und} \qquad a_{2l}(h) = \frac{(b-a)b_{2l}}{(2l)!} f^{(2l)}(\xi(h)). \tag{1.3}$$

Weil $f^{(2l)}(\xi)$ als stetige Funktion auf $[a,b]$ beschränkt ist, ist auch a_{2l} beschränkt, weshalb gilt

$$\lim_{h \to 0} T_h(f) = I(f).$$

Allerdings geht der Rechenaufwand für $h \to 0$ gegen Unendlich. Die Idee des Romberg-Verfahrens ist es nun, $T_0(f)$ durch „Extrapolation“ folgendermaßen abzuschätzen: Setze $\tau = h^2$ als das Quadrat der Intervalllänge.

1. Sei $g(\tau) := T_{\sqrt{\tau}}(f)$ für alle $\tau \neq 0$, $g(0) := T_0(f)$.
2. Bestimme $g(\tau_0), \ldots, g(\tau_l)$ für $l+1$ Stützstellen $\tau_j := h_j^2$ für Intervalllängen $h_0, \ldots, h_l$ mit $h_j = \frac{b-a}{m_j}$, $m_j \in \mathbb{N}$.
3. Interpoliere g an den $l+1$ Stützstellen durch ein Polynom $p \in \Pi_l$, also mit
$$p(\tau_j) = g(\tau_j) = T_{h_j}(f), \qquad j = 0, \ldots, l.$$
4. Approximiere
$$I(f) = \int_a^b f(x)dx = \lim_{h \to 0} T_h(f) \approx \lim_{h \to 0} p(h^2) = p(0).$$

Weil wir das Interpolationspolynom nicht selbst kennen müssen, sondern nur an seinem Wert an der Stelle 0 interessiert sind, bietet sich zur Berechnung das Verfahren von Neville-Aitken (siehe Numerik I) an.

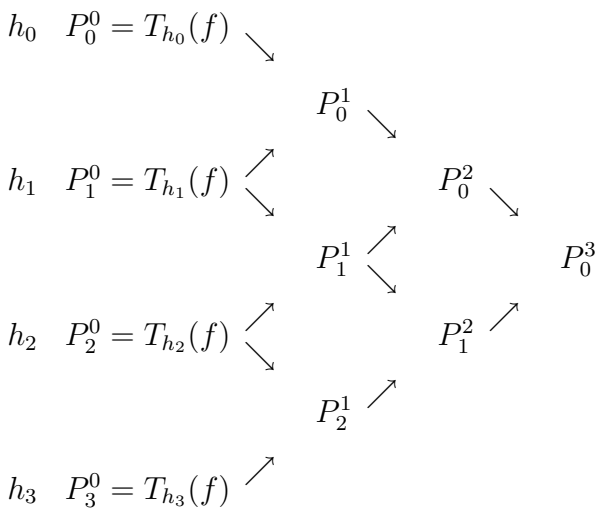
Nach Satz 6.6 aus Numerik I kann folgende Formel verwendet werden: sei $P_i^k(\tau)$ das Polynom, das g an den Stützstellen $\tau_i, \tau_{i+1}, \ldots, \tau_{i+k}$ interpoliert. Dann gilt:

$$P_i^{k+1}(\tau) = \frac{(\tau - \tau_i)P_{i+1}^k - (\tau - \tau_{i+k+1})P_i^k}{\tau_{i+k+1} - \tau_i}.$$

Für $P_i^k := P_i^k(0)$ gilt somit

$$P_i^k = \frac{\tau_i P_{i+1}^{k-1} - \tau_{i+k} P_i^{k-1}}{\tau_i - \tau_{i+k}}$$

und die Werte lassen sich berechnen durch folgendes Schema:



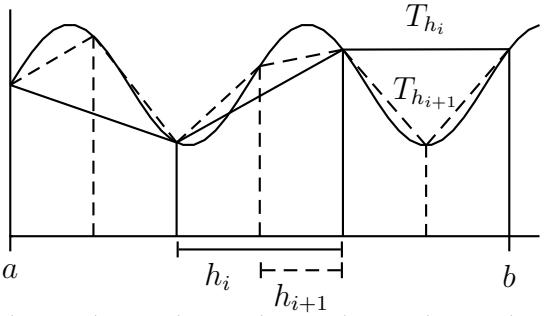
Jeder der Einträge stellt eine eigene Quadraturformel dar. Verwendet man z.B.

$$\tau_i = 2^{-2i} h_0^2 \text{ bzw. } h_i = 2^{-i} h_0,$$

so erhält man aus der ersten Spalte:

$$
\begin{aligned}
P_i^1 &= \frac{2^{-2i}h_0^2 P_{i+1}^0 - 2^{-2i-2}h_0^2 P_i^0}{2^{-2i}h_0^2 - 2^{-2i-2}h_0^2} \\
&= \frac{T_{h_{i+1}}(f) - \frac{1}{4}T_{h_i}(f)}{1-\frac{1}{4}} = \tfrac{4}{3}T_{h_{i+1}}(f) - \tfrac{1}{3}T_{h_i}(f) \\
&= \tfrac{4}{3}\left(\tfrac{1}{2}f(a) + \sum_{j=1}^{2m_i-1} f(a+\tfrac{1}{2}jh_i) + \tfrac{1}{2}f(b)\right)h_{i+1} \\
&\quad - \tfrac{1}{3}\left(\tfrac{1}{2}f(a) + \sum_{j=1}^{m_i-1} f(a+jh_i) + \tfrac{1}{2}f(b)\right)h_i \\
&= h_{i+1}\Big(\tfrac{1}{3}f(a) + \tfrac{4}{3}f(a+\tfrac{1}{2}jh_i) + \tfrac{2}{3}f(a+jh_i) \\
&\quad + \tfrac{4}{3}f(a+\tfrac{3}{2}jh_i) + \cdots + \tfrac{1}{3}f(b)\Big),
\end{aligned}
$$

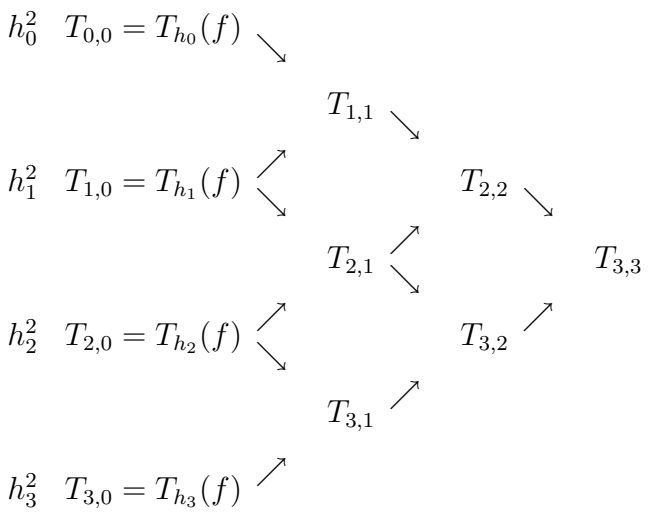
wobei wir im letzten Schritt $\frac{h_i}{h_{i+1}} = 2$ verwendet haben.



Gewichte:							
erster Term:	$\frac{4}{6}$	$\frac{4}{3}$	$\frac{4}{3}$	$\frac{4}{3}$	$\frac{4}{3}$	$\frac{4}{3}$	$\frac{4}{6}$
zweiter Term:	$-\frac{2}{6}$	0	$-\frac{2}{3}$	0	$-\frac{2}{3}$	0	$-\frac{2}{6}$
gesamt:	$\frac{1}{3}$	$\frac{4}{3}$	$\frac{2}{3}$	$\frac{4}{3}$	$\frac{2}{3}$	$\frac{4}{3}$	$\frac{1}{3}$

Man erhält also genau die zusammengesetzte Simpson-Regel.

Bemerkung: In der Literatur wird das Schema anders nummeriert, nämlich durch



mit $T_{i,k} = P_{i-k}^k$ bzw. $P_i^k = T_{i+k,k}$ und

$$T_{i,k} = \frac{h_{i-k}^2 T_{i,k-1} - h_i^2 T_{i-1,k-1}}{h_{i-k}^2 - h_i}.$$

Zum Abschluss untersuchen wir, wann Romberg-Quadraturen exakt sind.

Satz 1.30 *Die Romberg-Quadraturen $P_i^k(f)$ (bzw. $T_{i+k,k}(f)$) sind exakt für Polynome vom Grad kleiner gleich $2k$.*

Beweis: Ist $f \in \Pi_{2k}$, so folgt, dass $f^{(2k)}$ konstant ist, also ist $a_{2k}(h)$ in (1.3) auf Seite 26 konstant. Es gilt $a_{2k}(h) = a_{2k}$. Nach Satz 1.28 erhalten wir, dass

$$T_h(f) = I(f) + a_2 h^2 + a_4 h^4 + \cdots + a_{2k-2} h^{2k-2} + a_{2k} h^{2k}$$

ein Polynom vom Grad kleiner gleich k in der Variablen h^2 ist, beziehungsweise dass

$$g(\tau) = T_{\sqrt{\tau}}(f) = I(f) + a_2 \tau + a_4 \tau^2 + \cdots + a_{2k} \tau^k$$

ein Polynom aus Π_k ist. Aufgrund der Eindeutigkeit der Polynominterpolation folgt

$$p(\tau) \equiv g(\tau).$$

Insbesondere gilt $p(0) = g(0)$, also

$$P_i^k(f) = p(0) = g(0) = T_0(f) = I(f).$$

QED

Wie wählt man die Schrittweiten $h_k = \frac{b-a}{n_k}$? Dazu gibt es die

- klassische Romberg-Folge:

  $$n_k = 2^k \Rightarrow h_k = \tfrac{1}{2} h_{k-1}.$$

  Der Vorteil liegt darin, dass Funktionsauswertungen von einem Schritt i auf den nächsten Schritt $i+1$ wiederverwendet werden können. Der Nachteil ist, dass die Folge sehr schnell wächst!

- harmonische Folge:

  $$n_k = k + 1.$$

  Im Gegensatz zur klassischen Romberg-Folge wächst diese langsamer, doch sind alte Funktionsauswertungen im $(i+1)$-ten Schritt unbrauchbar. Daher wählt man als Kompromiss die

- Burlisch-Folge:

  $$\begin{aligned} n_0 &= 1 \\ n_{2k-1} &= 2^k \\ n_{2k} &= 3 \cdot 2^{k-1}. \end{aligned}$$

1.6 Zusammenfassung

Ziel:

- *Einfache Formel für*

$$I_\omega(f) = \int_a^b \omega(x) f(x) dx$$

- *"einfach": Quadraturformeln*

$$Q_n(f) := \sum_{i=0}^{n} a_i f(x_i)$$

Interpolationsquadraturen

Seien $x_0, \ldots, x_n$ gegeben. Sei

$$Q_n(f) := \int_a^b (L_n f)(x) dx.$$

- Quadratur Q_n Interpolationsquadratur $\Leftrightarrow$ Q_n $\forall p \in \Pi_n$ exakt
- $Q_n(f)$ ist eindeutig bestimmt
- Newton-Côtes Formeln: Trapez-Regel, Simpson-Regel, ...
  - n ungerade: exakt auf Π_n
  - n gerade: exakt auf Π_{n+1}

Zusammengesetzte Newton-Côtes Formeln

$$T_h(f) = h\left(\tfrac{1}{2} f(x_0) + \sum_{i=1}^{m-1} f(x_i) + \tfrac{1}{2} f(x_m)\right)$$

$$S_h(f) = \frac{h}{3}\left(\sum_{j=0}^{\frac{m}{2}-1} f(x_{2j}) + 4f(x_{2j+1}) + f(x_{2j+2})\right)$$

Gauß'sche Quadraturen

Wähle auch $x_0, \ldots, x_n$. Sei $Q_n(f)$ Gauß'sche Quadratur falls exakt $\forall p \in \Pi_{2n+1}$.

- $Q_n(f) := \int_a^b \omega(x)(L_n f)(x) dx$ Gauß'sche Quadratur
  $\Leftrightarrow \int_a^b \omega(x) h_{n+1}(x) p(x) dx (= (h_{n+1}, p)_\omega) = 0.$
- Konstruktion der ω-orthogonalen Polynome (Orthogonalbasis)

- $x_0, \ldots, x_n$ sind Nullstellen des Polynoms $p \in \Pi_n$, das $(p_{n+1}, f) = 0, \forall f \in \Pi_n$ erfüllt.
- Gewichte alle > 0!
  - $I = [-1, 1]$, $\omega \equiv 1 \Rightarrow$ Legendre-Polynome
  - $I = [-1, 1]$, $\omega(x) = \frac{1}{\sqrt{1-x^2}} \Rightarrow$ Tschebyscheff-Polynome

Fehleranalyse

- $Q_n(f) \to I_\omega(f), \forall f \in C[a, b]$, falls $Q_n(p) \to I_\omega(p), \forall p \in \Pi_\infty$ und $\sum_{j=0}^{n} |a_j^{(n)}| \leq C$, $\forall n$.
- Gauß-Quadraturen konvergieren
- Newton-Côtes nicht
- Restglied $R_n(f) = I_\omega(f) - Q_n(f)$
- Peano-Kern: $K_m(t) := \frac{1}{m!} R_n(z_{t,m}^+)$
- Q_n auf Π_m exakt. Dann
  - $R_n(f) = \int_a^b K_m(t) f^{(m+1)} t dt$
  - Wechselt K_m das Vorzeichen nicht, so existiert $\xi \in [a, b]$:

$$R_n(f) = \frac{f^{(m+1)}(\xi)}{(m+1)!} R_n(x^{m+1}).$$

- Trapez-Regel: $R_1(f) = -\frac{h^3}{12} f''(\xi)$
- Simpson-Regel: $R_2(f) = -\frac{h^5}{90} f^{(4)}(\xi)$
- zusammengesetzte Trapez-Regel: $R(f) = -\frac{h^2(b-a)}{12} f''(\xi)$
- zusammengesetzte Simpson-Regel: $R(f) = -\frac{h^4(b-a)}{180} f^{(4)}(\xi)$
- Gauß: $R_n(f) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \int_a^b \omega(x)(h_{n+1}(x))^2 dx$

Romberg-Verfahren

- Euler McLaurinsche Summenformel:

$$T_h(f) - I(f) = a_2 h^2 + a_4 h^4 + \cdots a_{2l+2} h^{2l-2} + a_{2l}(h) h^{2l}$$

- Extrapolation:

  1. $\tau = h^2$, $g(\tau) := T_h(f)$
  2. Bestimme $g(\tau_0), \ldots, g(\tau_l)$ mit $\tau_j = (\frac{b-a}{m_j})^2$
  3. Interpoliere g durch Polynom p (Neville-Aitken)
  4. $T_0(h) := p(0)$

- Romberg-Quadraturen P_i^k (Interpolationen $\tau_i, \ldots, \tau_{i+k}$)) exakt $\forall p \in \Pi_{2k}$

Kapitel 2

Approximationstheorie

In diesem Kapitel wollen wir eine Funktion f durch eine "einfache" Funktion u (mit $u \in U \subseteq C[a,b]$, z.B. $U = \Pi_n$) annähern. Bei der **Interpolation** sollte u an gegebenen Punkten mit f übereinstimmen (s. Numerik I, Kapitel 6). Bei der **Approximation** soll u die Funktion f im ganzen Definitionsbereich "gut" darstellen. Unter "gut" verstehen wir, dass $\|f-u\|$ klein ist und beschäftigen uns hauptsächlich mit der Tschebyscheff-Norm $\|f\|_\infty := \max_{x\in[a,b]} |f(x)|$.

2.1 Approximationssätze von Weierstraß

In diesem Abschnitt wollen wir den in Abschnitt 1.4 schon benutzten Satz von Weierstraß (Satz 1.19) beweisen. Dazu benutzen wir so genannte *Korovkin-Operatoren*.

Definition 2.1 *Eine Abbildung $K : C[a,b] \to C[a,b]$ heißt **monoton**, falls für alle $f, g \in C[a,b]$ gilt*

$$f(x) \le g(x), \quad \forall x \in [a,b] \quad \Rightarrow \quad Kf(x) \le Kg(x), \quad \forall x \in [a,b].$$

*Eine Folge $K_n : C[a,b] \to C[a,b]$, $n \in \mathbb{N}$ heißt **Korovkin-Folge**, falls*

(a) K_n ist monotoner, linearer Operator für alle $n \in \mathbb{N}$.

(b) $\lim_{n\to\infty} \|K_n f - f\|_\infty = 0$ für $f \in \{\mathbf{1}, x, x^2\}$ (gleichmäßige Konvergenz).

Bemerkung: Ist K_n Korovkin-Folge, so gilt

$$\lim_{n\to\infty} \|K_n f - f\|_\infty = 0, \quad \forall f \in \Pi_2,$$

denn: $f \in \Pi_2$ lässt sich schreiben als $f(x) = \alpha x + \beta x + \gamma \cdot \mathbf{1}$, also ist

$$\begin{aligned} \|K_n f - f\| &= \|\alpha K_n(x^2) + \beta K_n(x) + \gamma K_n(\mathbf{1}) - \alpha x^2 - \beta x - \gamma\|, \text{ da } K_n \text{ linear} \\ &\le \underbrace{|\alpha| \|K_n x^2 - x^2\|}_{\to 0} + \underbrace{|\beta| \|K_n x - x\|}_{\to 0} + \underbrace{|\gamma| \|K_n \mathbf{1} - \mathbf{1}\|}_{\to 0} \\ &\to 0 \text{ für } n \to \infty. \end{aligned}$$

Überraschenderweise folgt aus der gleichmäßigen Konvergenz auf Π_2 sogar die gleichmäßige Konvergenz für alle stetigen Funktionen!

Satz 2.2 *Ist $\{K_n\}$ eine Korovkin-Folge auf $C[a,b]$, so gilt*

$$\lim_{n\to\infty} \|K_n f - f\|_\infty = 0 \text{ für alle } f \in C[a,b].$$

Beweis: Ist f stetig auf $[a,b]$, so ist f sogar gleichmäßig stetig auf $[a,b]$, d.h. zu $\varepsilon > 0$ existiert ein $\delta > 0$, so dass

$$|f(x) - f(y)| \leq \frac{\varepsilon}{3} \text{ für alle } x, y \in [0,1] \text{ mit } |x-y| < \delta.$$

Sei nun $t \in [a,b]$ fest.

- Falls $|x-t| < \delta$ gilt also $|f(x) - f(t)| < \frac{\varepsilon}{3}$.
- Falls $|x-t| \geq \delta$, so gilt

$$\begin{aligned} |f(x) - f(t)| &\leq |f(x)| + |f(t)| \leq 2\|f\|_\infty \\ &\leq 2\|f\|_\infty \underbrace{\left(\frac{x-t}{\delta}\right)^2}_{\geq 1}. \end{aligned}$$

Zusammen erhalten wir

$$\forall x \in [a,b] : |f(x) - f(t)| \leq \underbrace{\frac{\varepsilon}{3}}_{\geq 0} + \underbrace{2\|f\|_\infty \left(\frac{x-t}{\delta}\right)^2}_{\geq 0}. \tag{2.1}$$

Seien nun

$$\begin{aligned} p_t(x) &= f(t) - \frac{\varepsilon}{3} - 2\|f\|_\infty \left(\frac{x-t}{\delta}\right)^2 \\ q_t(x) &= f(t) + \frac{\varepsilon}{3} + 2\|f\|_\infty \left(\frac{x-t}{\delta}\right)^2. \end{aligned}$$

Dann lässt sich (2.1) schreiben als

$$p_t(x) \leq f(x) \leq q_t(x), \quad \forall x \in [a,b]. \tag{2.2}$$

K_n ist monoton für alle n, also gilt $K_n p_t(x) \leq K_n f(x) \leq K_n q_t(x)$. Weil $p_t, q_t \in \Pi_2[a,b]$ konvergiert die Anwendung der K_n auf sie gleichmäßig (in x), d.h.

$$\begin{aligned} |K_n q_t(x) - q_t(x)| &\to 0 \text{ für } n \to \infty \\ |K_n p_t(x) - p_t(x)| &\to 0 \text{ für } n \to \infty \end{aligned}$$

für alle x und für alle t.
Wir möchten nun ein $N \in \mathbb{N}$, so wählen, dass für alle $n \geq N$, für alle $x \in [a,b]$ und für *alle* $t \in [a,b]$ gilt

$$\begin{aligned} |K_n q_t(x) - q_t(x)| &\leq \frac{\varepsilon}{3} \\ |K_n p_t(x) - p_t(x)| &\leq \frac{\varepsilon}{3}. \end{aligned} \tag{2.3}$$

Dazu ist gleichmäßige Konvergenz von $K_n q_t(x) - q_t(x)$ in x und in t nötig. Diese zeigt man für q_t wie folgt:

$$\begin{aligned} q_t(x) &= f(t) + \frac{\varepsilon}{3} + 2\|f\|_\infty \frac{(x-t)^2}{\delta^2} \\ &= f(t) + \frac{\varepsilon}{3} + \frac{2\|f\|_\infty}{\delta^2}(x^2 - 2tx + t^2) \\ &= \mathbf{1}\left(f(t) + \frac{\varepsilon}{3} + \frac{2t^2\|f\|_\infty}{\delta^2} \right) - 4tx\frac{\|f\|_\infty}{\delta^2} + 2x^2\frac{\|f\|_\infty}{\delta^2}. \end{aligned}$$

Man beachte, dass ein Polynom vom Grad zwei in x vorliegt. Aus letzterer Überlegung ergibt sich:

$$\begin{aligned} |K_n q_t(x) - q_t(x)| = \Bigg| &(K_n\mathbf{1} - \mathbf{1})\left[f(t) + \frac{\varepsilon}{3} + \frac{2t^2\|f\|_\infty}{\delta^2} \right] \\ &+ (K_n x - x)\left[\frac{-4t\|f\|_\infty}{\delta^2} \right] + (K_n x^2 - x^2)\left[\frac{2\|f\|_\infty}{\delta^2} \right] \Bigg| \\ \leq \|K_n\mathbf{1} - \mathbf{1}\|_\infty &\left(\|f\|_\infty + \frac{\varepsilon}{3} + \frac{2c^2\|f\|_\infty}{\delta^2} \right) \\ &+ \|K_n x - x\|_\infty \frac{4c\|f\|_\infty}{\delta^2} + \|K_n x^2 - x^2\|_\infty \frac{2\|f\|_\infty}{\delta^2} \end{aligned}$$

mit $c := \max\{|a|, |b|\}$. Dieser Ausdruck hängt weder von x noch von t ab und strebt gleichmäßig gegen Null. Für p_t erhält man analog einen ähnlichen Ausdruck.
Damit finden wir also $N \in \mathbb{N}$, so dass (2.3) gilt und erhalten daraus:

$$p_t(x) - \frac{\varepsilon}{3} \leq K_n f(x) \leq q_t(x) + \frac{\varepsilon}{3}. \tag{2.4}$$

Es folgt für alle x, t und $n > N$:

$$\begin{aligned} p_t(x) - q_t(x) - \frac{\varepsilon}{3} &\leq f(x) - q_t(x) - \frac{\varepsilon}{3}, \text{ denn } p_t(x) \leq f(x) \text{ nach (2.2)} \\ &\leq f(x) - K_n f(x), \text{ weil } K_n f(x) \leq q_t(x) + \tfrac{\varepsilon}{3} \text{ nach (2.4)} \\ &\leq f(x) - p_t(x) + \frac{\varepsilon}{3}, \text{ durch } p_t(x) - \tfrac{\varepsilon}{3} \leq K_n f(x) \text{ aus (2.4)} \\ &\leq q_t(x) - p_t(x) + \frac{\varepsilon}{3}, \text{ da } f(x) \leq q_t(x) \text{ nach (2.2)}. \end{aligned}$$

Insbesondere gilt das auch für $t = x$. Wegen

$$\begin{aligned} p_x(x) - q_x(x) &= f(x) - \tfrac{\varepsilon}{3} - 2\|f\|_\infty \cdot 0 - f(x) - \tfrac{\varepsilon}{3} - 2\|f\|_\infty \cdot 0 \\ &= -\tfrac{2}{3}\varepsilon \end{aligned}$$

gilt also

$$-\tfrac{2}{3}\varepsilon - \tfrac{\varepsilon}{3} \leq f(x) - K_n f(x) \leq \tfrac{2}{3}\varepsilon + \tfrac{\varepsilon}{3}$$

oder

$$|f(x) - K_n f(x)| \leq \varepsilon$$

für alle $n \geq N$ und $x \in [a, b]$. QED

Jetzt kann man zeigen, dass jede stetige Funktion beliebig gut durch Polynome approximiert werden kann, indem man eine Folge von Korovkin-Operatoren

$$K_n : C[a, b] \to \Pi_n$$

angibt, die jede stetige Funktion auf ein Polynom abbilden. Das wird durch die Bernstein-Operatoren erfüllt.

Notation 2.3

$$B_n : C[0, 1] \to \Pi_n(\mathbb{R}),$$

definiert durch

$$B_n f(x) := \sum_{j=0}^{n} \binom{n}{j} f\left(\frac{j}{n}\right) x^j (1 - x)^{n-j}, \quad x \in [0, 1]$$

nennt man ***Bernstein-Operatoren****.*

Satz 2.4 *Die Bernsteinoperatoren bilden eine Korovkin-Folge auf* $C[0, 1]$.

Beweis:

(a) Die B_n sind linear und monoton, da $x \geq 0$ und $1 - x \geq 0$ für alle $x \in [0, 1]$.

(b) Zu zeigen bleibt noch: $B_n f - f \to 0$ für $n \to \infty$ für $f \in \{\mathbf{1}, x, x^2\}$.
Wir betrachten zunächst den Fall $f(x) = \mathbf{1}$:

$$B_n \mathbf{1}(x) = \sum_{j=0}^{n} \binom{n}{j} x^j (1 - x)^{n-j} = 1 = \mathbf{1}(x),$$

nach dem Binomischen Lehrsatz, also ist $B_n \mathbf{1} = \mathbf{1}$.

Sei nun $f(x) = x$:

$$\begin{aligned} B_n x &= \sum_{j=1}^{n} \binom{n}{j} \frac{j}{n} x^j (1-x)^{n-j} \\ &= \sum_{j=0}^{n-1} \binom{n}{j+1} \frac{j+1}{n} x^{j+1} (1-x)^{n-j-1} \\ &= x \underbrace{\sum_{j=0}^{n-1} \binom{n-1}{j} x^j (1-x)^{(n-1)-j}}_{=1}, \text{ denn } \binom{x}{y}\frac{y}{x} = \binom{x-1}{y-1} \\ &= x = f(x). \end{aligned}$$

Wir betrachten abschließend $f(x) = x^2$. Nach etwas Rechnen erhält man

$$B_n f(x) = \frac{n-1}{n} x^2 + \frac{x}{n},$$

und somit

$$\begin{aligned} |f(x) - B_n f(x)| &= \left| x^2 - \frac{n-1}{n} x^2 - \frac{x}{n} \right| = \left| \frac{1}{n} x^2 - \frac{x}{n} \right| \\ &\leq \left| \frac{x^2}{n} \right| + \left| \frac{x}{n} \right| \leq \frac{2}{n} \to 0, \end{aligned}$$

also $\|f - B_n f\|_\infty \to 0$ für $n \to \infty$. QED

Damit folgt der Satz von Weierstraß:

Satz 2.5 (Weierstraß) *Zu jedem $f \in C[a,b]$ und jedem $\varepsilon > 0$ gibt es ein Polynom p so, dass $\|f - p\|_\infty < \varepsilon$.*

Beweis: Für $[a,b] = [0,1]$ folgt die Aussage aus Satz 2.2 und Satz 2.4. Im allgemeinen Fall sei $f \in C[a,b]$. Wir definieren

$$g(s) := f((b-a)s + a) \in C[0,1].$$

Zu g existiert ein Polynom q, so dass $\|g - q\|_\infty < \varepsilon$. Sei weiterhin $p(t) := q(\frac{t-a}{b-a})$, $t \in [a,b]$. Dann ist p ein Polynom und weil $t = (b-a)s + a$ äquivalent ist zu $\frac{t-a}{b-a} = s$ folgt

$$f(t) - p(t) = g\left(\frac{t-a}{b-a}\right) - q\left(\frac{t-a}{b-a}\right)$$

und daraus

$$\|f - p\|_\infty = \|g - q\|_\infty < \varepsilon,$$

also ist p das gesuchte Polynom für f. QED

Bemerkung: Für $f \in C[a,b]$ definiert man die Bernstein-Operatoren vermöge

$$\underline{B}_n f(x) = \sum_{j=0}^{n} \binom{n}{j} f\left(a + (b-a)\frac{j}{n}\right) \underbrace{\left(\frac{x-a}{b-a}\right)^j}_{=:y} \underbrace{\left(\frac{b-x}{b-a}\right)^{n-j}}_{=1-y}$$

$$= \frac{1}{(b-a)^n} \sum_{j=0}^{n} \binom{n}{j} f\left(a + (b-a)\frac{j}{n}\right)(x-a)^j(b-x)^j$$

indem man

$$[a,b] \to [0,1] \quad \text{via} \quad x \to \frac{x-a}{b-a}$$

abbildet.

Übungsaufgabe: Wandeln Sie $B_n f$ so ab, dass sie eine Korovkin-Folge auf $C[a,b]$ erhalten. (Das ist ein alternativer Beweis zum Satz 2.5).

In Satz 2.5 haben wir den Abstand zwischen der Funktion f und ihrer Approximation durch

$$\|f - p\|_\infty := \max_{x \in [a,b]} |f(x) - g(x)|$$

gemessen. Statt der Norm $\|\cdot\|_\infty$ verwenden wir im folgenden Satz die $L_p[a,b]$-Normen, die durch

$$\|f\|_p := \sqrt[p]{\int_a^b |f(x)|^p dx}$$

definiert sind.

Satz 2.6 *Zu jedem $f \in C[a,b]$ und jedem $\varepsilon > 0$ gibt es ein Polynom q so, dass $\|f - q\|_p < \varepsilon$.*

Beweis: Sei $\varepsilon > 0$. Nach Satz 2.5 gibt es ein Polynom q, so dass $\|f - q\|_\infty < \varepsilon' := \frac{\varepsilon}{(b-a)}$. Dann gilt

$$\begin{aligned} \|f - q\|_p^p &= \int_a^b |f(x) - q(x)|^p dx \\ &\leq \|f - q\|_\infty^p \int_a^b 1 dx \\ &= \|f - q\|_\infty^p (b-a) < (\varepsilon')^p (b-a) = \varepsilon^p, \end{aligned}$$

also $\|f - q\|_p \leq \varepsilon$. QED

Von Weierstraß stammt auch das folgende Approximationsresultat für trigonometrische Polynome, das wir ohne Beweis angeben:

Satz 2.7 *Zu jedem $f \in C(\mathbb{R})$ mit Periode 2π und jedem $\varepsilon > 0$ existiert ein trigonometrisches Polynom T so, dass $\|f - T\|_\infty < \varepsilon$ und $\|f - T\|_p < \varepsilon$ für alle $L_p[0, 2\pi]$-Normen.*

2.2 Existenzsätze

Wir verallgemeinern nun den Begriff der Approximation.

Definition 2.8 *Sei V ein normierter Vektorraum und $M \subseteq V$ eine Teilmenge von V. Sei $f \in V$. Dann heißt $u^* \in M$* ***beste Approximation*** *an f, falls*

$$\|f - u^*\| \leq \|f - u\| \text{ für alle } u \in M.$$

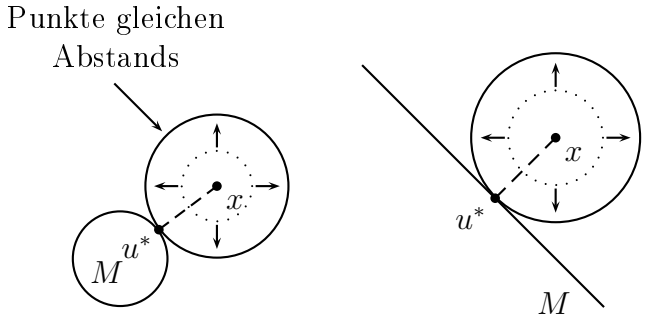
Man nennt $d(f, M) := \inf_{u \in M} \|f - u\|$ den (Minimal-)Abstand von f zu M.

Beispiele:

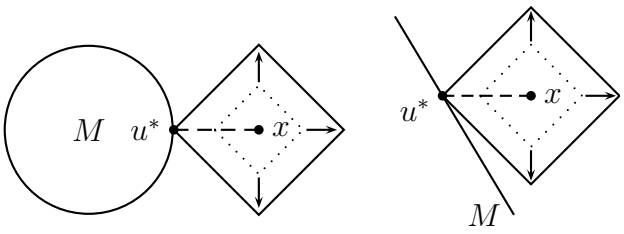
1. $V = C[a, b]$, $M = \Pi_4[a, b]$: Approximation einer stetigen Funktion $f \in V$ durch ein Polynom bis Grad 4.

2. $V = \mathbb{R}^n$, $M \subseteq \mathbb{R}^n$: Approximation eines Punktes durch einen (anderen) Punkt aus M. Hierbei ist $d(x, M)$ der Abstand des Punktes $x \in V$ von der Menge M.

   Für $\|\cdot\| = \|\cdot\|_2$ ist u^* die orthogonale Projektion von x auf M.



   Für $\|\cdot\| = \|\cdot\|_1$ ist u^* wie in der Abbildung.



3. In der linearen Ausgleichsrechnung (Numerik I, Kapitel 4.2) sind $A \in \mathbb{R}^{m,n}$ mit $m > n$ und $b \in \mathbb{R}^m$ gegeben. Gesucht ist ein $x \in \mathbb{R}^n$, sodass $\|Ax - b\|_2$ möglichst klein ist. Wir formulieren das Problem zu einer Approximationsaufgabe um: Seien $V = \mathbb{R}^m$, $M = \{Ax : x \in \mathbb{R}^n\}$ und $b \in V$ gegeben. Finde $u^* \in M$, sodass $\|b - u^*\|$ möglichst klein ist.

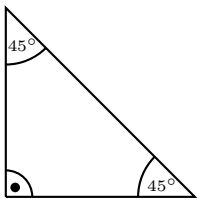
Definition 2.9 *Sei $M \subseteq V$, V normierter Vektorraum. M heißt* ***Existenzmenge****, falls es zu jedem $f \in V$ eine beste Approximation auf f gibt. M heißt Tschebyscheff-Menge, falls es zu jedem $f \in V$ genau eine beste Approximation gibt. M heißt* ***dicht in*** *V, falls $d(f, M) = 0$ für alle $f \in V$.*

Beispiele:

- $\Pi_\infty \subseteq C[a, b]$ ist keine Existenzmenge, aber
- der Satz von Weierstraß (Satz 2.5) besagt, dass für $M = \Pi_\infty$ – den Raum aller Polynome – gilt

$$\begin{aligned} d(f, M) &= \inf_{p \in M} \|f - q\|_\infty \\ &= 0 \text{ für alle } f \in C[a, b], \end{aligned}$$

  also liegt Π_∞ dicht in $C[a, b]$.
- Dagegen ist $\Pi_4[a, b]$ nicht dicht in $C[a, b]$.
- Jede konvexe, kompakte Menge $M \subseteq \mathbb{R}^n$ ist Tschebyscheff-Menge bzgl. $\|\cdot\|_2$.
- Bzgl. $\|\cdot\|_1$ ist z.B. ein gleichschenkliges Dreieck mit achsenparallelen Kanten keine Existenzmenge.



- $\mathbb{Q}$ liegt dicht in $\mathbb{R}$, ist aber keine Existenzmenge.

Lemma 2.10 *Sei M eine kompakte Teilmenge eines normierten Raums V. Dann ist M Existenzmenge.*

Beweis: $\|\cdot\|$ ist stetig, genauer: Sei $f \in V$. Betrachte

$$\begin{aligned} \varphi : V &\to \mathbb{R} \\ v &\mapsto \|f - v\|. \end{aligned}$$

Dann gibt es für jedes $\varepsilon > 0$ ein $\delta := \varepsilon$, sodass

$$|\varphi(v) - \varphi(u)| = |\|f - v\| - \|f - u\|| \leq \|u - v\| \leq \varepsilon$$

für alle u, v mit $\|u - v\| \leq \delta$. Also ist φ eine stetige Funktion auf einer kompakten Menge und nimmt entsprechend ihr Minimum an. QED

Lemma 2.11 *Es gilt:* $|d(f,M) - d(g,M)| \leq \|f-g\|$ *für alle* $f,g \in V$, V *normierter Vektorraum und* $M \subseteq V$, *d.h. der Minimalabstand hängt stetig von dem zu approximierenden Element ab.*

Beweis: Seien $f,g \in V, \varepsilon > 0$. Wähle $u(\varepsilon) \in M$ so, dass $\|g-u(\varepsilon)\| \leq d(g,M)+\varepsilon$. Dann gilt:

$$\begin{aligned} d(f,M) &\leq \|f-u(\varepsilon)\| \leq \|f-g\| + \|g+u(\varepsilon)\| \\ &\leq \|f-g\| + d(g,M) + \varepsilon \end{aligned}$$

also $d(f,M) - d(g,M) \leq \|f-g\| + \varepsilon$. Analog erhält man, wenn man f und g vertauscht:

$$d(g,M) - d(f,M) \leq \|f-g\| + \varepsilon.$$

Zusammen ergibt sich:

$$\begin{aligned} |d(g,M) - d(f,M)| &\leq \|f-g\| + \varepsilon \text{ für alle } \varepsilon > 0 \\ \text{also } |d(g,M) - d(f,M)| &\leq \|f-g\|. \end{aligned}$$

QED

Wir betrachten nun Mengen $M \subseteq V$ mit weiteren Eigenschaften:

1. M konvexe Teilmenge von V.
2. M Unterraum von V.

Wir erinnern uns:

$$M \text{ konvex} \Leftrightarrow \forall x,y \in M, \forall \lambda \in (0,1) : \lambda x + (1-\lambda)y \in M.$$

Es gilt:

- Jeder Unterraum ist konvex.
- $\varnothing$ ist konvex.
- M_1, M_2 konvex $\Rightarrow M_1 \cap M_2$ konvex.

Im Folgenden bezeichnet $\mathcal{U}^*_{(f)}$ die Menge der besten Approximationen an $f \in V$ aus M.

Satz 2.12 *Sei* V *normierter Vektorraum und* $M \subseteq V$ *konvex. Zu* $f \in V$ *existiere eine beste Approximation* $u^* \in M$, *d.h.* $\mathcal{U}^*_{(f)} \neq \varnothing$. *Dann gilt: Entweder* $\mathcal{U}^*_{(f)} = \{u^*\}$ *oder* $|\mathcal{U}^*_{(f)}| = \infty$ *und* $\mathcal{U}^*_{(f)}$ *ist konvex.*

Beweis: Seien u_1, u_2 beides beste Approximationen an f, also

$$d(f, u_1) = \|f - u_1\| = \|f - u_2\| = d(f, u_2).$$

Betrachte $u := tu_1 + (1-t)u_2 \in M$ für beliebiges $t \in [0, 1]$. Dann gilt

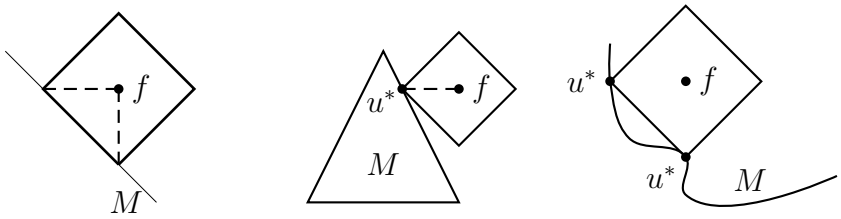
$$\begin{aligned} \|f - u\| &= \|(f - u_1)t + (f - u_2)(1 - t)\| \\ &\leq |t| \underbrace{\|f - u_1\|}_{=d(f,u_1)} + |1 - t| \underbrace{\|f - u_2\|}_{=d(f,u_2)} \\ &= d(f, u_1), \end{aligned}$$

also ist u auch beste Approximation an f und die Menge $\mathcal{U}^*_{(f)}$ ist konvex. Weil

$$|\{tu_1 + (1-t)u_2 : t \in [0, 1]\}| = \infty,$$

hat die Menge aller besten Approximationen – wie jede konvexe Menge mit mehr als einem Element – unendlich viele Elemente. QED

Beispiele: $V = \mathbb{R}^2$, $\|\cdot\| = \|\cdot\|_1$.



konvexe Menge beschreibt beste Approximation

beste Approximation eindeutig

Genau zwei beste Approximationen

Speziell für lineare Unterräume M gilt die folgende Aussage:

Satz 2.13 *Sei U ein endlich-dimensionaler Unterraum eines normierten Vektorraums V. Dann ist U eine Existenzmenge. Weiterhin ist für alle $f \in V$ die Menge der besten Approximationen $\mathcal{U}^*_{(f)}$ konvex und es gilt entweder $|\mathcal{U}^*_{(f)}| = 1$ oder $|\mathcal{U}^*_{(f)}| = \infty$.*

Beweis: Weil U ein Unterraum ist, gilt $0 \in U$. Sei

$$U_0 = \{u \in U : \|f - u\| \leq \|f - 0\|\}$$

die Menge aller Elemente aus U, die f mindestens genauso gut approximieren wie 0. Es ist also $\mathcal{U}^*_{(f)} \subset U_0$.

U_0 ist abgeschlossen (weil $\|\cdot\|$ stetig ist) und beschränkt (weil $\|u\| \leq \|u-f\| + \|f\| \leq 2\|f\|$ für alle $u \in U_0$). Zusammen folgt, dass U_0 eine kompakte Menge ist. Nach Lemma 2.10 existiert also eine beste Approximation an f aus U_0. Diese ist beste Approximation an f aus U.
Weil jeder Unterraum insbesondere konvex ist, folgt der zweite Teil aus Satz 2.12.
QED

Bemerkung: Die Voraussetzung "endlich-dimensional" ist nötig! Betrachte dazu $V = C[a,b]$ mit $\|\cdot\| = \|\cdot\|_\infty$ und $U = \Pi_\infty[a,b]$. Sei $f \in C[a,b] \setminus \Pi_\infty[a,b]$. Dann gilt zwar $d(f,U) = 0$, aber weil f kein Polynom ist, wird dieses Infimum nie angenommen.

Bemerkung: Man kann zeigen, dass die beste Approximation in Euklidischen Räumen – sofern sie existiert – immer eindeutig ist.

2.3 Tschebyscheff-Approximation in $C[a,b]$

Wir untersuchen nun wieder die Approximation einer stetigen Funktion $f \in C[a,b]$ durch $u^* \in U \subseteq C[a,b]$. Dabei betrachten wir als Abstand

$$\|u-f\|_\infty = \max_{x\in[a,b]} |u(x) - f(x)|.$$

Aus Satz 2.13 wissen wir, dass jeder endlich-dimensionale Unterraum $U \subseteq C[a,b]$ eine Existenzmenge ist. Um die Eindeutigkeit zu behandeln, betrachten wir unisolvente Räume (siehe auch Numerik I).

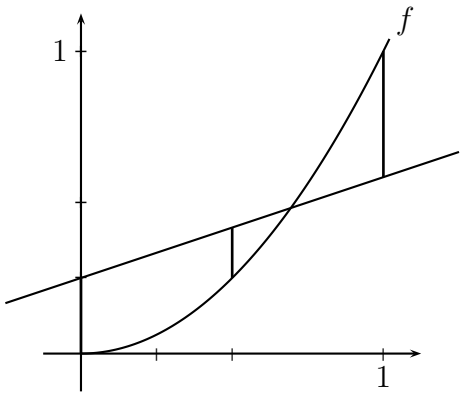
Definition 2.14 *Sei $U \subseteq C[a,b]$ ein Unterraum von $C[a,b]$ mit $\dim(U) = n$. Dann heißt U **Haar'scher Raum** der Dimension n, falls jedes $u \in U \setminus \{0\}$ höchstens $n-1$ Nullstellen in $[a,b]$ hat.*

Bemerkung: Ein Haar'scher Raum ist unisolvent bezüglich jeder Menge $X \subset [a,b]$ mit $|X| \geq n$.

Beispiel: Es ist $\Pi_n \subseteq C[a,b]$ ein Haar'scher Raum der Dimension $n+1$, denn jedes nicht-verschwindende Polynom vom Grad maximal n hat höchstens n Nullstellen in $[a,b]$.

Die Approximation einer Funktion bezüglich der $\|\cdot\|_\infty$-Norm soll zunächst an einem ausführlichen Beispiel demonstriert werden.

Beispiel: Betrachte $I = [0,1]$ und $f(x) = x^2 \in C[0,1]$.
Wir interessieren uns für die beste lineare Approximation, suchen also eine Funktion $u^*(x) = \alpha + \beta x$ mit minimalem Abstand $\|f-u^*\|_\infty$ zu f.



keine optimale Lösung

Für den Fehler gilt:

$$\begin{aligned}\|f - u^*\|_\infty &= \max_{x\in[0,1]} |f(x) - u^*(x)| \\ &= \max_{x\in[0,1]} |x^2 - \beta x - \alpha|.\end{aligned}$$

Es gilt

- $x^2 - \beta x - \alpha$ wird als konvexe Funktion am Rand maximal, also für $x = 0$ oder für $x = 1$ mit Maximalwerten $|\alpha|$ oder $|\beta + \alpha - 1|$.
- $-x^2 + \beta x + \alpha$ wird als konkave und differenzierbare Funktion am Rand maximal, oder falls ihr Gradient gleich Null ist, also falls

  $$-2x + \beta = 0 \Leftrightarrow x = \tfrac{1}{2}\beta.$$

  Der Maximalwert beträgt dann $|\frac{1}{4}\beta^2 - \frac{1}{2}\beta^2 - \alpha| = |\alpha + \frac{1}{4}\beta^2|$.

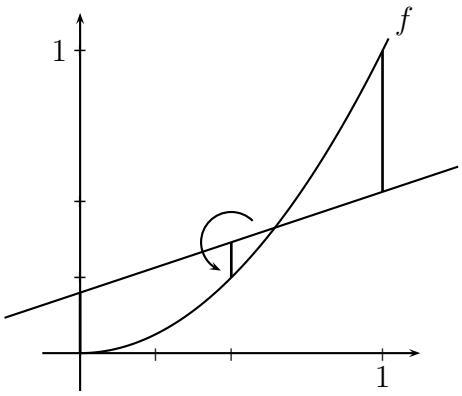
Wir erhalten also

$$\|f - u^*\|_\infty = \max\{|\alpha|, |\beta + \alpha - 1|, |\alpha + \tfrac{1}{4}\beta^2|\}.$$

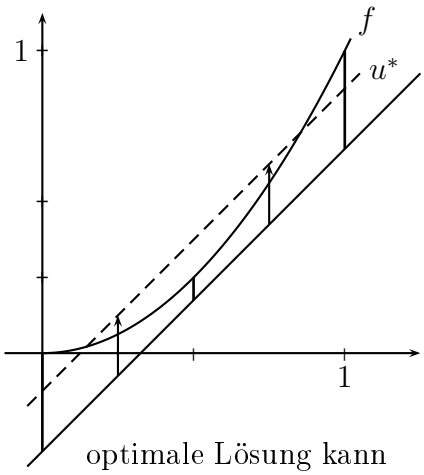
Für welche α, β wird dieser Ausdruck minimal?

- Dazu müssen alle drei Terme den gleichen Wert annehmen. Man kann sich das durch eine Fallunterscheidung leicht klarmachen: Sind die Werte nicht gleich, gilt also zum Beispiel $|\alpha| > |\beta + \alpha + 1|$ und $|\alpha| > |\alpha + \frac{1}{4}\beta^2|$, so kann die Lösung verbessert werden, indem man α (auf Kosten von β) etwas reduziert. (Analog in den anderen Fällen.)
- Außerdem müssen die Vorzeichen der drei Terme alternieren; sonst könnte man die Gerade ebenfalls verbessern (Skizze).

Durch eine Skizze lassen sich beide Aussagen veranschaulichen: Ist eine der drei Strecken länger als die beiden anderen, so kann man sie durch Verschieben und Drehen von u auf Kosten der anderen verkürzen und so u verbessern.



keine optimale Lösung



optimale Lösung kann durch Verschieben erreicht werden

In unserem Beispiel erhält man für den Fall

$$f(0) > u^*(0), \quad f(\tfrac{1}{2}\beta) < u^*(\tfrac{1}{2}\beta) \quad \text{und} \quad f(1) > u^*(1)$$

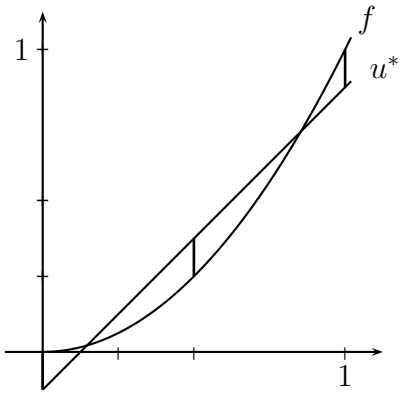
die Gleichungen

$$-\alpha = \alpha + \tfrac{1}{4}\beta^2 = 1 - \alpha - \beta, \text{ d.h. } 2\alpha + \tfrac{1}{4}\beta^2 = 0 \quad \text{und} \quad 1 - \beta = 0.$$

woraus folgt, dass $\beta = 1$, $\alpha = -\frac{1}{8}$ und $\|f - u\|_\infty = \frac{1}{8}$. Da das Vorzeichen alternieren muss, gibt es nur noch einen weiteren Fall: $f(0) < u^*(0), f(\frac{1}{2}\beta) > u^*(\frac{1}{2}\beta)$ und $f(1) < u^*(1)$. Dieser liefert keinen besseren Wert für $\|f - u^*\|_\infty$, also ist

$$u^*(x) = x - \tfrac{1}{8}$$

die beste Approximation.



beste Approximation

Das Beispiel motiviert die folgende Definition:

Definition 2.15 *Sei U ein Haar'scher Raum der Dimension n über $[a,b]$. Eine Menge X von $n+1$ Punkten $a \le x_1 < x_2 < \cdots < x_{n+1} \le b$ heißt* ***Alternante*** *für $f \in C[a,b]$ und $u \in U$, falls*

$$\operatorname{sign}\Big(f(x_j) - u(x_j)\Big) = \sigma(-1)^j, \qquad 1 \le j \le n+1$$

gilt mit einer Konstanten $\sigma \in \{-1,1\}$.

Eine Menge X ist also Alternante für f und u, wenn $f-u$ in den x_j alternierend das Vorzeichen wechselt.

Satz 2.16 *Sei U ein n-dimensionaler Haar'scher Raum über $[a,b]$. Gibt es zu $f \in C[a,b]$ und $u^* \in U$ eine Alternante X mit*

$$|f(x_j) - u^*(x_j)| = \|f - u^*\|_\infty, \qquad 1 \le j \le n+1,$$

so ist u^* eine beste Approximation an f aus U.

Beweis: Sei $X = \{x_1, \ldots, x_n, x_{n+1}\}$ Alternante mit

$$\operatorname{sign}(f(x_j) - u^*(x_j)) = \sigma(-1)^j \quad \text{für alle} \quad 1 \le j \le n+1$$

für ein festes $\sigma \in \{-1,1\}$. Sei $u \in U$. Wir wollen zeigen, dass

$$\|f - u^*\|_\infty \le \|f - u\|_\infty.$$

Dazu rechnen wir

$$\begin{aligned} \|f-u^*\|_\infty &= |f(x_j) - u^*(x_j)| \quad \text{für } j = 1,\ldots,n+1 \\ &= (f(x_j) - u^*(x_j))\sigma(-1)^j \quad \text{für } j = 1,\ldots,n+1 \\ &= (f(x_j) - u(x_j))\sigma(-1)^j + (u(x_j) - u^*(x_j))\sigma(-1)^j \\ &\qquad\qquad \text{für } j = 1,\ldots,n+1 \end{aligned} \tag{2.5}$$

Um diesen Ausdruck weiter abzuschätzen, zeigen wir zunächst, dass es ein $j_0 \in \{1,\ldots,n-1\}$ so gibt, dass

$$(u(x_{j_0}) - u^*(x_{j_0}))(-1)^{j_0}\sigma \le 0. \tag{2.6}$$

Dazu nehmen wir an, dass (2.6) für kein j_0 gültig ist. Das heißt,

$$(u(x_j) - u^*(x_j))(-1)^j\sigma > 0 \quad \text{für alle} \quad j \in \{1,2,\ldots,n,n+1\},$$

also würde $u-u^*$ in jedem der n Intervalle (x_j, x_{j+1}), $j = 1, \ldots, n$ das Vorzeichen wechseln. Nach dem Zwischenwertsatz hätte die stetige Funktion $u - u^*$ also mindestens n Nullstellen. Aber $u - u^* \in U$ und U haben wir als Haar'schen Raum der Dimension n vorausgesetzt. Somit gilt:

$$u - u^* \equiv 0 \text{ oder } u - u^* \text{ hat höchstens } n-1 \text{ Nullstellen.}$$

Daraus ergibt sich also $u \equiv u^*$; das aber ist ein Widerspruch zu $u(x_j) \neq u^*(x_j)$ an den Punkten $x_1, \ldots, x_{n+1}$.

Wir verwenden die eben gezeigte Aussage, um $\|f-u^*\|$ in (2.5) weiter abzuschätzen, indem wir für j den Index j_0 wählen, der (2.6) erfüllt. Wir erhalten:

$$\begin{aligned}\|f-u^*\|_\infty &= (f(x_{j_0}) - u(x_{j_0}))\sigma(-1)^{j_0} + \underbrace{(u(x_{j_0}) - u^*(x_{j_0}))\sigma(-1)^{j_0}}_{\leq 0 \text{ nach (2.6)}} \\ &\leq |f(x_{j_0}) - u(x_{j_0})| \leq \|f-u\|_\infty.\end{aligned}$$

QED

Um eine beste Approximation zu finden, macht es also Sinn, f zunächst auf einer diskreten Menge $X = (x_1, \ldots, x_{n+1})$ zu approximieren. Wir führen die folgenden Bezeichnungen ein.

Notation: *Einen Vektor $X = (x_1, \ldots, x_{n+1})^T \in \mathbb{R}^{n+1}$ mit $a \leq x_1 < x_2 < \cdots < x_n < x_{n+1} \leq b$ nennen wir **Referenz**. Wir definieren*

$$\|(u-f)|_X\|_\infty := \max_{i=1,\ldots,n+1} |u(x_i) - f(x_i)|.$$

Gilt für $u^* \in U$ dass

$$\|(u^*-f)|_X\|_\infty \leq \|(u-f)|_X\|_\infty$$

*für alle $u \in U$, so nennt man u^* **beste Approximation** an f aus U auf der Referenz X, oder **diskrete Approximation** auf X oder **Tschebyscheff-Approximation** an f aus U auf X.*

Korollar 2.17 *Sei U ein Haar'scher Raum der Dimension n über $[a,b]$. Gibt es zu $f \in C[a,b]$ und $u^* \in U$ eine Alternante X mit*

$$|f(x_j) - u^*(x_j)| = const \quad \text{für alle } 1 \leq j \leq n+1$$

(das heißt dann, dass $|f(x_j) - u^*(x_j)| = \|(f-u^*)|_X\|_\infty$ für alle $1 \leq j \leq n+1$), so ist u^* Tschebyscheff-Approximante auf X an f aus U.

Beweis: Der Beweis verläuft genau analog zu dem Beweis von Satz 2.16, nur betrachtet man statt $\|f - u^*\|_\infty$ den Ausdruck $\|(f - u^*)|_X\|_\infty$ beziehungsweise statt $\|f - u\|_\infty$ den Ausdruck $\|(f - u)|_X\|_\infty$. QED

Das ergibt folgende Idee, um eine beste Approximation iterativ anzunähern.

1. Starte mit Referenz X und bestimme $u^* \in U$ so, dass

   - X ist Alternante für f und u^*
   - $|f(x_i) - u^*(x_i)| = \text{const}$.

   Dann ist u^* beste diskrete Approximation an f aus U auf X (nach Korollar 2.17).

2. Gilt zusätzlich, dass $\|f - u^*\|_\infty = \text{const}(= \|(f - u^*)|_X\|_\infty)$, so ist u^* beste Approximante an f aus U auf ganz $[a, b]$ (nach Satz 2.16).

3. Sonst verändere die Referenz X und gehe zu 1.

Wir werden im Folgenden besprechen,

- wie man in Schritt 1 die diskrete Tschebyscheff-Approximante berechnen kann, und
- wie man in Schritt 3 die Referenz X geeignet modifiziert, so dass das Verfahren konvergiert.

Wir beginnen mit der Berechnung der diskreten Tschebyscheff-Approximante.

Sei $u_1, \ldots, u_n$ eine Basis von U. Sei X eine Referenz und bezeichne

$$\rho_X = d_X(f, U) = \inf_{u \in U} \|(f - u)|_X\|_\infty$$

den Minimalabstand von f und U bzgl. der Referenz $X = (x_1, \ldots, x_{n+1})^T$. Wir suchen

$$u^* = \sum_{j=1}^{n} \alpha_j u_j,$$

genauer also die Koeffizienten $\alpha_1, \ldots, \alpha_n$. Sei σ_X das Vorzeichen von $f(x_1) - u^*(x_1)$. Dann müssen die folgenden $n + 1$ Bedingungen erfüllt sein:

$$f(x_i) - u^*(x_i) = \rho_X \sigma_X (-1)^{i-1}, \qquad \forall i = 1, \ldots, n + 1.$$

Das schreiben wir um zu:

$$f(x_i) = \sum_{j=1}^{n} \alpha_j u_j(x_i) + \underbrace{(-1)^{i-1}}_{\substack{:= u_{n+1} \\ \text{bekannt}}} \underbrace{\sigma_X \rho_X}_{\substack{=: \alpha_{n+1} \\ \text{Variable}}} \qquad 1 \le i \le n + 1.$$

Als Gleichungssystem erhält man $n+1$ Gleichungen in $n+1$ Variablen, wobei wir zur Vereinfachung der Schreibweise

$$u_{n+1}(x_i) := (-1)^{i-1}$$

setzen. In Matrixform ergibt sich:

$$\underbrace{\begin{pmatrix} u_1(x_1) & \cdots & u_n(x_1) & u_{n+1}(x_1) \\ u_1(x_2) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ u_1(x_{n+1}) & \cdots & u_n(x_{n+1}) & u_{n+1}(x_{n+1}) \end{pmatrix}}_{=:A} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n+1} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{n+1}) \end{pmatrix}. \tag{2.7}$$

Ist dieses Gleichungssystem lösbar? Wir benutzen den Laplaceschen Entwicklungssatz für die letzte Spalte. Sei dazu

$$D_i = \begin{pmatrix} u_1(x_1) & \cdots & u_n(x_1) \\ \vdots & & \vdots \\ u_1(x_{i-1}) & \cdots & u_n(x_{i-1}) \\ u_1(x_{i+1}) & \cdots & u_n(x_{i+1}) \\ \vdots & & \vdots \\ u_1(x_{n+1}) & \cdots & u_n(x_{n+1}) \end{pmatrix} \in \mathbb{R}^{n,n}.$$

Dann gilt:

$$\begin{aligned} \det(A) &= \sum_{i=1}^{n+1} (-1)^i \underbrace{u_{n+1}(x_i)}_{=(-1)^{i-1}} \det(D_i) \\ &= \sum_{i=1}^{n+1} \det(D_i). \end{aligned}$$

D_i ist die zu den Punkten $x_1, \ldots, x_{i-1}, x_{x+1}, \ldots, x_n$ gehörende Interpolationsmatrix, daher ist $\det D_i \neq 0$ für alle i. Man kann sogar zeigen, dass alle $\det D_i$ das gleiche Vorzeichen haben, also gilt $\det A \neq 0$. Außerdem gilt der folgende Satz:

Satz 2.18 *Sei U ein Haar'scher Raum der Dimension n über $[a,b]$ und sei $X = a \leq x_1 < x_2 < \cdots \leq x_{n+1} = b$ eine Referenz. Dann gibt es zu jedem $f \in C[a,b]$ genau eine Lösung der Tschebyscheff-Approximation. Man kann sie durch Lösen des linearen Gleichungssystems* (2.7) *berechnen.*

Beweisskizze:

- Weil (2.7) eindeutig lösbar ist, folgt die Existenz der Tschebyscheff-Approximation.

- Um die Eindeutigkeit nachzuweisen, muss man zeigen, dass jede Tschebyscheff-Approximante auch Lösung von (2.7) ist. Weil (2.7) eindeutig lösbar ist, folgt daraus die Behauptung.

Übungsaufgabe: Sei U ein Haar'scher Raum der Dimension n über $[a,b]$ und sei $X \subseteq [a,b]$ mit $a \leq x_1 \leq \cdots \leq x_n \leq b$, also $|X| = n$. Bestimmen Sie $d_X(f,U)$!

Bevor wir das Remes-Verfahren formulieren, machen wir uns die Idee, die beste Approximation durch eine beste diskrete Approximation anzunähern, an folgendem Lemma klar:

Lemma 2.19 *Sei X eine Referenz. Dann gilt*

$$d_X(f,U) \leq d(f,U).$$

Beweis: Sei $u \in U$. Dann gilt

$$\|(f-u)|_X\|_\infty = \max_{i=1,\ldots,n+1} |f(x_i) - u(x_i)| \leq \|f-u\|_\infty$$

und folglich

$$\inf_{u\in U} \|(f-u)|_X\|_\infty \leq \inf_{u\in U} \|f-u\|_\infty,$$

also $d_X(f,U) \leq d(f,U)$. QED

Wenn man also $d(f,U)$ durch $d_X(f,U)$ annähern möchte, ist die Referenz X dafür besser geeignet als die Referenz X', falls $d_X(f,U) \geq d_{X'}(f,U)$, also falls die Fehlerfunktion $f - u_X^*$ für die Tschebyscheff-Approximante u_X^* bezüglich der Referenz X möglichst *groß* ist! Diese Beobachtung wird im Remes-Verfahren wie folgt ausgenutzt:

Algorithmus 1: Remes-Verfahren

Input: $f \in C[a,b]$, $U \subseteq C[a,b]$ Haar'scher Raum der Dimension n.

Schritt 1: Wähle Startreferenz $X^{(0)} = \{x_1^{(0)}, \ldots, x_{n+1}^{(0)}\}$, $j := 0$,

Schritt 2: Bestimme die Tschebyscheff-Approximation u_j^* auf $X^{(j)}$ an f. Sei $p_j = d_{X^{(j)}}(f,U) = \|(f-u_j^*)|_{X^{(j)}}\|_\infty$.

Schritt 3: Falls $d_{X^{(j)}}(f,U) = \|f-u_j^*\|_\infty$: STOP. Lösung sei v_j^*.

Schritt 4: Bestimme die neue Referenz $X^{(j+1)}$, die den folgenden drei Bedingungen genügt:

a) $\operatorname{sign}(f-u_j^*)(x_k^{(j+1)}) = -\operatorname{sign}(f-u_j^*)(x_{k+1}^{(j+1)})$ für alle $1 \le k \le n$.

b) $|(f-u_j^*)(x_k^{(j+1)})| \ge d_{X^{(j)}}(f,U)$ für alle $1 \le k \le n+1$.

c) $\|(f-u_j^*)|_{X^{(j+1)}}\|_\infty = \|f-u_j^*\|_\infty$.

Setze $j := j+1$ und gehe zu 2.

Zunächst analysieren wir Schritt 4:

- Bedingung a) bedeutet, dass die alte Fehlerfunktion $f-u_j^*$ auch auf der neuen Referenz $X^{(j+1)}$ alternieren soll.
- Bedingung b) besagt, dass die alte Fehlerfunktion $f-u_j^*$, angewendet auf die Punkte der neuen Referenz, nicht kleiner sein darf als an den Punkten der alten Referenz.
- Zusammen mit Bedingung c) heißt das sogar, dass die alte Fehlerfunktion, angewendet auf die neuen Punkte, maximal werden soll, also den Gesamtfehler $\|f-u_j^*\|_\infty$ an einem der neuen Punkte annehmen muss.

Man versucht also, gemäß der Aussage von Lemma (2.19), die neue Referenz so zu wählen, dass ihr Fehler möglichst groß wird. Bevor wir uns mit der Konvergenz des Remes-Verfahrens beschäftigen, zeigen wir, dass es in Schritt 4 immer eine passende neue Referenz gibt.

Lemma 2.20 *Es gibt eine Referenz $X^{(j+1)}$, die den Bedingungen von Schritt 4 genügt, falls $d_{X^{(j)}} < \|f-u_j^*\|_\infty$.*

Beweis: Die Referenz $X^{(j)}$ genügt den Randbedingungen a) und b). Wir werden daher nur einen Punkt aus $X^{(j)}$ gegen einen neuen austauschen. Dazu bestimmen

wir $\tilde{x}$ mit

$$\|f - u_j^*\| = f(\tilde{x}) - u_j^*(\tilde{x})$$

als einen Punkt in $[a, b]$, an dem der Fehler maximal wird. Wegen

$$d_{X^{(j)}} < \|f - u_j^*\|_\infty$$

ist $\tilde{x} \neq x_k$ für $k = 1, \ldots, n+1$. Wir unterscheiden drei Fälle:

1. Existiert ein k, so dass $x_k^{(j)} < \tilde{x} < x_{k+1}^{(j)}$, so setze $x_k^{(j+1)} := \tilde{x}$ falls $\text{sign}(f - u_j^*)(x_k^{(j+1)}) = \text{sign}(f - u_j^*)(\tilde{x})$, sonst $x_{k+1}^{(j)} := \tilde{x}$. Man ersetzt also entweder $x_k^{(j)}$ oder $x_{k+1}^{(j)}$ durch $\tilde{x}$.

2. Falls $\tilde{x} < x_1^{(j)}$, setze $x_1^{(j+1)} := \tilde{x}$. Falls $\text{sign}(f - u_j^*)(x_1) \neq \text{sign}(f - u_j^*)(\tilde{x})$, setze außerdem $x_{k+1}^{(j+1)} := x_k^{(j)}$ für $k = 1, \ldots, n$. (Im ersten Fall wird also $x_1^{(j)}$ aus der Referenz entfernt, im zweiten Fall $x_{n+1}^{(j)}$.)

3. Falls $\tilde{x} > x_{n+1}^{(j)}$ analog zu Fall 2.

QED

In der Praxis ersetzt man meistens mehr als einen Punkt aus $X^{(j)}$.

Jetzt können wir folgenden Satz über das Remes-Verfahren formulieren:

Satz 2.21 *Sei $U \subseteq C[a, b]$ ein Haar-Raum der Dimension n und sei $f \in C[a, b] \setminus U$. Dann existiert genau eine beste Approximation $u^* \in U$ auf f aus U auf $[a, b]$. Ferner bricht das Remes-Verfahren entweder nach endlich vielen Schritten mit u^* ab, oder es liefert Folgen $\{X^{(j)}\}$, $\{u_j^*\}$ und $\{p_j\}$ mit folgenden Eigenschaften:*

- *$\{p_j\}$ konvergiert mindestens linear gegen $\|f - u^*\|$. Genauer existiert eine Konstante $q \in (0, 1)$ mit*

$$\|f - u^*\| - p_{j+1} \leq q(\|f - u^*\| - p_j), \qquad j \in \mathbb{N}_0$$

- *$\{u_j^*\}$ konvergiert gleichmäßig auf I gegen die Lösung u^*.*

Beweis: Bricht das Verfahren nach endlich vielen Schritten ab, so gibt es ein $j \in \mathbb{N}_0$ mit

$$\rho_j = \|(f - u_j^*)|_{X^{(j)}}\|_\infty = \|f - u_j^*\|_\infty,$$

also ist u_j^* beste Approximation an f auf I nach Satz (2.16).
Nehmen wir also an, das Verfahren endet nicht. Dann kann man zeigen, dass die Folge (p_j) aufgrund der Bedingungen a), b) und c) streng monoton wachsend ist. Wegen Lemma 2.19 ist

$$\rho_j \leq d(f, U)$$

also ist die Folge nach oben beschränkt. Daraus folgt Konvergenz.
Der Nachweis der mindestens linearen Konvergenz wird hier nicht beschrieben.
Die Eindeutigkeit folgt folgendermaßen: Sei

$$\{X^{(j)}\} \subseteq \text{Menge aller Referenzen,}$$

dann besitzt diese eine konvergente Teilfolge, die gegen eine Referenz X^* konvergiert. Sei u^* die zugehörige Tschebyscheff-Approximante, die Lösung von $\inf_{u \in U} \|f - u\|_\infty$ ist. Sei $\tilde{u}$ eine weitere Lösung des Approximationsproblems, dann ist $\tilde{u}$ auch eine Tschebyscheff-Approximation an f aus V auf $[a.b]$. Nach Satz 2.18 ist diese eindeutig, also $u^* = \tilde{u}$. QED

Bemerkung: Die beste Approximation u^* ist eindeutig und $u_j^* \to u^*$. Aber die Folge der Referenzen $X^{(j)}$ hat nur eine konvergente Teilfolge, weil es zu u^* mehrere Alternanten geben kann, als Häufungspunkte der Referenzen auftreten können.

Als Folge des letzten Satzes erhalten wir die "Rückrichtung" zu Satz 2.16:

Satz 2.22 (Alternantensatz) *Sei U ein n-dimensionaler Haar'scher Raum über $[a, b]$. Ein Element $u^* \in U$ ist genau dann beste Approximation an $f \in C[a, b]$, wenn es eine Alternate X für f und u^* mit*

$$|f(x_j) - u^*(x_j)| = \|f - u^*\|_\infty, \qquad 1 \le j \le n + 1$$

gibt. Die beste Approximation u^* ist eindeutig bestimmt, die Alternante aber nicht.

Durch das Remes-Verfahren haben wir konstruktiv gezeigt, dass jeder Haar'sche Raum eine Tschebyscheff-Menge ist. Der nächste Satz sagt, dass Haar'sche Räume die einzigen Tschebyscheff-Mengen sind.

Satz 2.23 *Sei U ein n-dimensionaler Unterraum von $C[a, b]$. Dann gilt:*

$$U \text{ ist Haar'scher Raum} \Leftrightarrow U \text{ ist Tschebyscheff-Menge.}$$

2.4 Zusammenfassung

Ziel:
- *Nähere ein Objekt f (aus einem VR V) durch ein "einfacheres Objekt" an*
- *"einfacher": Aus einer Menge $M \subseteq V$*
- *"annähern": $\|f - u^*\| \leq \|f - u\|$, $\forall u \in M \Rightarrow u^*$ ist beste Annäherung (beste Approximation)*

Beispiele

- $V = C[a,b]$, $\|\cdot\|_\infty$, M z.B. Π_n: Approximation von Funktionen
- $V = \mathbb{R}^n$, $\|\cdot\|$ bel. Norm, $M \subseteq \mathbb{R}^n$: Projektion
- $V = \mathbb{R}^n$, $M = \{Ax : x \in \mathbb{R}^n\}$, $\|\cdot\|_2$: Ausgleichsrechnung ($A \in \mathbb{R}^{m,n}$)

Existenz- und Tschebyscheff-Mengen

- M Existenzmenge, falls eine beste Approximation $u^* \in M$ existiert
- M Tschebyscheff-Menge, falls genau eine beste Approximation existiert
- M kompakt $\Rightarrow$ M Existenzmenge
- M konvex $\Rightarrow$ $\exists$ keine, genau eine oder unendl. viele beste Approx. und bilden eine konvexe Menge.

Speziell für $C[a,b]$

Satz von Weierstrass

- Satz von Weierstrass:

$$\forall \varepsilon > 0, \forall f \in C[a,b] : \exists p \in \Pi_\infty : \|f - p\|_\infty < \varepsilon$$

- Beweisskizze:
  - $\{K_n\}$ Korovkin-Folge, falls K_n linear und monoton $\forall n$ und $\lim_{n\to\infty} \|K_n f - f\|_\infty = 0$ für $f \in \{\mathbf{1}, x, x^2\}$
  - Ist $\{K_n\}$ Korovkin-Folge, dann gilt $\lim_{n\to\infty} \|K_n f - f\|_\infty = 0$, $\forall f \in C[a,b]$
  - Ziel: Finde Korovkin-Folge $K_n : C[a,b] \to \Pi_\infty[a,b]$
  - Bernstein-Operatoren! $\Rightarrow$ Beweis

- Folge: Satz von Weierstraß gilt auch für $\|\cdot\|_{L_p}$-Normen;

$$\|f\|_{L_p} = \sqrt[p]{\int_a^b |f(x)|^p dx}$$

Tschebyscheff-Approx. in Haar'schen Räumen

- ..., d.h.

$$V = C[a,b], \|f\|_\infty = \max_{x\in[a,b]} |f(x)|, M \text{ ist Haar'scher Raum}$$

- U ist Haar'scher Raum der Dim. n, falls jedes $u \in U \setminus \{0\}$ höchstens $n-1$ Nullstellen hat.
- $X = \{x_1 < x_2 < \cdots < x_{n+1}\} \subseteq [a,b]$ heißt Alternante für f und u, falls $\text{sign}(f(x_j) - u(x_j)) = \sigma(-1)^j$, $\sigma \in \{-1, 1\}$
- Kriterium: $U \subseteq V$ Haar'scher Raum der Dim. n, $f \in [a,b]$, $u^* \in U$: Gibt es eine Alternante X mit $|f(x_j) - u^*(x_j)| = \|f - u^*\|_\infty$, $j = 1, \ldots, n-1$, so ist u^* beste Approx. an f (aus V)

Diskrete Approximation

- Gegeben: $X = \{x_1, \ldots, x_{n+1}\}$. Finde $u^* : \|(f-u^*)|_X\|_\infty \leq \|(f-u)|_X\|_\infty$, $\forall u \in U$.
- Kriterium: $u^* \in U$ ist beste diskrete Approx., falls

$$|f(x_j) - u^*(x_j)| \leq \|(f-u^*)|_X\|_\infty$$

  für eine Alternate X für f und u^*.
- Lösen durch ein Gleichungssystem ($n+1$ Var., $n+1$ Bed.)
- Lösung des diskreten Approx.-Prob. ist immer existent und eindeutig

Remes-Verfahren

- Starte mit einer Referenz $X^{(0)}$, $j = 0$
- diskrete Approx. $u^{(j)}$ an f in $X^{(j)}$
- Falls $u^{(j)}$ beste Approx. an f ist $\rightarrow$ STOP.
- sonst: neue Referenz durch Austauschen (eines) der Punkte in $X^{(j)}$
- Wichtig: $\|(f-u_j^*)|_{X^{(j)}}\|_\infty < \|(f-u_{j+1}^*)|_{X^{(j+1)}}\|_\infty$
- Es gilt: Konvergenz (sublinear) zu eindeutiger Lösung

Alternantensatz

- U Haar'scher Raum der Dim. n, $f \in C[a,b]$, $u^* \in U$ ist beste Approx. an f aus U bzgl. $\|\cdot\|_\infty \Leftrightarrow$ es ex. eine Alternante X mit $\|(f-u^*)|_X\|_\infty = \|f-u^*\|_\infty$
- Beweis: Kriterium + Eindeutigkeit durch Remes-Verfahren
- Bemerkung: Für Unterräume U gilt: Haar'scher Raum $\Leftrightarrow$ Tschebyscheff-Raum (Tschebyscheff-Menge)

Kapitel 3

Numerik gewöhnlicher Differentialgleichungen

3.1 Einführung und Notation

Wir beschäftigen uns in diesem Kapitel hauptsächlich mit gewöhnlichen, expliziten Differentialgleichungen erster Ordnung, gegeben durch

$$x'(t) = f(t, x(t)), \qquad t \in I = [a, b] \tag{3.1}$$

Dabei ist

- $x : I \to \mathbb{R}^d$ eine *gesuchte*, differenzierbare Funktion auf einem Intervall $I = [a, b] \subseteq \mathbb{R}$ (Kurve) und $x'(t) = \begin{pmatrix} x_1'(t) \\ \vdots \\ x_d'(t) \end{pmatrix}$ der Tangentialvektor von x an t.
- $f : D \subseteq (\mathbb{R} \times \mathbb{R}^d) \to \mathbb{R}^d$ eine *gegebene* Funktion.

Wir klären zunächst einige Begriffe.

Notation 3.1

- *Eine Differentialgleichung heißt* ***gewöhnlich****, wenn die unbekannte Funktion x nur von einer reellen Variablen abhängt. Hängt x von mehreren Variablen ab, d.h. gilt*
$$x : B \to \mathbb{R}^d, B \subseteq \mathbb{R}^k,$$
so liegt eine ***partielle*** *Differentialgleichung vor.*
- *Eine Differentialgleichung hat* ***die Ordnung*** *k, falls nur Ableitungen von x bis zur Ordnung k vorkommen. Sie hat die Ordnung 1, falls nur die erste Ableitung von x vorkommt.*

- *Man nennt eine Differentialgleichung* ***explizit****, falls der höchste Ableitungsterm isoliert auftaucht, ansonsten* ***implizit****.*
- *Für $d = 1$ nennt man die Differentialgleichung* ***skalar****, für $d > 1$ spricht man auch von einem* ***System von Differentialgleichungen****.*

Beispiele:

- $F(t, x(t), x'(t)) = 0$, $t \in I = [a, b]$ mit einer gegebenen Funktion $F : \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ ist eine gewöhnliche, implizite Differentialgleichung erster Ordnung.
- $y^{(k)}(t) = g(t, y(t), \ldots, y^{(k-1)}(t))$, $t \in I = [a, b]$ mit einer gesuchten Funktion $y : I \to \mathbb{R}^2$, die k-mal differenzierbar ist, ist eine gewöhnliche, explizite Differentialgleichung der Ordnung k.
- $x'(t) = x(t)$, $t \in [a, b]$ ist eine gewöhnliche, explizite, skalare Differentialgleichung erster Ordnung.

Notation 3.2 *Eine gewöhnliche Differentialgleichung der Form*

$$x'(t) = f(x(t)),$$

bei der die rechte Seite nicht explizit von t abhängt, heißt ***autonom****.*

Wir beschäftigen uns im Wesentlichen mit expliziten, gewöhnlichen Differentialgleichungen.

Beispiel:

$$\begin{aligned} x_1'(t) &= -x_2(t) \\ x_2'(t) &= x_1(t) \end{aligned}$$

ist eine gewöhnliche, explizite und autonome Differentialgleichung (bzw. ein System von Differentialgleichungen) der Form

$$\begin{aligned} x'(t) &= f(t, x(t)) \\ \text{mit } f(t, x(t)) &= \begin{pmatrix} -x_2(t) \\ x_1(t) \end{pmatrix}. \end{aligned}$$

Eine Lösung dieser Differentialgleichung ist

$$x(t) = \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix},$$

denn

$$\begin{array}{rcccccl} x_1'(t) & = & \cos'(t) & = & -\sin(t) & = & -x_2(t), \\ x_2'(t) & = & \sin'(t) & = & \cos(t) & = & x_1(t). \end{array}$$

Es gibt aber noch weitere Lösungen, nämlich

$$\tilde{x}(t) = C \cdot x(t - t_0) = \begin{pmatrix} C \cdot \cos(t - t_0) \\ C \cdot \sin(t - t_0) \end{pmatrix}$$

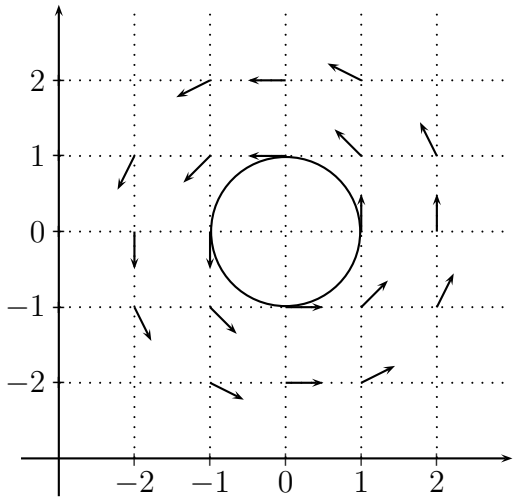
für alle $t_0 \in \mathbb{R}$ und $C \in \mathbb{R}$, denn

$$\tilde{x}'(t) = \begin{pmatrix} -C \cdot \sin(t - t_0) \\ C \cdot \cos(t - t_0) \end{pmatrix} = \begin{pmatrix} -\tilde{x}_2(t) \\ \tilde{x}_1(t) \end{pmatrix}.$$

Veranschaulichung:
Die rechte Seite der Differentialgleichung beschreibt ein Vektorfeld

$$f : \mathbb{R}^2 \to \mathbb{R}^2, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \begin{pmatrix} -x_2 \\ x_1 \end{pmatrix},$$

das man durch einen Vektor $\alpha \binom{-x_2}{x_1}$ in jedem Punkt $\binom{x_1}{x_2}$ skizzieren kann. Die Lösung $x(t) = \binom{\cos(t)}{\sin(t)}$ beschreibt eine Kurve im $\mathbb{R}^2$, zu der das Vektorfeld in jedem Punkt tangential ist.



Lemma 3.3 *Jede gewöhnliche, explizite Differentialgleichung der Ordnung k kann in eine äquivalente Differentialgleichung erster Ordnung transformiert werden.*

Beweis: Sei

$$y^{(k)}(t) = g(t, y(t), \ldots, y^{(k-1)}(t)), \qquad t \in I$$

mit einer gesuchten, k-mal differenzierbaren Funktion $y : I \to \mathbb{R}^d$ gegeben. Definiere $x_j : I \to \mathbb{R}^d$ durch

$$x_j(t) := y^{(j)}(t) \text{ für } j = 0, \ldots, k - 1.$$

Dann gilt:

$$x_j'(t) = y^{(j)'}(t) = y^{(j+1)}(t) = x_{j+1}(t) \text{ für } j = 0, \ldots, k-2$$

und

$$x_{k-1}'(t) = y^{(k-1)'}(t) = y^{(k)}(t) = g(t, y(t), \ldots, y^{(k-1)}(t)) = g(t, x_0(t), \ldots, x_{k-1}(t)),$$

also erhält man das System

$$\mathbb{R}^{kd} \ni x'(t) \left\{ \begin{array}{l} x_0'(t) = x_1(t) \\ x_1'(t) = x_2(t) \\ \quad \vdots \\ x_{k-2}'(t) = x_{k-1}(t) \\ x_{k-1}'(t) = g(t, x_0(t), \ldots, x_{k-1}(t)), \end{array} \right\} f(t, x(t))$$

in dem nur Ableitungen der Ordnung 1 vorkommen. Sei nun eine Lösung dieses Systems gegeben durch eine differenzierbare Funktionen x_j mit $j = 0, \ldots, k-1$. Dann ist

$$y(t) = x_0(t)$$

k-mal differenzierbar, da

$$y^{(j)}(t) = x_j(t) \text{ für } j = 0, \ldots, k-1$$

gilt und alle x_j mindestens einmal differenzierbar sind. Weiterhin gilt:

$$y^{(k)}(t) = y^{(k-1)'}(t) = x_{k-1}'(t) = g(t, x_0(t), \ldots, x_{k-1}(t)) = g(t, y(t), \ldots, y^{(k-1)}(t)).$$

QED

Die Lösung einer Differentialgleichung ist im Allgemeinen nicht eindeutig bestimmt. In dem Beispiel auf Seite 59 hatten wir zum Beispiel zwei Parameter C und t_0 zu wählen. Um die Eindeutigkeit zu erhalten, müssen die freien Variablen durch zusätzliche Bedingungen festgelegt werden.

Notation 3.4 *Ein **Anfangswertproblem (AWP)** einer gewöhnlichen Differentialgleichung erster Ordnung ist gegeben durch*

$$x'(t) = f(t, x(t)), \quad x(t_0) = x_0. \qquad \text{(AWP)}$$

*Ein **Randwertproblem** einer gewöhnlichen Differentialgleichung zweiter Ordnung ist gegeben durch*

$$x''(t) = f(t, x(t), x'(t)), \quad x(a) = r_a, \quad x(b) = r_b.$$

Dabei sind $x_0, r_a, r_b \in \mathbb{R}^d$.

Bemerkung: Die Gleichung $x(t) = x_0$ besteht aus d Bedingungen, sie legt also d Parameter fest (falls sie eindeutig lösbar ist).

Bemerkung: Die numerische Behandlung von Randwertproblemen und Anfangswertproblemen ist unterschiedlich. In dieser Vorlesung befassen wir uns mit Anfangswertproblemen.

Wir kommen noch einmal auf autonome Differentialgleichungen zurück.

Lemma 3.5 *Sei $x : I \to \mathbb{R}^d$ eine Lösung einer autonomen Differentialgleichung $x'(t) = f(x(t))$. Dann ist*

$$y : I \to \mathbb{R}^d, t \mapsto x(t - t_0)$$

auch eine Lösung der Differentialgleichung und zwar für alle $t_0 \in \mathbb{R}$.

Beweis:

$$y'(t) = x'(t - t_0) = f(x(t - t_0)) = f(y(t))$$

QED

Bemerkung: Im Beispiel auf Seite 59 haben wir die Aussage genutzt, um Lösungen zu erzeugen.

Bemerkung: Oft beschreibt der Parameter t die Zeit. Die Aussage des Lemmas lautet dann: Die Lösung einer autonomen Differentialgleichung ist invariant gegenüber Zeittransformationen.

Lemma 3.6 *Jedes Anfangswertproblem der Form $x'(t) = f(t, x(t))$, $x(t_0) = x_0$ lässt sich in ein äquivalentes, autonomes Anfangswertproblem transformieren.*

Beweis: Definiere $s(t) := t$ und $y(t) = \binom{s(t)}{x(t)}$. Betrachte das autonome System

$$y'(t) = \begin{pmatrix} s'(t) \\ x'(t) \end{pmatrix} = \begin{pmatrix} 1 \\ f(y(t)) \end{pmatrix}, \quad y(t_0) = \begin{pmatrix} s(t_0) \\ x(t_0) \end{pmatrix} = \begin{pmatrix} t_0 \\ x_0 \end{pmatrix}. \tag{3.2}$$

- Sei x eine Lösung von $x'(t) = f(t, x(t))$, $x(t_0) = x_0$. Mit $s(t) := t$ erhalten wir

  $$y'(t) = \begin{pmatrix} s'(t) \\ x'(t) \end{pmatrix} = \begin{pmatrix} 1 \\ f(t, x(t)) \end{pmatrix} = \begin{pmatrix} 1 \\ f(s(t), x(t)) \end{pmatrix} = \begin{pmatrix} 1 \\ f(y(t)) \end{pmatrix},$$

  also eine Lösung von (3.2).

- Sei nun $y(t) = \binom{s(t)}{x(t)}$ eine Lösung von (3.2). Dann gilt:

  $$s'(t) = 1, \text{ setze also } s(t) = t.$$

  Damit ist

  $$x'(t) = f(y(t)) = f(s(t), x(t)) = f(t, x(t))$$

  eine Lösung von $x'(t) = f(t, x(t))$.

Den Übergang eines Anfangswertproblemes zu (3.2) nennt man auch **Autonomisierung** des Anfangswertproblemes.

Zwei praktische Anwendungen

Bewegung eines Massepunktes. Die Bewegung eines Massepunktes zur Zeit t am Ort x kann durch die Differentialgleichung 2. Ordnung

$$m \cdot x''(t) = g(t, x)$$

beschrieben werden. Die Funktion g beschreibt dabei die Wirkung äußerer Kräfte, z.B. erhält man bei einer einseitig gespannten Feder $g(t, x) = -kx$, wobei k die Federkonstante bezeichnet. Weiterhin ist meist der Anfangspunkt $x_0 = x(t_0)$ und die Anfangsgeschwindigkeit $x'_0 = x'(t_0)$ vorgegeben.
Das System kann in das folgende äquivalente System 1. Ordnung verwandelt werden:

$$\begin{aligned} x'_1(t) &= x_2(t) \\ x'_2(t) &= -\tfrac{k}{m} x_1(t), \end{aligned}$$

mit Anfangsbedingungen

$$x_1(t_0) = x_0, \quad x_2(t_0) = x'_0.$$

Dieses System von Differentialgleichungen ist erster Ordnung, linear und autonom. Die Lösung ist gegeben durch

$$\begin{aligned} x(t) &= x_1(t) = x_0 \cos\left(\sqrt{\tfrac{k}{m}}\, t\right) + x'_0 \sin\left(\sqrt{\tfrac{k}{m}}\, t\right) \\ x'(t) &= x_2(t) \end{aligned}$$

Volterra-Lottka Zyklus. Betrachte ein ökologisches System mit zwei Arten, bei denen die eine Art der anderen als Nahrung dient. Entsprechend bezeichnen wir sie als "Jäger" und "Beute". Sei

$$\begin{aligned} x_J(t) &= \text{die Größe der Jäger-Population zur Zeit } t \text{ und} \\ x_B(t) &= \text{die Größe der Beute-Population zur Zeit } t. \end{aligned}$$

Die Wachstumsrate der Populationen ergibt sich aus der Differenz der Geburtenrate und der Sterberate. Dabei nehmen wir an, dass für die Beute-Population genügend Nahrung vorhanden sei, so dass sie sich (im ungestörten Fall) exponentiell vermehren würde, die Geburtenrate also konstant ist. Mit geeigneten Parametern $\alpha, \beta > 0$ ergibt sich dann

$$x'_B(t) = \alpha x_B(t) - \beta x_B(t) x_J(t).$$

Die Gleichung kann wie folgt interpretiert werden:

- das ungestörte eigene Wachstum der Beute-Population resultiert aus einem exponentiellen Wachstum $x_B = e^{\alpha x}$ und ist daher durch $x'_B = \alpha x_B$ beschrieben.
- die Anzahl der durch Jagd gestorbenen Beutetiere ist proportional zur Rate, mit der sich Jäger und Beute treffen, auf einem begrenzten Gebiet also proportional zu x_B und proportional zu x_J.

Für die Jäger-Population ergibt sich

$$x'_J(t) = \gamma x_J(t) x_B(t) - \delta x_J(t),$$

ebenfalls mit geeigneten Parametern $\gamma, \delta > 0$. Die Interpretation dieser Gleichung ist wie folgt:

- Die Jäger-Population wächst exponentiell mit Rate γ und proportional zur Beute-Population x_B,
- die natürliche Sterberate ist (bei exponentiellem Wachstum) $x'_J = -\delta x_J$.

Die Lösung dieses Systems von Differentialgleichungen führt zu periodischen Lösungen, die man auch *Volterra-Lottka-Zyklen* nennt. Bilder dazu finden sich z.B. in der Wikipedia.

Wir beenden diesen einführenden Abschnitt mit einer letzten Notation.

Notation 3.7 *Ein System von Differentialgleichungen heißt* ***linear****, falls*

$$x'(t) = f(t,x) := A(t)x + g(t)$$

gilt, wobei $g : I \to \mathbb{R}^d$ *eine stetige Funktion ist und* $A = (a_{ij})_{i,j=1,\ldots,d}$ *eine* $d \times d$*-Matrix mit stetigen Einträgen* $a_{ij} : I \to \mathbb{R}$*.*

Von den beiden oben beschriebenen Anwendungsbeispielen ist das erste linear, das Volterra-Lottka-System aber nichtlinear.

3.2 Existenz und Eindeutigkeit

In diesem Abschnitt wollen wir die Existenz und die Eindeutigkeit von Lösungen für Anfangswertprobleme der Form

$$\begin{aligned} x'(t) &= f(t, x(t)) \\ x(t_0) &= x_0 \end{aligned}$$

untersuchen. Wir zeigen zunächst zwei Beispiele.

- Das erste Beispiel zeigt, dass die Lösung im Allgemeinen nicht eindeutig sein muss. Sei folgendes Anfangswertproblem

$$\begin{aligned} x'(t) &= |x(t)|^\alpha \\ x(0) &= 0 \end{aligned}$$

für einen Parameter $\alpha \in (0,1)$ gegeben. Die Differentialgleichung hat die folgenden beiden Lösungen $\tilde{x}$ und x:

$$\begin{aligned} \tilde{x}(t) &\equiv 0 \\ x(t) &= \begin{cases} ((1-\alpha)t)^{\frac{1}{1-\alpha}} & \text{für } t \geq 0 \\ 0 & \text{für } t < 0 \end{cases} \end{aligned}$$

Für $\tilde{x}$ sieht man das direkt, für die zweite Lösung x rechnet man nach:

  - $x(0) = 0$,
  - $x'(t) = |x(t)|^\alpha$ für $t \geq 0$ und $x'(t) = 0$ für $t < 0$,
  - und $x(0) = x'(0) = 0$, also ist x stetig und differenzierbar.

- Das zweite Beispiel zeigt, dass keine Lösung auf ganz I existieren muss: Betrachten wir

$$\begin{aligned} x'(t) &= (x(t))^2 \\ x(0) &= 1. \end{aligned}$$

Die Lösung $x(t) = -\frac{1}{t-1}$ ist nur für $t \neq 1$ definiert und kann wegen

$$\lim_{t \to 1} x(t) = \infty$$

nicht als stetige Funktion für $t \geq 1$ fortgesetzt werden. Tatsächlich existiert in diesem Fall keine Lösung des Anfangswertproblemes für *alle* $t > 0$. Der Effekt wird auch "blow up" genannt.

Um die Frage nach Existenz und Eindeutigkeit von Lösungen für Anfangswertprobleme zu beantworten, formulieren wir (AWP) zu einer so genannten *Integralgleichung* um.

Lemma 3.8 *Sei $D \subseteq \mathbb{R}^{d+1}$ offen, $f : D \to \mathbb{R}^d$ stetig, $a \leq t_0 \leq b$ und $x : [a,b] \to \mathbb{R}^d$ eine Funktion. Es gelte*

$$\{(t, x(t)) : t \in [a,b]\} \subseteq D.$$

Dann sind die folgenden Aussagen äquivalent:

1. *x ist stetig differenzierbar und löst das (AWP)*

$$\begin{aligned} x'(t) &= f(t, x(t)),\ t \in [a, b] \\ x(t_0) &= x_0 \end{aligned}$$

2. *x ist stetig und erfüllt die Integralgleichung*

$$x(t) = x_0 + \int_{t_0}^{t} f(\tau, x(\tau))d\tau, \quad t \in [a, b]. \tag{3.3}$$

Beweis: $1 \Longrightarrow 2$: Sei $x'(t) = f(t, x(t))$, $x(t_0) = x_0$ eine Lösung des Anfangswertproblemes. Nach dem Hauptsatz der Differential- und Integralrechnung gilt dann

$$\begin{aligned} x(t) &= x(t_0) + \int_{t_0}^{t} x'(\tau)d\tau \\ &= x_0 + \int_{t_0}^{t} f(\tau, x(\tau))d\tau. \end{aligned}$$

$2 \Longrightarrow 1$: Sei nun $x(t) = x_0 + \int_{t_0}^{t} f(\tau, x(\tau))d\tau$. Da f und x beide stetig sind, ist $\int_{t_0}^{t} f(\tau, x(\tau))d\tau$ stetig nach t differenzierbar. Also ist x stetig differenzierbar und die Ableitung von x ist gegeben durch

$$x'(t) = \frac{d}{dt} \int_{t_0}^{t} f(\tau, x(\tau))d\tau = f(t, x(t))$$

nach dem Hauptsatz der Differential- und Integralrechnung. Weiter gilt:

$$x(t_0) = x_0 + \int_{t_0}^{t_0} f(\tau, x(\tau))d\tau = x_0$$

QED

Wozu hilft uns dieses Lemma? Der Vorteil liegt darin, dass wir durch die Integralgleichung eine Fixpunktgleichung in der unbekannten Funktion x gefunden haben. Diese sieht wie folgt aus:
Wir definieren den Operator F, den wir auf $x : I \to \mathbb{R}^d$ anwenden wollen durch

$$(F(x))(t) := x_0 + \int_{t_0}^{t} f(\tau, x(\tau))d\tau.$$

Dann kann man die Integralgleichung (3.3) schreiben als

$$x(t) = (F(x))(t)$$

oder, kürzer, als

$$x = F(x).$$

Unsere gesuchte Lösung x kann also als die Lösung einer Fixpunktgleichung in einem unendlich dimensionalen Raum aufgefasst werden. Wir wollen darauf nun den Banach'schen Fixpunktsatz anwenden. Dieser wurde in Numerik I behandelt. Zur Wiederholung erinnern wir daran, dass jeder vollständige und normierte Raum ein **Banach-Raum** ist, und dass für eine Teilmenge U eines Banachraumes X eine Abbildung $\Phi : U \to X$ **kontrahierend** ist, falls es einen reellen Kontraktionsfaktor $q < 1$ so gibt, dass

$$\|\Phi(x) - \Phi(y)\| \leq q\|x - y\| \text{ für alle } x, y \in U.$$

Der Banach'sche Fixpunktsatz lautet wie folgt:

Satz 3.9 (Banach'scher Fixpunktsatz) *Sei X ein Banach-Raum mit Norm $\|\cdot\|$ und $U \subseteq X$ eine abgeschlossene Teilmenge von X. Sei weiterhin $F : U \to U$ eine kontrahierende Abbildung mit Kontraktionsfaktor $q < 1$. Dann hat die Fixpunktgleichung $F(x) = x$ einen eindeutigen Fixpunkt x^*.*

Für den Beweis verweisen wir auf die Vorlesung Numerik I.
Im Folgenden bezeichne $\|\cdot\|_2$ die Euklidische Norm. Wir erinnern an die folgende Bezeichnung.

Notation 3.10

- *Sei $f : D \to \mathbb{R}^d$, $D \subseteq \mathbb{R}^{d+1}$. f ist **Lipschitzstetig bezüglich seiner letzten d Variablen**, falls zu jedem $(t_0, x_0) \in D$ eine Umgebung $U := U(t_0, x_0) \subseteq D$ und eine Konstante $L = L(t_0, x_0)$ so existiert, dass*

$$\|f(t, x) - f(t, y)\|_2 \leq L\|x - y\|_2 \text{ für alle } (t, x), (t, y) \in U.$$

- *Sei $f : D \to \mathbb{R}^d$, $D = I \times \mathbb{R}^d$. f ist **global Lipschitzstetig bezüglich seine letzten d Variablen**, falls es eine Konstante $L > 0$ so gibt, dass*

$$\|f(t, x) - f(t, y)\|_2 \leq L\|x - y\|_2 \text{ für alle } t \in I \text{ und } x, y \in \mathbb{R}^d.$$

Damit formulieren wir nun das Hauptergebnis dieses Abschnitts.

Satz 3.11 (Satz von Picard-Lindelöf) *Sei $D \subseteq \mathbb{R}^{d+1}$ offen und sei $f : D \to \mathbb{R}^d$ stetig und bezüglich der letzten d Variablen Lipschitzstetig. Dann existiert zu jedem $(t_0, x_0) \in D$ eine Umgebung I von t_0, auf der das Anfangswertproblem*

$$x'(t) = f(t, x(t)), \quad x(t_0) = x_0$$

eindeutig lösbar ist.

Bemerkung: Der Satz liefert nur die *lokale* Existenz von Lösungen, also auf kleinen Intervallen für t um t_0.

Beweis: Seien $(t_0, x_0) \in D$ gegeben. Weil f Lipschitzstetig ist, existiert $\bar{U} := U(t_0, x_0) \subseteq D$ und $L = L(t_0, x_0)$ so, dass

$$\|f(t,x) - f(t,y)\|_2 \leq L\|x - y\|_2, \text{ für alle } (t,x),\ (t,y) \in \bar{U}.$$

Wir wählen nun $\alpha, \beta > 0$ so, dass für

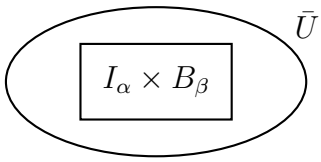
$$\begin{aligned} I_\alpha &= \{t \in \mathbb{R} : |t - t_0| \leq \alpha\} = [t_0 - \alpha, t_0 + \alpha] \\ \text{und} \quad B_\beta &= \{x \in \mathbb{R}^d : \|x - x_0\|_2 \leq \beta\} \end{aligned}$$

gilt:

$$I_\alpha \times B_\beta \subseteq \bar{U}.$$

Da f stetig auf der kompakten Menge $I_\alpha \times B_\beta$ ist, existiert

$$M := \max_{(t,x) \in I_\alpha \times B_\beta} \|f(t,x)\|_2.$$



Wir wählen α^* mit

$$0 < \alpha^* \leq \min\left(\frac{\beta}{M}, \alpha\right),$$

d.h. $\alpha^* > 0$, $\alpha^* \leq \alpha$ und $\alpha^* M \leq \beta$. Sei weiterhin

$$I^* := [-\alpha^* + t_0, \alpha^* + t_0].$$

Wir wollen den Banach'schen Fixpunktsatz anwenden und wählen dazu

- als Banachraum: $X := C(I^*, \mathbb{R}^d)$ als Menge der stetigen Funktionen von I^* nach $\mathbb{R}^d$,
- als Norm
$$\|x\|_B := \sup_{t \in I^*} e^{-2L|t-t_0|}\|x(t)\|_2, \text{ für alle } x \in X,$$
- als Teilmenge U den Unterraum
$$U := \{x \in X : \sup_{t \in I^*} \|x(t) - x_0\|_2 \leq \beta\}$$

- und als Abbildung $F : U \to X$, $x \mapsto F(x)$ den vorhin schon genannten Operator F, der durch

$$(F(x))(t) := x_0 + \int_{t_0}^{t} f(\tau, x(\tau))d\tau$$

definiert ist.

Jetzt überprüfen wir die Voraussetzungen des Banachschen Fixpunktsatzes.

1) X ist Vektorraum. Es ist leicht zu zeigen, dass $\|\cdot\|_B$ Norm auf X ist. Dass X vollständig ist, kann man mit Methoden der Analysis nachrechnen.

2) U ist abgeschlossen. Sei dazu (x_n) mit $x_n \in U$ eine Folge, die bezüglich $\|\cdot\|_B$ (gleichmäßig) gegen $x \in X$ konvergiert.

   Wir wollen zeigen, dass $x \in U$. Dazu berechnen wir:

$$\|x(t) - x_0\|_2 = \|\lim_{n\to\infty} x_n(t) - x_0\|_2 = \lim_{n\to\infty} \|x_n(t) - x_0\|_2,$$

   denn die Normfunktion ist stetig. Weil $x_n, x_0 \in U$ ist, gilt weiter

$$\|x_n(t) - x_0\|_2 \leq \beta \text{ für alle } t \in I^* \text{ und alle } n \in \mathbb{N},$$

   also

$$\|x(t) - x_0\|_2 = \lim_{n\to\infty} \underbrace{\|x_n(t) - x_0\|_2}_{\leq \beta \ \forall n} \leq \beta \text{ für alle } t \in I^*.$$

   Es folgt: $x \in U$.

3) Sei $F : U \to U$, sei $x \in U$. Dann ist $F(x) \in X$. Wir wollen zeigen, dass $F(x) \in U$, d.h.

$$\sup_{t\in I^*} \|(F(x))(t) - x_0\|_2 \leq \beta$$

   und berechnen dazu:

$$\begin{aligned} \|(F(x))(t) - x_0\|_2 &= \left\| \int_{t_0}^{t} f(\tau, x(\tau))d\tau \right\|_2 \\ &\leq |t - t_0| + \max_{\tilde{t}\in I^*, \tilde{x}\in B_\beta} \|f(\tilde{t}, \tilde{x})\|_2 \qquad (3.4) \\ &\leq \underbrace{\alpha^*}_{t,t_0\in I^*} \cdot \underbrace{M}_{\substack{\text{Def.} \\ \text{von } M}} \leq \underbrace{\beta}_{\substack{\text{Def.} \\ \text{von } \alpha^*}} \text{ für alle } t \in I^*. \end{aligned}$$

   In Abschätzung (3.4) darf man über der Menge $\tilde{t} \in I^*, \tilde{x} \in B_\beta$ maximieren, weil

   - mit $t, t_0 \in I^*$ das ganze Intervall zwischen t und t^* in I^* liegt, und weil

- aus $x \in U$ folgt, dass $\|x(\tau) - x_0\|_2 \leq \beta$ und entsprechend $\{x(\tau) : \tau \in [t_0, t]\} \subseteq B_\beta$.

Also folgt: $F(x) \in U$.

4) F ist Kontraktion. Wähle $x, y \in U$ und betrachte

$$e^{-2L|t-t_0|}\|(F(x))(t) - (F(y))(t)\|_2. \tag{3.5}$$

Der Übersicht halber betrachten wir zunächst nur:

$$\begin{aligned}
&\|(F(x))(t) - (F(y))(t)\|_2 \\
&= \left\| \int_{t_0}^{t} f(\tau, x(\tau)) - f(\tau, y(\tau)) d\tau \right\|_2 \\
&\leq \operatorname{sign}(t - t_0) \int_{t_0}^{t} \|f(\tau, x(\tau)) - f(\tau, y(\tau))\|_2 \, d\tau \\
&\leq L \operatorname{sign}(t - t_0) \int_{t_0}^{t} \|x(\tau) - y(\tau)\|_2 d\tau \\
&\leq L \operatorname{sign}(t - t_0) \int_{t_0}^{t} e^{2L|\tau - t_0|} \underbrace{e^{-2L|\tau - t_0|}\|x(\tau) - y(\tau)\|_2}_{\leq \|x - y\|_B} d\tau \\
&\leq L \operatorname{sign}(t - t_0) \int_{t_0}^{t} e^{2L|\tau - t_0|} d\tau \|x - y\|_B \\
&\leq L \operatorname{sign}(t - t_0) \left[\frac{1}{2L} e^{2L|t - t_0|} \right]_{t_0}^{t} \|x - y\|_B \\
&= L \operatorname{sign}(t - t_0) \frac{1}{2L} \operatorname{sign}(t - t_0)(e^{2L|t - t_0|} - 1)\|x - y\|_B \\
&= \frac{1}{2}(e^{2L|t - t_0|} - 1)\|x - y\|_B.
\end{aligned}$$

Dieses setzen wir jetzt in (3.5) ein und erhalten

$$\begin{aligned}
&e^{-2L|t-t_0|}\|(F(x))(t) - (F(y))(t)\|_2 \\
&\qquad \leq \tfrac{1}{2}\|x - y\|_B \underbrace{\left(1 - \underbrace{e^{-2L|t-t_0|}}_{\geq 0}\right)}_{\leq 1} \leq \tfrac{1}{2}\|x - y\|_B,
\end{aligned}$$

also gilt

$$\begin{aligned}
\|F(x) - F(y)\|_B &= \sup_{t \in I^*} e^{-2L|t-t_0|}\|(F(x))(t) - (F(y))(t)\|_2 \\
&\leq \tfrac{1}{2}\|x - y\|_B,
\end{aligned}$$

d.h. F ist Kontraktion mit $q = \frac{1}{2}$.

Somit sind alle Voraussetzungen des Banachschen Fixpunktsatzes erfüllt und wir erhalten:

$$x = F(x) \text{ besitzt eine eindeutige Lösung in } U.$$

Abschließend müssen wir noch ausschließen, dass F noch einen weiteren Fixpunkt $y \in X$ mit $\{(t, y(t)) : t \in I^*\} \subseteq D$ besitzt, der nicht in U liegt. Dazu ersetzen wir im vorhergehenden Beweis β durch $\beta/2$ und erhalten wie unter Punkt 2):

$$\|x(t) - x_0\|_2 \leq \frac{\beta}{2} \text{ für } |t - t_0| \leq \tilde{\alpha} := \min\left(\frac{\beta}{2M}, \alpha\right).$$

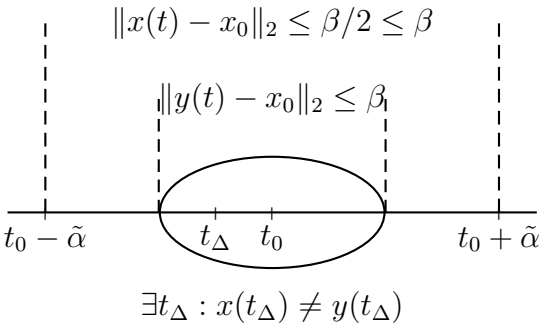
Sei weiterhin unser „neues“ U

$$\tilde{U} = \{x \in X : \sup_{t:|t-t_0|<\tilde{\alpha}} \|x(t) - x_0\|_2 \leq \beta/2\}.$$

Angenommen, so ein Fixpunkt $y \in X \setminus \tilde{U}$ existiert und löst damit (AWP). Wegen $y(t_0) = x_0$ gibt es α^{**} mit $0 < \alpha^{**} < \tilde{\alpha}$ und

$$\|y(t) - x_0\| \leq \beta \text{ für } |t - t_0| < \alpha^{**}.$$

sowie $x(t) \neq y(t)$ für mindestens ein t_Δ mit $|t_\Delta - t_0| < \alpha^{**}$. Weil $\|x(t) - x_0\| \leq \beta/2 \leq \beta$ für alle $|t - t_0| < \tilde{\alpha}$,



gäbe es aber auf $I^{**} := [-\alpha^{**} + t_0, \alpha^{**} + t_0]$ zwei verschiedene Lösungen $x,\ y$ der Fixpunktgleichung, die beide in

$$U^{**} = \{x \in X : \sup_{t \in I^{**}} \|x(t) - x_0\|_2 \leq \beta\}$$

liegen, was nach dem Banachschen Fixpunktsatz nicht sein kann. QED

Die globale Existenz einer Lösung liefert der folgende Satz:

Satz 3.12 *Sei $I \subseteq \mathbb{R}$ ein Intervall, sei $D = I \times \mathbb{R}^d$ und sei $f : D \to \mathbb{R}^d$ bezüglich der letzten d Variablen global Lipschitzstetig. Dann besitzt das Anfangswertproblem $x'(t) = f(t, x(t))$, $x(t_0) = x_0$ für alle $(t_0, x_0) \in D$ eine eindeutige Lösung $x : I \to \mathbb{R}^d$.*

Beweis: Im Beweis des Satzes von Picard-Lindelöf setzen wir $I_\alpha = I_* = I$ und wählen als Teilmenge U den ganzen Banachraum, also $U = X$. Die Konstanten $\alpha, \alpha^*, \beta, M$ werden nun nicht mehr benötigt. Die Details werden hier nicht ausgeführt. QED

Beispiel: Wir untersuchen die Voraussetzungen der Sätze 3.11 und 3.12 am zweiten Beispiel auf Seite 65,

$$\begin{aligned} x'(t) &= (x(t))^2 \\ x(0) &= 1. \end{aligned}$$

Wir erhalten $f(t,x) = x^2$ und entsprechend

$$\|f(t,x) - f(t,y)\|_2 = |x^2 - y^2| = |x+y| \cdot |x-y| \leq L \cdot |x-y| \text{ für alle } x, y \in I$$

falls

$$L \geq |x+y| \text{ für alle } x, y \in I$$

gilt.
Das ist auf jedem beschränkten Intervall erfüllt, nicht aber auf $I = [0, \infty)$ oder auf $I = \mathbb{R}$. Das Anfangswertproblem erfüllt daher die Voraussetzungen von Satz 3.11, aber nicht die von Satz 3.12, was zu dem so genannten "blow up" Effekt führt.

Unter den Voraussetzungen von Satz 3.12 kann dieser "blow up" Effekt nicht auftreten.

Bemerkung: Für lineare Differentialgleichungen

$$x'(t) = A(t)x(t) + g(t)$$

mit $t \in I$ und $f(t,x) = A(t)x + g(t)$ erhält man

$$\begin{aligned} \|f(t,x) - f(t,y)\|_2 &= \|A(t)x + g(t) - A(t)y - g(t)\|_2 \\ &= \|A(t)(x-y)\|_2 \leq \|A(t)\|_2 \|x-y\|_2 \\ &\leq L\|x-y\|_2, \end{aligned}$$

falls $L := \sup_{t \in I} \|A(t)\|_2 < \infty$.
Die Voraussetzungen von Satz 3.12 sind für lineare Differentialgleichungen also erfüllt, falls

$$\sup_{t \in I} \|A(t)\|_2 < \infty.$$

Das gilt insbesondere auf jedem kompakten Intervall I.

Der Banach'sche Fixpunktsatz liefert nicht nur theoretische Aussagen über Existenz und Eindeutigkeit, sondern mit dem Verfahren der sukzessiven Approximation auch ein konvergentes Verfahren zur Bestimmung des Fixpunktes. Dieses Verfahren lässt durch folgenden Iterationsschritt (so genannte *Picard-Iterationen*) auf Anfangswertprobleme anwenden:

$$x^{(n+1)}(t) := x^{(n)}(t_0) + \int_{t_0}^{t} f(\tau, x^{(n)}(\tau))d\tau$$

Als Startwert kann man z.B. $x^{(0)}(t) := x_0$ wählen – das resultierende Verfahren ist allerdings durch die dazu nötige numerische Auswertung der zahlreich auftretenden Integrale ineffizient und wird in der Praxis fast nicht verwendet.

Satz 3.13 (Globale Eindeutigeit) *Sind die Voraussetzungen von Satz 3.11 erfüllt und sind x und y Lösungen des (AWP)*

$$\begin{aligned} x'(t) &= f(t, x(t)) \\ x(t_0) &= x_0. \end{aligned}$$

auf einem beliebigen Intervall I mit $t_0 \in I$, so gilt $x(t) = y(t)$ für alle $t \in I$.

Beweis: Sei $I = [a, b]$, $t_0 \in I$ und seien x und y Lösungen des (AWP). Wähle $I' \subseteq I$ als das längste Intervall mit $x(t) = y(t)$ für alle $t \in I'$.
Wir möchten zeigen, dass $I = I'$.
Angenommen, dies ist gilt nicht, dann sei $I' = [a', b'] \subset I$. Dann ist ohne Beschränkung der Allgemeinheit $b' < b$. Wir betrachten das neue (AWP')

$$\begin{aligned} z'(t) &= f(t, z(t)) \\ z(b') &= x(b'). \end{aligned}$$

Nach Satz 3.11 existiert eine Umgebung $U = (b' - \alpha, b' + \alpha)$ mit $\alpha > 0$ auf der (AWP') eindeutig lösbar ist. Weil x und y beides Lösungen für (AWP') sind, folgt also $x(t) = y(t)$ für alle $t \in U$. Das ist ein Widerspruch zur Maximalität von I'.
QED

Abschließend geben wir noch ein Kriterium an, anhand dessen man die geforderte Lipschitz-Bedingung von Satz 3.11 nachweisen kann.

Lemma 3.14 *Ist $f : I \times \mathbb{R}^d \to \mathbb{R}^d$ bezüglich x stetig partiell differenzierbar, so erfüllt f die Lipschitz-Bedingung des Satzes 3.11 für alle $(t_0, x_0) \in I \times \mathbb{R}^d$.*

Beweis: (Vergleiche auch den Beweis von Lemma 5.7 aus Numerik I).
Weil f bezüglich x stetig partiell differenzierbar ist, existiert der Gradient

$$D_x f(t, x) : \mathbb{R} \times \mathbb{R}^d \to (\mathbb{R}^d)^*$$

und es gilt $L := \sup_{(t,x)\in\bar{U}(t_0,x_0)} \|D_x f(t,x)\|_2 < \infty$, wenn die Umgebung $\bar{U}$ kompakt gewählt wird. Wählt man $\bar{U}$ zusätzlich konvex, so kann man mittels

$$g(\xi) := f(t, x + \xi(y - x))$$

folgern, dass

$$\begin{aligned}
\|f(t,y) - f(t,x)\|_2 &= \|g(1) - g(0)\|_2 = \left\| \int_0^1 g'(\tau) d\tau \right\| \\
&= \left\| \int_0^1 D_x f(t, x + \tau(y-x)) \cdot (y - x) d\tau \right\|_2 \text{ multivariate Kettenregel} \\
&\leq \int_0^1 \|D_x f(t, x + \tau(y-x))\|_2 \cdot \|y - x\|_2 d\tau \\
&\leq \int_0^1 L\|y - x\|_2 d\tau = L\|y - x\|_2.
\end{aligned}$$

QED

Die Aussage von Satz 3.11 nutzen wir nun, um die *Evolution* zu definieren.

Definition 3.15 *Sei $D \subseteq \mathbb{R}^{d+1}$ offen, $f : D \to \mathbb{R}^d$ stetig und Lipschitzstetig bezüglich der letzten d Variablen. Seien $t_0, t \in I$ und $|t - t_0|$ hinreichend klein. Dann definiert man eine zweiparametrige Funktion*

$$\Phi^{t,t_0} : \mathbb{R}^d \to \mathbb{R}^d$$

durch $\Phi^{t,t_0}(x_0) := x(t)$, wobei $x(t)$ die eindeutige (lokale) Lösung des Anfangswertproblemes

$$\begin{aligned}
x'(t) &= f(t, x(t)) \\
x(t_0) &= x_0
\end{aligned}$$

*ist. Man nennt Φ die **Evolution** der Differentialgleichung $x'(t) = f(t, x(t))$.*

Φ^{t,t_0} bildet den Wert der Lösung x zur Zeit t_0 auf den Wert der gleichen Lösung zur Zeit t ab.

Beispiel: Betrachte $x'(t) = (x(t))^2$, also $f(t,x) = x^2$. Dann ist die eindeutige (lokale) Lösung zu (t_0, x_0) mit $t_0 = 0$, $x_0 > 0$ gegeben durch

$$x(t) = \frac{x_0}{1 - t x_0}, \quad \text{für } t < \frac{1}{x_0}.$$

Für die Evolution gilt entsprechend im Fall $t > 0$

$$\Phi^{t,0}(x_0) = \frac{x_0}{1 - t x_0} \text{ für } x_0 < \frac{1}{t}.$$

Lemma 3.16 *Die Evolution Φ der Differentialgleichung $x'(t) = f(t, x(t))$ besitzt die folgenden Eigenschaften:*

(Ev1) $\Phi^{t_0,t_0}(x_0) = x_0$

(Ev2) $\frac{\partial}{\partial\tau}\Phi^{t+\tau,t}(x_0)|_{\tau=0} = f(t, x_0)$

(Ev3) $\Phi^{t_2,t_0}(x_0) = \Phi^{t_2,t_1}(\Phi^{t_1,t_0}(x_0))$

für alle $(t_0, x_0) \in D$ und $|t_1 - t_0|$, $|t_2 - t_0|$ und $|t - t_0|$ hinreichend klein. Weiter ist Φ durch diese drei Bedingungen eindeutig charakterisiert.

Beweis: **(Ev1)** gilt weil $\Phi^{t_0,t_0}(x_0) = x(t_0) = x_0$.

(Ev2) Seien x_0, t fest. Sei x die Lösung des Anfangswertproblemes zum Startwert (t, x_0). Definiere

$$g(\tau) := \Phi^{t+\tau,t}(x_0) = x(t + \tau).$$

Dann gilt

$$\begin{aligned} & \frac{\partial}{\partial\tau}\Phi^{t+\tau,t}(x_0) = g'(\tau) = x'(t + \tau) = f(t + \tau, x(t + \tau)) \\ \Longrightarrow \quad & \frac{\partial}{\partial\tau}\Phi^{t+\tau,t}(x_0)|_{\tau=0} = g'(0) = f(t, x(t)) = f(t, x_0) \end{aligned}$$

(Ev3) Sei x Lösung von

$$\begin{aligned} x'(t) &= f(t, x(t)) \\ x(t_0) &= x_0, \end{aligned}$$

das heißt $\Phi^{t,t_0}(x_0) = x(t)$ für alle t nahe genug an t_0. Damit gilt:

$$\begin{aligned} \Phi^{t_2,t_1}(\Phi^{t_1,t_0}(x_0)) &= \Phi^{t_2,t_1}(x(t_1)) \\ &= x(t_2) = \Phi^{t_2,t_0}(x_0), \end{aligned}$$

wobei die vorletzte Gleichheit gilt, weil für $t_2 - t_0$ hinreichend klein x auch Lösung ist von dem Anfangswertproblem

$$\begin{aligned} y'(t) &= f(t, y(t)) \\ y(t_1) &= x(t_1). \end{aligned}$$

(Eindeutigkeit) Sei $\Psi^{t,t_0} : \mathbb{R}^d \to \mathbb{R}^d$ eine Funktion, die ebenfalls die drei Bedingungen (Ev1),(Ev2) und (Ev3) erfüllt. Sei (t_0, x_0) beliebig. Definiere

$$x(t) := \Psi^{t,t_0}(x_0).$$

Dann gilt

$$\begin{aligned} x'(t) &= \frac{\partial}{\partial \tau} \Psi^{t+\tau,t_0}(x_0)|_{\tau=0} \\ &= \frac{\partial}{\partial \tau} \left(\Psi^{t+\tau,t}\left(\Psi^{t,t_0}(x_0)\right)\right)|_{\tau=0} \text{ wegen (Ev3)} \\ &= f(t, \Psi^{t,t_0}(x_0)) \text{ wegen (Ev2)} \\ &= f(t, x(t)) \end{aligned}$$

und wegen (Ev1) ist außerdem $x(t_0) = \Psi^{t_0,t_0}(x_0) = x_0$. Also ist nach Satz 3.11

$$x(t) = \Phi^{t,t_0}(x_0)$$

die eindeutige (lokale) Lösung des Anfangswertproblemes

$$\begin{aligned} x'(t) &= f(t, x(t)) \\ x(t_0) &= x_0, \end{aligned}$$

und entsprechend gilt $\Psi^{t,t_0}(x_0) = \Phi^{t,t_0}(x_0)$ für alle $(t_0, x_0) \in D$ und alle t mit $|t - t_0|$ hinreichend klein. QED

Abschließend führen wir noch den Begriff der *Stabilität* ein. Dieser gibt an, wie stark sich zwei Lösungen $x(t)$ und $y(t)$ derselben Differentialgleichung unterscheiden, wenn die Anfangswerte $x(t_0)$ und $y(t_0)$ nur wenig voneinander abweichen. Dabei interessieren wir uns für die Zukunft, d.h. nur für Werte $t \geq t_0$.

Definition 3.17 *Sei $D \subset \mathbb{R}^d$, $t_0 \in \mathbb{R}$. Die Funktion $f : [t_0, \infty] \times D$ erfüllt eine einseitige Lipschitz-Bedingung mit Konstante $L^+ = L^+(t) \in \mathbb{R}$, falls*

$$(x - y)^T (f(t, x) - f(t, y)) \leq L^+ \|x - y\|_2^2 \quad \forall x, y \in D$$

und für alle $t \in [t_0, \infty]$. Kann $L^+ \leq 0$ gewählt werden, so nennt man f und die zugehörige Differentialgleichung $x' = f(t, x)$ ***dissipativ****.*

Bemerkung: Aus globaler Lipschitzstetigkeit für $t \geq t_0$ folgt die einseitige Lipschitz-Bedingung.

Dieses zeigen wir im Folgenden.
Sei $\|f(t, x) - f(t, y)\| \leq L \cdot \|x - y\|$ für alle $x, y \in D$ und alle $t \geq t_0$. Dann gilt nach der Cauchy-Schwarzschen Ungleichung:

$$\begin{aligned} (x - y)^T (f(t, x) - f(t, y)) &\leq \|x - y\|_2 \cdot \|f(t, x) - f(t, y)\|_2 \\ &\leq L^+ \cdot \|x - y\|_2^2 \text{ mit } L^+ = L. \end{aligned}$$

Die Umkehrung gilt aber nicht, wie das folgende Beispiel zeigt.

Beispiel: $f(t,x) = -x$ erfüllt die einseitige Lipschitz-Bedingung mit $L^+ = -1$, denn

$$(x-y)(f(t,x) - f(t,y)) = (x-y)(y-x) = -(x-y)^2 = -\|x-y\|_2^2.$$

Dagegen ergibt die globale Lipschitz-Bedingung

$$|f(t,x) - f(t,y)| = |y-x| \leq L \cdot |y-x|,$$

gilt also nur für $L \geq 1$.

Satz 3.18 *Erfüllt $f : [0,\infty] \times D \to \mathbb{R}^d$ eine einseitige Lipschitz-Bedingung mit Konstante L^+, so gilt für die Evolution Φ von $x' = f(t,x)$:*

$$\|\Phi^{t,t_0}(x_0) - \Phi^{t,t_0}(y_0)\|_2 \leq e^{L^+(t-t_0)}\|x_0 - y_0\|_2.$$

Für dissipative Systeme gilt insbesondere, dass

$$\|\Phi^{t,t_0}(x_0) - \Phi^{t,t_0}(y_0)\|_2 \leq \|x_0 - y_0\|_2.$$

Beweis: siehe Übungen.

3.3 Einschritt-Verfahren

3.3.1 Grundlagen

Obwohl eine Lösung bei stetigen Eingangsdaten immer existiert, ist sie im Allgemeinen selbst bei skalaren Differentialgleichungen mit $d = 1$ nicht in geschlossener Form darstellbar. Meist ist f auch nur durch Messwerte gegeben.
Die Grundidee der numerischen Lösung von Anfangswertproblemen ist, die Lösung x näherungsweise an diskreten Punkten zu ermitteln:

> gesucht werden Näherungswerte an den gesuchten Vektor $x(t)$ für $t \in \Delta := \{t_0, t_1, \ldots, t_N\}$ mit $t_0 < t_1 < \ldots < t_N = T$ auf dem Intervall $[t_0, T]$.

Notation 3.19 *$\Delta := \{t_0, t_1, \ldots, t_N\}$ mit $t_0 < t_1 < \ldots < t_N = T$ heißt **Gitter** auf $[t_0, T]$. Die Werte $T_j := t_{j+1} - t_j$ nennt man **Schrittweiten**. Die **Feinheit des Gitters** ist gegeben durch*

$$\tau_\Delta := \max_{j=0,\ldots,N-1} T_j.$$

Gesucht ist dann eine **Gitterfunktion** $x_\Delta : \Delta \to \mathbb{R}^d$, welche die Lösung von $x'(t) = f(t, x(t))$, $x'(t_0) = x_0$ auf dem Gitter möglichst gut approximiert.

Bei **Einschritt-Verfahren** ermittelt man x_Δ durch eine Zwei-Term-Rekursion:

$$x_\Delta(t_j) \to x_\Delta(t_{j+1}),$$

das heißt in die Berechnung von $x_\Delta(t_{j+1})$ geht nur $x_\Delta(t_j)$ ein, keine Werte von t_i mit $i < j$. Dagegen gehen bei Mehr-Term-Rekursionen mehrere Werte in die Berechnung von $x_\Delta(t_{j+1})$ mit ein, genauer für $m \in \mathbb{N}$:

$$x_\Delta(t_j), \ldots, x_\Delta(t_{j-m}) \to x_\Delta(t_{j+1}).$$

Diese Rekursionen führen zu **Mehrschritt-Verfahren**.

Im Folgenden wird die Evolution Φ der Differentialgleichung durch eine **diskrete Evolution** Ψ ersetzt.

korrekte Evolution:	Approximation durch diskrete Evolution:
$x(t_{j+1}) = \Phi^{t_{j+1},t_j}(x(t_j))$	$x_\Delta(t_{j+1}) := \Psi^{t_{j+1},t_j}(x_\Delta(t_j))$
$x(t_0) = x_0$	$x_\Delta(t_0) := x_0$

3.3.2 Beispiele

Um Einschritt-Verfahren herzuleiten benutzt man die Integraldarstellung des Anfangswertproblems aus Lemma 3.8:

$$x(t_0 + \tau) = x_0 + \int_{t_0}^{t_0+\tau} f(t, x(t))dt. \tag{3.6}$$

Explizites Euler-Verfahren

Seien zunächst

$$t_j := t_0 + j \cdot \tau$$

äquidistante Gitterpunkte. Man approximiert $x(t_j)$ aus (3.6) nun iterativ wie folgt:

$$x(t_1) = x(t_0 + \tau) = x_0 + \int_{t_0}^{t_0+\tau} f(t, \underbrace{x(t)}_{\text{unbekannt}})dt,$$

Um das Integral abzuschätzen, verwendet man die Rechteck-Regel mit Funktionsauswertung am linken Randpunkt und erhält:

$$\int_{t_0}^{t_0+\tau} f(t, x(t))dt \approx \tau \cdot f(t_0, x_0).$$

Das ergibt

$$x(t_1) \approx x_0 + \tau \cdot f(t_0, x_0)$$

bzw. für unsere Approximationsfunktion

$$x_\Delta(t_1) = x_0 + \tau \cdot f(t_0, x_0).$$

Diese Formel ergibt sich alternativ auch aus dem Differenzenquotienten durch

$$\tfrac{x(t_0+\tau)-x(t_0)}{\tau} \approx x'(t_0) = f(t_0, x_0).$$

Ist nun $x(t_1)$ approximativ bekannt, erhält man

$$\begin{aligned} x(t_2) &= x(t_1) + \int_{t_1}^{t_1+\tau} f(t, x(t))dt \\ &\approx x_\Delta(t_1) + \tau \cdot f(t_1, x_\Delta(t_1)) =: x_\Delta(t_2) \end{aligned}$$

und rekursiv

$$x_\Delta(t_{j+1}) = x_\Delta(t_j) + \tau \cdot f(t_j, x_\Delta(t_j)).$$

Die diskrete Evolution ergibt sich entsprechend zu

$$\Psi^{t+\tau,t}_{\text{E-Euler}}(x) = x + \tau \cdot f(t, x).$$

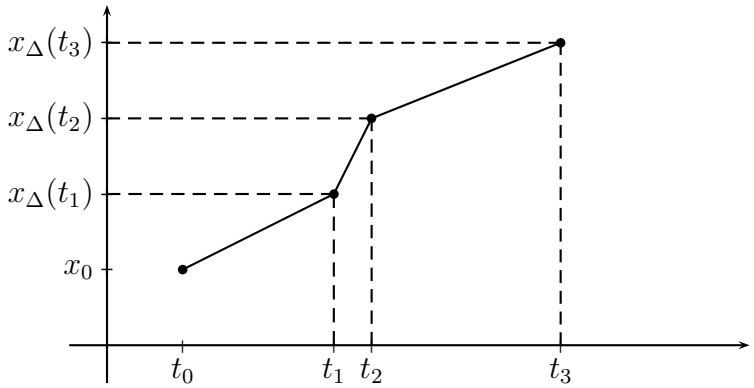
Etwas allgemeiner ist es mit $\tau_i := t_{i+1} - t_i$ nicht mehr nötig, äquidistante Stützstellen zu verwenden. Man erhält

$$x_\Delta(t_{j+1}) = \Psi^{t_{j+1},t_j}_{\text{E-Euler}}(x_\Delta(t_j)) := x_\Delta(t_j) + \tau_j \cdot f(t_j, x_\Delta(t_j)).$$

Interpretation:

> Um den Wert $x_\Delta(t_{j+1})$ an t_{j+1} zu bestimmen, verwendet man den Wert in $x_\Delta(t_j) + \tau_j \cdot x'(t_j, x_\Delta(t_j))$, also den Startwert und die Steigung an dem Ausgangspunkt $(t_j, x_\Delta(t_j))$.

Im skalaren Fall nennt man das explizite Euler-Verfahren daher auch Polygonzug-Verfahren.

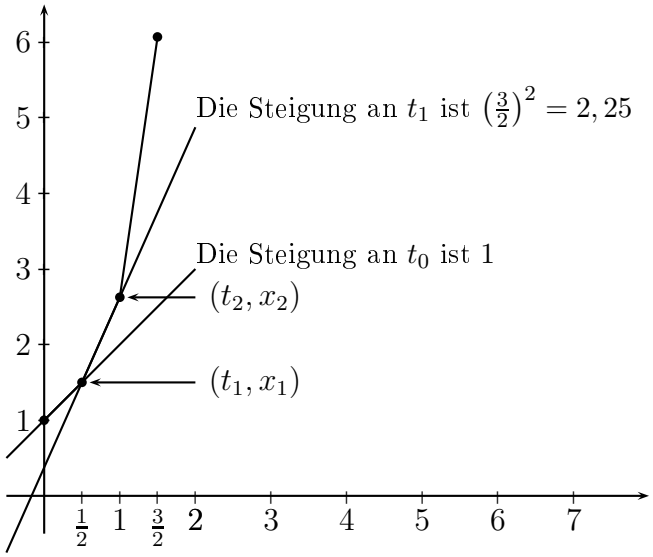


Beispiel: Sei folgendes Problem gegeben:

$$\begin{aligned} x'(t) &= (x(t))^2 \\ x(0) &= 1 \\ f(t,x) &= x^2 \\ \Delta &= \{0, \tfrac{1}{2}, 1, \tfrac{3}{2}\} \end{aligned}$$

Dann erhält man

$$\begin{aligned} x_\Delta(t_1) &= x_0 + \tfrac{1}{2} \cdot f(0, x_0) = 1 + \tfrac{1}{2} \cdot 1 = \tfrac{3}{2} \\ x_\Delta(t_2) &= x_\Delta(t_1) + \tfrac{1}{2} \cdot f(t_1, x_\Delta(t_1)) = \tfrac{3}{2} + \tfrac{1}{2} \cdot \tfrac{9}{4} = \tfrac{12}{8} + \tfrac{9}{8} = \tfrac{21}{8} = 2,625 \\ x_\Delta(t_3) &= x_\Delta(t_2) + \tfrac{1}{2} \cdot f(t_2, x_\Delta(t_2)) = \tfrac{21}{8} + \tfrac{1}{2} \cdot \left(\tfrac{21}{8}\right)^2 \approx 6,07. \end{aligned}$$



Implizites Euler-Verfahren

Das **implizite Euler-Verfahren** entsteht, wenn man das Integral durch die Rechteck-Regel am rechten Randpunkt approximiert:

$$\int_{t_j}^{t_j+\tau_j} f(t, x(t))dt \approx \tau_j \cdot f(t_j + \tau_j, x(t_j + \tau)).$$

Man erhält:

$$x_\Delta(t_{j+1}) = \Psi^{t_{j+1},t_j}_{\text{I-Euler}}(x_\Delta(t_j)) = \underbrace{x_\Delta(t_j)}_{\text{bekannt}} + \tau_j \cdot f(t_{j+1}, \underbrace{x_\Delta(t_{j+1})}_{\text{unbekannt}}).$$

Um $x_\Delta(t_{j+1})$ zu bestimmen, muss also ein (nichtlineares) Gleichungssystem mit d Unbekannten und d Gleichungen gelöst werden – und das in jedem Schritt!

Euler-Heun-Verfahren

Wählt man statt der Rechteck-Regel die Trapez-Regel zur Integralauswertung, so erhält man die Näherung:

$$\int_{t_j}^{t_j+\tau_j} f(t,x(t))dt \approx t_j \cdot \frac{f(t_j,x(t_j)) + f(t_j+\tau_j, x(t_j+\tau_j))}{2}$$

und es ergibt sich

$$\underbrace{x_\Delta(t_{j+1})}_{\text{unbekannt}} = \Psi^{t_{j+1},t_j}_{\text{E-Heun}}(x_\Delta(t_j)) := x_\Delta(t_j) + \frac{\tau_j}{2}(f(t_j,x_\Delta(t_j)) + f(t_{j+1}, \underbrace{x_\Delta(t_{j+1})}_{\text{unbekannt}})). \quad (3.7)$$

Auch dieses Verfahren ist implizit, weil in jedem Zeitschritt der Vektor $x_\Delta(t_{j+1})$ aus einem (nichtlinearen) Gleichungssystem ermittelt werden muss. In diesem Fall kann man dazu das Verfahren der sukzessiven Approximation benutzen:

Lemma 3.20 *Die Funktion $f(t,x)$ sei Lipschitzstetig bezüglich x mit Lipschitzkonstante L. Sei weiter $L \cdot \tau_j < 2$ für alle $j = 0,\ldots,N-1$. Dann lässt sich das Gleichungssystem (3.7) durch sukzessive Approximation*

$$x_\Delta^{(m+1)}(t_{j+1}) := x_\Delta(t_j) + \frac{\tau_j}{2}\left[f(t_j,x_\Delta(t_j)) + f(t_{j+1}, x_\Delta^{(m)}(t_{j+1}))\right], m \in \mathbb{N}_0$$

lösen.

Beweis: Die Fixpunktgleichung lautet $x = g(x)$ mit

$$g(x) = x_\Delta(t_j) + \frac{\tau_j}{2}\left[f(t_j,x_\Delta(t_j)) + f(t_{j+1},x)\right]$$

in jedem Schritt j. Wir müssen nachweisen, dass g eine Kontraktion ist. Es gilt:

$$\begin{aligned}\|g(x) - g(\tilde{x})\|_2 &= \frac{\tau_j}{2}\|f(t_{j+1},x) - f(t_{j+1},\tilde{x})\|_2 \leq \frac{\tau_j}{2} \cdot L \cdot \|x-\tilde{x}\|_2 \\ &= q \cdot \|x-\tilde{x}\|_2 \quad \text{mit } q = \frac{\tau_j}{2} \cdot L < 1.\end{aligned}$$

QED

Prädiktor-Korrektor Variante von Euler-Heun

Hier kombiniert man das explizite Euler-Verfahren und das Euler-Heun Verfahren mit jeweils einem Iterationsschritt der sukzessiven Approximation nach Lemma 3.20 wie folgt:

Im j-ten Schritt:

- bestimme den Startwert (Prädiktor) nach E-Euler:

$$\tilde{x}_\Delta(t_{j+1}) := x_\Delta(t_j) + \tau_j \cdot f(t_j, x_\Delta(t_j)) \qquad \text{(Prädiktor)}$$

- Wähle $\tilde{x}_\Delta(t_{j+1})$ als Startwert für die sukzessive Approximation von Euler-Heun und führe darin genau einen Schritt der sukzessiven Approximation nach Lemma 3.20 aus (Korrektor-Schritt):

$$x_\Delta(t_{j+1}) = \Psi^{t_{j+1},t_j}_{\text{Pre-Kor-V}}(x_\Delta(t_j)) = x_\Delta(t_j) + \frac{\tau_j}{2}\left[f(t_j, x_\Delta(t_j)) + f(t_{j+1}, \underbrace{\tilde{x}_\Delta(t_{j+1})}_{\text{aus (Prädiktor)}})\right].$$

Das Verfahren erreicht gewöhnlich eine höhere Genauigkeit als das explizite Euler-Verfahren.

3.3.3 Konsistenz und Eindeutigkeit

Wir untersuchen nun das Konvergenzverhalten von Einschritt-Verfahren theoretisch. Dazu fordern wir zunächst die ersten beiden der drei Eigenschaften einer Evolution (aus Lemma 3.16) auch für die diskrete Evolution Ψ.

Definition 3.21 *Eine diskrete Evolution Ψ heißt **konsistent** zur Differentialgleichung $x' = f(t,x)$, falls für alle $(t_0, x_0) \in D$ gilt:*

$$\Psi^{t_0,t_0}(x_0) = x_0 \tag{3.8}$$

$$\textit{und} \qquad \frac{d}{d\tau}\Psi^{t_0+\tau,t_0}(x_0)|_{\tau=0} = f(t_0, x_0). \tag{3.9}$$

*Ein Einschritt-Verfahren heißt **konsistent**, falls es jeder hinreichend glatten Funktion f eine konsistente diskrete Evolution $\Psi[f]$ zuordnet.*

Zwei äquivalente Konsistenzkriterien sind die folgenden.

Lemma 3.22 *Die diskrete Evolution $\Psi^{t_0+\tau,t_0}(x_0)$ sei für alle $(t_0, x_0) \in D$ und hinreichend kleines τ differenzierbar. Dann sind die folgenden Aussagen äquivalent:*

(i) *Ψ ist konsistent.*

(ii) *Es gibt eine bezüglich τ stetige Verfahrensfunktion $\phi = \phi(t_0, x_0, \tau)$ mit den Eigenschaften:*

$$\Psi^{t_0+\tau,t_0}(x_0) = x_0 + \tau \cdot \phi(t_0, x_0, \tau) \tag{3.10}$$

$$\phi(t_0, x_0, 0) = f(t_0, x_0) \tag{3.11}$$

(iii) Es gilt:

$$\lim_{\tau\to 0}\frac{1}{\tau}\left\|\Psi^{t_0+\tau,t_0}(x_0)-\Phi^{t_0+\tau,t_0}(x_0)\right\|=0. \tag{3.12}$$

Beweis:

$(i)\Longrightarrow(ii)$**:** Sei Ψ konsistent. Definiere

$$\phi(t_0,x_0,\tau):=\begin{cases}\frac{1}{\tau}\left(\Psi^{t_0+\tau,t_0}(x_0)-x_0\right) & \text{falls } \tau\neq 0.\\ f(t_0,x_0) & \text{falls } \tau=0\end{cases}$$

Dann sind (3.10) und (3.11) direkt erfüllt und es muss nur die Stetigkeit von ϕ gezeigt werden. Dazu betrachten wir

$$\begin{aligned}\lim_{\tau\to 0}\frac{1}{\tau}\left(\Psi^{t_0+\tau,t_0}(x_0)-x_0\right) &= \lim_{\tau\to 0}\frac{\Psi^{t_0+\tau,t_0}(x_0)-\Psi^{t_0,t_0}(x_0)}{\tau}, \text{ wegen (3.8)}\\ &= \tfrac{\partial}{\partial\tau}\Psi^{t_0+\tau,t_0}(x_0)|_{\tau=0}, \text{ wegen (3.9)}\\ &= f(t_0,x_0),\end{aligned}$$

also ist ϕ stetig.

$(ii)\Longrightarrow(iii)$**:** Sei ϕ eine Verfahrensfunktion, die (3.10) und (3.11) erfüllt. Dann gilt

$$\begin{aligned}&\lim_{\tau\to 0}\frac{1}{\tau}\|\Psi^{t_0+\tau,t_0}(x_0)-\Phi^{t_0+\tau,t_0}(x_0)\|\\ &=\lim_{\tau\to 0}\left\|\frac{\Psi^{t_0+\tau,t_0}(x_0)-x_0}{\tau}-\frac{\Phi^{t_0+\tau,t_0}(x_0)-x_0}{\tau}\right\|\\ &=\|\phi(t_0,x_0,0)-f(t_0,x_0)\| \text{ wegen (3.10) und [Ev2] im Lemma 3.16}\\ &=0 \text{ wegen (3.11)}\end{aligned}$$

$(iii)\Longrightarrow(i)$**:** Sei nun (3.12) erfüllt. Eine Taylorentwicklung bis zum Grad 1 liefert wegen [Ev2]

$$\Phi^{t_0+\tau,t_0}(x_0)=x_0+\tau f(t_0,x_0)+o(\tau) \text{ für } \tau\to 0.$$

Weiter ist Ψ nach Voraussetzung für hinreichend kleines τ differenzierbar bezüglich τ. Das ergibt

$$\Psi^{t_0+\tau,t_0}(x_0)=\Psi^{t_0,t_0}(x_0)+\tau\tfrac{\partial}{\partial\tau}\Psi^{t_0+\tau,t_0}(x_0)|_{\tau=0}+o(\tau) \text{ für } \tau\to 0.$$

Für $\tau\to 0$ sind die linken Seiten dieser beiden Gleichungen wegen (3.12) gleich, also auch die rechten Seiten und durch einen Koeffizientenvergleich folgt $x_0=\Psi^{t_0,t_0}(x_0)$ und $f(t_0,x_0)=\frac{\partial}{\partial\tau}\Psi^{t_0+\tau,t_0}|_{\tau=0}$; (3.8) und (3.9) gelten also und Ψ ist konsistent. QED

Ist eine diskrete Evolution konsistent, so ist der lokale Fehler, den wir in jedem Schritt bei der Berechnung der Gitterfunktion machen, klein. Interessanter ist aber der globale Fehler

$$\max_{t\in\Delta} \|x_\Delta(t) - x(t)\|,$$

der möglichst klein sein soll – zumindest wenn das Gitter Δ fein genug ist.

Notation 3.23 *Ein Einschritt-Verfahren heißt* ***konvergent****, falls*

$$\lim_{\tau\to 0} \sup_{\Delta:\tau_\Delta=\tau} \max_{t\in\Delta} \|x_\Delta(t) - x(t)\| = 0$$

Dabei bezeichnet $\tau_\Delta = \max_{j=0,\dots,N-1} t_{j+1} - t_j$ wie schon zu Beginn des Abschnittes 3.3.1 die Feinheit des Gitters $\Delta = \{t_0, \dots, t_N\}$.

Der folgende Satz zeigt, dass aus der Konsistenz unter einer zusätzlichen Stabilitätsannahme die Konvergenz von Einschritt-Verfahren folgt. Dabei müssen wir die Konsistenzbedingung allerdings verstärken: Wir verwenden (3.12) und verlangen, dass die Bedingung gleichmäßig erfüllt ist, also für alle $x(t)$ auf der Lösungskurve.

Satz 3.24 *Die diskrete Evolution Ψ sei in einer Umgebung U der Trajektorie $\{(t, x(t)) : t \in [t_0, T]\}$ definiert und genüge den folgenden Bedingungen.*

Stabilitätsbedingung: *Es gibt Konstanten $L_\Psi \geq 0$ und $\tau_0 > 0$ so, dass*

$$\|\Psi^{t+\tau,t}(x_1) - \Psi^{t+\tau,t}(x_2)\| \leq e^{L_\Psi \tau} \|x_1 - x_2\|$$

für alle $(t, x_1), (t, x_2) \in U$ und alle $0 \leq \tau \leq \tau_0$.

Konsistenzbedingung: *Es gibt eine monoton wachsende Funktion $\mathrm{err} : [0, \tau_0] \to [0, \infty)$ mit $\lim_{\tau\to 0} \mathrm{err}(\tau) = 0$ so, dass*

$$\|\Phi^{t+\tau,t}(x(t)) - \Psi^{t+\tau,t}(x(t))\| \leq \tau\ \mathrm{err}(\tau)$$

für alle $t \in [0, T]$.

Dann gibt es ein $\tau_1 \in [0, \tau_0]$ so, dass für jedes Gitter $\Delta = \{t_0, \dots, t_N\}$ auf $[t_0, T]$ mit Feinheit $\tau_\Delta \leq \tau_1$ die Gitterfunktion x_Δ durch die diskrete Evolution

$$x_\Delta(t_{j+1}) = \Psi^{t_{j+1},t_j}(x_\Delta(t_j)), \qquad x_\Delta(t_0) = x_0$$

wohldefiniert ist, und der Fehler für alle $t \in \Delta$ der Abschätzung

$$\|x_\Delta(t) - x(t)\| \leq r(\tau_\Delta) := \begin{cases} \mathrm{err}(\tau_\Delta)\frac{e^{L_\Psi(t-t_0)}-1}{L_\Psi} & \text{falls } L_\Psi > 0 \\ \mathrm{err}(\tau_\Delta)(t - t_0) & \text{falls } L_\Psi = 0 \end{cases}$$

genügt.

Der Satz sagt auf abstrakter Ebene, dass Konsistenz und Stabilität zusammen Konvergenz ergeben.

Beweis: Wir wählen τ_1 so klein, dass für alle $t \in [0,T]$ und für alle $x_1 \in \mathbb{R}^d$ gilt:

$$\|x_1 - x(t)\| \leq r(\tau_1) \Longrightarrow (t, x_1) \in U.$$

Sei Δ ein beliebiges Gitter mit $\tau_\Delta \leq \tau_1$. Wir möchten nachweisen, dass die Abschätzung

$$\|x_\Delta(t) - x(t)\| \leq r(\tau_\Delta)$$

für alle $t_0, t_1, \ldots, t_N$ des Gitters Δ erfüllt ist.
Insbesondere gilt dann $\|x_\Delta(t) - x(t)\| \leq r(\tau_1)$, woraus wir wegen der Definition von τ_1 folgern, dass $(t_j, x_\Delta(t_j)) \in U$. Entsprechend kann man also $x_\Delta(t_{j+1}) = \Psi^{t_{j+1},t_j}(x_\Delta(t_j))$ berechnen und $x_\Delta(t_j)$ ist wohldefiniert.

Zum Nachweis der Abschätzung verwenden wir Induktion nach j, gehen also der Reihe nach alle Punkte $t_0, t_1, \ldots, t_N$ des Gitters Δ durch.
Für $j = 0$ gilt $x_\Delta(t_0) = x_0 = x(t_0)$, die Abschätzung gilt also wegen $r(\tau_\Delta) \geq 0$.
Sei nun $\|x_\Delta(t_{j'}) - x(t_{j'})\| \leq r(\tau_\Delta)$ für alle $j' \leq j$ erfüllt. Wir betrachten t_{j+1}. Dazu unterscheiden wir zwei Fälle.

Fall 1: Sei $L_\Psi > 0$. Dann gilt

$$\begin{aligned}
& \|x_\Delta(t_{j+1}) - x(t_{j+1})\| \\
= \; & \|\Psi^{t_{j+1},t_j}(x_\Delta(t_j)) - \Phi^{t_{j+1},t_j}(x_\Delta(t_j))\| \\
\leq \; & \|\Psi^{t_{j+1},t_j}(x_\Delta(t_j)) - \Psi^{t_{j+1},t_j}(x(t_j))\| + \|\Psi^{t_{j+1},t_j}(x(t_j)) - \Phi^{t_{j+1},t_j}(x(t_j))\| \\
\leq \; & e^{L_\Psi(t_{j+1}-t_j)}\|x_\Delta(t_j) - x(t_j)\| + (t_{j+1} - t_j)\mathrm{err}(\tau_\Delta) \text{ wegen Stabilität und Konsistenz} \\
\leq \; & \frac{\mathrm{err}(\tau_\Delta)}{L_\Psi}\left(e^{L_\Psi(t_{j+1}-t_j)}\left(e^{L_\Psi(t_j-t_0)} - 1\right) + L_\Psi(t_{j+1} - t_j)\right) \text{ Induktionsannahme} \\
= \; & \frac{\mathrm{err}(\tau_\Delta)}{L_\Psi}\left(e^{L_\Psi(t_{j+1}-t_0)} \underbrace{-e^{L_\Psi(t_{j+1}-t_j)} + L_\Psi(t_{j+1} - t_j)}_{\leq -1, \text{ denn } e^a \geq a+1}\right) \\
\leq \; & \frac{\mathrm{err}(\tau_\Delta)}{L_\Psi}\left(e^{L_\Psi(t_{j+1}-t_0)} - 1\right) = r(\tau_\Delta).
\end{aligned}$$

Fall 2: Sei $L_\Psi = 0$. Dann geht man vor wie oben, allerdings ergibt die Induktionsvoraussetzung, dass

$$\begin{aligned}
\|x_\Delta(t_{j+1}) - x(t_{j+1})\| &\leq \mathrm{err}(\tau_\Delta)(t_j - t_0) + \mathrm{err}(\tau_\Delta)(t_{j+1} - t_j) \\
&= \mathrm{err}(\tau_\Delta)(t_{j+1} - t_0)
\end{aligned}$$

QED

Ein weiterer Begriff ist die Konsistenzordnung, welche hilft, die Konvergenzgeschwindigkeit eines Einschritt-Verfahrens abzuschätzen.

Definition 3.25

- *Eine diskrete Evolution Ψ für eine Differentialgleichung $x'(t) = f(t, x(t))$, $f : D \to \mathbb{R}^d$, besitzt die* ***Konsistenzordnung*** *$p > 0$, falls es für jede kompakte Teilmenge $K \subseteq D$ eine Konstante $C > 0$ so gibt, dass*

  $$\|\Psi^{t+\tau,t}(x) - \Phi^{t+\tau,t}(x)\| \leq C \cdot \tau^{p+1}$$

  für alle $(t, x) \in K$ und alle hinreichend kleinen $\tau \geq 0$.

- *Ein Einschritt-Verfahren besitzt die Konsistenzordnung $p > 0$, falls für jede rechte Seite $f \in C^\infty(D, \mathbb{R}^d)$ die zugeordnete diskrete Evolution $\Psi = \Psi[f]$ die Konsistenzordnung p besitzt.*

- *Ein Einschritt-Verfahren besitzt die Konvergenzordnung $p > 0$, falls für jede Lösung $x : [t_0, T] \to \mathbb{R}^d$ eines Anfangswertproblemes mit rechter Seite $f \in C^\infty(D, \mathbb{R}^d)$ der globale Fehler der durch das Verfahren bestimmten Lösung x_Δ auf einem Gitter Δ mit hinreichend kleiner Gitterfeinheit τ_Δ die Abschätzung*

  $$\max_{t \in \Delta} \|x_\Delta(t) - x(t)\| \leq \tilde{C} \cdot {\tau_\Delta}^p$$

  erfüllt, wobei $\tilde{C}$ nicht von Δ abhängt.

Lemma 3.26 *Besitzt ein Einschritt-Verfahren die Konsistenzordnung p und erfüllt es die Stabilitätsbedingung aus Satz 3.24, so besitzt es die Konvergenzordnung p.*

Beweis: Sei $f \in C^\infty(D, \mathbb{R}^d)$ beliebig. Weil das Verfahren die Konsistenzordnung p hat, gilt für die diskrete Evolution Ψ, dass

$$\|\Psi^{t+\tau,t}(x) - \Phi^{t+\tau,t}(x)\| \leq C \cdot \tau^{p+1}.$$

Die Funktion $\mathrm{err}(\tau) := C \cdot \tau^p$ erfüllt dann wegen $\lim_{\tau \to 0} C \cdot \tau^p = 0$ die Konsistenzbedingung aus Satz 3.24. Wir können also Satz 3.24 anwenden und erhalten

$$\|x_\Delta(t) - x(t)\| \leq r(\tau_\Delta) = \begin{cases} \mathrm{err}(\tau_\Delta) \cdot \frac{e^{L_\Psi (t-t_0)} - 1}{L_\Psi} & L_\Psi > 0 \\ \mathrm{err}(\tau_\Delta) \cdot (t - t_0) & L_\Psi = 0. \end{cases}$$

Es folgt nun

$$\max_{t \in \Delta} \|x_\Delta(t) - x(t)\| \leq \tilde{C} \cdot {\tau_\Delta}^p \text{ mit } \tilde{C} = \begin{cases} C \cdot \frac{e^{L_\Psi (T-t_0)} - 1}{L_\Psi} & L_\Psi > 0 \\ C \cdot (T - t_0) & L_\Psi = 0 \end{cases}.$$

QED

Satz 3.27 *Die diskrete Evolution des expliziten Euler-Verfahrens ist für stetig differenzierbare Seiten f konsistent von der Ordnung 1.*

Beweis: Übung.

3.3.4 Explizite Runge-Kutta-Verfahren

Euler-Verfahren

Approximiere das Integral durch die Rechteckregel, d.h.

$$\int_t^{t+\tau} f(s, \underbrace{\Phi^{s,t}(x)}_{x(s)})ds \approx \tau \cdot f(t,x).$$

Dabei ist der Fehler nach Satz 3.27 von der Größe $O(\tau^2)$.

Verfahren von Runge (explizite Mittelpunktregel)

Die Idee ist, dass man eine Quadraturformel höherer Ordnung verwendet, zum Beispiel die Mittelpunktregel:

$$\int_t^{t+\tau} f(s, \Phi^{s,t}(x))ds \approx \tau \cdot f(t+\tfrac{\tau}{2}, \Phi^{t+\frac{\tau}{2},t}(x))$$

mit einem Fehler von $O(\tau^3)$. Allerdings kennen wir den Wert $\Phi^{t+\frac{\tau}{2},t}(x)$ nicht. Es reicht aber, ihn mit einer Genauigkeit von $O(\tau^2)$ auszuwerten, weil er noch mit τ multipliziert wird. Dazu verwendet man

$$\Phi^{t+\frac{\tau}{2},t}(x) = x + \tfrac{\tau}{2} f(t,x)$$

nach dem Euler-Verfahren mit $O(\tau^2)$. Man erhält

$$\Psi^{t+\tau,t}(x) = x + \tau \cdot f(t+\tfrac{\tau}{2}, x + \tfrac{\tau}{2} f(t,x))$$

oder, algorithmisch:

$$\begin{aligned} k_1 &:= f(t,x) \\ k_2 &:= f(t+\tfrac{\tau}{2}, x + \tfrac{\tau}{2} \cdot k_1) \\ \Psi^{t+\tau,t}(x) &:= x + \tau \cdot k_2 \end{aligned}$$

Dieses Verfahren hat Konsistenzordnung 2.

Runge-Kutta-Verfahren

Seien

$$k_i = k_i(t,x,\tau) = f\left(t + c_i\tau, x + \tau \sum_{j=1}^{i-1} a_{ij} k_j\right), \qquad \text{für } i = 1, \ldots, s$$

$$\Psi^{t+\tau,t}(x) = x + \tau \sum_{j=1}^{s} b_j k_j(t,x,\tau) = x + \tau \sum_{j=1}^{s} b_j k_j.$$

k_i heißt die i-te Stufe des Runge-Kutta-Verfahrens. Man benutzt folgende Notation:

$$A = \begin{pmatrix} 0 & & & & 0 \\ a_{21} & 0 & & & \\ a_{31} & a_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ a_{s1} & \cdots & \cdots & a_{s,s-1} & 0 \end{pmatrix}, \qquad b = \begin{pmatrix} b_1 \\ \vdots \\ b_s \end{pmatrix}, \qquad c = \begin{pmatrix} c_1 \\ \vdots \\ c_s \end{pmatrix}.$$

Mit der Vereinbarung, dass $a_{ij} := 0$ für $j \geq i$ ist, vereinfachen wir die Summenschreibweise. Wir erhalten

$$k_i = f\left(t + c_i\tau, x + \tau \sum_{j=1}^{s} a_{ij} k_j\right), \qquad i = 1, \ldots, s.$$

Dabei heißt s die **Stufenzahl** des Runge-Kutta-Verfahrens und beschreibt die Tiefe der Schachtelungen von f-Auswertungen. Man gibt ein Verfahren oft durch folgendes **Butcher-Schema** an:

$$\begin{array}{c|c} c & A \\ \hline & b^t \end{array}$$

Beispiele:

1. Explizites Euler-Verfahren:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

   also

$$\begin{aligned} k_1 &= f(t + c_1\tau, x + 0) \\ &= f(t + 0, x + 0) = f(t, x) \end{aligned}$$

   und

$$\Psi^{t+\tau,t}(x) = x + \tau b_1 k_1 = x + \tau f(t, x).$$

2. Verfahren von Runge:

$$\begin{array}{c|cc} 0 & 0 & \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$$

3. „Klassisches“ Runge-Kutta-Verfahren der Ordnung 4:

$$\begin{array}{c|cccc} 0 & 0 & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Ausführliche Notation:

$$\begin{aligned} k_1 &:= f(t,x) \\ k_2 &:= f(t+\tfrac{1}{2}\tau, x+\tfrac{1}{2}\tau k_1) \\ k_3 &:= f(t+\tfrac{1}{2}\tau, x+\tfrac{1}{2}\tau k_2) \\ k_4 &:= f(t+\tau, x+\tau k_3) \\ \Psi^{t+\tau,t}(x) &:= x+\tau(\tfrac{1}{6}k_1+\tfrac{1}{3}k_2+\tfrac{1}{3}k_3+\tfrac{1}{6}k_4) \end{aligned}$$

Lemma 3.28 *Ein Runge-Kutta-Verfahren (A,b,c) ist genau dann konsistent für alle $f \in C(D,\mathbb{R}^d)$, falls*

$$\sum_{j=1}^{s} b_j = 1.$$

Beweis: Wir benutzen die beiden Bedingungen (3.10) und (3.11) aus Lemma 3.22 und definieren

$$\phi(t,x,\tau) := \sum_{j=1}^{s} b_j k_j(t,x,\tau).$$

Dann gilt (3.10), denn:

$$\begin{aligned} \Psi^{t+\tau,t}(x) &= x+\tau\sum_{j=1}^{s} b_j k_j(t,x,\tau) \\ &= x+\phi(t,x,\tau). \end{aligned}$$

Weiterhin gilt $k_j(t,x,0) = f(t,x)$ für alle j, also

$$\phi(t,x,0) = \sum_{j=1}^{s} b_j k_j(t,x,0) = f(t,x)\sum_{j=1}^{s} b_j.$$

Da die Bedingung (3.11) $\phi(t,x,0) = f(t,x)$ fordert, ist (3.11) genau dann erfüllt, wenn $\sum_{j=1}^{s} b_j = 1$ gilt. QED

Lemma 3.29 *Besitzt ein s-stufiges Runge-Kutta-Verfahren für alle $f \in C^\infty(D,\mathbb{R}^d)$ die Konsistenzordnung p, so gilt $p \le s$.*

Beweis: Betrachte (AWP)

$$x'(t) = x(t), \qquad x(0) = 1.$$

Die Lösung ist

$$\Phi^{\tau,0}(1) = e^\tau 1 = \tau + \tfrac{1}{2!}\tau^2 + \cdots + \tfrac{1}{p!}\tau^p + \mathcal{O}(\tau^{p+1}).$$

Für die Konsistenzordnung wollen wir $\Phi^{t+\tau,t}(x)$ mit $\Psi^{t+\tau,t}(x)$ vergleichen – für $f(t,x) = x(t)$ und $t = 0$, $x = 1$. Die echte Evolution Φ kennen wir schon. Um auch $\Psi^{t+\tau,t}(x)$ zu verstehen, betrachten wir die k_j.

Behauptung: $k_j(0,1,\tau)$ ist ein Polynom in τ vom Grad $\leq j-1$, also $k_j \in \Pi_{j-1}$.

Vollständige Induktion über j:

- $j=1$: $k(0,1,\tau) = f(t+c_1\tau, x) = x \in \Pi_0$, da konstant in τ.
- $j \mapsto j+1$:

$$k_{j+1}(0,1,\tau) = f\left(t + c_j\tau, x + \tau \sum_{l=1}^{j} a_{jl}k_l\right)$$
$$= x + \tau \underbrace{\sum_{l=1}^{j} a_{jl}k_l}_{\in \Pi_{j-1}} \in \Pi_j, \text{ da } k_l \in \Pi_{l-1} \text{ nach der Induktionsannahme.}$$

Also ist $\Psi^{\tau,0}(1) \in \Pi(s)$. Damit erhalten wir:

$$\| \underbrace{\Psi^{\tau,0}(1)}_{\in \Pi_s} - \underbrace{\Phi^{\tau,0}(1)}_{1+\tau+\cdots+\frac{1}{s!}\tau^s+\mathcal{O}(\tau^{s+1})} \| \leq c\,\tau^{s+1}.$$

Folglich kann die Konsistenzordnung höchstens s sein. QED

Bei der Konstruktion vom Runge-Kutta-Verfahren hat man also zunächst viele Wahlmöglichkeiten. Wir stellen aber die folgenden Bedingungen an das Verfahren:

1. Invarianz gegen Autonomisierung und
2. Konsistenzordnung p für vorgegebenes p.

Diese Bedingungen formulieren wir im Folgenden als Bedingungen an die Koeffizienten (A,b,c) des Verfahrens. Wir betrachten zuerst die Invarianz gegen Autonomisierung.
Seien $x'(t) = f(t,x(t))$ ein (AWP) im $\mathbb{R}^d$ und $x(t_0) = x_0$. Nach Lemma 3.6 lässt sich das in ein äquivalentes System im $\mathbb{R}^{d+1}$, nämlich in

$$\widehat{(\text{AWP})} \qquad y'(t) = \begin{pmatrix} 1 \\ f(y(t)) \end{pmatrix}, \qquad y^{(t_0)} = y_0 := \begin{pmatrix} t_0 \\ x_0 \end{pmatrix}$$

umwandeln. Dabei gelten die folgenden Aussagen:

- Ist x Lösung von (AWP), so ist $(t, x(t))^T$ eine Lösung von $\widehat{(\text{AWP})}$.
- Ist $(s,x)^T$ eine Lösung von $\widehat{(\text{AWP})}$, so folgt $s(t) = t$ und x ist Lösung von (AWP).

Formal lässt sich die Äquivalenz der beiden Anfangswertprobleme durch die Evolution $\hat{\Phi}$ von $y' = (1, f(y))^T$ und Φ von $x' = f(t,x)$ folgendermaßen schreiben:

$$\begin{pmatrix} t+\tau \\ \Phi^{t+\tau,t}(x) \end{pmatrix} = \hat{\Phi}^{t+\tau,t} \begin{pmatrix} t \\ x \end{pmatrix}.$$

Diese Eigenschaft soll dann auch für diskrete Evolutionen Ψ und $\hat{\Psi}$ gelten; sie soll also gewissermaßen vererbt werden. Für die Evolution Ψ (bzw. $\hat{\Psi}$ für das erweiterte System) bedeutet

$$\begin{pmatrix} t+\tau \\ \Psi^{t+\tau,t}(x) \end{pmatrix} = \hat{\Psi}^{t+\tau,t} \begin{pmatrix} t \\ x \end{pmatrix} \tag{3.13}$$

dass man das gleiche Ergebnis erhält, egal, ob man ein durch Ψ gegebenes Einschritt-Verfahren direkt auf die gegebene Differentialgleichung anwendet, oder ob man das gleiche Verfahren mittels $\hat{\Psi}$ auf die autonomisierte Differentialgleichung anwendet. Man nennt das Verfahren dann **invariant gegenüber Autonomisierung**.

Lemma 3.30 *Ein explizites Runge-Kutta Verfahren ist genau dann invariant gegen Autonomisierung, wenn es konsistent ist und es*

$$c_i = \sum_{j=1}^{s} a_{ij} \text{ für } j = 1, \ldots, s$$

erfüllt.

Beweis: Sei $y' = \hat{f}(y(t))$ die autonomisierte Differentialgleichung mit

$$\hat{f}\left(\begin{pmatrix} t \\ x \end{pmatrix}\right) = \left(\begin{pmatrix} 1 \\ f(t,x) \end{pmatrix}\right),$$

wobei $y(t) = \left(\begin{pmatrix} t \\ x(t) \end{pmatrix}\right)$ und $\hat{f}$ autonom ist, also $\hat{f}(t, y(t)) = \hat{f}(y(t))$ gilt. Bezeichnen wir nun mit $\hat{K}_i = \left(\begin{pmatrix} \hat{l}_i \\ \hat{k}_i \end{pmatrix}\right)$, $i = 1, \ldots, s$ die Stufen von $\hat{\Psi}$, so gilt:

$$\begin{aligned}
\hat{K}_i &= \hat{f}(t + c_i\tau, y + \tau \sum_{j=1}^{s} a_{ij}\hat{K}_j), \\
&= \hat{f}(y + \tau \sum_{j=1}^{s} a_{ij}\hat{K}_j) \\
&= \hat{f}\left(\begin{pmatrix} t \\ x \end{pmatrix} + \tau \sum_{j=1}^{s} a_{ij} \begin{pmatrix} \hat{l}_j \\ \hat{k}_j \end{pmatrix}\right) \\
&= \begin{pmatrix} 1 \\ f(t + \tau \sum_{j=1}^{s} a_{ij}\hat{l}_j, x + \tau \sum_{j=1}^{s} \hat{k}_j) \end{pmatrix} \quad i = 1, \ldots, s,
\end{aligned}$$

das heißt, $\hat{l}_i = 1$ und $\hat{k}_i = f(t + \tau \sum_{j=1}^{s} a_{ij}\hat{l}_j, x + \tau \sum_{j=1}^{s} a_{ij}\hat{k}_j)$ für $i = 1, \ldots, s$. Für ein Runge-Kutta Verfahren gilt weiter für die diskrete Evolution, dass

$$\hat{\Psi}^{t+\tau,t}\left(\begin{pmatrix} t \\ x \end{pmatrix}\right) = \left(\begin{pmatrix} t \\ x \end{pmatrix}\right) + \tau \sum_{j=1}^{s} b_j \left(\begin{pmatrix} \hat{l}_j \\ \hat{k}_j \end{pmatrix}\right) = \left(\begin{pmatrix} t + \tau \sum_{j=1}^{s} b_j \\ x + \tau \sum_{j=1}^{s} b_j \hat{k}_j \end{pmatrix}\right).$$

Nach (3.13) ist das Verfahren invariant gegen Autonomisierung genau dann wenn

$$\begin{aligned}
&\left(\begin{pmatrix} t+\tau \\ \Psi^{t+\tau,t}(x) \end{pmatrix}\right) = \hat{\Psi}^{t+\tau,t}\left(\begin{pmatrix} t \\ x \end{pmatrix}\right) \\
\iff\ & t + \tau = t + \tau \sum_{j=1}^{s} b_j \text{ und } \Psi^{t+\tau,t}(x) = x + \tau \sum_{j=1}^{s} b_j \hat{k}_j \\
\iff\ & \sum_{j=1}^{s} b_j = 1 \text{ und } x + \tau \sum_{j=1}^{s} b_j k_j = x + \tau \sum_{j=1}^{s} b_j \hat{k}_j \\
\iff\ & \text{konsistent und } k_i = \hat{k}_i \text{ für alle } i = 1, \ldots, s.
\end{aligned}$$

Letzteres ist genau dann der Fall, wenn

$$f\left(t + c_i\tau, x + \tau \sum_{j=1}^{s} a_{ij} k_j\right) = f\left(t + \tau \sum_{j=1}^{s} a_{ij}, x + \tau \sum_{j=1}^{s} a_{ij} k_j\right),$$

also genau dann wenn $c_i = \sum_{j=1}^{s} a_{ij}$. QED

Gegen Autonomisierung invariante Runge-Kutta Verfahren bezeichnen wir kurz mit (A, b) und wir schreiben dann auch $\Psi^{\tau}(x) = \Psi^{t+\tau,t}(x)$, da man c von der Matrix A abhängig ist.

Folgende Bedingungen an die Koeffizienten eines Runge-Kutta Verfahrens haben wir bisher erarbeitet:

- Das Verfahren ist genau dann konsistent, wenn $\sum_{i=1}^{s} b_i = 1$, und
- es ist genau dann invariant gegen Autonomisierung, wenn es konsistent ist und $c_i = \sum_{j=1}^{2} a_{ij}$.

Wir wollen nun den ersten der beiden Punkte verallgemeinern und für gegen Autonomisierung invariante Runge-Kutta-Verfahren genauere Forderungen an die Konsistenzordnung stellen. Diese werden als *Ordnungsbedingungen* bezeichnet.

Satz 3.31 *Ein autonomisierungsinvariantes Runge-Kutta-Verfahren besitzt für jede Differentialgleichung mit p-mal stetig differenzierbarer rechter Seite f die Konsistenzordnung*

- $p = 1$, *falls* $\sum_{i=1}^{s} b_i = 1$.
- $p = 2$, *falls zusätzlich* $\sum_{i=1}^{s} b_i c_i = \frac{1}{2}$.
- $p = 3$, *falls zusätzlich* $\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}$ *und* $\sum_{i,j=1}^{s} b_i a_{ij} c_j = \frac{1}{6}$.
- $p = 4$, *falls zusätzlich* $\begin{array}{ll} \sum_{i=1}^{s} b_i c_i^3 = \frac{1}{4} & \sum_{i,j=1}^{s} b_i c_i a_{ij} c_j = \frac{1}{8} \\ \sum_{i,j=1}^{s} b_i a_{ij} c_j^2 = \frac{1}{12} & \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k = \frac{1}{24} \end{array}$.

Beweis: Wir geben nur die Grundstruktur des Beweises an: Das Ziel besteht darin, zu zeigen, dass

$$\|\Psi^\tau(x) - \Phi^\tau(x)\| = \mathrm{O}(\tau^{p+1}) \text{ für } \tau \to 0.$$

Dazu geht man in drei Schritten vor:

1. Taylorentwicklung von der exakten Evolution $g_1(\tau) = \Phi^\tau(x)$ bis zur Ordnung p.
2. Taylorentwicklung von der diskreten Evolution $g_2(\tau) = \Psi^\tau(x)$ bis zur Ordnung p.
3. Koeffizientenvergleich der beiden Taylorentwicklungen.

Der Beweis kann z.B in den Skripten von G. Lube oder von T. Hohage nachgelesen werden.

Betrachten wir nun die Ordnungsbedingungen genauer:

$s = 1$**:** Das Schema für $s = 1$ lautet

$$\begin{array}{c|c} c_1 & a_{11} = 0 \\ \hline & b_1 \end{array}$$

Wegen $c_1 = a_{11} = 0$ folgt aus der geforderten Konsistenzordnung von $p = 1$, dass $b_1 = 1$ gelten muss. Das explizite Euler-Verfahren ist also das einzige einstufige, explizite, autonomisierungsinvariante Verfahren der Ordnung 1.

$s = 2$**:** Das Schema für $s = 2$ lautet

$$\begin{array}{c|cc} c_1 & 0 & \\ c_2 & a_{21} & 0 \\ \hline & b_1 & b_2 \end{array}$$

Wegen der Invarianz gegen Autonomisierung sind $c_1 = 0$ und $c_2 = a_{21}$ bereits festgelegt. Als Variablen verbleiben also a_{21}, b_1, b_2, wobei aber die folgenden Bedingungen beachtet werden müssen:

$$\begin{aligned} b_1 + b_2 &= 1 \\ b_1 \underbrace{c_1}_{=0} + b_2 c_2 &= \tfrac{1}{2}. \end{aligned}$$

Aus dem Gleichungssystem ergibt sich

$$\begin{aligned} b_1 &= 1 - b_2 \\ c_2 &= \frac{1}{2b_2}, \text{ falls } b_2 \neq 0. \end{aligned}$$

(Für $b_2 = 0$ ist das System nicht lösbar.) Man erhält das folgende Butcher-Schema für $b \neq 0$.

$$\begin{array}{c|cc} 0 & 0 & \\ \frac{1}{2b} & \frac{1}{2b} & 0 \\ \hline & 1-b & b \end{array}$$

Für $b = 1$ folgt beispielsweise die explizite Mittelpunktsregel und mit $b = \frac{1}{2}$ die explizite Trapezregel, zu der folgendes Butcher-Schema gehört:

$$\begin{array}{c|cc} 0 & 0 & \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$s = 4$: Für $s = 4$ ergeben sich 10 Unbekannte und 8 Gleichungen, siehe

$$\begin{array}{c|cccc} 0 & 0 & & & \\ c_2 & a_{21} & 0 & & \\ c_3 & a_{31} & a_{32} & 0 & \\ c_4 & a_{41} & a_{42} & a_{43} & 0 \\ \hline & b_1 & b_2 & b_3 & b_4 \end{array}$$

Man kann sich die c_i als Stützstellen der Quadraturformel vorstellen, also für die Simpson-Regel etwa $0, \frac{1}{2}, 1$, was man mit doppelter Stützstelle an $\frac{1}{2}$ als

$$c^t = (0, \tfrac{1}{2}, \tfrac{1}{2}, 1)$$

ausdrücken kann. Eine darauf beruhende Lösung ist das schon vorgestellte *klassische Runge-Kutta Verfahren.*

$s = 10$: In diesem Fall erhält man 1.205 Bedingungen und 55 Variablen.

$s = 20$: Für den Fall $s = 10$ erwarten uns 20.247.374 Bedingungen .

Man erkennt leicht, dass die Anzahl der Bedingungen mit steigendem p immer größer wird.

Beziehung zur numerischen Integration

Wir möchten kurz eine interessante Beziehung zu Kapitel 1 dieses Skriptes erläutern: Man kann die numerische Integration einer Funktion $f \in C([0,1],\mathbb{R})$ auf dem Intervall $[0,1]$ als Spezialfall des folgenden Anfangswertproblemes

$$\begin{aligned} x'(t) &= f(t) \\ x(0) &= 0 \end{aligned}$$

auffassen, denn dessen Lösung ist nach dem Hauptsatz der Differential- und Integralrechnung gegeben durch

$$x(t) = \int_0^t f(\tau)d\tau.$$

Es entspricht also $x(1)$ genau dem gesuchten Integral. Wendet man auf dieses (AWP) ein Runge-Kutta Verfahren an, so erhält man daraus eine Quadraturformel

$$\begin{aligned} \int_0^1 f(\tau)d\tau &= x(1) \approx \Psi^{t_0+\tau,t_0}(x_0) \\ &= \underbrace{x_0}_{=0} + \underbrace{\tau}_{=1} \sum_{j=1}^{s} b_j k_j(t,x,\tau) \\ &= \sum_{i=1}^{s} b_j f(\underbrace{t}_{=0} + c_j \underbrace{\tau}_{=1}) = \sum_{j=1}^{s} b_j f(c_j). \end{aligned}$$

Die jeweils erstgenannten Ordnungsbedingungen aus Satz 3.31 für $p = 1,2,3,4$ entsprechen der Forderung, dass die Monome $1, t, t^2, t^3$ mit Stützstellen c_j und Gewichten b_j exakt integriert werden.

Konvergenz von expliziten Runge-Kutta Verfahren

Bisher haben wir ausschließlich die Konsistenzordnung von Runge-Kutta Verfahren betrachtet. Wir wollen nun die *Konvergenz* der Runge-Kutta Verfahren diskutieren. Auch hierzu benötigen wir in den Voraussetzungen nicht nur die Konsistenz, sondern auch die Stabilität.

Satz 3.32 *Sei $f \in C(D_0,\mathbb{R}^d)$ und genüge der Lipschitz-Bedingung*

$$\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\| \text{ für alle } x_1, x_2 \in D_0.$$

Dann erfüllt die diskrete Evolution Ψ eines gegen Autonomisierung invarianten Runge-Kutta-Verfahrens die Stabilitätsbedingung aus Satz 3.24 mit Konstante $L_\Psi = \gamma L$, wobei $\gamma \geq 0$ nur von A und b abhängt. Ist speziell $p \leq 4$ und sind $b_i, a_{ij} \geq 0$ für alle i,j so ist $\gamma = 1$.

Beweis: Der Satz lässt sich durch wiederholtes Anwenden der Lipschitz-Bedingung im Ausdruck

$$\begin{aligned}\|k_i(t,x,\tau) - k_i(t,\tilde{x},\tau)\| &\leq \|f(x+\tau\sum_j a_{ij}k_j(t,x,\tau)) - f(\tilde{x}+\tau\sum_j a_{ij}k_j(t,\tilde{x},\tau))\| \\ &\leq L(\|x-\tilde{x}\| + \tau\sum_j a_{ij}\|k_j(t,x,\tau) - k_j(t,\tilde{x},\tau)\|)\end{aligned}$$

nachrechnen. Auf Details gegen wir hier nicht ein. QED

3.3.5 Implizite Runge-Kutta-Verfahren

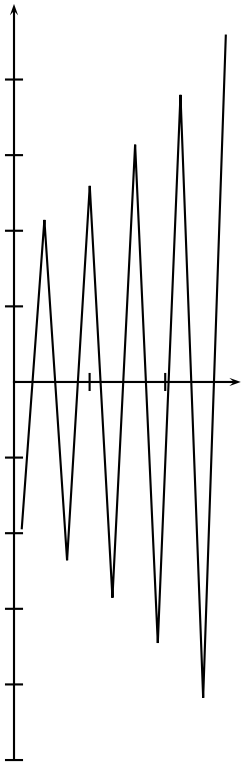
Als „Testproblem“ bekannt ist

$$\begin{aligned}x'(t) &= \lambda x(t) \\ x(0) &= 1\end{aligned}$$

mit Parameter $\lambda \in \mathbb{C}$. Die Lösung ist $x(t) = e^{\lambda x}$. Wir betrachten im Speziellen $\lambda \in \mathbb{R}$. Falls $\lambda < 0$ ist, gilt für $t \to \infty$, dass die Funktion $e^{\lambda t}$ und alle ihre Ableitungen gegen Null konvergieren. Die Hoffnung ist, dass unsere Verfahren schnell konvergieren. Leider ist das nicht so! Das Heun-Verfahren

$$x_\Delta(t_{j+1}) = x_\Delta(t_j) + \tau\lambda x_\Delta(t_j)$$

liefert eine oszillierende, immer weiter ausschlagende Funktion als Lösung.



Euler-Heun Verfahren zu

$$x(t) = -7x(t), \qquad x(0) = 1$$

auf dem Intervall 2 bis 5 mit Schrittweite $h = 0,3$.
Die echte Lösung der DGL ist

$$x(t) = e^{-7t} \approx 0,$$

verläuft also fast entlang der x-Achse. Die Näherung ist unbrauchbar.

Um vernünftige Ergebnisse zu erzielen braucht man *sehr* kleine Schrittweiten. Warum?

Wir erinnern uns an die Idee des Eulerverfahrens: Nutze die Rechteckregel,

$$\int_a^b f(\tau)d\tau \approx (b-a)f(a)$$

um das in der Integralgleichung

$$x(t) = x_0 + \int_a^b f(\tau, x(\tau))d\tau$$

auftretende Integral zu approximieren und erhalte

$$x_\Delta(t_{j+1}) = x_\Delta(t_j) + \tau f(t_j, x_\Delta(t_j)).$$

Benutzen wir stattdessen den rechten Rand des Integrationsintervalls, also

$$\int_a^b f(\tau)d\tau \approx (b-a)f(b),$$

so erhalten wir

$$x_\Delta(t_{j+1}) = x_\Delta(t_j) + \tau f(t_{j+1}, x_\Delta(t_{j+1})),$$

was $e^{\lambda t}$ auch schon für mittlere Schrittweiten recht gut approximiert. Diese Überlegung führt zum *impliziten Euler-Verfahren.*

Graphische Interpretation: Das *explizite* Euler-Verfahren nutzt die Tangente der Lösungskurve im jeweiligen Startpunkt t_j. Das *implizite* Euler-Verfahren nutzt die Tangente der Lösungskurve im jeweiligen Zielpunkt t_{j+1}. Das entspricht dem expliziten Eulerverfahren „von hinten", d.h. mit Startwert t_N.

Wir wenden beide Verfahren auf $f(x) = \lambda x$, $x(0) = 1$ mit $t_j = \tau j$, $j = 0, \ldots, N$ an.

- Explizites Eulerverfahren:

$$x_\Delta(t_{j+1}) = x_\Delta(t_j) + \tau\lambda x_\Delta(t_j).$$

Behauptung: $x_\Delta(t_j) = (1+\lambda\tau)^j$.

Beweis: (Induktion)
$j = 0 \Rightarrow x_\Delta(t_0) = x(0) = 1 = (1+\lambda\tau)^0$
$j \mapsto j+1$:

$$\begin{aligned} x_\Delta(t_{j+1}) &= x_\Delta(t_j) + \tau\lambda x_\Delta(t_j) \\ &= (1+\lambda\tau)^j(1+\lambda\tau) = (1+\lambda\tau)^{j+1}, \end{aligned}$$

woraus die Behauptung folgt.

- Implizites Eulerverfahren

$$x_\Delta(t_{j+1}) = x_\Delta(t_j) + \tau\lambda x_\Delta(t_{j+1}) \Rightarrow x_\Delta(t_{j+1}) = \frac{x_\Delta(t_j)}{1-\lambda\tau}, \qquad \tau\lambda \neq 1.$$

Behauptung: $x_\Delta(t_j) = \left(\frac{1}{1-\lambda\tau}\right)^j$

Beweis: (Induktion)

$j = 0 \Rightarrow x_\Delta(t_0) = x(0) = 1 = \left(\frac{1}{1-\lambda\tau}\right)^0$

$j \mapsto j+1$

$$\begin{aligned} x_\Delta(t_{j+1}) &= \frac{x_\Delta(t_j)}{1-\tau\lambda} = \frac{1}{(1-\lambda\tau)^j}\frac{1}{(1-\tau\lambda)} \\ &= \left(\frac{1}{1-\lambda\tau}\right)^{j+1}, \end{aligned}$$

woraus abermals die Behauptung folgt.

- Zum Vergleich: Die echte Lösung ist $x(t_j) = e^{\lambda t_j}$.

Wir untersuchen unsere Verfahren auf die Eigenschaft $x(t_j) \to 0$ für $j \to \infty$ der echten Lösung $x(t)$ für $\lambda < 0$.

- Im expliziten Euler-Verfahren erhalten wir: $x_\Delta(t_j) \to 0$ falls $|1-\lambda\tau| < 1$. Wegen $|1+\lambda\tau| = |\lambda|\tau - 1$ ist das für $\tau < \frac{2}{|\lambda|}$ erfüllt. Besonders für große λ sind also kleine Schrittweiten erforderlich.

- Im impliziten Eulerverfahren gilt dagegen

$$\left|\frac{1}{1-\lambda\tau}\right| = \frac{1}{|1+|\lambda|\tau|} < 1$$

für alle Schrittweiten τ, also gilt $x_\Delta(t_j) \to 0$ für $j \to \infty$ für jede Schrittweite τ. Das erklärt das bessere Konvergenzverhalten des impliziten Eulerverfahrens.

Bemerkung: Der eben beschriebene Effekt tritt bei dem AWP $x'(t) = \lambda x(t)$, $x(0) = 1$ auch bei allen anderen Runge-Kutta-Verfahren auf, genauer

$$\forall \tau > 0 : \lim_{|\lambda|\to\infty} |\Psi_\lambda^\tau(1)| = \infty,$$

wobei Ψ_λ^τ ein Runge-Kutta-Verfahren zu der Differentialgleichung $f(x) = \lambda x$ ist. (Die Aussage gilt, weil $\Psi_\lambda^\tau(1)$ ein Polynom $\in \Pi_s$ ist.)

Wir erinnern uns daran, dass die exakte Evolution einer Differentialgleichung die Stabilitätsbedingung

$$\|\Phi^{t,t_0}(x_0) - \Phi^{t,t_0}(y_0)\|_2 \leq e^{L_+(t-t_0)}\|x_0 - y_0\|_2$$

erfüllt (Satz 3.18), wobei L_+ die einseitige Lipschitzkonstante ist. Für explizite Runge-Kutta-Verfahren „erbt“ die diskrete Evolution Ψ diese Stabilitätseigenschaften, aber nur mit der Konstanten $L_\Psi = \gamma L$ (Satz 3.32), wobei L die Lipschitzkonstante von f ist. Diese Konstante geht exponentiell in die Fehlerabschätzung aus Satz 3.24

$$\|x_\Delta(t) - x(t)\| \leq r(\tau_\Delta) = \begin{cases} \mathrm{err}(\tau_\Delta)\frac{e^{L_\Psi(t-t_0)}-1}{L_\Psi} & L_\Psi > 0 \\ \mathrm{err}(\tau_\Delta)(t-t_0) & L_\Psi = 0 \end{cases}$$

ein. Daher wäre es gut, wenn $L_\Psi \approx L_+$ gilt.

Das ist bei expliziten Runge-Kutta-Verfahren nicht gegeben, falls $L_+ \ll L$. Solche Differentialgleichungen nennt man **steif**. Für steife Differentialgleichungen liefern explizite Runge-Kutta-Verfahren erst für extrem kleine Schrittweiten verlässliche Ergebnisse und sind daher unbrauchbar. Besser wären Verfahren, bei denen in die Fehlerabschätzung nur die einseitige Lipschitz-Konstante L_+ (und nicht L) einfließt.

Steife Differentialgleichungen treten in der Praxis sehr häufig auf und können (wie in unserem Beispiel) meistens gut mit impliziten Runge-Kutta Verfahren gelöst werden.
Ein s-stufiges implizites Runge-Kutta-Verfahren ist gegeben durch die Vorschrift

$$x_\Delta(t+\tau) := \Psi^{t+\tau,t}(x_\Delta(t)) := x_\Delta(t) + \tau\sum_{j=1}^{s} b_j k_j(t, x_\Delta(t), \tau)$$

mit

$$k_i(t,x,\tau) := f\left(t + c_i\tau, x + \tau\sum_{j=1}^{s} a_{ij}k_j(t,x,\tau)\right).$$

Die Werte c_i nennt man auch **Knoten**, die k_i **Steigungen**.
Das Butcher-Schema lautet:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

Notation 3.33

- *Für $a_{ij} = 0$, $i \leq j$ ergibt* $\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$ *ein* ***explizites*** *Runge-Kutta-Verfahren*
- *Für $a_{ij} = 0$, $i < j$ erhält man ein* ***diagonal-implizites*** *Runge-Kutta-Verfahren (DIRK). Gilt sogar $a_{ii} = y$, so spricht man von $SDIRK$-Verfahren.*
- *Gibt es ein $j > i$ mit $a_{ij} \neq 0$, so nennt man das Runge-Kutta-Verfahren* ***voll implizit****.*

Bei der Implementation von impliziten Runge-Kutta-Verfahren sind in jedem Schritt die Steigungen k_i durch Lösen von

$$k_i(t, x, \tau) = f\left(t + c_i\tau, x + \tau \sum_{j=1}^{s} a_{ij} k_j(t, x, \tau)\right), \qquad i = 1, \ldots, s$$

zu ermitteln. Leider funktionieren Fixpunktiterationen nur mit Schrittweitenbeschränkungen (vgl. Euler-Heun Verfahren, Lemma 3.20 auf Seite 81). Man benutzt daher das Newton-Verfahren (oder Varianten davon).

Wir wollen nun implizite Runge-Kutta Verfahren höherer Ordnung konstruieren.

- Das implizites Euler-Verfahren $\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$ hat Ordnung 1.
- Das Mittelpunktsverfahren

  $$x_\Delta(t_{j+1}) = x_\Delta(t_j) + \tau f\left(t_j + \frac{\tau}{2}, \frac{x_\Delta(t_j) + x_\Delta(t_{j+1})}{2}\right)$$

  mit dem Butcher-Schema $\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$ hat Konsistenzordnung $p = 2$!

Satz 3.34 *Es gelten sinngemäß die Bedingungen für Konsistenz und Invarianz gegen Autonomisierung sowie die Ordnungsbedingungen auch für implizite Runge-Kutta-Verfahren (Lemma 3.28, Lemma 3.30 und Satz 3.31).*

Zum Festlegen der $s^2 + 2s$ Parameter eines impliziten Runge-Kutta-Verfahrens werden häufig *Kollokationsverfahren* verwendet: Die Idee von Kollokationsverfahren ist es, die Lösung eines gegebenen Anfangswertproblemes durch ein Polynom ω zu approximieren. Dieses soll das Anfangswertproblem an vorgegebenen Stützstellen lösen. Als Stützstellen definiert man *Kollokationspunkte* $t_0 + c_i\tau$, $i = 1, \ldots, s$. Dann verlangt man

$$\omega'(t_0 + c_i\tau) = f(t_0 + c_i\tau, \omega(t_0 + c_i\tau)), \qquad i = 1, \ldots, s \tag{3.14}$$

$$\omega(t_0) = x_0 \tag{3.15}$$

für das vektorwertige Polynom $\omega \in (\Pi_s)^n$. Wir nennen die wesentlichen Resultate:

Lemma 3.35 *Seien für* $0 \leq c_1 < \cdots < c_s \leq 1$ *die Bedingungen* (3.14) *und* (3.15) *eindeutig lösbar. Dann wird durch die diskrete Evolution*

$$\Psi^{t_0+\tau,t}(x_0) := \omega(t_0 + \tau)$$

ein implizites Runge-Kutta-Verfahren definiert, das durch die Parameter

$$\begin{aligned} a_{ij} &= \int_0^{c_i} L_j(\tau)d\tau \text{ für } i,j = 1,\ldots,s \\ b_i &= \int_0^1 L_i(\tau)d\tau \text{ für } i = 1,\ldots,s \end{aligned}$$

gegeben ist.

Lemma 3.36 *Ein durch Kollokation definiertes, implizites Runge-Kutta-Verfahren ist konsistent und invariant gegen Autonomisierung.*

Der Beweis dieser Aussagen lässt sich relativ einfach mit den Standardmitteln dieser Vorlesung zu führen. Dagegen ist der folgende Satz ein etwas tiefliegenderes Ergebnis.

Satz 3.37 *Für gegebene Parameter* $c_1,\ldots,c_s$ *sei die Quadraturformel* $\int_0^1 g(t)dt \approx \sum_{i=1}^s b_i g(c_i)$ *exakt für alle Polynome in* Π_{p-1} *mit* $p \geq s$. *Dann hat das zu* $c_1,\ldots,c_s$ *gehörende, durch Kollokation gewonnene Runge-Kutta-Verfahren die Konsistenzordnung* p.

3.4 Zusammenfassung

Begriffe

- DGL: Differentialgleichung
- AWP: Anfangswerproblem = DGL + Startbedingungen
- gewöhnliche/partielle DGL
- explizite, implizite DGL
- autonom
- linear

Transformationen

- jede gewöhnliche, explizite DGL der Ordnung k kann in eine äquivalente DGL erster Ordnung überführt werden, d Gleichungen $\mapsto k \cdot d$ Gleichungen
- Autonomisierung: Eine gewöhnliche, explizite DGL kann man in eine äquivalente, autonome, gewöhnliche DGL überführen

Eindeutigkeit/Lösbarkeit

- Gegenbeispiel für eindeutige Lösbarkeit
- Gegenbeispiel für Existenz einer Lösung auf ganz I

Äquivalenz: AWP $\Leftrightarrow$ Integralgleichung

$$\begin{aligned} x'(t) &= f(t, x(t)) \\ x(t_0) &= x_0 \end{aligned} \qquad \Leftrightarrow \qquad x(t) = x_0 + \int_{t_0}^{t} f(\tau, x(\tau)) d\tau$$

- Anwendung vom Banach'schen Fixpunktsatz
- Konstruktion von Einschrittverfahren
- Picard-Lindelöf: f stetig + Lipschitzstetig bzgl. der letzten d Variablen. Dann ist jedes AWP auf einer Umgebung U um den Startwert eindeutig lösbar.
  - Die Lösung eines (AWP) ist global eindeutig
  - globale Lösbarkeit auf ganz I, falls Lipschitzstetigkeit global

- Folge: Definition der Evolution Φ einer DGL $x' = f(t,x)$ durch
$$\Phi^{t,t_0}(x_0) = x(t),$$
wenn x die eindeutige Lösung von (AWP)
$$x'(t) = f(t,x)$$
$$x(t_0) = x_0$$
- Evolutionen sind durch drei Eigenschaften eindeutig charakterisiert
- Stabilität einer Evolution
$$\|\Phi^{t,t_0}(x_0) - \Phi^{t,t_0}(x)\| \leq e^{L_+(t-t_0)}\|x_0 - x\|$$

Einschritt-Verfahren

- Gitter $\Delta = \{t_0, \ldots, t_N\}$ gesucht:
$$x_\Delta : \Delta \to \mathbb{R}^d$$
$$x_\Delta(t_{j+1}) := \Psi^{t_{j+1},t_j}(x_\Delta(t_j))$$
- explizites Eulerverfahren
- implizites Eulerverfahren
  - Euler-Heun-Verfahren (implizit, sukzessive Approximation)
  - Prädiktor-Korrektor-Variante
- explizites Runge-Kutta-Verfahren
- implizites Runge-Kutta-Verfahren
- Konsistenz von Ψ: drei äquivalente Bedingungen
$$\|\Psi^{t,t_0}(x_0) - \Psi^{t,t_0}(x_0)\| \to 0 \text{ für } t \to t_0$$
- Konvergenz
$$\|x_\Delta(t) - x(t)\| \to 0 \text{ falls } \tau \to 0, \text{ gleichmäßig}$$
- Konsistenz der Ordnung p + Stabilität $\Rightarrow$ Konvergenz der Ordnung p

Explizite Runge-Kutta-Verfahren

- Butcher-Schema
- Bedingung an Konsistenz und an Invarianz gegen Autonomisierung
- implizite Runge-Kutta-Verfahren für steife DGL
- Kollokationsverfahren

Kapitel 4

Optimierung

4.1 Begriffe und Überblick

Notation 4.1 *Sei $\mathcal{B} \subseteq \mathbb{R}^n$ und sei $f : \mathcal{B} \to \mathbb{R}$. Sei weiter $P \subseteq \mathcal{B}$. Ein Optimierungsproblem ist gegeben durch*

$$(P) \qquad \min_{x \in P} f(x).$$

Man nennt f Zielfunktion, $\mathcal{B}$ Grundmenge und P den zulässigen Bereich von (P).
Schreibweise: (P) wird geschrieben als $\min\{f(x) : x \in P\}$ *oder*

$$\begin{array}{ll} \min & f(x) \\ s.d. & x \in P \end{array}.$$

Bemerkung:

- Es gibt auch Optimierungsprobleme, in denen $\mathcal{B} \subseteq \mathbb{R}^n$ nicht gilt, zum Beispiel bei der Bestimmung einer Funktion.
- $\min_{x \in P} f(x)$ ist äquivalent zu $- - \max_{x \in P} -f(x)$, daher können wir uns o.B.d.A. auf Minimierungsprobleme beschränken.
- Da ein Minimum nicht existieren muss, müsste man eigentlich $\inf_{x \in P} f(x)$ schreiben - die Schreibweise mit min hat sich aber eingebürgert.

Notation 4.2 *Sei $\min_{x \in P} f(x)$ ein Optimierungsproblem.*

- *Jedes $x \in P$ heißt* zulässig.
- *Ist $P = \varnothing$, so nennt man das Optimierungsproblem unzulässig.*
- *$x \in P$ heißt* (global) optimal, *falls $f(x) \leq f(x')$ für alle $x' \in P$ gilt.*

- $x \in P$ *heißt* lokal optimal, *falls es eine „vernünftig definierte" Umgebung* $U(x) \subseteq \mathcal{B}$ *so gibt, dass* $f(x) \leq f(x')$ *für alle* $x' \in U(x)$. *Wenn* $\mathcal{B} = \mathbb{R}^n$ *gilt, so kann man immer* $U(x) = \{x' \in \mathbb{R}^n : \|x - x'\| \leq \varepsilon\}$ *mit einer Norm* $\|\cdot\|$ *wählen.*

Beispiel: Ein aus der Vorlesung schon bekanntes Optimierungsproblem ist die in Kapitel 2 behandelte Approximation in endlich-dimensionalen Räumen:

Gegeben: P „einfache Repräsentanten", $x \in X$

gesucht: $y \in P$ so, dass $\|x - y\|$ klein ist, das heißt

$$\min_{y \in P} f(y),$$

wobei $f(y) = \|x - y\|$.

Wir betrachten jetzt systematisch verschiedene Typen von Optimierungsproblemen.

Nicht-Restringierte, differenzierbare Optimierung

Definition: $\mathcal{B} = P = \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$ differenzierbar.

Ergebnisse:

x^* lokal optimal $\Rightarrow \nabla f(x^*) = 0$.

$\nabla f(x^*) = 0$ und die Hesse-Matrix $H(f)(x^*)$ ist positiv definit $\Rightarrow x^*$ ist lokal optimal.

Verfahren: Verfahren des steilsten Abstiegs („steepest descent"), Newton-Verfahren.

Bemerkung: Globale Optima zu finden ist nicht trivial.

Lineare Optimierung

Definition: $\mathcal{B} = \mathbb{R}^n$, $P \subseteq \mathbb{R}^n$ ist ein Polyeder, $f : \mathcal{B} \to \mathbb{R}$ ist linear.

Ergebnisse: Hat (P) eine Lösung, so gibt es eine Ecke von P, die (global) optimal ist.

Verfahren: Simplex-Verfahren (probiert alle Ecken durch), Innere-Punkte-Verfahren.

Bemerkung: Lineare Optimierung ist weitestgehend verstanden; Effizienz-Steigerung ist aber immer noch sinnvoll.

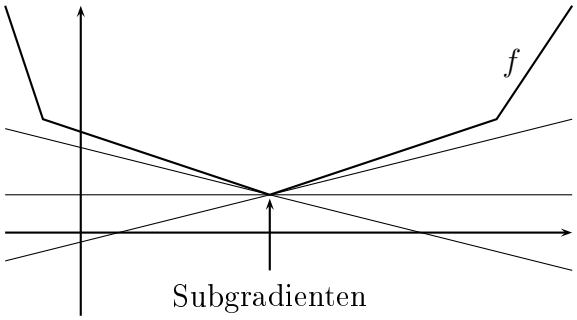
Konvexe Optimierung

Definition: $\mathcal{B} = \mathbb{R}^n$, $P \subseteq \mathbb{R}^n$ konvex, $f : \mathbb{R}^n \to \mathbb{R}$ konvex.

Ergebnisse:

Sei x^* lokales Minimum $\Rightarrow$ x^* ist globales Minimum.

x^* ist (global) optimal auf $\mathbb{R}^n \Leftrightarrow$ Es existiert ein Subgradient $\xi = 0$ an x^*. Auch für $P \subsetneq \mathbb{R}^n$ lassen sich globale Minima durch Subgradienten charakterisieren.



Verfahren: Subgradienten-Verfahren, Volume Algorithmus

Bemerkung: Für spezielle Probleme gibt es effizientere Verfahren.

Konkave Optimierung

Definition: $\mathcal{B} = \mathbb{R}^n$, $P \subseteq \mathbb{R}^n$ konvex, $f : \mathbb{R}^n \to \mathbb{R}$ konkav.

Ergebnisse: Hat (P) eine Lösung, so gibt es einen Extrempunkt von P, der optimal ist. Ist P ein Polyeder, so gibt es eine optimale Ecke. Insbesondere gibt es eine Optimallösung $x^* \in \partial P$.

Verfahren: Auffinden einer endlichen Kandidatenmenge (FDS = finite dominating set).

Ganzzahlige (lineare) Optimierung

Definition: $\mathcal{B} = \mathbb{Z}^n$, $P' \subseteq \mathbb{R}^n$ ist ein Polyeder, $P = P' \cap \mathcal{B}$, $f : \mathbb{R}^n \to \mathbb{R}$ ist linear.

Ergebnisse: Diese liegen vor allem in Spezialfällen vor, zum Beispiel als Ergebnisse im Bereich der Polyeder-Theorie.

Verfahren: Spezialverfahren, welche die Strukturen von P' ausnutzen (TU-Matrizen), ansonsten Gewinnung von oberen Schranken (durch Heuristiken, wie zum Beispiel allgemeine Heuristiken wie Simulated Annealing, genetische Algorithmen, Tabu-Suche) und unteren Schranken (durch Relaxationen).

Bemerkung: Das Problem ist NP-schwer, das heißt ein exaktes Verfahren mit polynomieller Laufzeit ist nicht zu erwarten.

Diskrete Optimierung

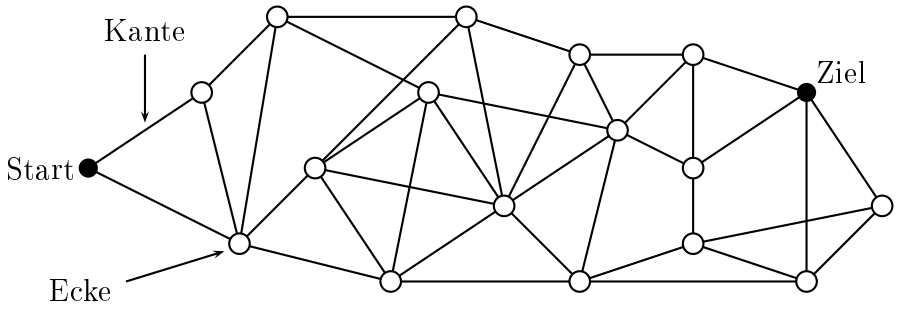
Definition: $\mathcal{B}$ endliche Menge, $P \subseteq \mathcal{B}$ beliebig, $f : \mathcal{B} \to \mathbb{R}$.

Ergebnisse: je nach Problem

Verfahren: je nach Problem oder im Allgemeinen wie Simulated Annealing

Bemerkung: Es gibt effizient lösbare und NP-schwere Probleme.

Beispiel: gegeben ist ein Graph mit Knoten und Kanten, wobei jede Kante eine (positive) Länge hat.



Aufgabe 1: Finde einen kürzesten Weg vom Start zum Ziel.

$\mathcal{B} = \{$alle möglichen Wege vom Start bis zum Ziel$\}$,
$P = \mathcal{B}$,
$f : \mathcal{B} \to \mathbb{R}, \quad f(\text{Weg}) = \text{Länge des Weges} = \sum_{\text{Kanten im Weg}} \text{Länge(Kante)}.$

Dieses Problem ist effizient lösbar in Zeit $\mathcal{O}(n^2)$ (n sei die Anzahl der Knoten im Graph).

Aufgabe 2: Finde den kürzesten Weg vom Start bis zum Ziel, der alle Knoten genau einmal besucht.

$\mathcal{B} = \{$alle möglichen Wege vom Start bis zum Ziel$\}$,
$P \subseteq \mathcal{B}$ enthält die Wege, die alle Knoten genau einmal besuchen
$f : \mathcal{B} \to \mathbb{R}, \quad f(\text{Weg}) = \text{Länge des Weges}.$

Für dieses Problem ist kein effizientes Verfahren bekannt. Es ist schon NP-schwer, herauszufinden, ob $P \neq \varnothing$, d.h. ob es überhaupt einen Weg vom Start bis zum Ziel gibt, der alle Knoten genau einmal besucht.

Die Umgebung eines Weges W kann man z.B. definieren als

$U(W) = \{\text{Wege } W', \text{ die durch Vertauschen von zwei Knoten auf dem Weg } W \text{ entstehen}\}.$

Ein Verfahren, das innerhalb von solchen „benachbarten“ zulässigen Lösungen Elemente einer Lösung paarweise tauscht nennt man auch *zwei-opt*. Das Ergebnis eines zwei-opt Verfahrens ist immerhin lokal optimal.

Kontinuierliche, restringierte Optimierung

Definition: $\mathcal{B} = \mathbb{R}^n$, $P \subseteq \mathbb{R}^n$, $f : \mathcal{B} \to \mathbb{R}$.

Ergebnisse: Diese existieren nicht in dieser Allgemeinheit.

Verfahren: Barriere-Verfahren, Penalty-Verfahren (exakt), allgemeine Heuristiken wie Simulated Annealing

Die genannten Klassen von Optimierungsproblemen sind allerdings keineswegs disjunkt. So lassen sich viele diskrete Probleme als ganzzahlige Programme formulieren, oder auch ganzzahlige Programme als nichtlineare Probleme.

Es soll auch nicht unerwähnt bleiben, dass es noch viele weitere Klassen von Optimierungsproblemen gibt. Darunter fallen u.a. quadratische Optimierungsprobleme, die beispielsweise mit dem Verfahren der *konjugierten Gradienten* gelöst werden können.

4.2 Iterative Optimierungsverfahren

In diesem Abschnitt soll auf einige iterative Verfahren zur Lösung von Optimierungsproblemen eingegangen werden. Dabei betrachten wir zuerst differenzierbare Probleme ohne Nebenbedingungen und stellen das Verfahren des steilsten Abstiegs und (kurz) das Newton-Verfahren vor. Danach diskutieren wir Verfahren, die man auf sehr allgemeine restringierte Probleme

$$\min\{f(x) : x \in P\}$$

anwenden kann, nämlich das Strafverfahren und Simulated Annealing.

4.2.1 Differenzierbare, nicht-restringierte Probleme

In diesem Abschnitt betrachten wir die Minimierung einer differenzierbaren Funktion $f : \mathbb{R}^n \to \mathbb{R}$ über dem gesamten $\mathbb{R}^n$. Wir gehen davon aus, dass uns schon Verfahren zur Minimierung von eindimensionalen Funktionen $f : \mathbb{R} \to \mathbb{R}$ zur Verfügung stehen. Solche Verfahren nennt man „Line Search“ Verfahren, darunter sind zum Beispiel

> Intervallhalbierungsverfahren, Dichotomous-Suche, Verfahren des goldenen Schnitts

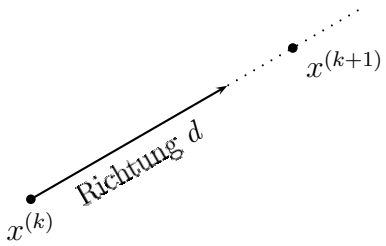
oder, für differenzierbare Funktionen

> Gradienten oder Newton-Verfahren.

Die Methode des steilsten Abstiegs in der mehrdimensionalen Optimierung beruht auf der Idee, eine Lösung $x \in P$ in jedem Schritt entlang einer fest gewählten Richtung d durch Lösen eines eindimensionalen Optimierungsproblems zu verbessern, also durch Lösen des eindimensionalen Problems

$$\min_{\lambda \geq 0} f(x + \lambda d)$$

mit Line Search, wobei f die Zielfunktion darstellt.



Wähle $x^{(k+1)}$ als den besten Punkt entlang der Richtung d. Wir diskutieren zunächst, wie man die Richtung d wählen kann.

Notation 4.3 *Sei $f : \mathbb{R}^n \to \mathbb{R}$ eine Funktion, sei $x \in \mathbb{R}$. Eine Richtung $d \in \mathbb{R}^n$ ist eine* Verbesserungsrichtung *an x bezüglich f, falls es ein $\delta > 0$ so gibt, dass*

$$f(x + \lambda d) < f(x) \quad \textit{für alle } \lambda \in (0, \delta).$$

Das folgende Lemma gibt ein Kriterium, an dem man Verbesserungsrichtungen leicht erkennen kann.

Lemma 4.4 *Sei $f : \mathbb{R}^n \to \mathbb{R}$ eine Funktion, seien $x, d \in \mathbb{R}^n$. Ist die Richtungsableitung $f'(x, d)$ von x in Richtung d echt kleiner als Null, so ist d eine Verbesserungsrichtung.*

Beweis: Es gilt

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

Wegen $f'(x,d) < 0$ gilt also $f'(x+\lambda d) < f(x)$ für alle hinreichend kleinen $\lambda > 0$.

QED

Man kann also jede Richtung d mit negativer Richtungsableitung wählen. Um eine möglichst große Verbesserung zu erzielen, macht es Sinn, eine Richtung d zu wählen, bei der die Richtungsableitung so klein wie möglich ist, also die „Richtung des steilsten Abstiegs“. Das folgende Lemma zeigt, wie man diese Richtung findet. Wir bezeichnen den Gradienten einer Funktion $f : \mathbb{R}^n \to \mathbb{R}$ an $x \in \mathbb{R}^n$ mit $\nabla f(x) \in (\mathbb{R}^n)^*$. Weiterhin sei $\|\cdot\|$ im Folgenden die Euklidische Norm.

Lemma 4.5 *Sei $f : \mathbb{R}^n \to \mathbb{R}$ differenzierbar und sei $\nabla f(x) \neq 0$. Dann ist*

$$\overline{d} = -\frac{(\nabla f(x))^t}{\|\nabla f(x)\|}$$

die normierte Richtung mit kleinster Richtungsableitung, das heißt

$$f'(x,\overline{d}) \leq f'(x,d) \quad \textit{für alle } d \in \mathbb{R}^n \textit{ mit } \|d\| = 1.$$

Beweis: Sei $d \in \mathbb{R}^n$ mit $\|d\| = 1$ beliebig. Dann gilt:

$$\begin{aligned}
|f'(x,d)| &= \left| \lim_{\lambda \to 0^+} \frac{f(x+\lambda d) - f(x)}{\lambda} \right| \\
&= |\nabla f(x) d|, \text{ weil } \lim_{\lambda \to 0} f(x+\lambda d) = f(x) + \lambda \nabla f(x) d \\
&\leq \|\nabla f(x)\| \cdot \|d\| \text{ nach Cauchy-Schwarz} \\
&= \|\nabla f(x)\| = \frac{|\nabla f(x)(\nabla f(x))^t|}{\|\nabla f(x)\|} = |\nabla f(x) \cdot \overline{d}| = |f'(x,\overline{d})|.
\end{aligned}$$

Weiterhin ist $\overline{d}$ eine Abstiegsrichtung wegen

$$f'(x,\overline{d}) = -\frac{\nabla f(x) \cdot (\nabla f(x))^t}{\|\nabla f(x)\|} = -\|\nabla f(x)\| < 0.$$

QED

Algorithmus „Steepest Descent“

Sei $x^{(0)} \in \mathbb{R}^n$ beliebig, $k = 0$.

1. Sei $d^{(k)} := -\nabla f(x^{(k)})$. Falls $\nabla f(x^{(k)}) = 0$, dann STOP.
2. Löse das eindimensionale Optimierungsproblem $\min_{\lambda \geq 0}\{f(x^{(k)} + \lambda d^{(k)})\}$. Sei x^* die Lösung.
3. $x^{(k+1)} := x^*$, gehe zu 1.

Bemerkung: Liegen alle $x^{(k)}$ in einer kompakten Menge, so konvergiert $x^{(k)} \to \overline{x}$ mit $\nabla f(\overline{x}) = 0$. In der Praxis macht das Verfahren in der Nähe des Minimums meistens nur sehr kleine und fast orthogonale Schritte. Man spricht auch von „Zick-Zack-Pfaden“.

Während das Verfahren des steilsten Abstiegs den Gradienten und damit eine lineare Approximation der Funktion f verwendet, nutzt das Newton-Verfahren die quadratische Approximation an die Funktion f. Um einen Punkt x mit $\nabla f(x) = 0$ zu finden, wird f in jedem Schritt durch seine quadratische Approximation ersetzt und eine Nullstelle ihrer Ableitung bestimmt.
Die quadratische Approximation von f an $x^{(k)}$ ist

$$f(x) \approx q(x) = f(x^{(k)}) + \nabla f(x^{(k)})(x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^t H(x^{(k)})(x - x^{(k)}),$$

wobei $H(x^{(k)})$ die Hesse-Matrix von f an $x^{(k)}$ ist. Wegen

$$\nabla q(x) = \nabla f(x^{(k)}) + (H(x^{(k)})(x - x^{(k)}))^t$$

gilt:

$$(\nabla q(x))^t = 0 \Leftrightarrow (\nabla f(x^{(k)}))^t + H(x^{(k)})(x - x^{(k)}) = 0,$$

also falls

$$x = x^{(k)} - (H(x^{(k)}))^{-1}(\nabla f(x^{(k)}))^t.$$

Entsprechend lautet das Verfahren

Newton-Verfahren

Sei $x^{(0)} \in \mathbb{R}^n$ beliebig, $k = 0$.

1. Falls $\nabla f(x^{(k)}) = 0$, dann STOP.
2. Sonst setze $x^{(k+1)} := x^{(k)} - (H(x^{(k)}))^{-1}(\nabla f(x^{(k)}))^t$, $k := k + 1$, gehe zu 1.

Unter gewissen Voraussetzungen kann quadratische Konvergenz gezeigt werden.

4.2.2 Restringierte Probleme

Wir betrachten

$$\min\{f(x) : x \in \mathcal{B}, x \in P\}.$$

Es gibt mehrere Möglichkeiten, die Verfahren aus dem letzten Abschnitt auch zum Lösen von restringierten Problemen zu nutzen. Eine Idee besteht darin, den berechneten Punkt $x^{(k)}$ in jedem Schritt zulässig zu machen, z.B. durch die Projektion von $x^{(k)}$ auf P, d.h. man wählt den Punkt x aus P, der $\|x - x^k\|$ minimiert. Das wird erfolgreich im Subgradienten-Verfahren für konvexe Probleme eingesetzt. Eine andere Variante ist es, unzulässige Lösungen zu bestrafen. Man spricht von **Strafverfahren**. Betrachten wir dazu

$$P = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \ldots, m \text{ und } h_j(x) = 0, j = 1, \ldots, l\}$$

als zulässige Menge unseres Optimierungsproblems. Wie kann man unzulässige Lösungen bestrafen?

Beispiel:

- Das Problem
$$\min f(x), \text{ s.d. } h(x) = 0$$
wird umgewandelt in $\min f(x) + \underbrace{\mu h^2(x)}_{\text{Strafterm}}$, μ groß.

- Das Problem
$$\min f(x), \text{ s.d. } g(x) \leq 0$$
wird umgewandelt in $\min f(x) + \underbrace{\mu(\max\{0, g(x)\})^p}_{\text{Strafterm}}$

Notation: *Sei*

$$(P) \qquad \min\{f(x) : x \in P\} \text{ mit } P \subseteq \mathcal{B}.$$

Dann heißt $\alpha : \mathbb{R}^n \to \mathbb{R}$ ***Straffunktion*** *für* (P)*, falls*

$$\alpha(x) = 0 \text{ für alle } x \in P \text{ und } \alpha(x) > 0 \text{ falls } x \notin P.$$

Das bezüglich α *und* $\mu \geq 0$ *relaxierte Problem ist dann*

$$(P_\mu) \qquad \min\{f(x) + \mu\alpha(x) : x \in \mathcal{B}\}.$$

Weiterhin sei $\theta(\mu) = \inf\{f(x) + \mu\alpha(x) : x \in \mathcal{B}\}$ *für* $\mu \geq 0$ *der Zielfunktionswert von* (P_μ)*.*

Es gilt:

Lemma 4.6

$$\min\{f(x) : x \in P\} \geq \theta(\mu) \text{ für alle } \mu \geq 0 \tag{4.1}$$

Beweis: Sei x^* eine Lösung von P. Dann ist $x^* \in \mathcal{B}$, also für (P_μ) zulässig, und erfüllt

$$f(x^*) = f(x^*) + \mu \underbrace{\alpha(x^*)}_{=0} \geq \min_{x \in \mathcal{B}} f(x) + \mu\alpha(x),$$

also ist die Lösung von (P_μ) mindestens so gut wie x^*. QED

Man kann aber noch mehr zeigen:

Lemma 4.7 *Sei $P \neq \varnothing$ und existiere eine optimale Lösung x_μ von (P_μ) für alle $\mu \geq 0$. Dann gilt:*

- *$\theta(\mu)$ ist monoton wachsend.*
- *$\alpha(X_\mu)$ ist monoton fallend.*
- *$f(X_\mu)$ ist monoton wachsend.*

Beweis: Übung.

Daraus folgt schließlich der folgende Satz:

Satz 4.8 *Sei $P \neq \varnothing$ und existiere eine optimale Lösung x_μ von (P_μ) für alle $\mu \geq 0$ so, dass alle x_μ in einer kompakten Teilmenge von $\mathcal{B}$ enthalten sind. Dann gilt*

$$\min\{f(x) : x \in P\} = \sup_{\mu \geq 0} \theta(\mu) = \lim_{\mu \to \infty} \theta(\mu).$$

Weiter sei $\lambda_k \geq 0$ und $\lambda_k \to \infty$ für $k \to \infty$. Ist $(x_{\lambda_k})_{k \in \mathbb{N}}$ konvergent, dann ist $x := \lim_{k \to \infty} x_{\lambda_k}$ eine optimale Lösung von (P).

Es ergibt sich der folgende Algorithmus:
Sei $\beta > 0$, $\mu_1 > 0$, $k = 1$.

1. Löse
$$(P_{\mu_k}) \qquad \min\{f(x) + \mu_k\alpha(x) : x \in \mathcal{B}\}$$
und erhalte x_{k+1} als optimale Lösung.

2. Falls $\mu_k\alpha(x_{k+1}) < \varepsilon$: x_{k+1} ist zulässig für (P) und nach (4.1) optimal. STOP.
Sonst: $\mu_{k+1} = \beta\mu_k$, $k := k + 1$, gehe zu 1.

Satz 4.8 garantiert Konvergenz zu einer Optimallösung.

Lokale Suche

Hat man bereits eine Lösung $x \in P$ gefunden, besteht die Möglichkeit, diese mittels einer lokalen Suche zu verbessern, um ein lokales Optimum zu erreichen. Dazu sucht man die „Nachbarschaft“ von x ab. Das Verfahren lässt sich auch gut auf diskrete Probleme anwenden.

Beispiel (Nachbarschaften):

- Für
$$\min\{f(x) : x \in P\}, P \subseteq \mathbb{R}^n$$
kann man $N(x) = U_\varepsilon(x) \cap P$ wählen.

- Betrachtet man
$$\min\{f(x) : x \in \{0,1\}^n\}$$
so kann man zum Beispiel

  $N(x) = \{x' \in \{0,1\}^n : x \text{ und } x' \text{ unterscheiden sich nur an höchstens } k \text{ Stellen}\}$

  wählen, wobei k (meistens klein) fest gewählt ist.

- Ist
$$\min\{f(P) : P \text{ Weg in Graph}\},$$
so bietet sich

  $N(P) = \{\text{Wege } P' \text{ die aus } P \text{ durch Vertauschen von zwei Knoten entstehen}\}$

  an.

Für die lokale Suche sei $x \in P$ gegen.

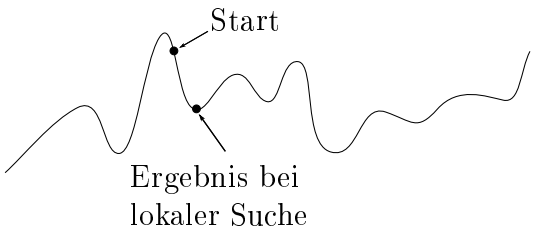
1. Teste, ob es $x' \in N(x)$ mit $f(x') < f(x)$ gibt.
2. Falls ja, setze $x := x'$ und gehe zu 1.

   Sonst: x' lokal optimal, STOP.

Das Verfahren macht Sinn, wenn Schritt 1 leicht zu lösen ist. Oft kann man sogar schnell
$$\min\{f(x') : x' \in N(x)\}$$
lösen, z.B. wenn die Funktion lokal konvex ist, die Mengen $N(x)$ konkave Bereiche sind oder lokale Konvergenz wie beim Newtonverfahren durch Durchprobieren im diskreten Fall vorliegt.

Simulated Annealing

Beim Simulated Annealing versucht man, die lokale Suche so abzuändern, dass man mit hoher Wahrscheinlichkeit ein globales Optimum findet. Man erlaubt dazu auch Schritte, in denen sich der Zielfunktionswert verschlechtert. Dabei soll die Wahrscheinlichkeit für eine Verschlechterung größer sein, wenn die Verschlechterung nur klein ist und im Laufe des Verfahrens abnehmen.



Wir erhalten folgenden Algorithmus:

ALgorithmus: Simulated Annealing
Input: $x \in P$, T_k die „Starttemperatur“, $0 < \alpha < 1$.
Solange T_k groß genug („nicht gefroren“).

1. Wähle zufälliges $x' \in N(x)$.
2. Ist $f(x') < f(x)$, setze $x = x'$ und gehe zu 1.
   Ist $f(x') \geq f(x)$, setze $x := x'$ mit Wahrscheinlichkeit
   $$e^{-\frac{f(x')-f(x)}{T}}.$$
   Setze $T_{k+1} := \alpha T_k$ und gehe zu 1.

Die Idee entstammt chemischen Abkühlungsprozessen, bei denen bei hoher Temperatur eine stabile Molekülbewegung zu beobachten ist, beim Abkühlen aber energieminimale Anordnungen entstehen. Dabei macht das Verfahren nur Sinn, wenn die Nachbarschafts-Definition die folgenden Bedingungen erfüllt:

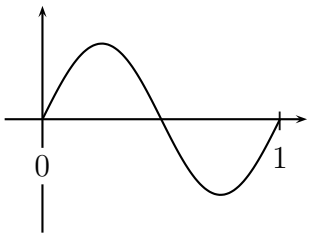
1. $x \in N(x)$, für alle x.
2. $x \in N(x') \Leftrightarrow x' \in N(x)$.
3. Für alle x, x' existiert eine Folge x_k, so dass $x \in N(x_1)$, $x_i \in N(x_{i+1})$, $i = 1, \ldots, k-1$, $x_k \in N(x')$, d.h. jeder Punkt ist von x aus erreichbar.

Kapitel 5

Eigenwertaufgaben

5.1 Motivation

Sei $u(x,t)$ die vertikale Auslenkung einer eingespannten Saite an der Position $x \in [0,1]$ zur Zeit t.



u erfüllt näherungsweise die Wellengleichung

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2}(x,t) &= \frac{1}{c^2} \cdot \frac{\partial^2 u}{\partial x^2}(x,t), \quad x \in (0,1), \quad t \in \mathbb{R} \\ u(0,t) &= u(1,t) = 0, \quad t \in \mathbb{R}, \end{aligned} \tag{5.1}$$

wobei c die Ausbreitungsgeschwindigkeit ist. Wir suchen zeitharmonische Lösungen, das heißt wir machen den Ansatz

$$u(x,t) = \mathrm{Re}(v(x)e^{i\omega t})$$

mit unbekanntem $\omega \in \mathbb{C}$. Einsetzen liefert die gewöhnliche Differentialgleichung

$$\begin{aligned} -v''(x) &= \left(\frac{\omega}{c}\right)^2 v(x), \quad x \in (0,1) \\ v(0) &= v(1) = 0 \end{aligned} \tag{5.2}$$

Das ist ein Eigenwertproblem für den Differentialoperator

$$A : \{u \in C^2([0,1]) | v(0) = v(1) = 0\} \to C([0,1]), \qquad u \mapsto -u''.$$

Diskretisiert man nun dieses Problem, so erhält man ein Matrix-Eigenwertproblem. Betrachten wir hierzu die Gitterpunkte

$$x_j = jh, \quad j = 0, \ldots, N, \qquad h = \frac{1}{N}$$

und approximieren die zweite Ableitung durch den Differenzenquotienten

$$-v''(x_j) \approx \frac{1}{h^2}[-\underbrace{v(x_{j-1})}_{=:v_{j-1}} + 2\underbrace{v(x_j)}_{=:v_j} - \underbrace{v(x_{j+1})}_{=:v_{j+1}}], \quad j = 1, \ldots, N-1.$$

Damit bekommt die Differentialgleichung 5.2 die Form

$$\frac{1}{h^2}(-v_{j-1} + 2v_j - v_{j+1}) = \left(\frac{\omega}{c}\right)^2 v_j, \quad j = 1, \ldots, N-1$$

und man erhält das Problem

$$\frac{c^2}{h^2}\begin{pmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ 0 & & & -1 & 2 \end{pmatrix}\begin{pmatrix} v_1 \\ \vdots \\ \vdots \\ \vdots \\ v_{N-1} \end{pmatrix} = \omega^2 \begin{pmatrix} v_1 \\ \vdots \\ \vdots \\ \vdots \\ v_{N-1} \end{pmatrix}.$$

Insgesamt haben wir also das Problem 5.1 in ein Matrix-Eigenwertproblem überführt.

5.2 Eigenwerte

Definition 5.1 *Sei $A \in \mathbb{R}^{n\times n}$. Eine Zahl $\lambda \in \mathbb{R}$ heißt Eigenwert zum Eigenvektor $x \in \mathbb{R}^n \setminus \{0\}$, falls*

$$Ax = \lambda x$$

gilt.

Die einfachste Berechnung für den Eigenwert λ benutzt das charakteristische Polynom

$$\varphi(\lambda) = \det(A - \lambda \,\mathrm{Id}).$$

Aus AGLA ist bekannt, dass $\varphi \in \Pi_n$ ein Polynom ist, dessen Wurzeln die Eigenwerte von A sind. Verfahren, die das charakteristische Polynom verwenden, heißen *direkte Verfahren* (z.B. Newton-Verfahren auf φ angewendet). Im Allgemeinen ist die Berechnung des charakteristischen Polynoms durch die Determinante jedoch sehr aufwändig, also werden wir im Folgenden Verfahren betrachten, welche die Berechung von φ vermeiden. Diese Verfahren heißen *iterative Verfahren.*
Grundsätzlich gibt es viele verschiedene Aufgabenstellungen:

- Berechnung des größten bzw. kleinsten Eigewertes
- Berechnung aller Eigenwerte
- Berechnung einiger Eigenwerte mit zugehörigen Eigenvektoren
- Berechnung aller Eigenwerte mit zugehörigen Eigenvektoren

In der Vorlesung werden wir die erste und die vierte Aufgabenstellung betrachten und für diese jeweils ein Beispiel angeben.

5.3 Lokalisierungssatz

Satz 5.2 (Lokalisierungssatz) *Ist $\|\cdot\|$ eine zu einer Vektornorm passende Matrixnorm, so gilt für jeden Eigenwert λ von A die Abschätzung*

$$|\lambda| \leq \rho(A) \leq \|A\| \quad \textit{(siehe Numerik I).}$$

Weiterhin gilt der folgenden Satz.

Satz 5.3 (Gerschgorin) *Für $A = (a_{jk}) \in \mathbb{K}^{n\times n}$ definieren wir die* Gerschgorin-Kreise *als*

$$G_j := \left\{ \lambda \in \mathbb{K} \,\middle|\, |\lambda - a_{jj}| \leq \sum_{\substack{k=1 \\ k\neq j}}^{n} |a_{jk}| \right\}, \quad j = 1, \ldots, n$$

und

$$G_k^* := \left\{ \lambda \in \mathbb{K} \,\middle|\, |\lambda - a_{kk}| \leq \sum_{\substack{j=1 \\ j\neq k}}^{n} |a_{jk}| \right\}, \quad k = 1, \ldots, n.$$

Dann gilt für alle Eigenwerte λ von A:

$$\lambda \in \bigcup_{j=1}^{n} G_j \quad \textit{und} \quad \lambda \in \bigcup_{k=1}^{n} G_k^*.$$

Beweis: Sei $Ax = \lambda x$ und $\|x\|_\infty = 1$. Wähle einen Index j mit $|x_j| = \|x\|_\infty = 1$. Dann gilt

$$|\lambda - a_{jj}| = |(\lambda - a_{jj})x_j| = |(Ax)_j - a_{jj}x_j| = \left| \sum_{\substack{k=1 \\ k\neq j}}^{n} a_{jk}x_k \right| \leq \sum_{\substack{k=1 \\ k\neq j}}^{n} |a_{jk}||x_k| \leq \sum_{\substack{k=1 \\ k\neq j}}^{n} |a_{jk}|.$$

Daraus folgt $\lambda \in \bigcup_{j=1}^{n} G_j$. Da A^* die komplex konjugierten Eigenwerte von A besitzt, folgt nun auch $\lambda \in \bigcup_{k=1}^{n} G_k^*$. QED

Im Folgenden wollen wir untersuchen, ob die Eigenwerte von A^* stetig von den Matrixeinträgen abhängen. Zudem werden wir untersuchen, was man über die Lage eines Eigenwertes sagen kann, wenn man „ungefähr“ einen Eigenvektor kennt. Wir werden hier nur den Fall von symmetrischen Matrizen untersuchen.
Die Resultate gelten in ähnlicher Form auch für normale Matrizen ($AA^T = A^TA$). Bei nicht-normalen Matrizen muss man mit extremer Empfindlichkeit der Eigenwerte bei ungenauen Daten rechnen.

Satz 5.4 (Rayleigh) *Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch. Seien $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n$ die Eigenwerte von A mit zugehörigen, orthonormalen Eigenvektoren $x_1, \ldots, x_n$. Sei $V_1 = \mathbb{R}^n$ und $V_j = \{x \in \mathbb{R}^n | x^t x_k = 0$ für alle $1 \leq k \leq j-1\}$. Dann gilt*

$$\lambda_j = \max_{\substack{x\in V_j \\ x\neq 0}} \frac{x^tAx}{x^tx} \text{ für alle } 1 \leq j \leq n.$$

Beweis: Sei $x \in V_j \setminus \{0\}$. Dann lässt sich x schreiben als $x = \sum_{k=j}^n c_k x_k$ mit $c_k = x^t x_k$, da der Raum V_j von den $x_j, \ldots, x_n$ aufgespannt wird und die $x_1, \ldots, x_n$ orthonormal sind. Also gelten $x^tx = \sum_{k=j}^n {c_k}^2$ und $Ax = \sum_{k=j}^n c_k\lambda_k x_k$. Man rechnet nun nach, dass

$$\frac{x^tAx}{x^tx} = \frac{\sum_{k=j}^n {c_k}^2\lambda_k}{\sum_{k=j}^n {c_k}^2} \leq \frac{\lambda_j \sum_{k=j}^n {c_k}^2}{\sum_{k=j}^n {c_k}^2} = \lambda_j.$$

Daraus folgt nun, dass

$$\max_{x\in V_j\setminus\{0\}} \frac{x^tAx}{x^tx} \leq \lambda_j$$

gilt. Für den Eigenvektor x_j zu λ_j gilt die Gleichheit, also wird das Maximum auch angenommen. QED

Satz 5.5 (Courant) *Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und seien $\lambda_1 \geq \ldots \geq \lambda_n$ die Eigenwerte von A. Dann gilt*

$$\lambda_j = \min_{U_j\in M_j} \max_{\substack{x\in U_j \\ x\neq 0}} \underbrace{\frac{x^tAx}{x^tx}}_{\substack{Rayleigh \\ Quotient}} \text{ für alle } 1 \leq j \leq n,$$

wobei M_j die Menge aller $(n+1-j)$-dimensionalen Unterräume von $\mathbb{R}^n$ bezeichnet.

Beweis: Seien $x_1, \ldots, x_n$ orthogonale Eigenvektoren und die V_j wie in Satz 5.4. Aus $V_j \in M_j$ folgt

$$\min_{U_j\in M_j} \max_{x\in U_j\setminus\{0\}} \frac{x^tAx}{x^tx} \leq \lambda_j.$$

Umgekehrt gibt es für jedes $U_j \in M_j$ ein $x \in U_j \setminus \{0\}$ mit $x^tx_k = 0$ für $j+1 \leq k \leq n$. Also wird das Minimum angenommen.

Korollar 5.6 *Seien $A, B \in \mathbb{R}^{n\times n}$ symmetrisch. Seien $\lambda_1(A), \ldots, \lambda_n(A)$ bzw. $\lambda_1(B), \ldots, \lambda_n(B)$ die zu A bzw. B gehörenden Eigenwerte. Dann gilt für jede beliebige natürliche Matrixnorm:*

$$|\lambda_j(A) - \lambda_j(B)| \leq \|A - B\|.$$

Beweis: Übung.
Tip: Zeige $\lambda_j(A) \leq \lambda_j(B) + \|A - B\|$ und verstausche die Rolle von A und B.

5.4 Verfahren von Mises

Sei $A \in \mathbb{R}^{n\times n}$ diagonalisierbar und habe einen dominanten Eigenwert, das heißt es gilt $|\lambda_1| \gg |\lambda_2| \geq \ldots \geq |\lambda_n|$ für einen Eigenwert λ_1. Sei $x_1, \ldots, x_n$ eine Basis aus Eigenvektoren, dann hat jedes $x \in \mathbb{R}^n$ eine eindeutige Darstellung $x = \sum_{j=1}^n \alpha_j x_j$ mit $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$. Nun gilt

$$A^m x = \sum_{j=1}^n \alpha_j A^m x_j = \sum_{j=1}^n \alpha_j \lambda_j x_j = \lambda_1^m \left(\alpha_1 x_1 + \underbrace{\sum_{j=2}^n \alpha_j \left(\frac{\lambda_j}{\lambda_1}\right)^m x_j}_{=:R_m} \right). \tag{5.3}$$

Man erkennt nun, dass $R_m \to 0$ für $m \to \infty$. Also erhalten wir, falls $\alpha_1 \neq 0$, dass

$$\frac{A^m x}{\lambda_1^m} \to \alpha_1 x_1.$$

Das Problem ist jetzt, dass λ_1 unbekannt ist. Außerdem konvergiert $A^m x$ nur für $|\lambda_1| < 1$. Ein Ausweg aus dieser Situation ist, dass man eine andere Normierung vornimmt. Wir betrachten

$$\|A^m x\|_2 = \left(\sum_{j,k=1}^n \alpha_j \alpha_k \lambda_j^m \lambda_k^m x_j{}^t x_k \right)^{\frac{1}{2}} =: |\lambda_1|^m (|\alpha_1| \|x_1\|_2 + r_m). \tag{5.4}$$

mit $\mathbb{R} \ni r_m \to 0$ für $m \to \infty$. Dann folgt

$$\frac{\|A^{m+1}x\|_2}{\|A^m x\|_2} = \frac{\|A^{m+1}x\|}{|\lambda_1|^{m+1}} \cdot \frac{|\lambda_1|^m}{\|A^m x\|} \cdot |\lambda_1| \to |\lambda_1| \text{ für } m \to \infty. \tag{5.5}$$

Definition 5.7 *Bei dem Mises-Verfahren (auch Potenzmethode genannt) wird ein Startvektor $x^{(0)} = \sum_{j=1}^n \alpha_j x_j$, $\alpha_1 \neq 0$ gewählt und $y^{(0)} = \frac{x^{(0)}}{\|x^{(0)}\|}$ gesetzt. Für $m \geq 1$ wird dann definiert*

$$x^{(m)} = A y^{(m-1)}$$

$$y^{(m)} = \frac{\sigma_m x^{(m)}}{\|x^{(m)}\|} \text{ mit } \sigma_m \in \{-1, 1\} \text{ so, dass } y^{(m)t} y^{(m-1)} \geq 0.$$

Dabei bedeutet die Vorzeichenwahl, dass der Winkel zwischen $y^{(m)}$ und $y^{(m-1)}$ im Intervall $[0, \frac{\pi}{2}]$ liegt, also dass es beim Übergang von $y^{(m-1)}$ zu $y^{(m)}$ keinen Sprung gibt. Um $\alpha_1 \neq 0$ müssen wir uns keine Sorgen machen, denn Rundungsfehler stellen die Bedingung meist sicher.

Satz 5.8 (Konvergenzbeweis für von Mises) *Sei $A \in \mathbb{R}^{n\times n}$ diagonalisierbar und habe einen dominanten Eigenwert λ_1, dann gilt:*

- $\|x^{(m)}\| \to |\lambda_1|$ *für* $m \to \infty$,
- $y^{(m)}$ *konvergiert für* $m \to \infty$ *gegen einen Eigenvektor von A zum Eigenwert* λ_1,
- $\sigma^{(m)} \to \operatorname{sign}(\lambda_1)$, *das heißt* $\sigma^{(m)} = \operatorname{sign}(\lambda_1)$ *für m groß genug.*

Beweis: Durch Induktion kann man zeigen, dass

$$y^{(m)} = \sigma^{(m)} \cdots \sigma^{(1)} \cdot \frac{A^{(m)}x^{(0)}}{\|A^{(m)}x^{(0)}\|_2} \text{ für } m = 1, 2, \ldots.$$

Einsetzen ergibt dann

$$x^{(m+1)} = Ay^{(m)} = \sigma^{(m)} \cdots \sigma^{(1)} \cdot \frac{A^{(m+1)}x^{(0)}}{\|A^{(m)}x^{(0)}\|_2}.$$

Aus (5.5) folgt nun, dass

$$\|x^{(m+1)}\|_2 \to |\lambda_1| \text{ für } m \to \infty$$

gilt. Wir nehmen nun ohne Einschränkungen an, dass $\|x_1\|_2 = 1$, dann gilt:

$$\begin{aligned} y^{(m)} &= \sigma^{(m)} \cdots \sigma^{(1)} \cdot \frac{\lambda_1^m(\alpha_1 x_1 + R_m)}{|\lambda_1|^m(|\alpha_1| + r_m)} \text{ mit (5.4) und (5.3)} \\ &= \sigma^{(m)} \cdots \sigma^{(1)} \cdot \operatorname{sign}(\lambda_1)^m \operatorname{sign}(\alpha_1) x_1 + \rho_m, \end{aligned}$$

wobei $\rho_m \to 0$ für $m \to \infty$. Daraus folgt, wenn $\sigma^{(m)}$ konstant ist für große m, dass $y^{(m)}$ gegen einen Eigenvektor von A zum Eigenwert λ_1 konvergiert. Dieses gilt, weil

$$\begin{aligned} 0 \leq y^{(m-1)^t} y^{(m)} &= \sigma^{(m)}\sigma^{(m-1)} \cdots \sigma^{(1)} \cdot \frac{\lambda_1^{2m-1}(\alpha_1 {x_1}^t + {R_{m-1}}^t)(\alpha_1 x_1 + R_m)}{|\lambda_1|^{(2m-1)}(|\alpha_1| + r_{m-1})(|\alpha_1| + r_m)} \text{ mit (5.4) und (5.3)} \\ &= \sigma^{(m)} \operatorname{sign}(\lambda_1) \cdot \underbrace{\frac{{\alpha_1}^2 + \alpha_1 {x_1}^t R_m + \alpha_1 {R_{m-1}}^t x_1 + {R_{m-1}}^t R_m}{|\alpha_1|^2 + |\alpha_1|(r_{m-1} + r_m) + r_{m-1} r_m}}_{\to 1 \text{ für } m\to\infty} \end{aligned}$$

Wielandt-Verfahren (Inverse Iteration, Nachiteration)

Sei A diagonalisierbar und λ_j ein einfacher Eigenwert von A. Sei λ kein Eigenwert von A und eine Näherung an λ_j, das heißt

$$|\lambda - \lambda_j| \ll |\lambda - \lambda_k| \text{ für } k \neq j.$$

Es folgt: $(A - \lambda \,\mathrm{Id})$ ist nichtsingulär und $(A - \lambda \,\mathrm{Id})^{-1}$ hat die Eigenwerte $\tilde{\lambda}_i$ mit $\tilde{\lambda}_i = \frac{1}{\lambda_i - \lambda}$. Also hat die Matrix $(A - \lambda \,\mathrm{Id})^{-1}$ einen dominanten Eigenwert $\tilde{\lambda}_j$ und die von Mises Iteration ist anwendbar.

Jakobiverfahren

Sei $A \in \mathbb{R}^{m \times m}$ symmetrisch. Wir betrachten die Frobenius-Norm

$$\|A\|_{\mathrm{F}} = \left[\sum_{i,j=1}^{n} |a_{ij}|^2 \right]^{\frac{1}{2}}.$$

Lemma 5.9

1. $\|A\|_F = \mathrm{spur}(A^T A) = \mathrm{spur}(A A^T)$
2. $\|A\|_F = \|Q^T A Q\|_F$, *Q orthogonal*

Beweis:

1. Da für die Spur eines Matrixprodukts AB mit $A, B \in \mathbb{R}^{m \times m}$ gilt

$$\mathrm{spur}(AB) = \sum_{i,j=1}^{n} a_{ij} b_{ji} = \sum_{i,j=1}^{n} b_{ij} a_{ji} = \mathrm{spur}(BA),$$

   bekommen wir insbesondere

$$\mathrm{spur}(A^T A) = \mathrm{spur}(A A^T) = \sum_{i,j=1}^{n} |a_{ij}|^2.$$

2. Wir nutzen die Eigenschaft einer orthogonalen Matrix $Q \in \mathbb{R}^{m \times m}$: $Q^{-1} = Q^T$.

$$\begin{aligned} \|Q^T A Q\|_{\mathrm{F}}^2 &= \mathrm{spur}(Q^T A Q Q^T A^T Q) = \mathrm{spur}(Q^T A A^T Q) = \mathrm{spur}(A^T Q Q^T A) \\ &= \mathrm{spur}(A^T A) = \|A\|_{\mathrm{F}}. \end{aligned}$$

QED

Ist $A \in \mathbb{R}^{m\times m}$ symmetrisch, so lässt sich A nach dem Spektralsatz mit einer orthogonalen Transformation auf Diagonalgestalt bringen. Zusammen mit dem Lemma folgern wir

$$\|A\|_{\mathrm{F}}^2 = \sum_{i,j=1}^{n} |a_{ij}|^2 = \sum_{i=1}^{n} |\lambda_i|^2.$$

Definition 5.10 *Eine* Außennorm *ist eine Abbildung*

$$\begin{aligned} N : \mathbb{R}^{m\times m} &\to \mathbb{R} \\ A &\mapsto \sum_{\substack{i,j=1 \\ i\neq j}}^{n} |a_{ij}|^2 = \|A\|_F^2 - \sum_{i=1}^{n} |a_{ii}|^2. \end{aligned}$$

Bemerkung: Die Außennorm ist keine Norm!

Trivialerweise verschwindet die Außennorm für Diagonalmatrizen. Sei a_{ij} ein Nichtdiagonalelement, d.h. $i \neq j$, ungleich Null. Wir betrachten eine Teilmatrix unserer symmetrischen Matrix A, nämlich:

$$\left(\begin{pmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{pmatrix} \right).$$

Wir wollen nur mithilfe von Rotationen die Nichtdiagonalelemente eliminieren.

$$\begin{aligned} \begin{pmatrix} b_{ii} & b_{ij} \\ b_{ij} & b_{jj} \end{pmatrix} &:= \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{pmatrix} \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \\ &= \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} a_{ii}\cos\varphi - a_{ij}\sin\varphi & a_{ij}\cos\varphi - a_{ii}\sin\varphi \\ a_{ij}\cos\varphi + a_{jj}\sin\varphi & a_{jj}\cos\varphi - a_{ij}\sin\varphi \end{pmatrix} \end{aligned}$$

Also lässt sich das transformierte Matrixelement b_{ij} auf folgende Art und Weise berechnen:

$$\begin{aligned} b_{ij} &:= a_{ij}\cos^2\varphi - a_{ii}\cos\varphi\sin\varphi + a_{jj}\sin\varphi\cos\varphi - a_{ij}\sin^2\varphi \\ &= (a_{jj} - a_{ii})\sin\varphi\cos\varphi + a_{ij}(\cos^2\varphi - \sin^2\varphi) \\ &= \tfrac{1}{2}(a_{jj} - a_{ii})\sin 2\varphi + a_{ij}\cos 2\varphi. \end{aligned}$$

Wollen wir es verschwinden lassen, folgt

$$\cot 2\varphi = \frac{a_{ii} - a_{jj}}{2a_{ij}}.$$

Um Kosinus und Sinus als Winkelfunktionen zu vermeiden, definiert man $\tau := \cos(2\varphi) = \cos^2\varphi - \sin^2\varphi$, $\varphi \in [-\pi/4, \pi/4]$. Dann gilt: $\cos\varphi = \sqrt{(1+\tau)/2}$, $\sin\varphi = \sigma\sqrt{(1-\tau)/2}$, $\sigma(\varphi) \in -1, 1$. Sei φ so gewählt, dass

$$a_{ij}\tau + (a_{jj} - a_{ii})\frac{\sigma}{2}\sqrt{1-\tau^2} = 0.$$

Eine Möglichkeit ist

$$\tau = \frac{a_{ii} - a_{jj}}{\sqrt{4a_{ij}^2 + (a_{ii} - a_{jj})^2}}, \qquad \sigma = \operatorname{sign}(a_{ij})$$

wählen. Das Vorzeichen σ ergibt sich wegen des Zählers von τ. Statt der Rotationsmatrix können wir also die Transformationsmatrix

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \quad \text{mit} \quad c = \sqrt{\frac{1+\tau}{2}} \text{ und } s = \sigma\sqrt{\frac{1-\tau}{2}}.$$

verwenden. Wir gehen von der Teilmatrix zur gesamten Matrix über.

Definition 5.11 *Sei $1 \leq i < j \leq n$. Dann nennen wir die Matrix*

$$\begin{pmatrix}
1 & & & & & & & & & \\
& \ddots & & & & & & & & \\
& & 1 & & & & & & & \\
& & & c & & & & -s & & \\
& & & & 1 & & & & & \\
& & & & & \ddots & & & & \\
& & & & & & 1 & & & \\
& & & s & & & & c & & \\
& & & & & & & & 1 & \\
& & & & & & & & & \ddots \\
& & & & & & & & & & 1
\end{pmatrix}$$

mit
$c = \cos\varphi$,
$s = -\sin\varphi$,
$g_{ii} = g_{jj} = \cos\varphi$,
$g_{ij} = -g_{ji} = \sin\varphi$,
ansonsten Identität

eine Givens-Rotation *bzw. eine* Jakobi-Transformation.

Offensichtlich gilt:

Lemma 5.12 *Die* Givens-Rotation *ist orthogonal.*

Satz 5.13 *Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch, $i \neq j$ mit $a_{ij} \neq 0$. Die Matrix*

$$B = G_{ij} A G_{ij}^T$$

1. *ist wieder symmetrisch,*
2. *es gilt $b_{ij} = 0$,*
3. *A und B unterscheiden sich nur in der i-ten bzw. j-ten Spalte/Zeile*
4. *und $N(B) = N(A) - 2a_{ij}^2$.*

Beweis: Aussagen 1-3 folgen aus den bisherigen Überlegungen. Weil die Frobeniusnorm unter orthogonalen Transformationen invariant ist, gilt

$$\left\| \begin{pmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{pmatrix} \right\|_{\mathrm{F}} = \left\| \begin{pmatrix} b_{ii} & b_{ij} \\ b_{ij} & b_{jj} \end{pmatrix} \right\|_{\mathrm{F}}.$$

Über diese Gleichheit und $b_{ij=0}$ erhalten wir durch Quadrieren $a_{ii}^2 + a_{jj}^2 + 2a_{ij}^2 = b_{ii}^2 + b_{jj}^2$. Wir können – da alle anderen Diagonalelemente von A und B gleich bleiben – auf die Außennorm zurückschließen.

$$\begin{aligned} N(B) &= \|B\|_{\mathrm{F}}^2 - \sum_{k=1}^{n} |b_{kk}|^2 = \|A\|_{\mathrm{F}}^2 - \sum_{k=1}^{n} |b_{kk}|^2 \\ &= N(A) + \sum_{k=1}^{n} (|a_{kk}|^2 - |b_{kk}|^2) \\ &= N(A) - 2a_{ij}^2. \end{aligned}$$

QED

Aus diesen Ergebnissen formulieren wir das Jakobi-Verfahren:

Definition 5.14 *Sei $A \in \mathbb{R}^{n \times n}$ symmetrisch. Im* klassischen Jakobi-Verfahren *wird zunächst $A^{(0)} = A$ gesetzt und für $m = 1, 2, \ldots$ iteriert mit $A^{(m)} = (a_{lk}^{(m)})$:*

1. *Suche $i \neq j$ mit $|a_{ij}^{(m)}| = \max_{l \neq k} |a_{lk}^{(m)}|$ und setze $G^{(m)} := G_{ij}$,*
2. *setze $A^{(m+1)} = G^{(m)} A^{(m)} G^{(m)^T}$.*

Das Verfahren sucht also das größte Element aus der Matrix heraus und transformiert es auf Null. Weil wir mit symmetrischen Matrizen arbeiten, müssen wir nur eine obere Dreicksmatrix durchsuchen. Obwohl bei jeder Transformation Nullen wieder verschwinden können, liegt Konvergenz vor.

Satz 5.15 *Das klassische Jakobi-Verfahren konvergiert zumindest linear in der Außennorm.*

Beweis: Wir betrachten ein festes m und erwähnen es daher nicht. Da wir $|a_{ij}| = \max_{l \neq k} |a_{lk}|$ gesetzt haben, können wir damit $N(A)$ abschätzen:

$$N(A) = \sum_{\substack{l,k=1 \\ l \neq k}}^{n} |a_{lk}|^2 \leq n(n-1)|a_{ij}|^2,$$

woraus folgt

$$a_{ij} \geq \frac{N(A)}{n(n-1)}.$$

Jetzt betrachten wir einen Iterationsschritt

$$N(B) = N(A) - 2|a_{ij}|^2 \leq \underbrace{\left(1 - \frac{2}{n(n-1)}\right)^1}_{=:q<1} N(A)$$

und stellen lineare Konvergenz fest, da der Exponent von q gleich 1 ist. QED

Zwar wissen wir nun, dass die Außennormen beim Jakobi-Verfahren gegen Null konvergieren, doch wissen wir nicht, ob dann auf der Diagonalen auch wirklich die Eigenwerte stehen. Dieses Problem wollen wir nun klären:

Korollar 5.16 *Sind $\lambda_1 \geq \cdots \geq \lambda_n$ die Eigenwerte der symmetrischen Matrix $A \in \mathbb{R}^{n\times n}$ und ist $\tilde{a}_{11}^{(m)} \geq \cdots \geq \tilde{a}_{nn}^{(m)}$ eine Umsortierung der Diagonalelemente von $A^{(m)}$, so gilt*

$$|\lambda_i - \tilde{a}_{ii}^{(m)}| \leq \sqrt{N(A^{(m)}} \to 0 \text{ für } m \to \infty.$$

Beweis: Aus Korollar 5.6 mit $A = A^{(m)}$ und $B = \operatorname{diag}(a_{11}^{(m)}, \ldots, a_{nn}^{(m)})$ sowie der euklidischen Norm erhalten wir, da A und $A^{(m)}$ die gleichen Eigenwerte besitzen:

$$|\lambda_i - \tilde{a}_{ii}^{(m)}| = |\lambda_i(A_m) - \lambda_i(B)| \leq \|A^{(m)} - B\|_2 \leq \|A^{(m)} - B\|_{\mathrm{F}} = \sqrt{N(A^{(m)}}.$$

QED

Auf die Eigenvektoren können wir schließen, da sich $A^{(m)}$ schreiben lässt als

$$A^{(m+1)} = G^{(m)} A^{(m)} G^{(m)^T} = \cdots = G^{(m)} \cdot \ldots \cdot G^{(1)} \cdot A \cdot G^{(1)^T} \cdot \ldots \cdot G^{(m)^T} =: Q^{(m)} A Q^{(m)^T},$$

wobei $Q^{(m)}$ orthogonal ist und $A^{(m+1)}$ näherungsweise diagonal. Also bestehen die Zeilen von $Q^{(m)}$ näherungsweise aus Eigenvektoren von A. Es gibt noch weitere Verfeinerungen des Verfahrens:

- Gerade, da das Aufsuchen des Maximums in jedem Schritt mit $n(n-1)$ Vergleichen $\mathcal{O}(n^2)$ wiegt, bei großen Matrizen sehr teuer sein kann. Beispielsweise kann man die Reihenfolge, in der die Paare (i, j) durchlaufen werden, vorher festlegen. Dies nennt man *zyklisches Jakobi-Verfahren.*
- Setzt man zusätzlich einen Schwellenwert, ab dem man sich mit dem a_{ij} zufrieden gibt, spricht man vom *zyklischen Jakobi-Verfahren mit Schwellenwert.*