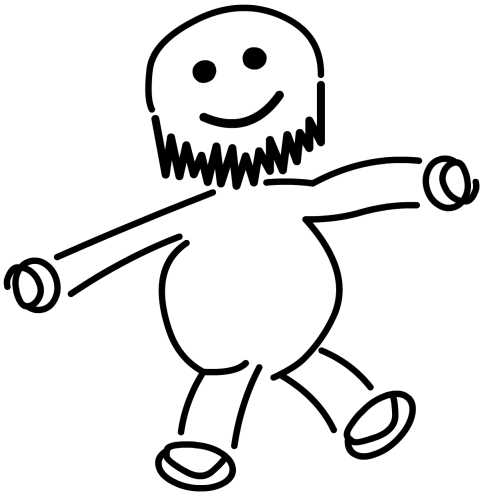
Numerische Mathematik für Physiker



Daniel Scholz und Lothar Nannen im Sommer 2008

Überarbeitete Version vom 23. Januar 2009.

Inhaltsverzeichnis

1 Einleitung **5**

1.1 Bezeichnungen 9

2 Iterative Verfahren **11**

2.1 Sukzessive Approximation 11

2.2 Matrixnorm 15

2.3 Banachscher Fixpunktsatz 20

2.4 Newton-Verfahren 25

2.5 Ausblick 33

3 Lineare Gleichungssysteme **34**

3.1 Notationen und Grundlagen 34

3.2 Gauß-Verfahren und LU-Zerlegung 36

3.3 Kondition von Gleichungssystemen 46

3.4 QR-Zerlegung 49

3.5 Cholesky-Verfahren 52

3.6 Gleichungssysteme mit Tridiagonalmatrizen 56

3.7 Iterative Verfahren für lineare Gleichungssysteme 58

3.8 CG-Verfahren 65

3.9 Ausblick 69

4 Eigenwertaufgaben **71**

4.1 Grundlagen zu Eigenwerten 71

4.2 Lokalisation von Eigenwerten 74

4.3 Vektoriteration 78
4.4 QR-Verfahren 82
4.5 Lanczos-Verfahren 84
4.6 Singulärwertzerlegung 87
4.7 Ausblick 90

5 Interpolation **91**
5.1 Polynominterpolation 91
5.2 Spline-Interpolation 100
5.3 Ausblick 109

6 Numerische Integration **110**
6.1 Interpolationsquadraturen 112
6.2 Gaußsche Quadraturformeln 119
6.3 Ausblick 127

7 Anfangswertprobleme **128**
7.1 Notationen und Grundlagen 128
7.2 Existenz und Eindeutigkeit 134
7.3 Evolutionen 141
7.4 Euler-Verfahren 144
7.5 Konsistenz und Konvergenz 147
7.6 Runge-Kutta-Verfahren 152
7.7 Adaptive Schrittweitensteuerung 167
7.8 Eingebettete Runge-Kutta-Verfahren 172
7.9 Implizite Runge-Kutta-Verfahren 174
7.10 Ausblick 179

8 Randwertprobleme **181**
8.1 Notationen und Grundlagen 181
8.2 Schießverfahren 186
8.3 Methode der finiten Differenzen 188

8.4 Methode der finiten Elemente 193
8.5 Ausblick 198

Literaturverzeichnis **200**

Stichwortverzeichnis **202**

1 Einleitung

Die numerische Mathematik beschäftigt sich mit der Konstruktion und Analyse von Algorithmen für kontinuierliche mathematische Probleme. Das Interesse an numerischen Methoden zu bestimmten Problemen besteht meist aus einem der folgenden Gründe:

(1) Es gibt keine explizite Lösungsdarstellung.

(2) Es gibt zwar eine Lösungsdarstellung, diese ist jedoch nicht geeignet, um die Lösung schnell auszurechnen.

(3) Es gibt zwar eine Lösungsdarstellung, diese liegt aber in einer Form vor, in welcher sich Rechenfehler stark bemerkbar machen.

Damit besteht auch in der Physik außerordentlicher Nutzen in der numerischen Mathematik, wie die folgenden Beispiel zeigen:

(1) Die Navier-Stokes-Gleichung besitzt oft keine explizite Lösungsdarstellung, trotzdem gibt es numerische Verfahren, welche eine Lösung sehr gut approximieren können.

(2) Auch Lösungen der Schrödinger-Gleichung können teilweise nur durch numerische Methoden ausreichend gut beschrieben werden.

(3) Das Schwingverhalten von Saiten einer Gitarre oder des Felles einer Trommel kann auf das Lösen einer Eigenwertaufgabe zurückgeführt werden. Da diese Probleme oft sehr viele Unbekannte haben, werden effiziente numerische Verfahren benötigt.

(4) Die Bahn eines Satelliten im Weltraum kann durch ein System von Differentialgleichungen beschrieben werden. Auch hierzu kann die Existenz einer eindeutigen Bahn gezeigt werden, trotzdem eignen sich nur Mittel der numerischen Mathematik diese tatsächlich zu berechnen.

Auch wenn es bei diesen Beispielen oft um das Lösen von Differentialgleichungen geht, werden für die entsprechenden numerischen Methoden viele

Bereiche der Analysis und Algeba benötigt. Daher werden wir uns in diesem Skript zunächst mit Iterationsverfahren beschäftigen, welche eine Grundlage vieler der späteren Verfahren bieten. Anschließend werden wir uns mit Lösungmethoden von linearen Gleichungssystemen und Eigenwertaufgaben befassen. Schließlich werden wir die numerische Interpolation und Integration besprechen, was uns eine Grundlage zur numerischen Lösung von gewöhnlichen Differentialgleichungen liefern wird.

Bevor wir mit diesem Vorhaben beginnen, wollen wir jedoch noch anhand einiger Beispiele verdeutlichen, dass es *gute* und *weniger gute* numerische Verfahren zum Lösen eines Problemes geben kann.

Beispiel 1.1. *In diesem Beispiel wollen wir $\sqrt{2}$ numerisch bestimmen und betrachten dazu die Gleichung $x^2 = 2$ für $x > 0$.*

Ein Verfahren zur Lösung solcher nichtlinearer Gleichungen, welches wir später noch genauer untersuchen werden, ist das Verfahren der sukzessiven Approximation. Dazu bringen wir die Gleichung in Fixpunktgestalt und erhalten die beiden äquivalenten Formulierungen

$$f_1(x) \;:=\; \frac{1}{x} + \frac{x}{2} \;=\; x \qquad \text{und} \qquad f_2(x) \;:=\; \frac{2}{x} \;=\; x.$$

Beginnen wir mit dem Startwert $x_0 = 5$ und nutzen die Iterationsvorschrift $x_{k+1} = f(x_k)$, so erhalten wir für $f_1(x)$ die Folge

$$5.0 \;\longrightarrow\; 2.7 \;\longrightarrow\; 1.72 \;\longrightarrow\; 1.441 \;\longrightarrow\; 1.415 \;\longrightarrow\; \ldots$$

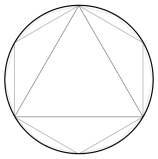
und für $f_2(x)$ die Folge

$$5.0 \;\longrightarrow\; 0.4 \;\longrightarrow\; 5.0 \;\longrightarrow\; 0.4 \;\longrightarrow\; 5.0 \;\longrightarrow\; \ldots .$$

Wir sehen also, dass die zweite äquivalente Darstellung keineswegs geeignet ist $\sqrt{2}$ zu approximieren, die erste Darstellung jedoch schon.

Beispiel 1.2. *Wir betrachten den Einheitskreis in der Ebene und die eingeschriebenen regelmäßigen n-Ecke mit dem Umfang u_n. Der Anschauung entnehmen wir sofort*

$$u_n \;<\; u_{2n} \;<\; 2\pi.$$



Durch einfache geometrische Überlegungen lassen sich zwei Rekursionsformel zur Bestimmung von u_n herleiten, nämlich

$$u_{2n} \;=\; \sqrt{4n \cdot \left(2n - \sqrt{4n^2 - u_n^2}\right)} \;=\; \sqrt{\frac{4n \cdot u_n^2}{\left(2n + \sqrt{4n^2 - u_n^2}\right)}}$$

mit dem Startwert $u_3 = 3\sqrt{3}$. Im folgenden haben wir beide Rekursionsformel in Mathematica 5.0 eingegeben und einige Iterationsschritte berechnen

lassen. Wie Tabelle 1.1 zeigt, liefert die erste Vorschrift bis zur 16. Iteration eine gegen 2π monoton wachsende Folge während sie für spätere Iterationen ein nicht mehr beschreibbares Verhalten zeigt und schließliche den konstanten Wert 0 zurückgibt. Die zweite Vorschrift konvergiert offenbar gegen 2π und zeigt keine derart unschönes Verhalten.

Iteration	erste Vorschrift	zweite vorschrift
1	6.00000	6.00000
2	6.21166	6.21166
3	6.26526	6.26526
4	6.27870	6.27870
⋮	⋮	⋮
15	6.28319	6.28319
16	6.28318	6.28319
⋮	⋮	⋮
26	6.00000	6.28319
27	6.92820	6.28319
28	0.00000	6.28319
⋮	⋮	⋮
40	0.00000	6.28319

Tabelle 1.1: Numerische Berechnung von 2π.

Nochmals sehen wir, dass es stabile und weniger stabile Algorithmen gibt. Obwohl theoretisch beide Rekursionsformel gegen 2π konvergieren, machen sich hier bei der ersten Vorschrift Rechenungenauigkeiten derart bemerktbar, dass die Lösung unbrauchbar wird.

Speziell kommt es in Beispiel 1.2 zur sogenannten ***Auslöschung***. Bei der ersten Iterationsvorschrift werden für große n zwei fast gleich große Zahlen voneinander subtrahiert. Da jede reelle Zahl im Rechner aber nur mit endlich vielen Nachkommastellen genau angezeigt werden kann, werden während der rekursion sehr kleine Zahlen auf 0 gerundet, obwohl ihr eigentlicher Wert durchaus bedeutsam für das exakte Ergebnis wäre. Dieses Problem tritt bei der zweiten Vorschrift nicht auf.

Beispiel 1.3. *Auch bei quadratischen Gleichungen kann es zur Auslöschung kommen. Wird zur Lösung einer Gleichung der Form*

$$x^2 + p \cdot x + q \;=\; 0$$

die pq-Formel

$$x_1 \;=\; -\frac{p}{2} + \sqrt{\left(\frac{p}{2}\right)^2 - q} \qquad \text{und} \qquad x_2 \;=\; -\frac{p}{2} - \sqrt{\left(\frac{p}{2}\right)^2 - q}$$

verwendet, so kann es auch hier vorkommen, dass zwei gleich große Zahlen voneinander subtrahiert werden, was zu numerischen Ungenauigkeiten führen kann. Daher ist es numerisch sinnvoller, die Formel

$$x_1 \;=\; -\frac{p}{2} - \operatorname{sgn}(p) \cdot \sqrt{\left(\frac{p}{2}\right)^2 - q} \qquad \text{und} \qquad x_2 \;=\; \frac{q}{x_1}$$

zu verwenden, welche nach dem Satz von Vieta die gleichen Lösungen liefert.

Beim letzten Beispiel kann es auch beim Term unter der Wurzel zu Problemen kommen. Diese lassen sich jedoch nicht so einfach vermeiden, wie folgende Fehleranalyse zeigt.

Wenn wir annehmen, dass an die Stelle der exakten Werte p und q die fehlerbehafteten Werte $\tilde{p} = p + \varepsilon_p$ und $\tilde{q} = q + \varepsilon_q$ treten, so erhalten wir als Nullstellen

$$\tilde{x}_{1,2} \;=\; f_{1,2}(\tilde{p}, \tilde{q}) \qquad \text{mit} \qquad f_{1,2}(p,q) \;=\; -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}.$$

Eine Taylorentwicklung der Funktion $f(x,y)$ um (p,q) ergibt

$$f_{1,2}(\tilde{p}, \tilde{q}) \;=\; f_{1,2}(p,q) \pm \frac{1}{2\sqrt{\left(\frac{p}{2}\right)^2 - q}} \cdot \left(\frac{p}{2}\varepsilon_p - \varepsilon_q\right) \pm \ldots .$$

Damit ist der Fehler $\Delta x = (\tilde{x}_{1,2} - x_{1,2})$ in erster Näherung

$$\tilde{x}_{1,2} - x_{1,2} \;\approx\; \pm \frac{1}{2\sqrt{\left(\frac{p}{2}\right)^2 - q}} \cdot \left(\frac{p}{2}\varepsilon_p - \varepsilon_q\right).$$

Selbst wenn die Anfangsfehler ε_p und ε_q sehr klein sind, wird der resultierende Fehler in den Nullstellen sehr groß, wenn $\left(\frac{p}{2}\right)^2 \approx q$ gilt. In solchen Fällen nennt man das Problem ***schlecht gestellt***.

Im Gegensatz zur Auslöschung ist diese Eigenschaft mit dem Problem selber und nicht mit dem numerischen Verfahren verbunden. Im Klartext: Gleich welches numerische Verfahren man bei schlecht gestellten Problemen verwendet, relativ kleine Eingangsfehler können zu sehr großen Fehlern in der Lösung führen. Einziger Ausweg ist eine Umformulierung des Problems wie zum Beispiel Regularisierung.

Diese Beispiele verdeutlichen uns, dass mit numerischen Ergebnissen durchaus kritisch umgegangen werden muss. Trotzdem werden wir nun damit beginnen möglichst stabile Algorithmen zur Lösung von Problemen der Algebra und der Analysis herzuleiten.

Weiterhin sei bemerkt, dass große Teile dieser Arbeit den Skripten Schöbel (2006), Schöbel (2007), Lube (2005b), Hohage (2005) und Hohage (2006) entnommen sind, zum Teil wörtlich.

1.1 Bezeichnungen

In diesem Abschnitt wollen wir kurz einige Bezeichnungen einführen, die im folgenden immer wieder auftreten werden.

Da es in vielen Kapiteln egal ist, ob wir uns im Körper der reellen oder der komplexen Zahlen befinden, schreiben wir oft nur $\mathbb{K}$ und meinen damit $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$.

Seien weiter $a_1, \ldots, a_m \in \mathbb{K}^n$ gegeben. Dann bezeichnen wir den kleinsten Unterraum von $\mathbb{K}^n$, der die Vektoren $\{a_1, \ldots, a_m\}$ enthält, mit

$$\operatorname{span}(a_1, \ldots, a_m) := \left\{ \sum_{k=1}^{m} \lambda_k a_k \ : \ \lambda_1, \ldots, \lambda_m \in \mathbb{K} \right\}.$$

Eine Diagonalmatrix mit den Diagonalelementen $d_1, \ldots, d_n$ schreiben wir auch als

$$\operatorname{diag}(d_1, \ldots, d_n) := \begin{pmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{pmatrix}.$$

Matrizen, die nur auf ihrer Diagonalen und auf ihren beiden Nebendiagonalen von Null verschiedene und jeweils identische Einträge haben, bezeichnen wir mit

$$\operatorname{tridiag}(a, b, c) := \begin{pmatrix} b & c & 0 & \cdots & 0 \\ a & b & c & & \vdots \\ 0 & a & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & c \\ 0 & \cdots & 0 & a & b \end{pmatrix}.$$

Geschlossene Intervalle werden gegeben durch $[a, b]$ und offene Intervalle schreiben wir als (a, b).

Der lineare Raum aller stetigen Funktionen auf einem Intervall $[a, b]$ sei $\mathcal{C}([a, b])$. Der Raum aller k-mal stetig differentierbaren Funktionen sei analog $\mathcal{C}^k([a, b])$.

Im folgenden wiederholen wir einige Definitionen, die dem Leser sicherlich schon bekannt sind.

Definition 1.1. Eine Matrix $A \in \mathbb{K}^{n\times n}$ heißt ***hermitesch***, falls

$$A^* \;=\; A$$

gilt. Ist speziell $\mathbb{K} = \mathbb{R}$, so nennt man A auch ***symmetrisch*** und wir schreiben A^T statt A^*.

Dabei ist $A^* = (a^*_{ji})$ die konjugiert komplexe Matrix zu A, es gilt also

$$a^*_{ji} \;=\; \overline{a}_{ij} \qquad \text{für alle} \quad 1 \le i, j \le n.$$

Definition 1.2. Eine Matrix $Q \in \mathbb{K}^{n\times n}$ heißt ***unitär***, falls

$$Q^* \cdot Q \;=\; Q \cdot Q^* \;=\; I$$

gilt. Ist speziell $\mathbb{K} = \mathbb{R}$, so nennt man Q auch ***orthogonal*** und wir schreiben Q^T statt Q^*.

Definition 1.3. Eine hermitesche Matrix $A \in \mathbb{K}^{n\times n}$ heißt ***positiv definit***, falls

$$x^T \cdot A \cdot x \;>\; 0 \qquad \text{für alle} \quad x \in \mathbb{R}^n \setminus \{0\}.$$

A heißt ***positiv semi-definit***, falls

$$x^T \cdot A \cdot x \;\ge\; 0 \qquad \text{für alle} \quad x \in \mathbb{R}^n.$$

Definition 1.4. Eine Menge $U \subset \mathbb{R}^n$ heißt ***konvex***, wenn

$$\lambda \cdot x + (1 - \lambda) \cdot y \;\in\; U$$

für alle $x, y \in U$ und $\lambda \in [0, 1]$.

2 Iterative Verfahren

In diesem Kapitel beginnen wir mit einer der wichtigsten Grundlage für viele numerischen Lösungsansätze, nämliche mit iterativen Verfahren. Wir beginnen sehr einfach mit der sukzessiven Approximation, führen anschließend Normen und Matrixnormen ein, um schließlich den Banachschen Fixpunktsatz genau verstehen und beweisen zu können. Abschließend diskutieren wir noch das mehrdimensional Newton-Verfahen – ein Anwendung der sukzessiven Approximation.

2.1 Sukzessive Approximation

In diesem Abschnitt betrachten die Grundlagen zu einer Klasse von Verfahren, die man zur Lösung von linearen und nichtlinearen Gleichungssystemen verwenden kann. Wir betrachten dazu relativ allgemein Funktionen $f_1, \ldots, f_m$ mit

$$f_i : \mathbb{R}^n \to \mathbb{R} \qquad \text{für} \quad i = 1, \ldots, m$$

und bezeichnen das System

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0 \\ f_2(x_1, \ldots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, \ldots, x_n) &= 0 \end{aligned} \tag{2.1}$$

als ***nichtlineares Gleichungssystem*** mit den Variablen $x_1, \ldots, x_n$. Definieren wir weiter

$$F : \mathbb{R}^n \to \mathbb{R}^m \qquad \text{mit} \qquad F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{pmatrix},$$

so können wir das Gleichungssystem in Kurzform auch schreiben als

$$F(x) = 0.$$

Gilt $F(x) = 0$ für ein $x \in \mathbb{R}^n$, so nennt man x eine Lösung des Gleichungssystems. Dass wir in dem Gleichungssystem die rechte Seite zu Null gesetzt haben ist keine Einschränkung, da ein Gleichungssystem $F(x) = b$ mit $b = (b_1, \ldots, b_m) \in \mathbb{R}^m$ jederzeit zu $G(x) = F(x) - b = 0$ umformt werden kann.

Nichtlineare Gleichungssysteme lassen sich im Allgemeinen nicht durch algebraische Manipulationen exakt auflösen. Wir betrachten daher ***iterative Verfahren***, die eine gegebene Lösung in jedem Schritt verbessern, bis eine vorgegebene Genauigkeit erreicht ist. Dazu untersuchen wir Gleichungssysteme, die in Fixpunktgestalt vorliegen.

Definition 2.1. Gegeben sei $\Phi : \mathbb{R}^n \to \mathbb{R}^n$. Dann heißt die Gleichung

$$\Phi(x) \;=\; x$$

Fixpunktgleichung. Jedes $x \in \mathbb{R}^n$, für welches $\Phi(x) = x$ gilt, wird als ***Fixpunkt*** von Φ bezeichnet.

Ein Zusammenhang zwischen Fixpunktgleichungen und nichtlineares Gleichungssystem wird im folgenden Lemma beschrieben. Diese Aussage wird in Abschnitt 3.7 noch wichtig sein.

Lemma 2.1. *Zunächst sei $m \leq n$ und das Gleichungssystem $F(x) = 0$ wie in Gleichung (2.1) gegeben. Weiter sei $M : \mathbb{R}^m \to \mathbb{R}^n$ eine lineare und injektive Abbildung. Zudem definieren wir*

$$\Phi(x) \;=\; M(F(x)) + x. \tag{2.2}$$

Dann ist x genau dann ein Fixpunkt von Φ, wenn x das Gleichungssystem $F(x) = 0$ löst.

Nun sei andererseits die Abbildung $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ gegeben und wir definieren

$$F(x) \;=\; \Phi(x) - x.$$

Dann ist das Gleichungssystem $F(x) = 0$ äquivalent zu der Fixpunktgleichung $\Phi(x) = x$.

Beweis. Aus $\Phi(x) = x$ folgt $M(F(x)) = 0$. Da M aber injektiv ist, folgt sofort $F(x) = 0$ und damit die erste Aussage.

Die zweite Aussage ist trivial. □

Gleichungssystem $F(x) = 0$ mit $m \leq n$ können wir also lösen, wenn wir Fixpunkte bestimmen können. Damit werden wir uns im folgenden beschäftigen.

Die grundlegende Idee besteht darin zu einem gegebenem $x^{(0)} \in \mathbb{R}^n$ die Folge

$$x^{(k+1)} := \Phi\left(x^{(k)}\right) \qquad \text{für} \qquad k = 0, 1, 2, \ldots$$

zu bestimmen. Angenommen Φ ist stetig und die Folge der $x^{(k)}$ konvergiert, dann gibt es auch einen Grenzwert

$$x^* = \lim_{k \to \infty} x^{(k)}.$$

Für diesen Grenzwert würde dann $x^* = \Phi(x^*)$ gelten, x^* wäre also ein Fixpunkt von Φ. Dies motiviert die folgende Definition.

Definition 2.2. Die Iterationsvorschrift

$$x^{(k+1)} := \Phi\left(x^{(k)}\right) \tag{2.3}$$

heißt ***Verfahren der sukzessiven Approximation***.

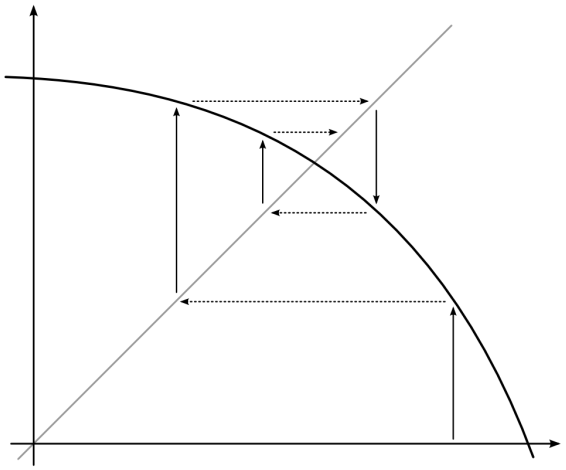


Abbildung 2.1: Das Verfahren der sukzessiven Approximation.

Ähnlich zum ersten Beispiel in der Einleitung können bei der sukzessiven Approximation mehrere Fälle auftreten, wie das folgende Beispiel noch einmal verdeutlicht.

Beispiel 2.1. *Wir betrachten die Gleichung*

$$f(x) = 2x - \tan(x) = 0$$

und formen diese Gleichung in Fixpunktgestalt um.

(1) *In der ersten Variante schreiben wir*

$$\phi(x) \;=\; f(x) + x \;=\; 3x - \tan(x)$$

und suchen einen Fixpunkt von ϕ mittels der Folge

$$x^{(k+1)} \;=\; 3x^{(k)} - \tan(x^{(k)}).$$

(2) *In einer zweiten Variante schreiben wir*

$$x \;=\; \frac{1}{2}\tan(x)$$

und erhalten die Folge

$$x^{(k+1)} \;=\; \frac{1}{2}\tan(x^{(k)}).$$

(3) *Schließlich lässt sich auch*

$$x \;=\; \arctan(2x)$$

verwenden, was uns zur Folge

$$x^{(k+1)} \;=\; \arctan(2x^{(k)})$$

führt.

Verwenden wir in allen drei Fällen den Startwert $x^{(0)} = 1.2$, so konvergiert die Folge aus **(1)** *gegen ∞. Die zweite Folge hingegen konvergiert gegen 0, nur die dritte Vorschrift geht gegen die eigentlich gesuchte Lösung*

$$x^* \approx 1.1656\,.$$

Natürlich besteht unser anliegen auf den nächsten Seiten darin Aussagen zu finden, wann das Verfahren der sukzessiven Approximation gegen eine gesuchte Lösung konvergiert. Dazu betrachten wir zunächst nur den skalaren Fall.

Satz 2.2. *Sei $I \subseteq \mathbb{R}$ ein abgeschlossenes Intervall, $q \in [0,1)$ und $\phi : I \to I$ eine Funktion, die*

$$|\phi(x) - \phi(y)| \;\leq\; q \cdot |x - y| \qquad \text{für alle} \quad x, y \in I \tag{2.4}$$

erfüllt. Besitzt ϕ einen Fixpunkt $x^* \in I$, so konvergiert die Folge

$$x^{(k+1)} \;=\; \phi(x^{(k)})$$

für jeden Startwert $x^{(0)} \in I$ gegen x^* und es gilt

$$|x^{(k)} - x^*| \;\leq\; q^k \cdot |x^{(0)} - x^*| \qquad \text{für} \quad k = 0, 1, 2, \ldots\,.$$

Beweis. Zunächst ist die Iterationsformel für $x^{(k)}$ wohldefiniert, da $x^{(k)} \in I$ für alle k. Die Aussage lässt sich nun leicht durch Induktion zeigen. Der Induktionsanfang für $k = 0$ ist klar. Den Induktionsschritt rechnen wir leicht nach:

$$\begin{aligned} |x^{(k+1)} - x^*| &= |\phi(x^{(k)}) - \phi(x^*)| \leq q \cdot |x^{(k)} - x^*| \\ &\leq q \cdot q^k \cdot |x^{(0)} - x^*| \\ &= q^{(k+1)} \cdot |x^{(0)} - x^*|, \end{aligned}$$

wobei wir im zweiten Schritt die Induktionsannahme verwendet haben. □

Aufgabe 2.1. Verwende Satz 2.2, um alle drei Fälle aus Beispiel 2.1 zu erklären.

Bevor wir Satz 2.2 verallgemeinern können, müssen wir uns mit Matrixnormen beschäftigen.

2.2 Matrixnorm

Zunächst wiederholen wir kurz den Begriff einer Norm.

Definition 2.3. Sei V ein Vektorraum über $\mathbb{K}$. Dann heißt eine Abbildung

$$\|\cdot\| : V \to \mathbb{R}$$

eine ***Norm*** auf V, falls sie die folgenden vier Bedingungen erfüllt:

(1) $\|x\| \geq 0$ für alle $x \in V$.

(2) $\|x\| = 0 \Leftrightarrow x = 0$ für alle $x \in V$.

(3) $\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$ für alle $\alpha \in \mathbb{K}$ und alle $x \in V$.

(4) $\|x + y\| \leq \|x\| + \|y\|$ für alle $x, y \in V$.

Der Raum $(V, \|\cdot\|)$ heißt dann ***normierter Raum***. Weiterhin ist

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

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

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

$$\begin{aligned}
\|x\|_1 &= \sum_{i=1}^{n} |x_i|, \\
\|x\|_\infty &= \max_{i=1,\ldots,n} |x_i|, \\
\|x\|_2 &= \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2} = \sqrt{x^T x}, \\
\|x\|_p &= \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \qquad \text{für} \qquad 1 \leq p \leq \infty.
\end{aligned}$$

Schließlich definieren wir Konvergenz bezüglich einer Norm.

Definition 2.4. Sei $(V, \|\cdot\|)$ ein normierter Raum.

Dann heißt eine Folge $(x_n) \in V$ ***konvergent*** bezüglich der Norm $\|\cdot\|$, falls es ein Element $x^* \in V$ mit der folgenden Eigenschaft gibt:

Für jedem $\varepsilon > 0$ existiert eine natürliche Zahl $N \in \mathbb{N}$, so dass

$$\|x^* - x_n\| \; < \; \varepsilon$$

für alle $n \geq N$.

In diesem Fall nennen wir x^* den ***Grenzwert*** der Folge (x_n). Eine nicht-konvergente Folge heißt ***divergent***.

Die Frage ist nun, in wie weit sich Konvergenz-Definitionen für verschiedene Normen unterscheiden. Dazu ist die folgende Definition hilfreich.

Definition 2.5. Zwei Normen $\|\cdot\|_a$ und $\|\cdot\|_b$ auf einem Vektorraum V heißen ***äquivalent***, wenn es positive reelle Zahlen $c, C > 0$ gibt, so dass

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

für alle $x \in V$ gilt.

Nach dieser Definition gelten die folgenden beiden Aussagen, die wir hier aber nicht beweisen wollen.

Satz 2.3. *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.*

Satz 2.4. *Auf einem endlich-dimensionalen Vektorraum V sind alle Normen äquivalent.*

Damit sind allen folgenden Konvergenzaussagen in endlich dimensionalen Räumen unabhängig von der Norm.

Trotzdem darf nicht vergessen werden, dass auf Räumen unendlicher Dimension nicht alle Normen äquivalent sind. Betrachten wir zum Beispiel den Raum der stetigen Funktionen $\mathcal{C}([a,b])$ über einem Intervall $[a,b]$, so können wir für eine Funktion $f \in \mathcal{C}([a,b])$ die folgenden Normen definieren:

$$\begin{aligned} \|f\|_p &= \left(\int_a^b |f(x)|^p \, \mathrm{d}x \right)^{1/p} \qquad \text{für} \qquad 1 \leq p < \infty, \\ \|f\|_\infty &= \max_{x \in [a,b]} |f(x)|. \end{aligned}$$

Hier sind zum Beispiel $\|f\|_1$ und $\|f\|_\infty$ nicht äquivalent.

Nachdem wir Vektornormen zusammenfassend diskutiert haben, wollen wir nun Normen für Abbildungen und Matrizen definieren.

Definition 2.6. 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

$$\|F(v)\|_W \;\leq\; C \cdot \|v\|_V$$

für alle $v \in V$ gilt.

Wir untersuchen zunächst die Stetigkeit solcher linearen Abbildungen.

Lemma 2.5. *Sei $F : V \to W$ eine lineare Abbildung zwischen normierten Vektorräumen $(V, \|\cdot\|_V)$ und $(W, \|\cdot\|_W)$. Dann ist F genau dann beschränkt, wenn F stetig ist.*

Beweis. Ist F beschränkt, so folgt aus

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

direkt die Stetigkeit von F.

Nun sei F stetig. Dann gibt es zu jedem $\varepsilon > 0$ ein $\delta > 0$, sodass

$$\|F(v) - Fw\|_W \;<\; \varepsilon$$

für alle $v, w \in V$ mit $\|v - w\|_V \leq \delta$. Für $\varepsilon = 1$ und $w = 0$ erhält man wegen $F(0) = 0$ also ein $\delta > 0$ so, dass

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

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

$$\left\| \delta \cdot \frac{v}{\|v\|_V} \right\|_V \;\leq\; \delta \qquad \Rightarrow \qquad \left\| F\left(\delta \cdot \frac{v}{\|v\|_V} \right) \right\|_W \;\leq\; 1$$

und somit folgt

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

Damit ergibt sich die Beschränktheit mit $C = 1/\delta$. □

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

Lemma 2.6. *Sei $F : V \to W$ eine lineare Abbildung zwischen zwei endlich-dimensionalen und normierten Vektorräumen $(V, \| \cdot \|_V)$ und $(W, \| \cdot \|_W)$. Dann ist F beschränkt und stetig.*

Beweis. Sei $\{v_1, \ldots, v_n\}$ eine Basis von V. Dann gilt für

$$v \;=\; \sum_{k=1}^{n} \alpha_k v_k \;\in\; V$$

zunächst

$$F(v) \;=\; F\left(\sum_{k=1}^{n} \alpha_k v_k \right) \sum_{k=1}^{n} \alpha_k F(v_k)$$

und damit folgt

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

wobei beim letzten Schritt ausgenutzt wurde, dass alle Normen auf V äquivalent sind. Damit ist F also beschränkt und nach Lemma 2.5 auch stetig. □

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

Definition 2.7. Sei $F : V \to W$ eine beschränkte und lineare Abbildung zwischen normierten Vektorräumen $(V, \|\cdot\|_V)$ und $(W, \|\cdot\|_W)$. Dann definieren wir die zu $\|\cdot\|_V$ und $\|\cdot\|_W$ ***zugeordnete Norm*** durch

$$\|F\|_{V,W} := \sup_{v \in V \setminus \{0\}} \frac{\|F(v)\|_W}{\|v\|_V} = \sup_{\substack{v \in V \\ \|v\|_V = 1}} \|F(v)\|_W.$$

Gilt $V = W$ und $\|\cdot\|_V = \|\cdot\|_W$, so schreiben wir auch einfach nur $\|F\|_V$.

Weil F als beschränkt vorausgesetzt wurde, gilt

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

für alle $v \in V \setminus \{0\}$ und wir erhalten

$$\|F\|_{V,W} < \infty.$$

Die Norm der Abbildung F ist also die kleinstmögliche Konstante C, mit der man die Beschränktheit der Abbildung abschätzen kann.

Mit diesen allgemeinen Betrachtungen kommen wir nun endlich zu Matrizen zurück.

Satz 2.7. *Wir betrachten die normierten Vektorräume $(\mathbb{K}^n, \|\cdot\|_p)$ und $(\mathbb{K}^m, \|\cdot\|_p)$ mit $p \in \{1, \infty\}$. Weiter sei $A = (a_{ik}) \in \mathbb{K}^{m \times n}$ eine lineare Abbildung von $\mathbb{K}^n$ nach $\mathbb{K}^m$. Dann gilt*

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

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

Die Norm $\|A\|_1$ wird daher als ***Spaltensummennorm*** und die Norm $\|A\|_\infty$ als ***Zeilensummennorm*** bezeichnet.

Der Beweis zu diesen Satz lässt sich durch einfache aber längere Rechnungen führen und kann in jedem Buch zur numerischen Mathematik nachgeschlagen werden, zum Beispiel Hanke-Bourgeois (2006).

Sind $\mathbb{K}^n$ und $\mathbb{K}^m$ mit der selben Norm $\|\cdot\|$ versehen, so sagen wir die Matrixnorm $\|A\|$ ist der Vektornorm $\|\cdot\|$ zugeordnet.

Schließlich wollen wir noch die Matrixnorm zur Euklidischen Norm $\|\cdot\|_2$ für quadratische Matrizen untersuchen und benötigen dafür einige Definitionen.

Definition 2.8. Eine Zahl $\lambda \in \mathbb{K}$ heißt ***Eigenwert*** einer Matrix $A \in \mathbb{K}^{n\times n}$, falls es einen Vektor $x \in \mathbb{K}^n \setminus \{0\}$ gibt mit

$$A \cdot x \;=\; \lambda \cdot x.$$

Ein solcher Vektor x heißt ***Eigenvektor*** von A zum Eigenwert λ.

Definition 2.9. Der ***Spektralradius*** $\rho(A)$ einer Matrix $A \in \mathbb{K}^{n\times n}$ ist der beträgsmäßig größte Eigenwert von A, also

$$\rho(A) \;=\; \max\{|\lambda| \;:\; \lambda \in \mathbb{K} \text{ ist ein Eigenwert von } A\}.$$

Damit können wir schließlich die $\|A\|_2$ Norm berechnen, wobei der Beweis auch hier wieder in Hanke-Bourgeois (2006) nachgeschlagen werden kann.

Satz 2.8. *sei $A = (a_{ik}) \in \mathbb{K}^{m\times n}$ eine lineare Abbildung von $\mathbb{K}^n$ nach $\mathbb{K}^m$. Dann gilt*

$$\|A\|_2 \;=\; \sup_{\substack{x\in\mathbb{K}^n \\ \|x\|_2=1}} \|Ax\|_2 \;=\; \sqrt{\rho(AA^*)}.$$

Ist speziell $A \in \mathbb{K}^{n\times n}$ hermitesch, so folgt

$$\|A\|_2 \;=\; \rho(A).$$

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

2.3 Banachscher Fixpunktsatz

In diesem Abschnitt werden wir die Konvergenzeigenschaften der sukzessiven Approximation weiter untersuchen. Unser Ziel ist eine Verallgemeinerung von Satz 2.2, bei der wir die Existenz eines Fixpunktes nicht mehr voraussetzen müssen. Außerdem wird das neue Ergebnis – der Banachsche Fixpunktsatz – nicht nur für skalare Funktionen ϕ sondern auch für Operatoren Φ in beliebigen Banachräumen X gelten. Damit kann die Unbekannte $x \in X$ nicht nur ein Vektor, sondern sogar eine Funktion sein. Dies wird

später bei der numerischen Untersuchung von Anfangswertproblemen wichtig sein.

Bevor wir den Banachschen Fixpunktsatz formulieren, wiederholen wir noch die Definition eines Banachraumes.

Definition 2.10. Sei $(X, \|\cdot\|)$ ein normierter Raum. Dann heißt eine Folge $(x_n) \in X$ ***Cauchy-Folge*** bezüglich der Norm $\|\cdot\|$, falls es zu jedem $\varepsilon > 0$ eine natürliche Zahl $N \in \mathbb{N}$ gibt, so dass

$$\|x_n - x_m\| \ < \ \varepsilon$$

für alle $n, m \geq N$. Ein Raum, in dem jede Cauchy-Folge zusätzlich auch konvergiert, siehe Definition 2.4, heißt ***vollständig***.

Definition 2.11. Ein vollständiger und normierter Raum $(X, \|\cdot\|)$ heißt ***Banachraum***.

Damit besitzt jede Cauchy-Folge in einem Banachraum einen Grenzwert. Weiterhin übertragen wir auch Gleichung (2.4) allgemein auf normierte Räume.

Definition 2.12. Sei X ein Banachraum 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 $0 \leq q < 1$ gibt, so dass

$$\|\Phi(x) - \Phi(y)\| \ \leq \ q \cdot \|x - y\| \qquad \text{für alle} \quad x, y \in U$$

gilt.

Nun können wir den Banachschen Fixpunktsatz formulieren und beweisen.

Satz 2.9 (Banachscher Fixpunktsatz). *Sei $(X, \|\cdot\|)$ ein Banachraum und $U \subseteq X$ eine abgeschlossene Teilmenge von X. Weiter sei $\Phi : U \to U$ eine Selbstabbildung und zusätzlich sei Φ kontrahierend mit Kontraktionsfaktor $q < 1$. Dann gelten die folgenden Asusagen.*

(1) *Φ besitzt genau einen Fixpunkt $x^* \in U$.*

(2) *Die Iterationsvorschrift der sukzessiven Approximation*

$$x^{(k+1)} \ = \ \Phi\left(x^{(k)}\right)$$

konvergiert für jeden Startwert $x^{(0)} \in U$ gegen x^*.

(3) *Es gilt die* **a priori** *Fehlerschranke*

$$\|x^{(k)} - x^*\| \;\leq\; \frac{q^k}{1-q} \cdot \|x^{(1)} - x^{(0)}\| \qquad \text{für alle} \quad k \in \mathbb{N}. \tag{2.5}$$

(4) *Es gilt die* **a posteriori** *Fehlerschranke*

$$\|x^{(k)} - x^*\| \;\leq\; \frac{q}{1-q} \cdot \|x^{(k)} - x^{(k-1)}\| \qquad \text{für alle} \quad k \in \mathbb{N}. \tag{2.6}$$

Beweis. Zunächst ist die Folge $x^{(k)}$ wohldefiniert, da $x^{(k)} \in U$ für alle $k \in \mathbb{N}$.

Für den Beweis nutzen wir die zentrale Aussage, dass in einem Banachraum alle Cauchy-Folgen konvergieren.

Schritt 1 (Cauchy-Folge) Dazu zeigen wir zunächst, dass $x^{(k)}$ eine Cauchy-Folge ist. Für alle $0 \leq j \leq k-1$ gilt

$$\begin{aligned} \|x^{(k)} - x^{(k-1)}\| &= \|\Phi(x^{(k-1)}) - \Phi(x^{(k-2)})\| \\ &\leq q\|x^{(k-1)} - x^{(k-2)}\| \;\leq\; \ldots \;\leq \\ &\leq q^j\|x^{(k-j)} - x^{(k-j-1)}\|. \end{aligned}$$

und somit erhalten wir

$$\begin{aligned} & \|x^{(l)} - x^{(k)}\| \\ \leq\; & \|x^{(l)} - x^{(l-1)}\| + \|x^{(l-1)} - x^{(l-2)}\| + \ldots + \|x^{(k+1)} - x^{(k)}\| \\ \leq\; & q^{l-k}\|x^{(k)} - x^{(k-1)}\| + q^{l-k-1}\|x^{(k)} - x^{(k-1)}\| + \ldots + q\|x^{(k)} - x^{(k-1)}\| \\ =\; & \|x^{(k)} - x^{(k-1)}\| \cdot \sum_{j=1}^{l-k} q^j \;\leq\; \|x^{(k)} - x^{(k-1)}\| \cdot \sum_{j=1}^{\infty} q^j \\ =\; & \|x^{(k)} - x^{(k-1)}\| \cdot \frac{q}{1-q} \qquad (2.7) \\ \leq\; & q^{k-1}\|x^{(1)} - x^{(0)}\| \cdot \frac{q}{1-q} \;=\; \frac{q^k}{1-q} \cdot \|x^{(1)} - x^{(0)}\|. \qquad (2.8) \end{aligned}$$

Da $q^k/(1-q) \to 0$ für $k \to \infty$, ist $x^{(k)}$ eine Cauchy-Folge.

Schritt 2 (Existenz des Fixpunktes) Weil $x^{(k)}$ eine Cauchy-Folge ist, gibt es ein x^* mit

$$x^* \;=\; \lim_{k\to\infty} x^{(k)}.$$

Für x^* gilt dann aber

$$\|\Phi(x^*) - \Phi(x^{(k)})\| \;\leq\; q \cdot \|x^* - x^{(k)}\| \qquad \text{für} \quad k \to \infty,$$

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 x^* $\tilde{x}$ und seien zwei Fixpunkt von Φ. Dann gilt

$$\|x^* - \tilde{x}\| \;=\; \|\Phi(x^*) - \Phi(\tilde{x})\| \;\leq\; q \cdot \|x^* - \tilde{x}\|.$$

Da $q < 1$, folgt $\|x^* - \tilde{x}\| = 0$ und somit $x^* = \tilde{x}$.

Schritt 4 (Fehlerschranken) Nutzen wir die in Schritt 1 aufgestellte Ungleichungskette und speziell die Gleichungen (2.7) und (2.8), so erhalten wir

$$\begin{aligned} \|x^* - x^{(k)}\| &= \lim_{l\to\infty} \|x^{(l)} - x^{(k)}\| \\ &\leq \frac{q}{1-q} \cdot \|x^{(k)} - x^{(k-1)}\| \\ &\leq \frac{q^k}{1-q} \cdot \|x^{(1)} - x^{(0)}\|. \end{aligned}$$

Damit ist der Satz gezeigt. □

Die Bedeutung des Banachschen Fixpunktsatzes für die numerische Mathematik liegt im konstruktiven Charakter seines Beweises. Neben dem Nachweis der Existenz und der Eindeutigkeit eines Fixpunktes beschreibt er gleichzeitig mit dem Verfahren der sukzessiven Approximation einen einfachen Algorithmus zur näherungsweisen Bestimmung des Fixpunktes. Zudem sollte man nie vergessen, dass der Banachsche Fixpunktsatz auch in unendlich-dimensionalen Banachräumen gilt.

Weiterhin bemerken wir, dass die Umgebung U auch sehr klein werden kann.

Wir beschäftigen uns nun noch mit dem Nachweis der Kontraktionseigenschaft von Φ. Dazu bezeichnen wir für eine total differenzierbare Funktion $\Phi = (\Phi_1, \ldots, \Phi_n) : \mathbb{R}^n \to \mathbb{R}^m$ die ***Jacobi-Matrix*** von Φ an einer Stelle $a \in \mathbb{R}^n$ mit

$$D\Phi(a) \;=\; \begin{pmatrix} \frac{\partial \Phi_1}{\partial x_1}(a) & \ldots & \frac{\partial \Phi_1}{\partial x_n}(a) \\ \vdots & & \vdots \\ \frac{\partial \Phi_m}{\partial x_1}(a) & \ldots & \frac{\partial \Phi_m}{\partial x_n}(a) \end{pmatrix}.$$

Für $\Phi : \mathbb{R}^n \to \mathbb{R}$ schreiben wir auch

$$\operatorname{grad} \Phi(a) \;=\; \left(\frac{\partial \Phi}{\partial x_1}(a), \ldots, \frac{\partial \Phi}{\partial x_n}(a) \right)$$

und für eine skalare Funktion $\phi : \mathbb{R} \to \mathbb{R}$ wie gehabt einfach nur $f'(a)$.

Lemma 2.10. *Sei $U \subseteq \mathbb{R}^n$ eine konvexe Menge und $\Phi : U \to \mathbb{R}^n$ stetig differenzierbar mit*

$$\|D\Phi(x)\| \;\leq\; q \;<\; 1 \qquad \text{für alle} \qquad x \in U,$$

wobei die Matrixnorm hier einer Vektornorm zugeordnete sein soll.

Dann ist Φ kontrahierend mit Kontraktionsfaktor q.

Beweis. Zu zwei festen $x, y \in U$ definieren wir eine Abbildung $f : \mathbb{R} \to \mathbb{R}^n$ durch

$$f(t) \;=\; \Phi(x + t(y - x)) \qquad \text{für} \qquad t \in [0, 1].$$

Damit gilt nach dme Hauptsatz der Differential- und Integralrechnung und der Kettenregel

$$\begin{aligned}
\|\Phi(y) - \Phi(x)\| \;&=\; \|f(1) - f(0)\| \;=\; \left\| \int_0^1 f'(t)\,\mathrm{d}t \right\| \\
&=\; \left\| \int_0^1 D\Phi(x + t(y - x)) \cdot (y - x)\,\mathrm{d}t \right\| \\
&\leq\; \int_0^1 \|D\Phi(x + t(y - x))\| \cdot \|y - x\|\,\mathrm{d}t \\
&\leq\; \|y - x\| \cdot \int_0^1 q\,\mathrm{d}t \;=\; q \cdot \|x - y\|.
\end{aligned}$$

Somit ist die Kontraktionseigenschaft gezeigt □

Wollen wir den Banachschen Fixpunktsatz anwenden und nach Lemma 2.10 zeigen, dass Φ kontrahierend ist, so müssen wir für $\|D\Phi(x)\|$ die Matrixnom verwenden, die der Vektornorm des Banachraumes zugeordnet ist. Damit muss die Wahl der Norm bedacht werden, um mittels Lemma 2.10 zu zeigen, dass Φ eine Kontraktion ist.

Abschließend untersuchen wir noch, wie wir bei der Approximation des Fixpunktes eine Genauigkeit von $\varepsilon > 0$ garantieren können, also

$$\|x^{(k)} - x^*\| \;\leq\; \varepsilon$$

in der k-ten Iteration. Dazu können wir während des Verfahrens die a posteriori Fehlerschranke aus dem Banachschen Fixpunktsatz verwenden, also

$$\|x^{(k)} - x^*\| \;\leq\; \frac{q}{1 - q} \cdot \|x^{(k)} - x^{(k-1)}\|.$$

Dies führt zum Abbruchkriterium

$$\frac{q}{1-q} \cdot \|x^{(k)} - x^{(k-1)}\| \ \leq \ \varepsilon.$$

Leider ist der Kontraktionsfaktor q oft nicht bekannt und man behilft sich mit der Abschätzung

$$\hat{q}_k \ = \ \frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k-1)} - x^{(k-2)}\|}.$$

Damit gilt $\hat{q}_k \leq q$, denn

$$\hat{q}_k \ = \ \frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k-1)} - x^{(k-2)}\|} \ = \ \frac{\|\Phi(x^{(k-1)}) - \Phi(x^{(k-2)})\|}{\|x^{(k-1)} - x^{(k-2)}\|} \ \leq \ q.$$

Somit brechen wir das Verfahren der sukzessiven Approximation ab, wenn

$$\frac{\hat{q}_k}{1-\hat{q}_k} \cdot \|x^{(k)} - x^{(k-1)}\| \ \leq \ \varepsilon$$

gilt.

Es bleibt zu bemerken, dass es sich hierbei um ein heuristisches Abbruchkriterium handel, da die Fehlerschranke $\|x^{(k)} - x^*\| \leq \varepsilon$ im Allgemeinen nicht garantiert werden kann, was an der Abschätzung

$$\frac{\hat{q}_k}{1-\hat{q}_k} \ \leq \ \frac{q}{1-q}$$

liegt. Meistens konvergiert $\hat{q}_k$ jedoch gegen q, sodass das Abbruchkriterium in der Regel ausreichend gut funktioniert.

2.4 Newton-Verfahren

Mit dem Verfahren der sukzessiven Approximation und dem Banachschen Fixpunktsatz haben wir eine Methode zur Nullstellenbestimmung von nichtlinearen Gleichungssystemen kennengelernt, sofern gewissen Konvergenzkriterien erfüllt sind. Wir haben aber bereits auch bemerkt, dass das Verfahren sehr langsam konvergieren kann, wenn der Kontraktionsfaktor nahe an 1 liegt. Daher wollen wir nun ein weiteres Verfahren vorstellen, welches wieder unter gewissen Voraussetzung eine bessere Konvergenz liefert.

Dazu betrachten wir zunächst den skalaren Fall und kehren anschließend zum mehrdimensionalen Fall zurück. Wir beginnen somit mit einer reellen Funktion $f : \mathbb{R} \to \mathbb{R}$.

Gesucht ist eine Nullstelle x^* von f. Angenommen wir haben schon eine Schätzung der Nullstelle $x^{(0)}$ und die Funktion f ist stetig differenzierbar, so besteht die Idee des Newton-Verfahrens darin, f durch seine Tangente

$$f(x) \approx f(x^{(0)}) + f'(x^{(0)})(x - x^{(0)})$$

durch den Punkt $x^{(0)}$ zu ersetzen. Dies entspricht damit auch der linearen Taylorentwicklung in $x^{(0)}$.

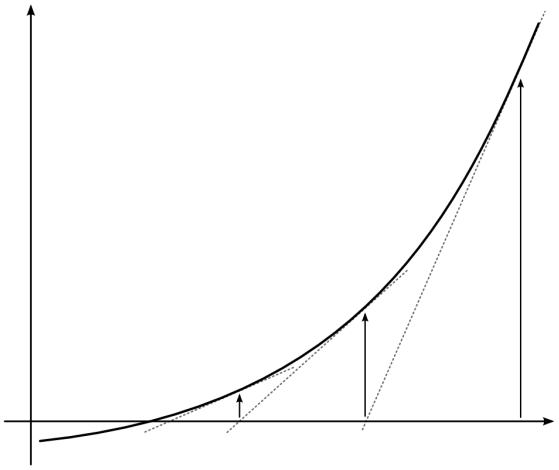


Abbildung 2.2: Das skalare Newton-Verfahren.

Wir approximieren nun die gesuchte Nullstelle x^* durch die Nullstelle der linearen Näherung, also

$$f(x^{(0)}) + f'(x^{(0)})(x - x^{(0)}) = 0 \qquad \Leftrightarrow \qquad x = -\frac{f(x^{(0)})}{f'(x^{(0)})} + x^{(0)},$$

falls $f'(x^{(0)}) \neq 0$. Wir gehen nun davon aus, dass diese Nullstelle eine bessere Approximation von x^* ist als $x^{(0)}$ und setzen daher

$$x^{(1)} = x^{(0)} - \frac{f(x^{(0)})}{f'(x^{(0)})}.$$

Wiederholen wir dieses Vorgehen, so erhalten wir das ***Newton-Verfahren***. Für $f'(x) \neq 0$ können wir somit die Fixpunktgleichung

$$g(x) = x - \frac{f(x)}{f'(x)}$$

definieren und erhalten das Iterationsverfahren $x^{(k+1)} = g(x^{(k)})$. Dazu gilt die folgende theoretische Konvergenzaussage.

Satz 2.11. *Sei x^* eine einfache Nullstelle einer Funktion $f : \mathbb{R} \to \mathbb{R}$. Weiterhin sei f in einer Umgebung von x^* zweimal stetig differenzierbar. Dann konvergiert das Newton-Verfahren für jeden Startwert $x^{(0)}$, der hinreichend nahe bei x^* liegt.*

Beweis. Da x^* eine einfache Nullstelle von f ist, gilt $f'(x^*) \neq 0$ und entsprechend gibt es eine Umgebung $U := U(x^*)$, sodass $f'(x) \neq 0$ für alle $x \in U$. Die Verfahrensvorschrift $x^{(k+1)} = g(x^{(k)})$ mit

$$g(x) \;=\; x - \frac{f(x)}{f'(x)}$$

ist somit für alle $x \in U$ definiert. Die Ableitung von g ist

$$g'(x) \;=\; 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} \;=\; \frac{f''(x)}{(f'(x))^2} \cdot f(x),$$

und damit folgt $g'(x^*) = 0$. Aufgrund der Stetigkeit von g' existieren reelle Zahlen $\delta > 0$ und $q < 1$ mit

$$|g'(x)| \;\leq\; q \;<\; 1$$

für alle $x \in U' = [x^* - \delta, x^* + \delta] \cap U$. Daraus folgt

$$|g(x) - x^*| \;=\; |g(x) - g(x^*)| \;\leq\; q \cdot |x - x^*| \;\leq\; \delta \qquad \text{für alle} \quad x \in U',$$

also ist g eine Kontraktion sowie eine Selbstabbildung auf U'. Der Banachsche Fixpunktsatz liefert schließlich die Behauptung. □

Damit kommen wir nun zum mehrdimensionalen Fall mit $F : \mathbb{R}^n \to \mathbb{R}^n$.

Die Motivation hierfür ist die gleiche wie für skalare Funktionen. Anstatt die Nullstelle $F(x) = 0$ einer stetig differenzierbaren Funktion $F : \mathbb{R}^n \to \mathbb{R}^n$ zu suchen, nutzen wir die lineare Taylorentwicklung

$$F(x) \;\approx\; F(x^{(0)}) + (DF(x^{(0)})) \cdot (x - x^{(0)}).$$

Existiert $(DF(x^{(0)}))^{-1}$, so können wir diese Gleichung nach x auflösen und erhalten

$$x^{(1)} \;=\; x^{(0)} - (DF(x^{(0)}))^{-1} \cdot F(x^{(0)})$$

als nächste Iterierte. Das entspricht der Fixpunktgleichung (2.2),

$$G(x) \;=\; x + (DF(x))^{-1} \cdot F(x),$$

und somit erhalten wir das Iterationsverfahren $x^{(k+1)} = G(x^{(k)})$. Wir fassen das Ergebnis noch einmal zusammen.

Definition 2.13. 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 Jacobi-Matrix $DF(x)$. Dann heißt das Verfahren

$$x^{(k+1)} \;=\; x^{(k)} - (DF(x^{(k)}))^{-1} \cdot F(x^{(k)})$$

mit einem Startwert $x^{(0)} \in U$ ***Newton-Verfahren***.

Um das Newton-Verfahren numerisch zu realisieren, wird zur Bestimmung von $x^{(k+1)}$ in jedem Schritt das lineare Gleichungssystem

$$DF(x^{(k)}) \cdot (x^{(k+1)} - x^{(k)}) \;=\; -\,F(x^{(k)})$$

gelöst. Dies geschieht durch das Lösen des Systems

$$DF(x^{(k)}) \cdot w^{(k)} \;=\; -\,F(x^{(k)})$$

und anschließendes Berechnen von

$$x^{(k+1)} \;=\; x^{(k)} + w^{(k)}.$$

Da die Berechnung der Jacobi-Matrix für große n sehr aufwändig wird und um den Konvergenzbereich des Newton-Verfahrens zu vergrößern, wird in der Praxis oft eine der folgenden Varianten verwendet:

(1) ***Frozen Newton***. Es wird nur einmal die Jacobi-Matrix berechnet, für die dann mittels LU-Zerlegung (siehe nächstes Kapitel) alle in den Iterationen auftretende Gleichungssysteme effizient lösbar sind.

(2) ***Quasi Newton***. Die Jacobi-Matrix wird in jedem Schritt approximativ angepasst.

(3) ***Gedämpftes Newton***. Es wir die Iterationsvorschrift

$$x^{(k+1)} \;=\; x^{(k)} + \lambda_k \cdot w^{(k)}$$

mit $\lambda_k \in [0, 1]$ verwendet.

Das Konvergenzverhalten des mehrdimensionalen Newton-Verfahrens lässt sich nicht so einfach analysieren wie im eindimensionalen Fall. Trotzdem beweisen wir im folgenden Satz sogar mehr, nämlich dass das Newton-Verfahren quadratisch konvergiert. Dazu die folgende Definition.

Definition 2.14. Eine gegen x^* konvergente Folge $(x^{(k)})$ in einem normierten Raum $(X, \|\cdot\|)$ besitzt die ***Konvergenzordnung*** $p \geq 1$, wenn es eine positive Konstante $C > 0$ gibt mit

$$\|x^{(k+1)} - x^*\| \leq C \cdot \|x^{(k)} - x^*\|^p \qquad \text{für alle} \qquad k \in \mathbb{N}.$$

Den Fall $p = 1$ bezeichnet man als ***lineare Konvergenz*** und 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 2.12. *Sei $U \subseteq \mathbb{R}^n$ offen und konvex und sei $F : U \to \mathbb{R}^n$ stetig differenzierbar. Für alle $x^{(0)} \in U$ gelten zudem die folgenden vier Bedingungen mit einer beliebigen Norm $\|\cdot\|$ auf dem $\mathbb{R}^n$:*

(1) *F besitzt eine Nullstelle $x^* \in U$.*

(2) *$DF(x)$ ist für alle $x \in U$ regulär.*

(3) *Es gibt ein $\omega > 0$, sodass für alle $x, y \in U$ und $\rho := \|x^* - x^{(0)}\|$ gilt:*

$$\|(DF(x))^{-1} \cdot (DF(y) - DF(x))\| \leq \omega \cdot \|x - y\| \qquad \text{und} \qquad \frac{\omega}{2} \cdot \rho < 1.$$

(4) *Es gilt $B_\rho(x^*) := \{x \in \mathbb{R}^n : \|x - x^*\| < \rho\} \subset U$.*

Dann gelten für das Newton-Verfahren

$$x^{(k+1)} = x^{(k)} - (DF(x^{(k)}))^{-1} \cdot F(x^{(k)})$$

die folgenden Aussagen:

(1) *Für alle $k \in \mathbb{N}$ ist $x^{(k)} \in B_\rho(x^*)$.*

(2) *$x^{(k)}$ konvergiert gegen x^*.*

(3) *Für alle $k \in \mathbb{N} \cup \{0\}$ gilt die* **a priori** *Fehlerschranke*

$$\|x^{(k)} - x^*\| \leq \rho \cdot \left(\frac{\omega\rho}{2}\right)^{2^k - 1}.$$

(4) *Für alle $k \in \mathbb{N} \cup \{0\}$ gilt die* **a posteriori** *Fehlerschranke*

$$\|x^{(k+1)} - x^*\| \leq \frac{\omega}{2} \cdot \|x^{(k)} - x^*\|^2.$$

Beweis. Der Beweis gliedert sich in mehrere Teile.

Schritt 1 (A posteriori Fehlerschranke) Wir zeigen zunächst, dass aus $x^{(k)} \in U$ die a-posteriori Fehlerschranke für k folgt. Dazu definieren wir die Funktion $g : [0,1] \to \mathbb{R}^n$ mit

$$g(t) \;=\; F(x^{(k)} + t(x^* - x^{(k)})).$$

Durch die mehrdimensionale Kettenregel erhalten wir

$$g'(t) \;=\; DF(x^{(k)} + t(x^* - x^{(k)})) \cdot (x^* - x^{(k)}),$$

woraus nach dem Hauptsatz der Differential- und Integralrechnung wie in Lemma 2.10 folgt, dass

$$\begin{aligned} F(x^*) - F(x^{(k)}) \;&=\; g(1) - g(0) \;=\; \int_0^1 g'(t)\mathrm{d}t \\ &=\; \int_0^1 DF(x^{(k)} + t(x^* - x^{(k)})) \cdot (x^* - x^{(k)})\mathrm{d}t \end{aligned}$$

gilt. Nun setzen wir voraus, dass $x^{(k)} \in U$ für alle $k \in \mathbb{N}$. Nach der Definition des Newton-Verfahrens gilt dann

$$\begin{aligned} A \;&=\; x^{(k+1)} - x^* \\ &=\; x^{(k)} - (DF(x^{(k)}))^{-1} \cdot F(x^{(k)}) - x^* \\ &=\; x^{(k)} - x^* - (DF(x^{(k)}))^{-1} \cdot (F(x^{(k)}) - F(x^*)) \\ &=\; (DF(x^{(k)}))^{-1} \cdot \Big(F(x^*) - F(x^{(k)}) - DF(x^{(k)}) \cdot (x^* - x^{(k)})\Big) \\ &=\; (DF(x^{(k)}))^{-1} \cdot \Big(g(1) - g(0) - DF(x^{(k)}) \cdot (x^* - x^{(k)})\Big) \\ &=\; (DF(x^{(k)}))^{-1} \cdot \int_0^1 DF(x^{(k)} + t(x^* - x^{(k)})) \cdot (x^* - x^{(k)})\,\mathrm{d}t \\ &\qquad -\; (DF(x^{(k)}))^{-1} \cdot \Big(DF(x^{(k)}) \cdot (x^* - x^{(k)})\Big) \\ &=\; \int_0^1 (DF(x^{(k)}))^{-1} \cdot \Big(DF(x^{(k)} + t(x^* - x^{(k)})) - DF(x^{(k)})\Big) \\ &\qquad \cdot (x^* - x^{(k)})\,\mathrm{d}t. \end{aligned}$$

Gehen wir zur Norm davon über, so erhalten wir

$$\begin{aligned} \|A\| \;&=\; \|x^{(k+1)} - x^*\| \\ &\leq\; \int_0^1 \Big\|(DF(x^{(k)}))^{-1} \cdot \Big(DF(x^{(k)} + t(x^* - x^{(k)})) - DF(x^{(k)})\Big)\Big\| \\ &\qquad \cdot \|x^* - x^{(k)}\|\,\mathrm{d}t. \end{aligned}$$

Nach der Voraussetzung **(3)** gilt aber

$$\begin{aligned} &(DF(x^{(k)}))^{-1} \cdot \Big(DF(x^{(k)} + t(x^* - x^{(k)})) - DF(x^{(k)})\Big) \\ \leq\; &\omega \cdot \|x^{(k)} - (x^{(k)} + t(x^* - x^{(k)}))\| \\ =\; &\omega t \cdot \|x^{(k)} - x^*\|. \end{aligned}$$

Setzen wir dies in die obige Ungleichung ein, so erhalten wir

$$\begin{aligned} \|A\| &= \|x^{(k+1)} - x^*\| \\ &\leq \int_0^1 \omega t \cdot \|x^{(k)} - x^*\| \cdot \|x^{(k)} - x^*\| \, \mathrm{d}t \\ &= \int_0^1 \omega t \cdot \|x^{(k)} - x^*\|^2 \, \mathrm{d}t \\ &= \frac{\omega}{2} \cdot \|x^{(k)} - x^*\|^2. \end{aligned}$$

Damit ist die a posteriori Fehlerschranke gezeigt, sofern $x^{(k)} \in U$ gilt.

Schritt 2 (Induktion) Alle weiteren Aussagen zeigen wir nun per Induktion.

Für $k = 0$ sind nur die beiden Fehlerschranken zu zeigen: Mit

$$\left(\frac{\omega\rho}{2}\right)^{2^k-1} = 1 \qquad \text{für} \quad k = 0$$

folgt direkt $\|x^{(0)} - x^*\| = \rho$ nach der Definition von ρ. Da weiter $x^{(0)} \in U$ gilt, können wir Schritt 1 des Beweises verwenden und erhalten

$$\|x^{(1)} - x^*\| \leq \frac{\omega}{2} \cdot \|x^{(0)} - x^*\|^2.$$

Damit ist der Induktionsanfang gezeigt.

Nun machen wir die Induktionsannahme, dass alle Aussagen für $k \in \mathbb{N}$ gelten und wir wollen im Induktionsschritt von k auf $k + 1$ schließen.

Nach Schritt 1, der Induktionsannahme und nach Voraussetzung **(3)** erhalten wir

$$\begin{aligned} \|x^{(k+1)} - x^*\| &\leq \frac{\omega}{2} \cdot \|x^{(k)} - x^*\|^2 \\ &\leq \frac{\omega}{2} \cdot \left(\frac{\omega\rho}{2}\right)^{2(2^k-1)} \cdot \rho^2 \\ &= \left(\frac{\omega\rho}{2}\right)^{2^{k+1}-1} \cdot \rho < \rho. \end{aligned}$$

Aus der letzten Zeile folgt genau die a priori Fehlerschranke für $k+1$, sowie die Aussage $x^{(k+1)} \in B_\rho(x^*)$.

Somit ist die Folge wohldefiniert und nach Voraussetzung **(4)** haben wir $x^{k+1} \in U$. Schritt 1 liefert damit auch die a posteriori Fehlerschranke für $k + 1$ und schließlich erhalten wir direkt die Konvergenz gegen x^*. □

Es sei bemerkt, dass wir durch die a posteriori Fehlerschranke auch die quadratische Konvergenz des Newton-Verfahrens gezeigt haben. Zum Abschluss

geben wir noch einen weiteren Satz mit strengeren Voraussetzung an, der eine lokale Konvergenzordnung 2 in der Nähe eine Nullstelle verspricht.

Satz 2.13. *Sei $U \subseteq \mathbb{R}^n$ offen und $F : U \to \mathbb{R}^n$ eine zweimal stetig differenzierbare Funktion. Weiter sei $x^* \in U$ eine Nullstelle von F mit* $\det(DF(x^*)) \neq 0$.

Dann existiert ein $\rho > 0$, sodass das Newton-Verfahren für alle Startwerte $x^{(0)} \in U := B_\rho(x^*)$ quadratisch konvergiert.

Beweis. Wir führen den Beweis auf Satz 2.12 und untersuchen dessen Voraussetzungen. Zunächst gilt **(1)** nach Voraussetzung. Da die Funktion

$$h(x) \;=\; (DF(x))^{-1}$$

als Matrixinversion stetig ist, gibt es ein $\rho > 0$, sodass

$$\|h(x)\| - \|h(x^*)\| \;\leq\; \|h(x) - h(x^*)\| \;\leq\; \varepsilon$$

für alle $x \in B_\rho(x^*)$. Mit $\varepsilon = \|(DF(x^*))^{-1}\|$ erhalten wir also

$$\|(DF(x))^{-1}\| - \|(DF(x^*))^{-1}\| \;\leq\; \|(DF(x^*))^{-1}\|$$

für alle $x \in B_\rho(x^*)$. Dies ist aber eine äquivalente Aussage zu

$$\|(DF(x))^{-1}\| \;\leq\; 2 \cdot \|(DF(x^*))^{-1}\|$$

für alle $x \in B_\rho(x^*)$ und somit folgt Voraussetzung **(2)** und mit $U = B_\rho(x^*)$ trivialerweise auch **(4)**.

Als letztes müssen wir noch Voraussetzung **(2)** zeigen. Hierfür nutzen wir aus, dass $DF(x)$ nach Voraussetzung für $x \in U$ differenzierbar ist. Somit gibt es eine Lipschitzkonstante $L > 0$, sodass

$$\|DF(x) - DF(y)\| \;\leq\; L \cdot \|x - y\| \qquad \text{für alle} \quad x, y \in U.$$

Wählen wir $\omega = 2L \cdot \|(DF(x^*))^{-1}\|$, so folgt

$$\begin{aligned} \|(DF(x))^{-1} \cdot (DF(y) - DF(x))\| &\leq \|(DF(x))^{-1}\| \cdot \|(DF(y) - DF(x))\| \\ &\leq 2 \cdot \|(DF(x^*))^{-1}\| \cdot L \cdot \|x - y\| \\ &= \omega \cdot \|x - y\| \end{aligned}$$

und dies ist der erste Teil von **(3)**. Da wir weiter stets $\rho < 2/\omega$ wählen können, folgt mit

$$\frac{\omega}{2} \cdot \rho \;<\; 1$$

auch der zweite Teil. Somit liefert Satz 2.12 die Behauptung. □

Abschließend sei noch bemerkt, dass wir bei allen Konvergenzaussagen zum Newton-Verfahren nur lokale Konvergenz gezeigt haben. Damit benötigen wir theoretisch eine hinreichend gute Approximation an die gesuchte Nullstelle x^*, um Konvergenz garantieren zu können. Praktisch kommt man jedoch oft mit einer groben Näherung aus.

2.5 Ausblick

In diesem Kapitel haben wir begonnen Gleichungssystem $F(x) = 0$ mit $m \leq n$ zu lösen. Natürlich lassen sich auch Probleme mit $m > n$ behandeln, dies führt zu linearen Ausgleichsproblemen.

Zudem gibt es eine Vielzahl von weiteren Newton-Varianten, die alle spezielle Vor- und Nachteile besitzen.

3 Lineare Gleichungssysteme

Grundsätzlich werden bei der numerischen Lösung von linearen Gleichungssystemen zwei Verfahren angewandt. Während direkte Verfahren eine exakte Lösung in endlich vielen Rechenschritten bestimmen, wird die Lösung bei iterativen Verfahren durch wiederholtes anwenden der gleichen Rechenvorschrift sukzessive angehähert. Dabei werden iterative Verfahren vor allem für große oder speziell strukturierte Gleichungssysteme eingesetzt. Wir werden in diesem Abschnitt sowohl direkte als auch iterative Verfahren diskutieren.

3.1 Notationen und Grundlagen

Bevor wir Lösungsverfahren für lineare Gleichungssysteme besprechen, führen wir zunächst die nötigen Notationen ein und wiederholen einige Begriffe und Ergebnisse aus der Linearen Algebra.

Notation 3.1. Mit $A \in \mathbb{K}^{m\times n}$ bezeichnen wir eine reelle oder komplexe $(m \times n)$-***Matrix***, d.h. eine Matrix mit m Zeilen und n Spalten.

Wir schreiben

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

Dabei bezeichnen A_j die Spalten der Matrix und a_i ihre Zeilen für $i = 1,\dots,m$ und $j = 1,\dots,n$.

Gilt $m = n$ so nennen wir die Matrix ***quadratisch***.

Damit definieren wir nun formal, was ein lineares Gleichungssystem ist.

Definition 3.2. Ein ***lineares Gleichungssystem***

$$A \cdot x \;=\; b$$

ist gegeben durch eine Matrix $A \in \mathbb{K}^{m\times n}$, einen Vektor $b = (b_1, \ldots, b_m) \in \mathbb{K}^m$ und n Variablen $x_1, \ldots, x_n$, kurz $x = (x_1, \ldots, x_n) \in \mathbb{K}^n$. Ausgeschrieben erhalten wir m Gleichungen

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

Falls $b = 0$ gilt, 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.

Sicherheitshalber erinnern wir noch an den Begriff der linearen Unabhängigkeit.

Definition 3.3. 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***. Daher sprechen wir nur von dem ***Rang*** einer Matrix.

Weiterhin wiederholen wir die folgenden Aussagen aus der linearen Algebra ohne Beweis.

Satz 3.1. *Sei $A \in \mathbb{K}^{n\times n}$ eine quadratische Matrix. Dann sind die folgenden Aussagen äquivalent:*

(1) *A ist invertierbar.*

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

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

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

Eine invertierbar Matrix nennen wir ***regulär*** oder ***nicht singulär***. Ist eine Matrix nicht invertierbar, so nennen wir sie ***singulär***.

Zudem sollte klar sein, dass wir eine $(m \times n)$-Matrix A als lineare Abbildung

$$A : \mathbb{K}^n \to \mathbb{K}^m \qquad \text{mit} \qquad A(x) \;=\; A \cdot x$$

auffassen können und dementsprechend auch vom ***Kern***

$$\ker(A) \;=\; \{x \in \mathbb{K}^n \;:\; A \cdot x = 0\}$$

der Matrix sprechen können.

Bevor wir numerische Verfahren zur Lösung eines linearen Gleichungssystems entwickeln, fassen wir einige Ergebnisse über die Lösbarkeit linearer Gleichungssysteme zusammen.

Satz 3.2. *Es gelten die folgenden Aussagen:*

(1) *Das Gleichungssystem* $A \cdot x = b$ *hat genau dann mindestens eine Lösung, wenn* $b \in \operatorname{span}\{A_1, \ldots, A_n\}$ *gilt.*

(2) *Das Gleichungssystem* $A \cdot x = b$ *hat genau dann höchstens eine Lösung, wenn* $\{A_1, \ldots, A_n\}$ *linear unabhängig sind.*

(3) *Das Gleichungssystem* $A \cdot x = b$ *hat genau dann eine eindeutig Lösung, wenn die Matrix* A *regulär ist. In diesem Fall ist*

$$x \;=\; A^{-1} \cdot b$$

die eindeutige Lösung.

3.2 Gauß-Verfahren und LU-Zerlegung

In diesem Abschnitt stellen wir ein direktes Verfahren zur Lösung von quadratischen linearen Gleichungssystemen vor, d.h. wir haben n Gleichungen mit n Unbekannten. Die grundlegende Idee besteht in der Beobachtung, dass Gleichungssysteme mit Dreiecksmatrizen besonders einfach gelöst werden können, nun aber alles der Reihe nach.

Definition 3.4. 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 $i = 1, \ldots, n$.

Es sei bemerkt, dass eine $(n \times n)$-Dreiecksmatrix genau dann regulär ist, wenn $a_{ii} \neq 0$ für alle $i = 1, \ldots, n$ gilt. Lineare Gleichungssysteme mit Dreiecksmatrizen lassen sich besonderns einfach lösen, wie die folgenden beiden einfache Ergebnisse zeigen.

Lemma 3.3 (Rückwärtselimination). *Sei $A \in \mathbb{K}^{n\times n}$ eine obere Dreiecksmatrix mit $a_{ii} \neq 0$ für $i = 1, \ldots, n$ und $b \in \mathbb{K}^n$. Dann lässt sich die Lösung des lineare Gleichungssystems $A \cdot x = b$ sukzessive durch*

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

bestimmen.

Lemma 3.4 (Vorwärtselimination). *Sei $A \in \mathbb{K}^{n\times n}$ eine untere Dreiecksmatrix mit $a_{ii} \neq 0$ für $i = 1, \ldots, n$ und $b \in \mathbb{K}^n$. Dann lässt sich die Lösung des lineare Gleichungssystems $A \cdot x = b$ sukzessive durch*

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

bestimmen.

Um den ***Aufwand*** von derartigen Algorithmen in Abhängigkeit der Eingabegröße n wiederzugeben, werden die ***wesentlichen Rechenoperationen*** gezählt. Zu den wesentlichen Rechenoperationen zählen wir Addition, Subtraktion, Multiplikation, Division und Vergleiche, nicht aber Zuweisungen.

Bei der Rückwärts- bzw. Vorwärtselimination benötigen wir n Divisionen, $\frac{1}{2}n(n-1)$ Multiplikationen und $\frac{1}{2}n(n-1)$ Subtraktionen und erhalten damit

$$n + n(n-1) \;=\; n^2$$

wesentliche Rechenoperationen. Um das asymptotische Verhalten für $n \to \infty$ besser verdeutlichen zu können, kann das ***Landau-Symbol*** $\mathcal{O}$ verwendet werden.

Definition 3.5. Seien (a_n) und (b_n) zwei reelle Folgen. Dann definieren wir $a_n = \mathcal{O}(b_n)$, wenn es ein $C > 0$ und ein $N \in \mathbb{N}$ gibt mit

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

Wir bezeichnen daher den Aufwand der Rückwärts- bzw. Vorwärtselimination mit $\mathcal{O}(n^2)$.

Beispiel 3.1. *Um das Landau-Symbol besser verstehen zu können, geben wir einige Beispiele.*

(1) *Ein Algorithmus mit*

$$3n^3 + \frac{1}{2}n^2 - 3n$$

wesentlichen Rechenoperationen, besitzt den Aufwand $\mathcal{O}(n^3)$.

(2) *Ein Algorithmus mit*

$$\binom{n}{2} + n$$

wesentlichen Rechenoperationen, besitzt den Aufwand $\mathcal{O}(\frac{1}{2}n^2)$.

(3) *Ein Algorithmus mit*

$$n! + 2^n$$

wesentlichen Rechenoperationen, besitzt den Aufwand $\mathcal{O}(n!)$.

Weiterhin sei bemerkt, dass es neben $\mathcal{O}$ auch noch weitere Landau-Symbole wie o, Ω und ω gibt. Für unsere Betrachtungen beschränken wir uns aber auf $\mathcal{O}$.

Unser Ziel im folgenden ist es eine quadratische Matrix als Produkt von zwei Dreiecksmatrizen zu schreiben, um dann durch Rückwärts- bzw. Vorwärtselimination eine Lösung des Gleichungssystems zu erhalten.

Definition 3.6. Eine Faktorisierung einer Matrix $A \in \mathbb{K}^{n\times n}$ der Form

$$A \;=\; L \cdot U$$

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 Gleichungssystemen mit

Dreiecksmatrizen bestimmen. Durch Vorwärtselimination lösen wir zuerst das Gleichungssystem

$$L \cdot z \;=\; b$$

und anschließend durch Rückwärtselimination

$$U \cdot x \;=\; z.$$

Die so erhaltene Lösung x erfüllt dann

$$A \cdot x \;=\; L \cdot U \cdot x \;=\; L \cdot z \;=\; b$$

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

Bevor wir uns mit der Berechnung einer LU-Zerlegung beschäftigen, machen wir noch eine theoretische Aussage zu Dreiecksmatrizen. Der Beweis sei dem Leser als Übung überlassen.

Satz 3.5. *Die Folgende Mengen von Matrizen bilden eine Gruppe bezüglich der Matrixmultiplikation als Verknüpfung.*

(1) *Die Menge aller regulären oberen Dreiecksmatrizen.*

(2) *Die Menge aller regulären unteren Dreiecksmatrizen.*

(3) *Die Menge aller normierten oberen Dreiecksmatrizen.*

(4) *Die Menge aller normierten unteren Dreiecksmatrizen.*

Damit erhalten wir sofort das folgende Lemma.

Lemma 3.6. *Hat eine reguläre Matrix A eine LU-Zerlegung mit normierter unterer Dreiecksmatrix L, so ist diese eindeutig.*

Beweis. Da A regulär ist, sind auch die Dreiecksmatrizen jeder LU-Zerlegung regulär. Seien nun $L_1 \cdot U_1 \;=\; L_2 \cdot U_2 \;=\; A$ zwei LU-Zerlegung von A mit normierten unteren Dreiecksmatrizen. Dann ist dies äquivalent zu

$$U_1 \cdot U_2^{-1} \;=\; L_1^{-1} \cdot L_2.$$

Nach Satz 3.5 steht links eine obere und rechts eine normierte untere Dreiecksmatrix. Um Gleichheit zu gewähren, muss also

$$U_1 \cdot U_2^{-1} \;=\; I \;=\; L_1^{-1} \cdot L_2$$

gelten und dementsprechend folgt $L_1 = L_2$ und $U_1 = U_2$. □

Mit dieser Eindeutigkeitsaussage kommen wir nun zur Berechnung einer LU-Zerlegung. Die Idee des im folgenden diskutierte ***Gauß-Verfahren*** besteht darin die gegeben Matrix A durch elementare Zeilenoperationen in eine Matrix in Dreiecksform zu transformieren.

Definition 3.7. Für einen Vektor

$$l^{(k)^T} \;=\; (0,\ldots,0,t_{k+1},\ldots,t_n) \;\in\; \mathbb{K}^n$$

mit $1 \leq k \leq n$ und dem k-ten Einheitsvektor $e_k \in \mathbb{K}^n$ definieren wir die ***Gauß-Matrix*** M_k durch

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

Wir sammeln zunächst einige Eigenschaften der Gauß-Matrizen.

Lemma 3.7. *Für die Gauß-Matrizen M_k gilt*

$$\det(M_k) \;=\; 1 \qquad \text{und} \qquad M_k^{-1} \;=\; I_n + l^{(k)} \cdot e_k^T.$$

Beweis. Da Gauß-Matrizen normierte untere Dreiecksmatrizen sind, ist die Aussage $\det(M_k) = 1$ trivial. Den zweiten Teil rechnen wir leicht nach:

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

wobei im letzten Schritt ausgenutzt wurde, dass $e_k^T \cdot l^{(k)} = 0$ gilt. □

Multiplizieren wir eine Gauß-Matrix M_k von links mit einer Matrix A, so erhalten wir 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. In

Formeln liefert dies

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

Man nennt diese Operation auch die ***Anwendung elementarer Zeilenoperationen***. Lemma 3.7 besagt dabei, dass die Anwendung von elementaren Zeilenoperationen die Determinante der Matrix nicht verändert.

Setzen wir für einen Vektor $c = (c_1, \ldots, c_n) \in \mathbb{K}^n$ und ein $k \in \{1, \ldots, n\}$ mit $c_k \neq 0$

$$l^{(k)} = \left(0, \ldots, 0, \frac{c_{k+1}}{c_k}, \ldots, \frac{c_n}{c_k}\right),$$

so erhalten wir

$$M_k \cdot c = (c_1, c_2, \ldots, c_k, 0, \ldots, 0). \tag{3.1}$$

Genau das wird im Gauß-Verfahren zur Transformation einer Matrix auf Dreiecksform ausgenutzt, siehe Algorithmus 3.1.

Input: Matrix $A \in \mathbb{K}^{n \times n}$.

$A^{(1)} := A$;
`for` $k = 1, \ldots, n-1$ `do`

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

`end for`.

Output: LU-Zerlegung von A mit $U = A^{(n)}$ und $L = M_1^{-1} \cdot \ldots \cdot M_{n-1}^{-1}$.

Algorithmus 3.1: Gauß-Verfahren ohne Spaltenpivotisierung.

Beispiel 3.2. *In diesem Beispiel wollen wir die LU-Zerlegung der Matriz*

$$A = \begin{pmatrix} 2 & 1 & 3 \\ -2 & 0 & -2 \\ 4 & 4 & 10 \end{pmatrix}$$

in der Matrixversion vorführen. Dazu multiplizieren wir die Matrix A mit der Matrix M_1, sodass wir im Produkt in der ersten Spalte von A die -2 und 4 eliminieren können:

$$M_1 \cdot A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 1 & 3 \\ -2 & 0 & -2 \\ 4 & 4 & 10 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 3 \\ 0 & 1 & 1 \\ 0 & 2 & 4 \end{pmatrix} = A^{(1)}.$$

Nun multiplizieren wir $A^{(1)}$ analog mit M_2, um eine obere Dreiecksmatrix $A^{(2)}$ zu erhalten:

$$M_2 \cdot A^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 1 & 3 \\ 0 & 1 & 1 \\ 0 & 2 & 4 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} = A^{(2)}.$$

Damit erhalten wir direkt

$$U = A^{(2)} = \begin{pmatrix} 2 & 1 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

und auch dir normierte Matrix $L = (l_{ij})$ können wir durch Ändern der Vorzeichen für $i > j$ direkt aus M_1 und M_2 ablesen:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 2 & 1 \end{pmatrix}.$$

Wir wollen nun zeigen, dass Algorithmus 3.1 tatsächlich eine LU-Zerlegung von A liefert. Zudem wollen wir uns mit der Durchführbarkeit des Verfahren beschäftigen. Dazu die folgende Definition.

Definition 3.8. Sei $A = (a_{ij}) \in \mathbb{K}^{n\times n}$ eine quadratische $(n \times n)$-Matrix. Dann bezeichnen wir mit

$$A^{[k]} = \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix} \tag{3.2}$$

für $k = 1, \dots, n$ den k-ten ***Hauptminor*** von A.

Satz 3.8. *Sei $A \in \mathbb{K}^{n\times n}$ mit $\det(A^{[k]}) \neq 0$ für $k = 1, \dots, n-1$. Dann ist Algorithmus 3.1 durchführbar und es gilt:*

(1) $a_{kk}^{(k)} \neq 0$ *für alle* $k = 1, \dots, n-1$.

(2) *U ist eine obere Dreiecksmatrix.*

(3) *L ist eine untere Dreiecksmatrix.*

(4) $A = L \cdot U$.

Beweis. Zunächst zeigen wir die ersten beiden Aussagen. Nach der Wahl der Vektoren $l^{(k)}$ und nach Gleichung (3.1) folgt iterativ

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

und damit ist bereits **(2)** gezeigt. Weiter wissen wir nach Lemma 3.7, dass

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

gilt. Wenden wir die elementaren Zeilenoperationen ausschließlich auf Submatrizen der Form (3.2) an, gilt diese Gleichung weiterhin und wir erhalten

$$\det \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix} = \det \begin{pmatrix} a_{11}^{(k)} & \dots & a_{1k}^{(k)} \\ \vdots & & \vdots \\ a_{k1}^{(k)} & \dots & a_{kk}^{(k)} \end{pmatrix}.$$

Nach Voraussetzung folgt schließlich

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

also insbesondere $a_{kk}^{(k)} \neq 0$. Damit ist auch **(1)** gezeigt.

Die Matrix L ist definiert als Produkt der M_k^{-1}. Da die M_k alle untere Dreiecksmatrizen sind, sind nach Satz 3.5 auch ihre Inversen Dreiecksmatrizen, ebenso die Produkte ihrer Inversen, also auch L und damit ist **(3)** bewiesen.

Algorithmus 3.1 liefert uns

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

Wegen $L = M_1^{-1} \cdot \ldots \cdot M_{n-1}^{-1}$ gilt weiter

$$L^{-1} \;=\; M_{n-1} \cdot \ldots \cdot M_1,$$

also $U = L^{-1} \cdot A$ und damit folgt **(4)**. □

Lemma 3.9. *Ist Algorithmus 3.1 durchführbar, so gilt*

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

Beweis. Nach Definition von L und Lemma 3.7 folgt

$$L \;=\; M_1^{-1} \cdot \ldots \cdot M_{n-1}^{-1} \;=\; (I + l^{(1)} e_1^T) \cdot \ldots \cdot (I + l^{(n-1)} e_{n-1}^T).$$

Zu zeigen bleibt damit, dass für alle $m = 1, \ldots, n-1$

$$I + \sum_{k=1}^{m} l^{(k)} e_k^T \;=\; (I + l^{(1)} e_1^T) \cdot \ldots \cdot (I + l^{(m)} e_m^T).$$

gilt. Für $m = 1$ ist die Behauptung klar. Weiter gelte die Behauptung für $m-1$. Dann erhalten wir

$$\begin{aligned}
& (I + l^{(1)} e_1^T) \cdot \ldots \cdot (I + l^{(m)} e_m^T) \\
= \;& \left(I + \sum_{k=1}^{m-1} l^{(k)} e_k^T \right) \cdot (I + l^{(m)} e_m^T) \\
= \;& I + l^{(m)} e_m^T + \sum_{k=1}^{m-1} l^{(k)} e_k^T + \sum_{k=1}^{m-1} l^{(k)} \underbrace{e_k^T l^{(m)}}_{=0} e_m^T \\
= \;& I + \sum_{k=1}^{m} l^{(k)} e_k^T,
\end{aligned}$$

womit der Induktionsschritt bewiesen ist. □

Diese Beobachtung hilft uns, das Gauß-Verfahren effizient zu organisieren. Wir speichern die Vektoren $l^{(1)}$ bis $l^{(n-1)}$ über die erzeugten Nullen im unteren Teil der Matrix A, während der obere Teil die Matrix U enthält. Dann können wir später die Matrix L einfach ablesen.

Bei dieser Grundversion des Gauß-Verfahrens treten oft zwei Probleme auf.

(1) Das Verfahren ist nicht für alle regulären Matrizen durchführbar. Es scheitert zum Beispiel schon an einer einfachen Matrix wie

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

(2) Bei kleinen, aber von Null verschiedenen Elementen $a_{kk}^{(k)}$ können große Rundungsfehler auftreten.

Erfreulicherweise lassen sich die beiden aufgeführten Schwierigkeiten durch das nun zu beschreibende Verfahren der ***Pivotisierung*** vermeiden. Im einfachsten Fall der ***Spaltenpivotisierung*** vertauschen wir während des k-ten Schritts des Gauß-Verfahrens die k-te Zeile mit einer darunterliegenden Zeile, und zwar mit 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 $j \in \{k, k+1, \ldots, n\}$, sodass

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

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

Definition 3.9. 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 gibt mit

$$P \cdot e_i \;=\; e_{\pi(i)} \qquad \text{für alle} \quad i = 1, \ldots, n.$$

P entsteht also durch Permutation der Spalten der Einheitsmatrix. Wir stellen ohne Beweis einige Eigenschaften zusammen.

Satz 3.10. *P sei eine Permutationsmatrix zur Permutation π. Dann gilt:*

(1) *P ist invertierbar.*

(2) *P^{-1} ist die Permutationsmatrix zur Permutation π^{-1}.*

(3) *P ist orthogonal, es gilt also $P^{-1} = P^T$.*

Um das Gauß-Verfahren zu verbessern, benötigen wir spezielle Permutationen π, nämlich solche, die genau zwei Elemente r und s vertauschen, also

$$\begin{aligned} \pi(r) &= s, \\ \pi(s) &= r, \\ \pi(i) &= i \qquad \text{für alle} \quad i \in \{1, \ldots, n\} \setminus \{r, s\}. \end{aligned}$$

Entsprechend erhalten wir die Permutationsmatrix

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

Wir halten fest: Die Matrixmultiplikation $P_{rs} \cdot A$ vertauscht die r-te mit der s-ten Zeile von A.

Das Gauß-Verfahren mit Spaltenpivotisierung liefert schließlich das folgende Ergebnis.

Satz 3.11. *Sei $A \in \mathbb{K}^{n\times n}$ eine regulär Matrix. Dann ist das Gauß-Verfahren mit Spaltenpivotisierung durchführbar und wir erhalten eine Zerlegung der Form*

$$P \cdot A \;=\; L \cdot U,$$

wobei P Permutationsmatrizen, L eine normierte untere Dreiecksmatrix L und U eine obere Dreiecksmatrix ist.

Der Beweis hierzu kann in Hanke-Bourgeois (2006) gefunden werden.

Abschließend sei noch bemerkt, dass zur effizienten Implementierung ein Gauß-Verfahren gewählt werden sollte, welches ohne Matrixmultiplikationen auskommt. Der Rechenaufwand eines derartigen Verfahrens ist von der Größenordnung $\mathcal{O}(\frac{1}{3}n^3)$.

3.3 Kondition von Gleichungssystemen

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, siehe Beispiel 1.2. Die ***Kondition*** eines Problems ist hingegen der im ungünstigsten Fall auftretende Vergrößerungsfaktor für den Einfluß von relativen Eingangsfehlern auf relative Ergebnisfehler. 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. Wir interessieren uns für den absoluten Fehler

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

Mit der Dreiecksungleichung gilt folgt

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

Hierbei sagt der erste Fehlerterm 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.

In diesen Abschnitt untersuchen wir nun die Kondition bei der Lösung eines linearen Gleichungssystems $Ax = b$. Dazu müssen wir lediglich die Kondition einer Matrix definieren.

Definition 3.10. Sei A eine reguläre Matirx und $\|\cdot\|$ eine Matrixnorm. Dann heißt

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

die ***Kondition*** von A.

Zunächst halten wir fest, dass stets

$$1 = \|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = \text{cond}(A)$$

gilt. Nach Satz 2.8 erhalten wir

$$\text{cond}_2(A) = \frac{\lambda_{max}}{\lambda_{min}},$$

wobei λ_{max} und λ_{min} die Beträge der betragsgrößten und betragskleinsten Eigenwerte von A sind. Weiter soll der Index verdeutlichen, dass wir die Spektralnorm verwenden.

Folgendes ***Störungslemma*** gibt Auskunft über die Lösbarkeit eines gestörten Gleichungssystems:

Lemma 3.12. *Sei $A \in \mathbb{K}^{n\times n}$ eine reguläre Matrix und es gelte für eine beliebige einer Vektornorm zugeordneten Matrixnorm die Ungleichung $\|A\| < 1$. Dann ist die Matrix $I - A$ regulär, und es gilt*

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

Beweis. Für alle $x \in \mathbb{K}^n$ mit $x \neq 0$ gilt wegen der umgekehrten Dreiecksungleichung

$$\|(I-A)x\| \;\geq\; \|x\| - \|Ax\| \;\geq\; \|x\| - \|A\|\|x\| = (1-\|A\|)\|x\| \;\geq\; 0.$$

Damit ist $(I-A)x = 0$ nur für $x = 0$ möglich und die Matrix $I - A$ ist invertierbar. Weiter gilt

$$\begin{aligned} 1 \;=\; \|(I-A)^{-1}(I-A)\| \;&\geq\; \|(I-A)^{-1}\| - \|(I-A)^{-1}\|\|A\| \\ &=\; (1-\|A\|)\|(I-A)^{-1}\| \end{aligned}$$

und daraus die Behauptung. □

Mit diesem Lemma können wir nun den Fehler abschätzen, den wir beim Lösen eines zum Beispiel durch Rundungsfehler gestörten Gleichungssystems machen.

Satz 3.13. *Sei $A \in \mathbb{K}^{n\times n}$ eine reguläre Matrix und $b \in \mathbb{K}^n \setminus \{0\}$. Weiter sei x eine Lösung $Ax = b$. Dann gilt für die fehlerbehaftet Lösung $x + \delta x$ des fehlerbehafteten Problems*

$$(A+\delta A)\cdot(x+\delta x) \;=\; (b+\delta b)$$

die Abschätzung

$$\frac{\|\delta x\|}{\|x\|} \;\leq\; \frac{\operatorname{cond}(A)}{1-\operatorname{cond}(A)\cdot\frac{\|\delta A\|}{\|A\|}} \cdot \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right),$$

sofern $\|A^{-1}\|\|\delta A\| < 1$ gilt.

Beweis. Mit Hilfe des vorigen Lemmas ist die Matrix $A+\delta A = A(I+A^{-1}\delta A)$ invertierbar. Aus

$$(I+A^{-1}\delta A)\cdot(x+\delta x) \;=\; A^{-1}(b+\delta b)$$

können wir somit

$$x+\delta x \;=\; (I+A^{-1}\delta A)^{-1}\cdot(x+A^{-1}\delta b)$$

sowie

$$\delta x \;=\; (I + A^{-1}\delta A)^{-1} \cdot A^{-1} \cdot (\delta b - \delta A x)$$

eliminieren. Damit erhalten wir

$$\|\delta x\| \;\leq\; \|(I + A^{-1}\delta A)^{-1}\| \cdot \|A^{-1}\| \cdot (\|\delta b\| + \|\delta A\| \cdot \|x\|)$$

und es folgt

$$\frac{\|\delta x\|}{\|x\|} \;\leq\; \|(I + A^{-1}\delta A)^{-1}\| \cdot \mathrm{cond}(A) \cdot \left(\frac{\|\delta b\|}{\|A\| \cdot \|x\|} + \frac{\|\delta A\|}{\|A\|} \right).$$

Unter Beachtung von $\|A\| \cdot \|x\| \geq \|b\|$ und

$$\|(I + A^{-1}\delta A)^{-1}\| \;\leq\; \frac{1}{1 - \mathrm{cond}(A) \cdot \frac{\|\delta A\|}{\|A\|}}$$

folgt schließlich die Behauptung. □

Eine kleine Kondition heißt also, dass das zugehörige lineare Gleichungssystem gut konditioniert ist. Eine große Kondition bedeutet, dass das zugehörige lineare Gleichungssystem schlecht konditioniert ist.

Später wird das folgende Lemma über unitäre Matrizen noch wichtig werden.

Lemma 3.14. *Sei $Q \in \mathbb{K}^{n\times n}$ unitäre und $A \in \mathbb{K}^{n\times n}$ regulär. Dann gilt*

$$\mathrm{cond}_2(Q) \;=\; 1 \qquad \text{sowie} \qquad \mathrm{cond}_2(QA) \;=\; \mathrm{cond}_2(A) \;=\; \mathrm{cond}_2(AQ).$$

Die Multiplikation von A mit einer unitären Matrix Q hat also keine Auswirkung auf die Kondition von A.

3.4 QR-Zerlegung

In diesen Abschnitt beschäftigen wir uns nun mit einem weiteren Eliminationsverfahren. Diese Faktorisierung führt meistens zu Matrizen mit kleinerer Kondition als bei der LU-Zerlegung und ist daher weniger anfällig für Rundungsfehler.

Definition 3.11. Eine Faktorisierung einer Matrix $A \in \mathbb{K}^{m\times n}$ der Form

$$A \;=\; Q \cdot R$$

mit einer unitären Matrix $Q \in \mathbb{K}^{m\times m}$ und einer oberen Dreiecksmatrix $R \in \mathbb{K}^{m\times n}$ heißt ***QR-Zerlegung*** von A.

Es sei bemerkt, dass die Matrix A nicht quadratisch sein muss.

Satz 3.15. *Sei $A \in \mathbb{K}^{n\times n}$ regulär. Dann gibt es eine QR-Zerlegung von A.*

Beweis. Der Beweis erfolgt mit dem Schmidtschen-Orthonormalisierungsverfahren. Da A regulär ist, sind die Spalenvektoren $a_1, \ldots, a_n$ linear unabhängig und können orthonormalisiert werden:

$$\begin{aligned} q_1 &= \frac{1}{\|a_1\|} a_1, \\ q_2 &= \frac{1}{\|a_2 - a_2^* q_1 q_1\|} \left(a_2 - a_2^* q_1 \; q_1\right), \\ &\vdots \\ q_n &= \frac{1}{\|\ldots\|} \left(a_n - \sum_{j=1}^{n-1} a_n^* q_j \; q_j \right). \end{aligned}$$

Die Matrix Q mit den Spalen $q_1, \ldots, q_n$ ist unitär und es gilt $Q = A \cdot R$ mit einer oben Dreiecksmatrix R. Da auch R^{-1} eine obere Dreiecksmatrix ist folgt die Behauptung. □

Bemerkung 3.1. Wenn eine QR-Zerlegung von A bekannt ist, so kann wegen $Q^{-1} = Q^*$ das Gleichungssystem $Ax = b$ sehr einfach durch Rückwärtelimination von $Rx = Q^*b$ gelöst werden.

Mit dem Schmidtsche Orthogonalisierungsverfahren haben wir theoretisch bereits ein Verfahren zur Bestimmung einer QR-Zerlegung gefunden. Da dieses jedoch empfindlich gegenüber dem Einfluß von Rundungsfehlern ist, nutzen wir Householder Matrizen.

Definition 3.12. Eine Matrix $H \in \mathbb{K}^{n\times n}$ der Form

$$H \;=\; I - \frac{2}{v^* v} v v^*$$

mit einem von 0 verschiedenen Vektor v heißt ***Householder***-Matrix.

Satz 3.16. *Householder-Matrizen H sind unitär und erfüllen $H = H^*$.*

Beweis. Beide Behauptungen können leicht nachgerechnet werden. Wir erhalten mit normiertem v

$$H^* \;=\; I^* - 2(vv^*)^* \;=\; I - 2vv^* \;=\; H$$

sowie

$$H \cdot H^* \;=\; (I - 2vv^*) \cdot (I - 2vv^*) \;=\; I - 4vv^* + 4vv^*vv^* \;=\; I$$

wobei wir $v^*v = 1$ genutzt haben. □

Es sei bemerkt, dass eine Householder-Matrix einer Spiegelung an der Ebene durch den Koordinatenursprung senkrecht zu v entspricht.

Wir möchten Householder-Matrizen ähnlich wie Gauß-Matrizen verwenden. Dazu müssen wir uns überlegen, ob es zu einem gegebenen Vektor $x \in \mathbb{K}^n \setminus \{0\}$ einen Vektor $v \in \mathbb{K}^n \setminus \{0\}$ gibt, so dass $Hx = \gamma e_1$ mit einer Zahl $\gamma \in \mathbb{K}$, d.h.

$$x - v\frac{2v^*x}{v^*v} = \gamma e_1.$$

Hinreichend hierfür sind die Gleichungen

$$\frac{2v^*x}{v^*v} = 1 \qquad \text{und} \quad v = x + ce_1 \qquad \text{mit } c \in \mathbb{R}.$$

Die erste Gleichung lässt sich umformen zu

$$v^*(2x - v) = 0.$$

Einsetzen der zweiten Gleichung in diese Gleichung liefert dann

$$v^*(2x - v) = (x + ce_1)^*(x - ce_1) = \|x\|_2^2 - c^2 = 0,$$

d.h. $c = \pm\|x\|_2$. Um Auslöschung zu vermeiden entscheiden wir uns für $c = \operatorname{sign}(x_1)\|x\|_2$. Zudem wird so garantiert, dass $v \neq 0$ gilt.

Lemma 3.17. *Ist $x \in \mathbb{K}^n \setminus \{0\}$ und setzt man $v := x + \operatorname{sign}(x_1)\|x\|_2 e_1$ und $H := I - \frac{2}{v^*v}vv^*$, so gilt*

$$Hx = -\operatorname{sign}(x_1)\|x\|_2 e_1.$$

Wir werden dieses Lemma nun iterativ auf die Spaltenvektoren $a_1, \ldots, a_n$ der Matrix A anwenden. Nach der Transformation des ersten Spaltenvektors a_1 erhalten wir eine Matrix der Form

$$H^{(1)}A = \begin{pmatrix} -\operatorname{sign}(a_1)\|a\|_2 & * \\ 0 & \boxed{(n-1) \times (n-1)} \end{pmatrix}.$$

Die beschriebene Elimination wird auf die verbleibende $(n-1) \times (n-1)$-Matrix angewandt. Das fehlende obere Diagonalelement wird dabei auf 1 ergänzt, um wieder eine n-dimensionale Householder-Matrix $H^{(2)}$ zu erhalten. Fortgesetztes Eliminieren ergibt nach $n-1$ Schritten

$$R \;=\; H^{(n-1)} \cdot \ldots \cdot H^{(1)} \cdot A$$

und weiterhin

$$Q \;=\; (H^{(n-1)} \cdot \ldots \cdot H^{(1)})^* \;=\; H^{(1)} \cdot \ldots \cdot H^{(n-1)}$$

und damit die gesuchte QR-Zerlegung.

Input: Matrix $A \in \mathbb{K}^{n \times n}$.

$A^{(0)} := A$;
`for` $k = 1, \ldots, n-1$ `do`

$$\begin{aligned} d &:= a_{k,k}^{(k-1)} + \operatorname{sign}(\mathrm{a}_{\mathrm{k,k}}^{(\mathrm{k-1})}) \sum_{\mathrm{i=k}}^{\mathrm{n}} \left|\mathrm{a}_{\mathrm{i,k}}^{(\mathrm{k-1})}\right|^2 \\ v^{(k)} &:= (\underbrace{0, \ldots, 0}_{k-1 \text{ mal}}, d, a_{k+1,k}^{(k-1)}, \ldots, a_{n,k}^{(k-1)})^T \\ H^{(k)} &:= I_n - \frac{2}{\left(v^{(k)}\right)^* v^{(k)}} v^{(k)} \left(v^{(k)}\right)^* \\ A^{(k)} &:= H^{(k)} A^{(k-1)} \end{aligned}$$

`end for`.

Output: $A = QR$ mit $R = A^{(n)}$ und $Q = H^{(1)} \cdot \ldots \cdot H^{(n-1)}$.

Algorithmus 3.2: QR-Zerlegung nach Householder

Die Elimination mittels unitärer Householder-Matrizen ist im allgemeinen etwas weniger empfindlich gegen Rundungsfehlern als die LU-Zerlegung mit dem Gauß-Verfahren.

Allerdings ist der Rechenaufwand von der Größenordnung $\mathcal{O}(\frac{2}{3}n^3)$ und damit doppelt so groß wie beim Gauß-Algorithmus.

3.5 Cholesky-Verfahren

Wir betrachten auch in diesem Abschnitt Gleichungssysteme $Ax = b$, allerdings nehmen wir nun an, dass A eine hermitesche und positiv definite Matrix ist. Dazu fassen wir zunächst einige Eigenschaften zusammen.

Lemma 3.18. *Sei $A \in \mathbb{K}^{n\times n}$ eine hermitesche Matrix. Dann gilt:*

(1) *A ist genau dann positiv definit, wenn alle ihre Eigenwerte echt positiv sind.*

(2) *A ist genau dann positiv semi-definit, wenn alle ihre Eigenwerte größer oder gleich Null sind.*

(3) *A ist genau dann positiv definit, wenn ihre Hauptminoren positiv sind, d.h.* $\det(A^{[k]}) > 0$ *für alle* $k = 1, \ldots, n$.

Insbesondere sind damit positiv definite Matrizen regulär.

Definition 3.13. Eine Faktorisierung einer hermiteschen Matrix $A \in \mathbb{K}^{n\times n}$ der Form

$$A \;=\; L \cdot D \cdot L^*$$

mit einer normierten unteren Dreiecksmatrix L und einer Diagonalmatrix D heißt ***LDL-Zerlegung*** von A .

Satz 3.19. *Sei $A \in \mathbb{K}^{n\times n}$ eine positiv definite hermitesche Matrix. Dann existiert eine eindeutig bestimmte LDL-Zerlegung von A.*

Beweis. Da A positiv definit ist und damit $\det(A^{[k]}) \neq 0$ gilt für $k = 1, \ldots, n$, kann nach Satz 3.8 eine LU-Zerlegung mit normierter unterer Dreiecksmatrix L ohne Spaltenpivotisierung bestimmt werden. Da A weiterhin regulär ist, ist diese Zerlegung sogar eindeutig.

Sei daher $A = L \cdot U$ eine LU-Zerlegung von A mit normierter unterer Dreiecksmatrix L und oberer Dreiecksmatrix U.

Nun setzen wir

$$D \;=\; diag(u_{11}, \ldots, u_{nn})$$

als die Diagonalmatrix mit den Einträgen aus der Hauptdiagonalen von U. Da U regulär ist, ist auch D regulär, sodass wir

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

definieren können und es gilt

$$L \cdot D \cdot \tilde{U} \;=\; L \cdot U \;=\; A.$$

Wir wollen nun zeigen, dass $\tilde{U} = L^*$ gilt. Dazu betrachten wir

$$A \;=\; A^* \;=\; (L \cdot D \cdot \tilde{U})^* \;=\; \tilde{U}^* \cdot D^* \cdot L^* \;=\; \tilde{U}^* \cdot (D^* \cdot L^*). \qquad (3.3)$$

$\tilde{U}$ ist nach Konstruktion eine normierte obere Dreiecksmatrix, also ist $\tilde{U}^*$ eine normierte untere Dreiecksmatrix. Weiter ist $D^* \cdot L^*$ eine obere Dreiecksmatrix, also haben wir auch mit Gleichung (3.3) eine LU-Zerlegung von A mit normierter unterer Dreiecksmatrix.

Wegen der Eindeutigkeit der LU-Zerlegung von A folgt damit $L = \tilde{U}^*$ und damit haben wir die LDL-Zerlegung von A gefunden. Es bleibt zu zeigen, dass die gefundene LDL-Zerlegung eindeutig ist.

Sei daher $A = L' \cdot D' \cdot (L')^*$ eine weitere LDL-Zerlegung von A mit normierter unterer Dreiecksmatrix L. Dann können wir wiederum

$$A \;=\; L' \cdot (D' \cdot (L')^*)$$

als LU-Zerlegung auffassen und die Eindeutigkeit der LU-Zerlegung liefert

$$L' \;=\; L \qquad \text{und} \qquad D' \cdot (L')^* \;=\; D \cdot L^*.$$

Durch die Invertierbarkeit von $L = L'$ folgt auch $D' = D$. □

Der Beweis des Satzes zeigt damit auch, wie die LDL-Zerlegung einer hermiteschen und positiv definiten Matrix A aus der LU-Zerlegung ohne Pivotisierung bestimmt werden kann. Damit kommen kommen wir nun zur Cholesky-Zerlegung.

Definition 3.14. Eine Faktorisierung einer hermiteschen Matrix $A \in \mathbb{K}^{n\times n}$ der Form

$$A \;=\; L \cdot L^*$$

mit einer unteren Dreiecksmatrix L heißt ***Cholesky-Zerlegung*** von A .

Satz 3.20. *Sei $A \in \mathbb{K}^{n\times n}$ eine positiv definite hermitesche Matrix. Dann existiert eine Cholesky-Zerlegung von A mit positiven Diagonalelementen von L. Unter dieser Nebenbedingung ist L eindeutig bestimmt.*

Beweis. Nach Satz 3.19 gibt es eine eindeutig bestimmt LDL-Zerlegung

$$A \;=\; L \cdot D \cdot L^*.$$

Weiterhin bezeichnen wir im folgenden mit $A^{[k]}$ und $D^{[k]}$ die Hauptminoren von A und D. Weil L eine untere Dreiecksmatrix ist, gilt

$$A^{[k]} \;=\; L^{[k]} \cdot D^{[k]} \cdot (L^{[k]})^*$$

und insbesondere $\det(A^{[k]}) = \det(D^{[k]})$. Wegen der positiven Definitheit von A gilt $\det(A^{[k]}) > 0$ und zusammen 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

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

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

$$\begin{aligned} \tilde{L} \cdot \tilde{L^*} &= L \cdot \mathrm{diag}(\sqrt{d_{11}}, \ldots, \sqrt{d_{nn}}) \cdot \mathrm{diag}(\sqrt{d_{11}}, \ldots, \sqrt{d_{nn}}) \cdot L^* \\ &= L \cdot D \cdot L^* \;=\; A \end{aligned}$$

gilt. Damit haben wir die Cholesky-Zerlegung gefunden.

Um die Eindeutigkeit zu zeigen, sei neben $A = \tilde{L} \cdot \tilde{L}^*$ auch $A = \tilde{L}' \cdot (\tilde{L}')^*$ eine Cholesky-Zerlegung mit Diagonalelementen

$$\lambda_1, \lambda_2, \ldots, \lambda_n \;>\; 0.$$

Mit $D' := \mathrm{diag}(\lambda_1^2, \ldots, \lambda_n^2)$ und

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

erhalten wir

$$A \;=\; L' \cdot D' \cdot (L')^*,$$

also eine weitere LDL-Zerlegung von A mit normierter unterer Dreiecksmatrix L'. Aus der Eindeutigkeit der LDL-Zerlegung 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

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

Zusammen ergibt sich

$$\tilde{L}' \;=\; L' \cdot \mathrm{diag}(\lambda_1, \ldots, \lambda_n) \;=\; L \cdot \mathrm{diag}(\sqrt{d_{11}}, \ldots, \sqrt{d_{nn}}) \;=\; \tilde{L}$$

und genau dies war noch zu zeigen. □

Nachdem Existenz und Eindeutigkeit einer Cholesky-Zerlegung nachgewiesen ist, wollen wir nun noch einen effizienten Algorithmus herleiten. Dazu nutzen wir die Dreicksstruktur von L um aus $A = LL^*$ folgende Gleichungen abzuleiten:

$$a_{kk} \;=\; \sum_{j=0}^{n} l_{kj}\bar{l}_{kj} = \sum_{j=0}^{k} |l_{kj}|^2, \qquad k = 1, \ldots, n, \tag{3.4}$$

$$a_{ik} \;=\; \sum_{j=0}^{n} l_{ij}\bar{l}_{kj} = \sum_{j=0}^{k} l_{ij}\bar{l}_{kj}, \qquad k = 1, \ldots, n,\ i = k+1, \ldots, n \tag{3.5}$$

Daraus können nun iterativ die Einträge l_{ik} von L bestimmt werden und wir erhalten folgenden Algorithmus:

Input: $A \in \mathbb{K}^{n\times n}$ hermitesch und positiv definit

`for` $k = 1, \ldots, n$ `do`
  $A(k,k) = \sqrt{A(k,k) - \sum_{j=1}^{k-1} |A(k,j)|^2}$
  `for` $i = k+1, \ldots, n$ `do`
    $A(i,k) = \frac{1}{A(k,k)} \left(A(i,k) - \sum_{j=1}^{k-1} A(i,j)\overline{A(k,j)} \right)$
  `end for`.
`end for`.

Output: Cholesky-Zerlegung von $A = LL^*$ mit L in der unteren Hälfte von A abgespeichert.

Algorithmus 3.3: Cholesky-Zerlegung

3.6 Gleichungssysteme mit Tridiagonalmatrizen

In diesen Abschnitt wollen wir noch die LU-Zerlegung von Tridiagonalmatrizen diskutieren, welche zum Beispiel bei der Methode der finiten Differenzen zur Lösung von Randwertproblemen gewöhnlicher Differentialgleichungen auftreten.

Wir untersuchen also Matrizen $A \in \mathbb{K}^{n\times n}$ mit

$$A = \begin{pmatrix} a_1 & d_1 & & & \\ c_1 & a_2 & d_2 & & \\ & c_2 & \ddots & \ddots & \\ & & \ddots & \ddots & d_{n-1} \\ & & & c_{n-1} & a_n \end{pmatrix}$$

und nehmen an, dass eine LU-Zerlegung von A ohne Spaltenpivotisierung existiert. Dann erhalten wir eine Zerlegung der Form

$$\begin{pmatrix} a_1 & d_1 & & & \\ c_1 & a_2 & d_2 & & \\ & c_2 & \ddots & \ddots & \\ & & \ddots & \ddots & d_{n-1} \\ & & & c_{n-1} & a_n \end{pmatrix} \tag{3.6}$$

$$= \begin{pmatrix} 1 & & & & \\ l_1 & 1 & & & \\ & l_2 & \ddots & & \\ & & \ddots & \ddots & \\ & & & l_{n-1} & 1 \end{pmatrix} \cdot \begin{pmatrix} m_1 & r_1 & & & \\ & m_2 & r_2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & r_{n-1} \\ & & & & m_n \end{pmatrix}, \quad (3.7)$$

wobei wir alle Unbekannten durch Koeffizientenvergleich iterativ berechnen können. Es ergibt sich sofort

$$\begin{aligned} m_1 &= a_1, \\ r_1 &= d_1, \\ l_1 \cdot m_1 &= c_1, \\ l_1 \cdot r_1 + m_2 &= a_2, \\ r_2 &= d_2 \end{aligned}$$

und so weiter. Damit ist die Gültigkeit von Algorithmus 3.4 leicht einzusehen.

Input: Tridiagonalmatrix $A \in \mathbb{K}^{n \times n}$ wie in Gleichung (3.6).

$m_1 := a_1$;
`for` $k = 1, \ldots, n-1$ `do`

$$\begin{aligned} r_k &:= d_k; \\ l_k &:= c_k / m_k; \\ m_{k+1} &:= a_{k+1} - l_k \cdot r_k; \end{aligned}$$

`end for`.

Output: LU-Zerlegung von A wie in Gleichung (3.7).

Algorithmus 3.4: LU-Zerlegung für Tridiagonalmatrizen.

In jeden Schritt des Algorithmus haben wir nur 3 wesentliche Rechenoperationen. Da wir genau $(n-1)$ Schritte durchlaufen müssen, sind in Algorithmus 3.4 $3(n-1)$ wesentlich Rechenoperationen durchzuführen. Wir erhalten also einen Aufwand von $\mathcal{O}(n)$, also einen äußerst günstigen Fall.

Zudem lässt sich auch die Rückwärts- bzw. Vorwärtselimination mit der berechneten LU-Zerlegung jeweils mit dem Aufwand $\mathcal{O}(n)$ durchführen, sodass wir Gleichungssysteme mit Tridiagonalmatrizen insgesamt mit einem linearen Aufwand $\mathcal{O}(n)$ lösen können, sofern eine LU-Zerlegung ohne Spaltenpivotisierung existiert.

Es sei noch bemerkt, dass sich auch Spaltenpivotisierungen bei Tridiagonalmatrizen mit einem Aufwand von $\mathcal{O}(n)$ durchführen lassen. Darauf wollen wir aber nicht weiter eingehen.

3.7 Iterative Verfahren für lineare Gleichungssysteme

In diesem Abschnitt untersuchen wir die Anwendung des Banachschen Fixpunktsatzes zur Lösung von linearen Gleichungssystemen $Ax = b$ mit einer Matrix $A \in \mathbb{K}^{n\times n}$ und $b \in \mathbb{K}^b$.

Dazu zerlegen wir die Matrix A in zwei Teilmatrizen

$$A \;=\; M + N.$$

Ist M invertierbar, so können wir die Gleichunng $Ax = b$ in die äquivalente Fixpunktgleichung

$$x \;=\; -M^{-1}Nx + M^{-1}b$$

umformen. Dies führt auf die Fixpunktiteration

$$x^{(k+1)} \;:=\; -M^{-1}Nx^{(k)} + M^{-1}b.$$

Zur Implementierung eines solchen Verfahrens sollte wiederum M^{-1} nicht ausgerechnet werden, sondern im $(k+1)$-ten Iterationsschritt das Gleichungssystem

$$Mx^{(k+1)} \;=\; -Nx^{(k)} + b$$

gelöst werden. Dabei können wir nur dann einen Effizienzgewinn erhoffen, wenn sich ein Gleichungssystem mit der Matrix M leichter lösen lässt als ein Gleichungssystem mit der Matrix A.

Zur theoretischen Analyse der Verfahren untersuchen wir zunächst affin lineare Abbildungen

$$\phi(x) \;=\; Bx - z$$

mit $B \in \mathbb{K}^{n\times n}$ und $z \in \mathbb{K}^b$. Weiter sei $\|\cdot\|$ eine Norm auf $\mathbb{K}^n$ mit der zugeordneten Matrixnorm $\|\cdot\|$ auf $\mathbb{K}^{n\times n}$. Damit gilt

$$\|\phi(x) - \phi(y)\| \;=\; \|B(x-y)\| \;\leq\; \|B\| \cdot \|x - y\|.$$

Nach dem Banachschen Fixpunktsatz konvergiert damit die Iteration

$$x^{(k+1)} \;:=\; Bx^{(k)} + z$$

bezüglich der Vektornorm $\|\cdot\|$, falls $\|B\| < 1$ gilt.

Da aber alle Normen auf dem $\mathbb{K}^n$ äquivalent sind, müssen wir nur eine Norm $\|\cdot\|_*$ mit $\|B\|_* < 1$ finden, sodass die Folge $(x^{(k)})$ bezüglich jeder Norm konvergiert.

Dazu geben wir ohne Beweis das folgende Ergebnis an, welches jede natürlichen Matrixnormen von B durch den Spektralradius nach unten beschränkt.

Lemma 3.21. *Für jede natürliche Matrixnorm $\|\cdot\|$ auf $\mathbb{K}^{n\times n}$ und jede Matrix $B \in \mathbb{K}^{n\times n}$ gilt*

$$\rho(B) \;\leq\; \|B\|,$$

wobei $\rho(B)$ der Spektralradius von B ist. Umgekehrt gibt es zu jeder Matrix $B \in \mathbb{K}^{n\times n}$ und jedem $\epsilon > 0$ eine natürliche Matrixnorm $\|\cdot\|$ auf $\mathbb{K}^{n\times n}$ mit

$$\|B\| \;\leq\; \rho(B) + \epsilon.$$

Mit diesen Ergebnis erhalten wir schließlich den für die folgenden Aussagen wichtigen Satz.

Satz 3.22. *Sei $B \in \mathbb{K}^{n\times n}$. Dann konvergiert die lineare Fixpunktiteration*

$$x^{(k+1)} \;=\; Bx^{(k)} + z$$

genau dann für alle Startwerte $x^{(0)} \in \mathbb{K}^n$ und alle $z \in \mathbb{K}^n$, wenn der Spektralradius von B die Ungleichung

$$\rho(B) \;<\; 1$$

erfüllt.

Beweis. Falls $\rho(B) < 1$ existiert nach dem vorigen Lemma eine natüliche Matrixnorm $\|\cdot\|$, so dass $\|B\| < 1$. Deshalb ist die Abbildung $\phi(x) = Bx+z$ kontrahierend auf $\mathbb{K}^n$ bezüglich der Norm $\|\cdot\|$ mit Kontraktionsfaktor $\|B\|$. Damit konvergiert Folge $(x^{(k)})$ nach dem Banachschen Fixpunktsatz bezüglich der Norm $\|\cdot\|$.

Da aber alle Normen auf $\mathbb{K}^n$ äquivalent sind, konvergiert die Folge $(x^{(k)})$ auch bezüglich jeder anderen Vektornorm.

Ist $\rho(B) \geq 1$, so existiert ein Eigenwert λ von B mit $|\lambda| \geq 1$. Sei x ein zugehöriger Eigenvektor. Dann ist die Iterationsfolge zum Startwert $x^{(0)} = x$ mit der rechten Seite $z = x$ gegeben durch

$$x^{(k)} \;=\; \left(\sum_{j=0}^{k} \lambda^j\right) x,$$

wie man leicht durch Induktion nach k zeigt. Da aber $|\lambda| \geq 1$ gilt, ist diese Folge nicht konvergent. □

Damit besprechen wir nun die zwei einfachsten Verfahren zur iterativen Lösung von linearen Gleichungssystemen.

Sei dazu $A = (a_{ij}) \in \mathbb{K}^{n\times n}$ gegeben. Dann zerlegen wir A in

$$A \;=\; A_D + A_L + A_U \tag{3.8}$$

mit der unteren Diagonalmatrix

$$A_D \;=\; \operatorname{diag}(a_{11}, \ldots, a_{nn}),$$

mit der untere Dreiecksmatrix

$$A_L \;=\; \begin{pmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ a_{31} & a_{31} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ a_{n1} & a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix}$$

und mit der oberen Dreiecksmatrix

$$A_U \;=\; \begin{pmatrix} 0 & a_{12} & \cdots & a_{1,n-1} & a_{1n} \\ & \ddots & \ddots & \vdots & \vdots \\ & & 0 & a_{n-2,n-1} & a_{n-2,n} \\ & & & 0 & a_{n-1,n} \\ & & & & 0 \end{pmatrix}.$$

Wie oben erwähnt, sollte die Matrix M in einer Zerlegung der Form

$$A \;=\; M + N$$

leicht invertierbar sein. Die einfachste Wahl ist $M = A_D$. Diese Wahl führt uns auf das ***Jacobi-Verfahren*** oder ***Gesamtschrittverfahren***

$$x^{(k+1)} \;=\; -A_D^{-1}(A_L + A_U)x^{(k)} + A_D^{-1}b. \tag{3.9}$$

Um die Durchfürbarkeit dieses Verfahrens zu gewährleisten, müssen wir offenbar annehmen, dass alle Diagonalelemente von A von 0 verschieden sind.

Wir wollen nun die Konvergenz des Jacobi-Verfahrens untersuchen und definieren daher

$$B \;=\; -A_D^{-1}(A_L + A_U).$$

Nach Satz 3.22 erhalten wir mit $\rho(B) < 1$ ein theoretisches Ergebnis zur Konvergenz des Verfahrens, allerdings ist der Spektralradius von B meist schwer zu berechnen.

Stattdessen schätzen wir die Norm von B bezüglich der gebräuchlichsten natürlichen Matrixnormen ab. Mit

$$b_{jl} \;=\; -\frac{a_{jl}}{a_{jj}} \qquad \text{für} \qquad j \neq l$$

Input: Matrix $A \in \mathbb{K}^{n\times n}$ und rechte Seite $b \in \mathbb{K}^n$.

Startvektor: $x^{(0)}$
`for` $k = 1, \ldots$ `do`
  `for` $l = 1, \ldots$,n `do`

$$x_l^{(k)} = \frac{1}{a_{ll}} \left(b_l - \sum_{j\neq l}^{n} a_{lj} x_j^{(k-1)} \right)$$

  `end for`
`end for`.

Output: Angenäherter Lösungsvektor $\tilde{x} = x^{(k)}$ des Gleichungssystems $Ax = b$.

Algorithmus 3.5: Jacobi-Verfahren.

und $b_{jj} = 0$ sonst definieren wir

$$\begin{aligned} q_\infty &:= \|B\|_\infty = \max_{j=1,\ldots,n} \sum_{\substack{l=1 \\ l\neq j}}^{n} \left| \frac{a_{jl}}{a_{jj}} \right|, \\ q_1 &:= \|B\|_1 = \max_{l=1,\ldots,n} \sum_{\substack{j=1 \\ l\neq j}}^{n} \left| \frac{a_{jl}}{a_{jj}} \right|, \\ q_2 &:= \|B\|_F = \sqrt{\sum_{\substack{j,l=1 \\ l\neq j}}^{n} \left| \frac{a_{jl}}{a_{jj}} \right|^2}. \end{aligned}$$

Somit haben wir $\rho(B) \leq \min\{q_1, q_2, q_\infty\}$ und durch Anwendung des Banachschen Fixpunktsatzes erhalten wir damit den folgenden Konvergenzsatz für das Jacobi-Verfahren.

Satz 3.23. *Gegeben sei eine Matrix $A \in \mathbb{K}^{n\times n}$. Mit den oben eingeführten Bezeichnungen gelte $q_\mu < 1$ für mindestens ein $\mu \in \{1, 2, \infty\}$.*

Dann konvergiert das Jacobi-Verfahren

$$x^{(k+1)} = -A_D^{-1}(A_L + A_U)x^{(k)} + A_D^{-1}b$$

für jede rechte Seite $b \in \mathbb{K}^n$ und jeden Startwert $x^{(0)} \in \mathbb{K}^n$ gegen die eindeutige Lösung x des Gleichungssystems $Ax = b$. Ist dies der Fall, so gelten die a-priori und a-posteriori Fehlerschranken

$$\begin{aligned} \|x^{(k)} - x^*\|_\mu &\leq \frac{q_\mu^k}{1-q_\mu} \|x^{(1)} - x^{(0)}\|_\mu, \\ \|x^{(k)} - x^*\|_\mu &\leq \frac{q_\mu}{1-q_\mu} \|x^{(k)} - x^{(k-1)}\|_\mu \end{aligned}$$

für $\mu \in \{1, 2, \infty\}$ *mit* $q_\mu < 1$.

Zunächst sei bemerkt, dass die Bedingungen $q_1 < 1$, $q_2 < 1$ und $q_\infty < 1$ nicht äquivalent sind.

Zudem lässt sich die Bedinung $q_\infty < 1$ umformulieren als

$$\sum_{\substack{l=1 \\ l \neq j}}^{n} |a_{jl}| \; < \; |a_{jj}| \qquad \text{für alle} \quad j = 1, \ldots, n.$$

Dieses Kriterium wird als ***starkes Zeilensummenkriterium*** bezeichngen. Analog ist die Bedingung $q_1 < 1$ äquivalent zu

$$\sum_{\substack{j=1 \\ j \neq l}}^{n} |a_{jl}| \; < \; |a_{ll}| \qquad \text{für alle} \quad l = 1, \ldots, n.$$

Dieses Kriterium heißt ***starkes Spaltensummenkriterium***.

Da sich Gleichungssysteme mit unteren Dreiecksmatrizen ebensfalls sehr effizient durch Vorwärtssubstitution lösen lassen, liegt neben $M = A_D$ auch die Wahl $M = A_D + A_L$ nahe. Das resultierende Verfahren

$$x^{(k+1)} = -(A_D + A_L)^{-1} A_U x^{(k)} + (A_D + A_L)^{-1} b \tag{3.10}$$

wird als ***Gauß-Seidel-Verfahren*** oder ***Einzelschrittverfahren*** bezeichnen.

Wiederum müssen alle Diagonalelemente von A von 0 verschieden sein, um die Durchführbarkeit dieses Verfahrens zu gewährleisten.

Input: Matrix $A \in \mathbb{K}^{n \times n}$ und rechte Seite $b \in \mathbb{K}^n$.

Startvektor: x
`for` $k = 1, \ldots$ `do`
  `for` $l = 1, \ldots$,n `do`

$$x_l = \frac{1}{a_{ll}} \left(b_l - \sum_{j \neq l}^{n} a_{lj} x_j \right)$$

  `end for`
`end for`.

Output: Angenäherter Lösungsvektor $\tilde{x} = x$ des Gleichungssystems $Ax = b$.

Algorithmus 3.6: Gauß-Seidel-Verfahren.

Die Konvergenzuntersuchung ist diesmal ein wenig aufwendiger als beim Jacobi-Verfahren.

Satz 3.24. *Die Matrix $A \in \mathbb{K}^{n\times n}$ erfülle das* **Sassenfeld-Kriterium**

$$p \;:=\; \max_{j=1,\dots,n} p_j \;<\; 1,$$

wobei die Zahlen p_j rekursiv definiert sind durch

$$p_1 \;:=\; \sum_{l=2}^{n} \left|\frac{a_{1l}}{a_{11}}\right|$$

sowie durch

$$p_j \;:=\; \sum_{l=1}^{j-1} \left|\frac{a_{jl}}{a_{jj}}\right| p_l + \sum_{l=j+1}^{n} \left|\frac{a_{jl}}{a_{jj}}\right| \qquad \text{für} \qquad j = 2,\dots,n.$$

Dann konvergiert die Gauß-Seidel-Verfahren

$$x^{(k+1)} = -(A_D + A_L)^{-1} A_U x^{(k)} + (A_D + A_L)^{-1} b$$

für jede rechte Seite $b \in \mathbb{K}^n$ und jeden Startwert $x^{(0)} \in \mathbb{K}^n$ gegen die eindeutige Lösung x des Gleichungssystems $Ax = b$. Ist dies der Fall, so gelten die a-priori und a-posteriori Fehlerschranken

$$\begin{aligned} \|x^{(k)} - x^*\|_\infty &\;\le\; \frac{p^k}{1-p}\|x^{(1)} - x^{(0)}\|_\infty, \\ \|x^{(k)} - x^*\|_\infty &\;\le\; \frac{p}{1-p}\|x^{(k)} - x^{(k-1)}\|_\infty. \end{aligned}$$

Beweis. Die Aussage des Satzes folgt wieder aus dem Banachschen Fixpunktsatz, falls wir für die Iterationsmatrix die Abschätzung

$$\|(A_D + A_L)^{-1} A_U\|_\infty \;\le\; p$$

zeigen können. Dazu betrachten wir die Gleichung

$$(A_D + A_L)x \;=\; -A_U z$$

für ein $x \in \mathbb{K}^n$ mit $\|x\|_\infty = 1$. Komponentenweise lautet diese Gleichung

$$x_j \;=\; -\sum_{l=1}^{j-1} \frac{a_{jl}}{a_{jj}} x_l - \sum_{j+1}^{n} \frac{a_{jl}}{a_{jj}} z_l \qquad \text{für} \quad j = 1,\dots,n.$$

Durch Induktion zeigt man leicht, dass $x_j \le p_j$ für $j = 1,\dots,n$ und daher gilt $\|x\|_\infty \le p$. Somit folgt

$$\|(A_D + A_L)^{-1} A_U\|_\infty \;\le\; p$$

und damit ergibt sich die Behauptung. □

Falls die Matrix A das starke Zeilensummen-Kriterium erfüllt, so erfüllt A auch das Sassenfeld-Kriterium und das Gauß-Seidel-Verfahren konvergiert. Das folgende Beispiel zeigt jedoch, dass die Umkehrung im Allgemeinen falsch ist.

Beispiel 3.3. *Gegeben sei die Matrix*

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}.$$

Offenbar ist $q_\infty = 1$, das starke Zeilensummenkriterium ist also nicht erfüllt. Weiter gilt

$$\begin{aligned} p_1 &= \frac{1}{2} \\ p_j &= \frac{1}{2}p_{j-1} + \frac{1}{2} \qquad \text{für} \quad j = 2, \ldots, n-1 \\ p_n &= \frac{1}{2}p_{n-1}. \end{aligned}$$

Daraus folgt durch Induktion

$$p_n = \frac{1}{2} - \frac{1}{2^n}$$

und daher

$$p = 1 - \frac{1}{2^{n-1}} < 1,$$

also ist das Sassenfeld-Kriterium hingegen erfüllt.

Obwohl das Gauß-Seidel-Verfahren in diesem Beispiel konvergiert, ist die Konvergenz extrem langsam, da der Kontraktionsfaktor p sehr nahe bei 1 liegt. Dieses Phänomen ist leider typisch für lineare Gleichungssystem, die durch Diskretisierung von Randwertproblemen für elliptische Differentialgleichungen entstehen.

Abschließend wollen wir noch einen weiteren Konvergenzsatz für das Gauß-Seidel-Verfahren ohne Beweis angeben.

Satz 3.25. *Sei $A \in \mathbb{K}^{n\times n}$ hermitesch und positiv definit. Dann konvergiert das Gauß-Seidel-Verfahren für jede rechte Seite $b \in \mathbb{K}^n$ und jeden Startwert $x^{(0)} \in \mathbb{K}^n$ gegen die eindeutige Lösung x des Gleichungssystems $Ax = b$.*

3.8 CG-Verfahren

In diesem Abschnitt betrachten wir lineare Gleichungssysteme $Ax = b$ mit einer hermiteschen und positiv definiten Matrix $A \in \mathbb{K}^{n\times n}$. Zunächst benötigen wir die folgende Bezeichnung.

Satz 3.26. *Sei $A \in \mathbb{K}^{n\times n}$ symmetrisch und positiv definit. Dann wird durch*

$$\langle x, y\rangle_A \;=\; x^* \cdot A \cdot y$$

für $x, y \in \mathbb{K}^n$ ein Skalarprodukt auf $\mathbb{K}^n$ definiert. Die zugehörige Norm

$$\|x\|_A \;=\; \sqrt{\langle x, x\rangle_A}$$

heißt **Energienorm**.

Beweis. Offensichtlich ist $\langle\,\cdot\,,\,\cdot\,\rangle_A$ antilinear im ersten und linear im zweiten Argument. Da weiter $A^* = A$ gilt, folgt

$$\overline{\langle x, y\rangle_A} \;=\; y^* \cdot A^* \cdot x \;=\; \langle y, x\rangle_A.$$

Schließlich gilt $\langle x, x\rangle_A = x^* \cdot A \cdot x > 0$ für $x \neq 0$, da A positiv definit. □

Der Einfachkeit halber werden wir bei der folgenden Herleitung nur den Fall $\mathbb{K} = \mathbb{R}$ betrachten, die Ergebnisse gelten jedoch auch für $\mathbb{K} = \mathbb{C}$.

Die Lösung $\hat{x} = A^{-1}b$ des Gleichungssystems $Ax = b$ ist das eindeutige Minimum des quadratischen Funktionals

$$\Phi(x) \;=\; \frac{1}{2} \cdot x^* Ax - x^* b$$

für $x \in \mathbb{R}^n$, denn es gilt

$$\begin{aligned}
\Phi(x) - \Phi(\hat{x}) &= \frac{1}{2} \cdot x^* Ax - x^* b - \frac{1}{2} \cdot \hat{x}^* A\hat{x} + \hat{x}^* b \\
&= \frac{1}{2} \cdot (x - \hat{x})^* \cdot A \cdot (x - \hat{x}) + x^* A\hat{x} - x^* b - \hat{x}^* A\hat{x} + \hat{x}^* b \\
&= \frac{1}{2} \cdot (x - \hat{x})^* \cdot A \cdot (x - \hat{x}) \\
&= \frac{1}{2} \cdot \|x - \hat{x}\|_A^2.
\end{aligned}$$

Geometrisch bedeutet dies, dass der Graph der Funktion $\Phi(x)$ bezüglich der Energienorm ein kreisförmiges Paraboloid ist, dessen Mittelpunkt über $\hat{x}$

liegt. Diese Beobachtung ist die Grundlage unserer Herleitung des Verfahrens der oder kurz ***CG-Verfahren***.

Ausgehend von einer Näherungslösung x_k, bestimmen wir im k-ten Iterationsschritt zunächst eine ***Suchrichtung*** $d_k \in \mathbb{R}^n \setminus \{0\}$ und wählen die nächste Iterierte über den Ansatz

$$x_{k+1} \;=\; x_k + \alpha_k \cdot d_k.$$

Dabei wählen wir die ***Schrittweite*** $\alpha_k \in \mathbb{R}$ als Minimum der Funktion

$$\begin{aligned} f(\alpha) &= \Phi\left(x_k + \alpha d_k\right) \\ &= \Phi\left(x_k\right) + \alpha \cdot \left(d_k^* A x_k - d_k^* b\right) + \frac{\alpha^2}{2} \cdot d_k^* A d_k. \end{aligned}$$

Da A positiv definit ist, folgt $d_k^* A d_k > 0$ und somit erhalten wir das Minimum von $f(\alpha)$ durch Auflösen der Gleichung $f'(\alpha_k) = 0$ nach α_k und erhalten

$$\alpha_k \;=\; \frac{(b - A x_k)^* \cdot d_k}{d_k^* \cdot A \cdot d_k} \;=\; \frac{r_k^* \cdot d_k}{d_k^* \cdot A \cdot d_k},$$

wobei wir hier und im folgenden

$$r_k \;=\; b - A x_k$$

als das ***Residuum*** im k-ten Iterationsschritt definieren.

Offen ist noch die Wahl der Suchrichtungen d_k. Eine naheliegende Möglichkeit wäre die Richtung des steilsten Abstiegs von $\Phi(x)$ zu verwenden, also

$$d_k \;=\; -\operatorname{grad}\Phi(x_k) \;=\; -\left(A x_k - b\right) \;=\; -\,r_k.$$

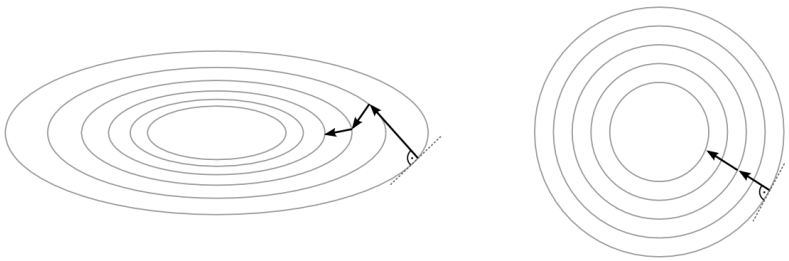


Abbildung 3.1: Suchrichtung als negativer Gradient (links) und Suchrichtung in der Energienorm beim CG-Verfahren.

Dies muss allerdings nicht die beste Wahl der Suchrichtung sein, wie Abbildung 3.1 an einem zweidimensionalen Beispiel veranschaulicht.

Bezüglich der Energienorm, also in der $\|\cdot\|_A$-Geometrie, sind die Niveaulinien aber Kreise. Aufgrund der Minimalitätseigenschaft von α_k ist x_{k+1} der Punkt, in dem die Gerade

$$\{x_k + \alpha \cdot d_k \ : \ \alpha \in \mathbb{R}\}$$

eine Niveaulinie tangential berührt. Wählen wir daher d_{k+1} orthogonal zu d_k bezüglich des Skalarproduktes $\langle \cdot, \cdot \rangle_A$, so liegt x_{k+2} genau im Kreismittelpunkt. Dies führt uns auf den Ansatz

$$d_{k+1} \ = \ r_{k+1} + \beta_k \cdot d_k,$$

wobei wir β_k so bestimmen wollen, dass

$$\langle d_{k+1}, d_k \rangle_A \ = \ 0$$

gilt. Durch einfache Rechnungen erhalten wir damit die Gleichung

$$\beta_k \ = \ - \frac{r_{k+1}^* \cdot A \cdot d_k}{d_k^* \cdot A \cdot d_k}.$$

Vektoren, die bezüglich des Skalarproduktes $\langle \cdot, \cdot \rangle_A$ orthogonal sind, werden auch ***A-konjugiert*** genannt. Da wir die Suchrichtungen d_k zueinander A-konjugiert gewählt haben, ergibt sich der Name ***Verfahren der konjugierten Gradienten***. Algorithmus 3.7 zeigt dieses Verfahren in seiner Grundversion.

Dieser Verfahren ist eine der am häufigsten verwendete Methode zur Lösung großer linearer Gleichungssysteme mit hermiteschen und positiv definiten Matrizen. Wir werden nun zunächst das Abbruchkriterium $r_k = 0$ diskutieren.

Dazu lässt sich zeigen, dass nicht nur aufeinanderfolgende, sondern alle Suchrichtungen paarweise A-konjugiert zueinander sind. Weiterhin sind auch alle Residuen paarweise orthogonal bezüglich des Euklidischen Skalarprodukts. Da Orthogonalsysteme aber linear unabhängig sind, muss spätestens nach n Schritten die Abbruchbedingung $r_k = 0$ erfüllt sein:

Satz 3.27. *Das CG-Verfahren bricht nach spätestens n Schritten mit der exakten Lösung ab.*

Für die Praxis ist dieses Resultat allerdings nur von eingeschränkter Bedeutung, da das CG-Verfahren meist nur mit deutlich weniger als n Schritten

Input: Hermitesche und positiv definite Matrix $A \in \mathbb{K}^{n\times n}$, Vektor $b \in \mathbb{K}^n$ und Startvektor $x_0 \in \mathbb{K}^n$.

$k := 0$;
$r_0 := b - Ax_0$;
$d_0 := r_0$;
`while` $r_k \neq 0$

$$\begin{aligned} \alpha_k &:= \frac{r_k^* \cdot d_k}{d_k^* \cdot A \cdot d_k}\,; \\ x_{k+1} &:= x_k + \alpha_k \cdot d_k\,; \\ r_{k+1} &:= b - A \cdot x_{k+1}\,; \\ \beta_k &:= -\frac{r_{k+1}^* \cdot A \cdot d_k}{d_k^* \cdot A \cdot d_k}\,; \\ d_{k+1} &:= r_{k+1} + \beta_k \cdot d_k\,; \\ k &:= k+1\,; \end{aligned}$$

`end`.

Output: Exakte Lösung x_k zum Gleichungssystem $Ax = b$.

Algorithmus 3.7: Das CG-Verfahren in seiner Grundversion.

effizient ist. Zudem treten durch Rundungsfehler Verluste der Orthogonalitätseigenschaften auf. Für die Interpretation des CG-Verfahrens als iteratives Verfahren gilt aber die Fehlerabschätzung

$$\|x_k - \hat{x}\|_A \;\leq\; 2 \cdot \left(\frac{\sqrt{\mathrm{cond}_2(A)} - 1}{\sqrt{\mathrm{cond}_2(A)} + 1} \right)^k \cdot \|x_0 - \hat{x}\|_A.$$

Diese Ungleichung kann mit Hilfe einer Optimalitätseigenschaft des CG-Verfahrens bezüglich der Energienorm bewiesen werden, ein genauer Beweis kann dem Lehrbuch Hanke-Bourgeois (2006) entnommen werden.

Abschließend benötigen wir noch die folgende allgemeine Definition, um eine weitere theoretische Aussage zum CG-Verfahren angeben zu können.

Definition 3.15. Sei $A \in \mathbb{K}^{n\times n}$ eine Matrix und $b \in \mathbb{K}^n$ ein Vektor. Dann definieren wir den zugehörigen k-ten ***Krylov-Raum*** als kleinsten Unterraum von $\mathbb{K}^n$, der die Vektoren $\{b, Ab, A^2b, \ldots A^{k-1}b\}$ enthält, also

$$\mathcal{K}_k(A, b) \;:=\; \mathrm{span}\{b, Ab, A^2b, \ldots A^{k-1}b\}.$$

Damit erhalten wir das folgende entscheidene Ergebnis.

Satz 3.28. *Für die k-te Iterierte des CG-Verfahrens gilt*

$$x_k \in x_0 + \mathcal{K}_k(A, r_0),$$

sofern das Verfahren nicht vorher abbricht. Zudem ist x_k in diesem affinen Unterraum das eindeutige Minimum von $\Phi(x)$.

In jedem Schritt k des CG-Verfahrens lösen wir somit das Gleichungssystem, welches durch Projektion des gegebenen Systems auf den Krylov-Raum $\mathcal{K}_k(A, r_0)$ entsteht.

Es sei noch bemerkt, dass für eine Implementation des CG-Verfahrens für große Matrizen die Werte α_k und β_k auf eine alternative Weise berechnet werden sollten, da die neuen Formeln etwas stabiler sind. Auch hier sei für weitere Ausführungen auf Hanke-Bourgeois (2006) verwiesen.

3.9 Ausblick

In diesem Kapitel haben wir viele zum Teil stark unterschiedliche Verfahren zur lösung von linearen Gleichungssystemen $Ax = b$ kennengelernt. Dabei sind wir fast immer einer quadratische Matrix A mit einer eindeutigen Lösung von $Ax = b$ vorausgesetzt. Angenommen $Ax = b$ ist nicht lösbar, so besteht die Aufgabe von linearen Ausgleichsproblemen darin, das Problem

$$\min_{x \in \mathbb{K}^n} \|Ax - b\|_2$$

zu lösen. Auch hierzu kann die QR-Zerlegung von A verwendet werden, welche wir auch für nicht quadratische Matrizen untersucht haben.

Im folgenden Stellen wir noch einmal zusammen, unter welchen Annahmen welches Verfahren zur lösung eines linearen Gleichungssystems verwendet werden sollte.

(1) Für voll besetzte quadratische Matrizen A mit nicht zu großer Dimension liefert uns das Gauß-Verfahren einen einfachen Algorithmus zur Lösung von $Ax = b$. Zudem bricht der Algorithmus automatisch ab, falls A nicht regulär ist.

(2) Für reguläre quadratische Matrizen A lässt sich auch dass QR-Verfahren verwenden. Der Vorteil hierbei ist, dass wir uns eine günstigere Konditionen der Matrizen erhoffen als bei der LU-Zerlegung.

(3) Für schwach besetzte Matrizen ist die LU- bzw. QR-Zerlegung hingegen ungeeignet, da wir hier Matrizen L und U bzw. Q und R erhalten, die meist voll besetzt sind. Daher sollten hier nach Möglichkeit iterative Verfahren bevorzugt werden.

(4) Für quadratische Matrizen in großer Dimension hingegen liefert das CG-Verfahren sehr schnell eine gute Approximation der optimalen Lösung.

4 Eigenwertaufgaben

In diesen Kapitel beschäftigen wir uns mit der numerischen Berechnung von Eigenwerten. Dabei stehen uns sehr unterschiedliche Mittel und Methoden zur Verfügung. Nachdem wir einige Aussagen über die Lokalisation der Eigenwerte einer Matrix zusammengestellt haben, werden wir verschiedene Verfahren herleiten. Dabei beginnen wir mit der Berechnung des größten bzw. kleinsten Eigenwertes durch Vektoriteration, bestimmen einige Eigenwerte mit zugehörigem Eigenvektor und berechnen schließlich alle Eigenwerte. Hierzu werden wir einige Grundlagen der Algebra wiederholen und schließlich auf zuvor besprochene Inhalte zurückgreifen, so zum Beispiel auf die QR-Zerlegung. Abschließend werden wir kurz ein Verfahren vorstellen, welches sich auch auf sehr große und dünn besetzte Matrizen anwenden lässt, das Lanczos-Verfahren.

4.1 Grundlagen zu Eigenwerten

Im folgenden steht $\mathbb{K}$ wie auch in einigen Kapitel zuvor für den Körper $\mathbb{R}$ der reellen Zahlen oder den Körper $\mathbb{C}$ der komplexen Zahlen. Zunächst wiederholen wir einige Definitionen.

Definition 4.1. Eine Zahl $\lambda \in \mathbb{K}$ heißt ***Eigenwert*** einer Matrix $A \in \mathbb{K}^{n\times n}$, falls es einen Vektor $x \in \mathbb{K}^n \setminus \{0\}$ gibt mit

$$A \cdot x \;=\; \lambda \cdot x.$$

Ein solcher Vektor x heißt ***Eigenvektor*** von A zum Eigenwert λ.

Die folgenden Ergebnisse sollten aus der linearen Algebra bekannt sein und werden daher an dieser Stelle ohne Beweis wiederholt.

Satz 4.1. *Die Eigenwerte einer Matrix $A \in \mathbb{K}^{n\times n}$ sind die Nullstellen des* **charakteristischen Polynoms**

$$p(\lambda) \;=\; \det(A - \lambda I). \tag{4.1}$$

Insbesondere besitzt A höchstens n Eigenwerte und eine komplexe Matrix besitzt mindestens einen Eigenwert.

Beweis. Nach Definition ist $\lambda \in \mathbb{K}$ genau dann Eigenwert von A, falls die Gleichung $(A - \lambda I)x = 0$ eine von 0 verschiedene Lösung x besitzt. Dies ist genau dann der Fall, wenn die Matrix $A - \lambda I$ nicht regulär ist bzw. wenn $\det(A - \lambda I) = 0$ gilt.

Nach Definition der Determinante ist $p(\lambda)$ ein Polynom vom Grad n und damit folgen alle Aussage aus dem Fundamentalsatz der Algebra. □

Das folgende Resultat ist die Grundlage für mehrere Algorithmen zur numerischen Berechnung von Eigenwerten.

Satz 4.2. *Sei $Q \in \mathbb{K}^{n\times n}$ eine reguläre Matrix und seu $A \in \mathbb{K}^{n\times n}$. Dann besitzen die Matrizen*

$$A \qquad \text{und} \qquad Q^{-1} \cdot A \cdot Q$$

die gleichen Eigenwerte.

Beweis. Für alle $\lambda \in \mathbb{K}$ gilt nach den Rechenregeln für Determinanten

$$\begin{aligned} \det(Q^{-1}AQ - \lambda I) &= \det(Q^{-1}(A - \lambda I)Q) = \det Q^{-1} \det(A - \lambda I) \det Q \\ &= \det(A - \lambda I) \det Q^{-1}Q = \det(A - \lambda I). \end{aligned}$$

Damit folgt die Behauptung aus Satz 4.1. □

Die Grundidee vieler numerischer Algorithmen zur Bestimmung der Eigenwerte einer Matrix $A \in \mathbb{K}^{n\times n}$ besteht darin, eine Folge von Matrizen (Q_n) zu konstruieren, so dass die Folge

$$A_k = Q_k^{-1} \cdot A_{k-1} \cdot Q_k \qquad \text{mit} \qquad A_0 = A$$

gegen eine Matrix D konvergiert, deren Eigenwerte leicht zu bestimmen sind. Dies ist zum Beispiel dann der Fall, wenn D eine Diagonalmatrix oder eine Dreiecksmatrix ist.

Auch die folgenden Aussagen sollten aus der linearen Algebra bekannt sein und werden nur kurz wiederholt, da wir sie später verwenden werden.

Satz 4.3 (Schur-Zerlegung). *Sei $A \in \mathbb{C}^{n\times n}$ beliebig. Dann gibt es eine unitäre Matrix $U \in \mathbb{C}^{n\times n}$ und eine obere Dreiecksmatrix $R \in \mathbb{C}^{n\times n}$ mit*

$$A = U \cdot R \cdot U^*.$$

Eine derartige Zerlegung von A heißt **Schur-Zerlegung**.

Da die Eigenwerte einer oberen Dreiecksmatrix R deren Diagonalelemente sind, können wir an einer Schur-Zerlegung einer Matrix A deren Eigenwerte ablesen. Eine einfache Folgerung aus der Schur-Zerlegung ist das folgende Ergebnis.

Satz 4.4 (Hauptachsentransformation). *Sei $A \in \mathbb{C}^{n\times n}$ hermitesch. Dann gibt es eine unitäre Matrix $U \in \mathbb{C}^{n\times n}$ und eine Diagonalmatrix $D \in \mathbb{R}^{n\times n}$ mit*

$$A \;=\; U \cdot D \cdot U^*.$$

Dabei ist zu beachten, dass D nur reelle Einträge besitzt.

Schreiben wir die Gleichung $AU = UD$ spaltenweise auf, so ergibt sich

$$A \cdot u_j \;=\; \lambda_j \cdot u_j,$$

wobei u_j die j-te Spalte von U und $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ ist. Wir können Korollar 4.4 daher auch folgendermaßen formulieren:

Zu jeder hermiteschen Matrix A gibt es eine Orthonomalbasis $\{u_1, \ldots, u_n\}$ aus Eigenvektoren von A und alle Eigenwerte von A sind reell.

Bevor wir mit der numerischen Berechnung von Eigenwerten so richtig loslegen, wollen wir die Bestimmung von Eigenwerten noch an einem Beispiel verdeutlichen.

Beispiel 4.1. *Sei $u(t,x)$ eine Funktion, die die vertikale Auslenkung einer Saite an der Position $x \in [0,1]$ zur Zeit t beschreibt.*

Damit wissen wir, dass $u(t,x)$ näherungsweise die **Wellengleichung**

$$\frac{\partial^2 u}{\partial x^2}(t,x) \;=\; \frac{1}{c^2} \cdot \frac{\partial^2 u}{\partial t^2}(t,x) \tag{4.2}$$

für alle $x \in (0,1)$ und $t \in \mathbb{R}$ erfüllt, wobei c die Ausbreitungsgeschwindigkeit ist. Wir nehmen an, dass die Saite an den beiden Endpunkten 0 und 1 fest eingespannt ist, es gelten also die Randbedingungen

$$u(t,0) \;=\; u(t,1) \;=\; 0$$

für alle $t \in \mathbb{R}$. Wir suchen nun zeitharmonische Lösungen dieser Differentialgleichung, d.h. wir machen den Ansatz

$$u(t,x) \;=\; \operatorname{re}\left(v(x) \cdot \mathrm{e}^{i\omega t}\right)$$

mit einer unbekannten und gesuchten Frequenz $\omega \in \mathbb{C}$. Setzen wir diesen Ansatz in die Differentialgleichung ein, so erhalten wir für alle $x \in (0,1)$

$$-v''(x) \;=\; \lambda \cdot v(x) \qquad \text{und} \qquad v(0) \;=\; v(1) \;=\; 0$$

mit $\lambda = (\omega/c)^2$. Dies ist ein Eigenwertproblem für den Differentialoperator $Av = -v''$, welchen wir nun in eine Matrix überführen wollen.

Dazu führen wir den Raum

$$U \;=\; \{v \in \mathcal{C}^2([0,1]) \;:\; v(0) = v(1) = 0\}$$

ein und definieren die Abbildung

$$A : U \to C([0,1]) \qquad \text{mit} \qquad A(v) \;=\; -\,v''.$$

Weiter betrachten wir die Gitterpunkte $x_k = k \cdot h$ für $k = 0, \ldots, m$ mit dem Abstand $h = 1/m$ und approximieren die zweite Ableitungen durch die Differenzenquotienten

$$-v''(x_k) \;\approx\; \frac{1}{h^2} \cdot \left(-v(x_{k-1}) + 2v(x_k) - v(x_{k+1})\right).$$

Wählen wir die Unbekannten $v_k \approx v(x_k)$, so erhalten wir das System

$$\frac{1}{h^2} \cdot (-v_{k-1} + 2v_k - v_{k+1}) \;=\; \lambda \cdot v_k$$

für $k = 1, \ldots, m$. In Matrixform ergibt sich das Eigenwertproblem

$$\frac{1}{h^2} \cdot \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{m-2} \\ v_{m-1} \end{pmatrix} \;=\; \lambda \cdot \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{m-2} \\ v_{m-1} \end{pmatrix}.$$

Durch Approximation der zweiten Ableitung durch einen Differenzenquotienten haben wir damit ein Eigenwertproblem für einen Differenzialoperator näherungsweise in ein Matrix-Eigenwertproblem überführt.

4.2 Lokalisation von Eigenwerten

Bevor wir uns mit weiteren Eigenschaften von Eigenwerten beschäftigen, liefert uns Abschnitt 2.2 direkt die folgende Aussage.

Korollar 4.5. *Sei $\|\ \|$ eine zu einer Vektornorm passende Matrixnorm. Dann gilt für jeden Eigenwert λ von A die Abschätzung*

$$|\lambda| \ \leq \ \varrho(A) \ \leq \ \|A\|.$$

Im folgenden sei $A \in \mathbb{K}^{n\times n}$ eine hermitesche Matrix. Die der Größe nach geordneten Eigenwerte von A bezeichnen wir mit

$$\lambda_0(A) \ \geq \ \lambda_1(A) \ \geq \ \cdots \ \geq \ \lambda_{n-1}(A).$$

Dabei wird es sich als praktisch herausstellen, die Zählung der Eigenwerte bei 0 zu beginnen. Nun definieren wir die Funktion

$$R_A(x) \ := \ \frac{x^*Ax}{x^*x} \qquad \text{für} \quad x \in \mathbb{K}^n \setminus \{0\} \tag{4.3}$$

als ***Rayleigh-Quotienten*** von A.

Satz 4.6 (Rayleigh). *Sei $A \in \mathbb{K}^{n\times n}$ hermitesch. Dann gilt*

$$\lambda_0(A) \ = \ \max_{x\in\mathbb{R}\setminus\{0\}} R_A(x) \qquad \text{und} \qquad \lambda_{n-1}(A) \ = \ \min_{x\in\mathbb{R}\setminus\{0\}} R_A(x)$$

bzw. anders geschrieben

$$\lambda_0(A) \ = \ \max_{\|x\|_2=1} x^*Ax \qquad \text{und} \qquad \lambda_{n-1}(A) \ = \ \min_{\|x\|_2=1} x^*Ax\,.$$

Beweis. Nach der Hauptachsentransformation gibt es ein Orthonormalsystem $\{u_0,\ldots,u_{d-1}\}$ von Eigenvektoren von A mit zugehörigen Eigenwerten $\lambda_0(A),\ldots,\lambda_{n-1}(A)$. Für einen Vektor $x \in \mathbb{K}^n$ der Länge $\|x\|_2 = 1$ gilt daher

$$x \ = \ \sum_{k=0}^{n-1}(u_k^*x)u_k \qquad \text{und} \qquad \sum_{k=0}^{n-1}|u_k^*x|^2 \ = \ 1.$$

Es folgt

$$x^*Ax \ = \ x^*\cdot\sum_{k=0}^{n-1}(u_k^*x)Au_k \ = \ \sum_{k=0}^{n-1}\lambda_k|u_k^*x|^2 \ \leq \ \lambda_0\sum_{k=0}^{n-1}|u_k^*x|^2 \ = \ \lambda_0.$$

Dies bedeutet aber gerade

$$\sup_{\|x\|_2=1} x^*Ax \ \leq \ \lambda_0.$$

Da das Supremum für $x = u_0$ angenommen wird, folgt die erste Behauptung. Die zweite Teil verläuft analog. □

Satz 4.7 (Courantsches Minimum-Maximum-Prinzip). *Sei $A \in \mathbb{K}^{n\times n}$ hermitesch. Dann gilt*

$$\lambda_k(A) \;=\; \min_{\substack{P^*=P\\ \operatorname{rang}(P)=k}} \max_{\|x\|_2=1} x^*(A-P)x \tag{4.4}$$

für $k = 0, \ldots, n-1$, wobei $\operatorname{rang}(P)$ den Rang von P bezeichne.

Beweis. Wieder gibt es nach der Hauptachsentransformation ein Orthonormalsystem $\{u_0, \ldots, u_{d-1}\}$ aus Eigenvektoren von A mit zugehörigen Eigenwerten $\lambda_0(A), \ldots, \lambda_{n-1}(A)$, wobei wir teilweise einfach λ_k statt $\lambda_k(A)$ schreiben.

Wir bezeichnen die rechte Seite von Gleichung (4.4) mit $\alpha_k(A)$.

Zunächst zeigen wir, dass $\alpha_k(A) \le \lambda_k(A)$. Es sei bemerkt, dass nach der Hauptachsentransformation

$$A \;=\; \sum_{i=0}^{n-1} \lambda_i u_i u_i^*$$

gilt. Nun wähle

$$P \;=\; \sum_{i=0}^{k-1} \lambda_i u_i u_i^*.$$

Dann gilt $\operatorname{rang}(P) = k$ sowie $P^* = P$ und für alle $x \in \mathbb{K}^n$ mit $\|x\|_2 = 1$ gilt die Ungleichung

$$x^*(A-P)x \;=\; \sum_{i=k}^{n-1} \lambda_i |u_i^* x|^2 \;\le\; \lambda_k \sum_{i=k}^{n-1} |u_i^* x|^2 \;\le\; \lambda_k \sum_{i=1}^{n-1} |u_i^* x|^2 \;=\; \lambda_k.$$

Daraus folgt

$$\max_{\|x\|_2=1} x^*(A-P)x \;\le\; \lambda_k$$

und durch Bildung des Minimums über P ergibt sich $\alpha_k(A) \le \lambda_k(A)$.

Nun zeigen wir, dass auch $\alpha_j(A) \ge \lambda_j(A)$ gilt. Dazu sei eine hermitesche Matrix P mit Rang $\operatorname{rang}(P) = k$ gegeben. Aufgrund der Rangbedingung besitzt die Abbildung

$$\Phi : \operatorname{span}\{u_0, \ldots, u_j\} \to \mathbb{K}^n \qquad \text{mit} \qquad \Phi(x) \;=\; P \cdot x$$

einen nichttrivalen Nullraum, es existiert also ein Vektor $\tilde{x}$ der Form

$$\tilde{x} \;=\; \sum_{i=0}^{k} \beta_i u_i$$

mit $\beta_i \in \mathbb{K}$, mit $P\tilde{x} = 0$ und mit

$$\sum_{i=0}^{k} |\beta_i|^2 \;=\; \|\tilde{x}\|_2^2 \;=\; 1.$$

Es folgt

$$\begin{aligned} \sup_{\|x\|_2=1} x^*(A-P)x \;&\geq\; \tilde{x}^*(A-P)\tilde{x} \;=\; \tilde{x}^*A\tilde{x} \\ &=\; \sum_{i=0}^{k} \lambda_i|\beta_i|^2 \;\geq\; \lambda_k \cdot \sum_{i=0}^{k} |\beta_i|^2 \;=\; \lambda_k. \end{aligned}$$

Wieder durch Bildung des Minimums über P ergibt sich $\alpha_k(A) \geq \lambda_k(A)$, was die Behauptung schließlich zeigt. □

Wie bereits erwähnt, verwenden viele numerische Verfahren zur Berechnung der Eigenwerten einer Matrix A eine Folge, die gegen eine Diagonalmatrix konvergiert. Wir wollen nun untersuchen, welche Aussagen man nach endlich vielen Schritten des Verfahrens über die Lage der Eigenwerte von A machen kann. Dazu werden wir Abschätzungen für Eigenwerte von Matrizen herleiten, bei denen alle Nicht-Diagonalelemente klein sind.

Satz 4.8. *Seien $A, B \in \mathbb{K}^{n\times n}$ hermitesche Matrizen. Dann gilt*

$$|\lambda_k(A) - \lambda_k(B)| \;\leq\; \|A-B\|_2$$

für $k = 0, \ldots, n-1$. Insbesondere hängen die Eigenwerte einer hermiteschen Matrix damit bezüglich der Euklidischen Norm stetig von der Matrix ab.

Beweis. Sei P eine hermitesche Matrix vom Rang k und $x \in \mathbb{K}^n$ ein Vektor mit $\|x\|_2 = 1$. Dann gilt

$$x^*(A-P)x \;=\; x^*(B-P)x + x^*(A-B)x \;\leq\; x^*(B-P)x + \|A-B\|_2.$$

Bildet wir in dieser Ungleichung zuerst das Supremum über x auf der rechten Seiten und anschließend auf der linken Seite, so erhalten wir

$$\sup_{\|x\|_2=1} x^*(A-P)x \;\leq\; \sup_{\|x\|_2=1} x^*(B-P)x + \|A-B\|_2.$$

Nun bilden wir das Infimum über P, zuerst auf der linken und anschließend auf der rechten Seite. Aus dem Courantschen Minimum-Maximum-Prinzip folgt

$$\lambda_k(A) - \lambda_k(B) \;\leq\; \|A-B\|_2.$$

Durch Vertauschung der Rollen von A und B ergibt sich analog

$$\lambda_k(B) - \lambda_k(A) \;\leq\; \|A - B\|_2,$$

was die Behauptung zeigt. □

Schließlich erhalten wir noch eine weitere Aussage über die Lage der Eigenwerte von nicht notwendigerweise hermiteschen Matrizen.

Satz 4.9 (Gershgorin). *Sei $A = (a_{ij})\mathbb{K}^{n\times n}$ beliebig. Weiter definieren wir die* **Gershgorin-Kreise**

$$G_j \;:=\; \left\{ \lambda \in \mathbb{K} \;:\; |\lambda - a_{jj}| \;\leq\; \sum_{\substack{k=1\\k\neq j}}^{n} |a_{jk}| \right\}$$

für $j = 1, \ldots, n$ und

$$G_k^* \;:=\; \left\{ \lambda \in \mathbb{K} \;:\; |\lambda - a_{kk}| \;\leq\; \sum_{\substack{j=1\\j\neq k}}^{n} |a_{jk}| \right\}$$

für $k = 1, \ldots, n$. Dann gilt für alle Eigenwerte λ von A

$$\lambda \;\in\; \bigcup_{j=1}^{n} G_j \qquad \text{und} \qquad \lambda \;\in\; \bigcup_{k=1}^{n} G_k^*.$$

Beweis. Sei $Ax = \lambda x$ mit $\|x\|_\infty = 1$ und wähle einen Index j mit $|x_j| = \|x\|_\infty = 1$. Dann gilt

$$\begin{aligned} |\lambda - a_{jj}| \;&=\; |(\lambda - a_{jj})x_j| \;=\; |(Ax)_j - a_{jj}x_j| \\ &=\; \left| \sum_{\substack{k=1\\k\neq j}}^{n} a_{jk}x_k \right| \;\leq\; \sum_{\substack{k=1\\k\neq j}}^{n} |a_{jk}||x_k| \;\leq\; \sum_{\substack{k=1\\k\neq j}}^{n} |a_{jk}|. \end{aligned}$$

Daraus folgt $\lambda \in \bigcup_{j=1}^{n} G_j$. Da A^* die konjugiert komplexen Eigenwerte von A besitzt, folgt direkt auch $\lambda \in \bigcup_{k=1}^{d} G_k^*$. □

4.3 Vektoriteration

Das ***Verfahren von Mises*** ist eines der einfachsten Verfahren, welches in der Regel aber nur sehr langsam konvergiert. Trotzdem ist sein Verständnis

grundlegend für viele folgenden Methoden, so dass wir es hier besprechen werden.

Zunächst benötigen wir noch eine Definition.

Definition 4.2. Eine Matrix $A \in \mathbb{K}^{n\times n}$ heißt ***diagonalisierbar***, wenn es eine reguläre Matrix $Q \in \mathbb{K}^{n\times n}$ gibt, so dass $Q^{-1}AQ$ eine Diagonalmatrix ist.

In diesem Falle bilden die Spalten $q_1, \ldots, q_n$ von Q ein vollständiges System von Eigenvektoren von A.

Wir nehmen an, dass $A \in \mathbb{K}^{n\times n}$ diagonalisierbar ist mit $Q = (q_1, \ldots, q_n)$, es gelte also $A = QDQ^{-1}$ mit einer Diagonalmatrix D. Weiter gelte für die zugehörigen Eigenwerte

$$|\lambda_1| \; > \; |\lambda_2| \; \geq \; |\lambda_3| \; \geq \; \ldots \; \geq \; |\lambda_n|. \tag{4.5}$$

Sei nun $\xi = (\xi_1, \ldots, \xi_n) \in \mathbb{K}^n$ ein Vektor mit $\xi_1 \neq 0$ und definiere

$$x \; = \; \xi_1 q_1 + \ldots + \xi_n q_n.$$

Damit folgt

$$A^k \cdot x \; = \; \sum_{i=1}^{n} \xi_i A^k q_i \; = \; \sum_{i=1}^{n} \xi_i \lambda_i^k q_i \; = \; \xi_1 \lambda_1^k q_1 + \ldots + \xi_n \lambda_n^k q_n.$$

Aufgrund der Annahme (4.5) dominiert auf der rechten Seite für $n \to \infty$ der erste Summand, der ein Vielfaches des ersten Eigenvektors ist. Damit haben wir ein einfaches Verfahren zur Berechnung des größten Eigenwertes einer Matrix samt zugehörigem Eigenvektor gefunden, siehe Algorithmus 4.1.

Input: Diagonalisierbare Matrix A, Startvektor x_0.

$i := 0$;
`repeat`
  $x_{i+1} := 1/\|x_i\| \cdot A \cdot x_i$;
  $i := i + 1$;
`until` stop.

Output: Eigenvektor $x_1 \approx x_i$ zum Eigenwert $|\lambda_1| \approx \|x_i\|_2$.

Algorithmus 4.1: Das Verfahren von Mises.

Um einen Over- und Underflow zu vermeiden, wird dabei der Vektor x_i in jedem Schritt normiert. Wir wollen nun auch eine Konvergenzaussage beweisen.

Satz 4.10. *Sei $A \in \mathbb{K}^{n\times n}$ eine diagonalisierbare Matrix, deren Eigenwerte die Annahme (4.5) erfüllen. Weiter sei $x_0 = \sum_{j=1}^{n} \xi_j q_j$ mit $\xi_1 \neq 0$. Dann gilt*

$$\left\| x_i - \frac{\lambda_1^i}{|\lambda_1|^{i-1}} \cdot \frac{\xi_1}{|\xi_1|} \cdot \frac{q_1}{\|q_1\|_2} \right\| \;=\; \mathcal{O}\left(\left| \frac{\lambda_2}{\lambda_1} \right|^i \right) \qquad \text{für} \quad i \to \infty, \tag{4.6}$$

d.h. x_i konvergiert linear mit der Geschwindigkeit $|\lambda_2/\lambda_1|$ gegen einen Eigenvektor zum Eigenwert λ_1.

Beweis. Per Induktion nach i folgt

$$x_{i+1} \;=\; \frac{1}{\|A^i x_0\|_2} \cdot A^{i+1} \cdot x_0$$

und somit

$$x_i \;=\; \left(\frac{\lambda_1^{i+1}}{|\lambda_1|^i}\xi_1 q_1 + \ldots + \frac{\lambda_n^{i+1}}{|\lambda_1|^i}\xi_n q_n \right) / \left(\left\| \frac{\lambda_1^i}{|\lambda_1|^i}\xi_1 q_1 + \ldots + \frac{\lambda_n^i}{|\lambda_1|^i}\xi_n q_n \right\|_2 \right).$$

Die Behauptung folgt nun, da

$$\left| \frac{\lambda_j^n}{|\lambda_1|^n} \right| \;=\; \mathcal{O}\left(\left| \frac{\lambda_2}{\lambda_1} \right|^n \right)$$

für $j = 2, \ldots, n$. □

Da das Verfahren von Mises stets gegen den größten Eigenwert konvergiert, wollen wir nun ein ähnliches Verfahren für andere Eigenwerte untersuchen.

Dazu nehmen wir an, dass bereits eine Näherung μ eines Eigenwertes λ_k von A mit

$$|\mu - \lambda_k| \;<\; |\mu - \lambda_j| \qquad \text{für alle} \quad j \neq k \tag{4.7}$$

bekannt ist und wollen einen zugehörigen Eigenvektor berechnen.

Dazu betrachten wir die Matrix

$$B \;:=\; (A - \mu I)^{-1}.$$

Offenbar besitzt B dieselben Eigenvektoren wie A, und die Eigenwerte von B sind gegeben durch

$$\frac{1}{\lambda_j - \mu} \qquad \text{für alle} \quad j = 1, \ldots, n.$$

Input: Näherung μ eines Eigenwertes λ_k von A, Startvektor x_0.

$i := 0$;
`repeat`
  $x_{i+1} := 1/\|x_i\| \cdot (A - \mu I)^{-1} \cdot x_i$;
  $i := i + 1$;
`until` stop.

Output: Eigenvektor $x_k \approx x_i$ zum Eigenwert λ_k.

Algorithmus 4.2: Das Verfahren von Wieland.

Mit der Annahme (4.7) ist $1/(\lambda_k - \mu)$ der größte Eigenwert von B und somit können wir das Verfahren von Mises auf die Matrix B anwenden. Diese Methode wird als ***Verfahren von Wieland*** oder ***inverse Vektoriteration*** bezeichnet, siehe Algorithmus 4.2

Es ist zu bemerken, dass die Konvergenzgeschwindigkeit von der Näherung μ des Eigenwertes λ_k abhängt. Weiterhin lässt sich das Verfahren von Wieland durch die folgenden Beobachtungen noch verbessern.

Ist nämlich q_k ein Eigenvektor zum Eigenwert λ_k, so gilt für den Rayleigh-Quotienten aus Gleichung (4.3)

$$R_A(q_k) \;=\; \lambda_k.$$

Damit liegt es nahe, die gegebene Näherung μ von λ_k im Laufe des Algorithmus durch die Rayleigh-Quotienten $R_A(x_n)$ zu ersetzen. Dies führt zur ***Rayleigh-Quotienten-Iteration***, siehe Algorithmus 4.3.

Input: Näherung μ eines Eigenwertes λ_k von A, Startvektor x_0.

$i := 0$;
`repeat`
  $x_{i+1} := (A - \mu I)^{-1} \cdot x_i$;
  $i := i + 1$;
  $x_i := x_i/\|x_i\|2$;
  $\mu := x_i^* \cdot A \cdot x_i$;
`until` stop.

Output: Eigenvektor $x_k \approx x_i$ zum Eigenwert $\lambda_k \approx \mu$.

Algorithmus 4.3: Die Rayleigh-Quotienten-Iteration.

4.4 QR-Verfahren

Während wir im Abschnitt zuvor Verfahren zu Bestimmung eines Eigenwertes bzw. Eigenvektors kennen gelernt haben, beschäftigen wir uns nun mit Methoden, welche uns alle Eigenwerte einer Matrix liefern werden. Dazu verwenden wir die bereits mehrfach angesprochenen Ähnlichkeitstransformationen der Schur-Zerlegung aus Satz 4.3.

Lemma 4.11. *Sei $A_0 = A \in \mathbb{K}^{n\times n}$ eine beliebige Matrix. Weiter definieren wir die Folge*

$$A_k \;=\; R_{k-1}Q_{k-1},$$

wobei $A_{k-1} = Q_{k-1}R_{k-1}$ eine QR-Zerlegung von A_{k-1} ist, siehe Abschnitt 3.4. Zudem sei

$$\mathbf{Q}_k \;=\; Q_0 \cdot \ldots \cdot Q_{k-1} \qquad \text{und} \qquad \mathbf{R}_k \;=\; R_{k-1} \cdot \ldots \cdot R_0.$$

Dann gilt:

(1) $A_k = Q_{k-1}^* \cdot A_{k-1} \cdot Q_{k-1}$.

(2) $A_k = \mathbf{Q}_k^* \cdot A \cdot \mathbf{Q}_k$.

(3) $A^k = \mathbf{Q}_k \cdot \mathbf{R}_k$.

Beweis. Behauptung **(1)** folgt aus der folgenden Gleichungskette:

$$A_k \;=\; R_{k-1} \cdot Q_{k-1} \;=\; Q_{k-1}^* \cdot Q_{k-1} \cdot R_{k-1} \cdot Q_{k-1} \;=\; Q_{k-1}^* \cdot A_{k-1} \cdot Q_{k-1}.$$

Aussage **(2)** folgt per Induktion aus **(1)** und auch **(3)** ergibt sich per Induktion:

$$\begin{aligned} A^{k+1} \;&=\; A \cdot A^k \;=\; A \cdot \mathbf{Q}_k \cdot \mathbf{R}_k \;=\; \mathbf{Q}_k\mathbf{Q}_k^* A \cdot \mathbf{Q}_k \cdot \mathbf{R}_k \\ &=\; \mathbf{Q}_k \cdot A^k \cdot \mathbf{R}_k \;=\; \mathbf{Q}_k \cdot Q_k \cdot R_k \cdot \mathbf{R}_k \;=\; \mathbf{Q}_{k+1} \cdot \mathbf{R}_{k+1}, \end{aligned}$$

wobei wir auch noch **(2)** verwendet haben. □

Damit konnten wir nachweisen, dass die in Lemma 4.11 definierte Folge Ähnlichkeitstransformationen liefert. Das folgende Ergebnis liefert die Korrektheit des QR-Verfahrens, welches wir anschließend vorstellen werden. Der Beweis zu dieser folgenden Aussage ist jedoch erheblich schwieriger als das vorherige Ergebnis, daher verweisen wir den interessierten Leser hier zum Beispiel auf Hanke-Bourgeois (2006).

Satz 4.12. *Sei $A \in \mathbb{K}^{n\times n}$ diagonalisierbar und seien $\lambda_1, \ldots, \lambda_n \in \mathbb{K}$ die Eigenwerte von A. Weiter gelte*

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n| > 0$$

und $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ sei die Eigenwert-Matrix und $Q = (q_1, \ldots, q_n)$ die zugehörige Eigenvektor-Matrix, also $A = Q\Lambda Q^{-1}$. Wir nehmen weiterhin an, dass eine LU-Zerlegung $Q^{-1} = LU$ von Q^{-1} existiert.

Dann konvergiert die in Lemma 4.11 definierte Folge von Matrizen A_n gegen eine obere Dreiecksmatrix und ihre Diagonalelemente konvergieren mindestens linear gegen die Eigenwerte $\lambda_1, \ldots, \lambda_n$.

Die Voraussetzung, dass Q^{-1} eine LU-Zerlegung besitzt, ist eine Verallgemeinerung der Voraussetzung bei der Vektor-Iteration, dass der Startvektor nicht eine Linearkombination der übrigen Eigenvektoren ist. Wesentlich einschränkender ist die Voraussetzung

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n| > 0.$$

Diesie schließt insbesondere mehrfache Eigenwerte und konjugiert-komplexe Eigenwerte von rellen Matrizen aus. Andererseits gibt es im allgemeinen keine reelle Schur-Zerlegung mit einer oberen Dreiecksmatrix und bei mehrfachen Eigenwerten kann man nicht erwarten, dass das folgende Verfahren gegen eine Schur-Zerlegung konvergiert.

Trotzdem liefern und die beiden Aussagen nun ein grundlegendes Verfahren zur Berechnung aller Eigenwerte einer Matrix $A \in \mathbb{K}^{n\times n}$, sofern die Voraussetzungen aus Satz 4.12 erfüllt sind, siehe Algorithmus 4.4.

Input: Matrix $A \in \mathbb{K}^{n\times n}$.

$i := 0$;
$A_0 := A$;
`repeat`
  Berechne QR-Zerlegung $A_i = Q_i \cdot R_i$;
  $A_{i+1} := R_i \cdot Q_i$;
  $i := i + 1$;
`until` stop.

Output: Eigenwerte $x_k \approx (A_i)_{kk}$ für $k = 1, \ldots, n$.

Algorithmus 4.4: Das QR-Verfahren in seiner Grundversion.

Natürlich ist das QR-Verfahren in seiner Grundversion sehr aufwendig, da die Berechnung der QR-Zerlegung in jedem Schritt von der Größenordnung

$\mathcal{O}(n^3)$ ist. Der Aufwand kann erheblich reduziert werden, wenn wir die Matrix A zunächst durch eine unitäre Ähnlichkeitstransformation in sogenannte ***Hessenberg-Form*** bringen. Das QR-Verfahren kann nun deutlich schneller durchgeführt werden, das sich die QR-Zerlegung einer Hessenber-Matrix sehr einfach durch ***Givens-Rotationen*** bestimmen lässt. Weiterhin kann das Verfahren auch durch ***Shift-Operationen*** beschleunigt werden. Für all diese Falle verweisen wir aber wiederum auf Hanke-Bourgeois (2006).

4.5 Lanczos-Verfahren

Im Abschnitt zuvor haben wir gezeigt, dass sich mit dem QR-Verfahren sämtliche Eigenwerte einer Matrix berechnen lassen, dafür wächst der Aufwand kubisch mit der Größe der Matrix. In vielen Anwendungen, insbesondere bei Eigenwertproblemen für gewöhnliche und partielle Differentialoperatoren wie in Beispiel 4.1 kennengelernt, ist man jedoch nur an einigen wenigen extremalen, also besonders großen oder kleinen, Eigenwerten interessiert. Zudem sind die auftretenden Matrizen oft sehr groß, aber nur dünn besetzt.

Im Gegensatz zum QR-Verfahren sind wir daher an Verfahren interessiert, die nur Matrix-Vektor-Multiplikationen benötigen. Mit der Vektor-Iteration haben wir bereits ein solches Verfahren kennengelernt, allerdings konnten wir damit nur eine Näherung des betragsgrößten Eigenvektors berechnen. Wir werden nun das ***Lanczos-Verfahren*** einführen, mit welchem sich auch weitere Eigenwerte berechnen lassen.

Im folgenden sei $A \in \mathbb{K}^{n\times n}$ eine hermitesche Matrix mit Eigenwerten

$$\lambda_0(A) \;\geq\; \lambda_1(A) \;\geq\; \cdots \;\geq\; \lambda_{n-1}(A).$$

Sind keine Verwechselungen möglich, schreiben wir auch einfach nur

$$\lambda_0 \;\geq\; \lambda_1 \;\geq\; \cdots \;\geq\; \lambda_{n-1}.$$

Bei der Vektoriteration wurden Vektoren der Form

$$x_j \;=\; A^j \cdot x_0$$

berechnet und als Näherung des größten Eigenwertes λ_0 von A die Zahlen

$$\underline{\mu}^{(j+1)} \;=\; \frac{x_j^* \cdot A \cdot x_j}{x_j^* \cdot x_j}$$

verwendet, siehe Algorithmus 4.3. Weiterhin gilt nach Satz 4.6

$$\lambda_0 \;=\; \max_{x\in\mathbb{R}\setminus\{0\}} \frac{x^*Ax}{x^*x}.$$

Beim Lanczos-Verfahren betrachten wir den Krylov-Raum zur Matrix A und zum Vektor x_0, also

$$\mathcal{K}_k \;=\; (A, x_0) \;=\; \operatorname{span}\{x_0, x_1, x_2, \ldots, x_{k-1}\},$$

siehe Definition 3.15.

Die Idee das Lanczos-Verfahrens besteht nun darin, als Approximation von λ_0 das Maximum des Rayleigh-Quotienten über dem Krylov-Raum $\mathcal{K}_k$ zu verwendet:

$$\mu^k \;=\; \max_{x \in \mathcal{K}_k \setminus \{0\}} \frac{x^* A x}{x^* x}.$$

Natürlich gilt $\{x_0\} \subset \mathcal{K}_k \subset \mathbb{K}^n$ und damit erhalten wir

$$\underline{\mu}^k \;\leq\; \mu^k \;\leq\; \lambda_0.$$

Natürlich ist dieses Verfahren nur dann sinnvoll, wenn das Maximum des Rayleigh-Quotienten über dem Krylov-Raum $\mathcal{K}_k$ und damit μ^k effiziente berechnet werden kann. Genau dies wird nun das Ziel der folgenden Betrachtungen sein. Zudem werden wir auch sehen, dass wir nicht nur λ_0 approximieren, sondern gleich mehrere extremale Eigenwerte.

Wir können die Lanczos-Approximationen $\mu_0^{(k)}$ des größten Eigenwertes von A auch interpretieren, indem wir die Einschränkungen der durch A gegebenen linearen Abbildung auf den k-ten Krylov-Raum betrachten, also

$$\mathcal{A}_k : \mathcal{K}_k \to \mathcal{K}_k \qquad \text{mit} \quad \mathcal{A}_k(z) \;=\; P_k \cdot A \cdot z.$$

Dabei sei $P_k \in \mathbb{K}^{n \times n}$ die eindeutig bestimmte orthogonale Projektion auf $\mathcal{K}_k$, d.h. P_k ist hermitesch und $P_k^2 = P_k$.

Nun ist $\mathcal{A}_k$ bezüglich des inneren Produktes $\langle x, y \rangle = x^* y$ selbstadjungiert, da

$$\langle \mathcal{A}_k z, y \rangle \;=\; \langle \mathcal{A}_k P_k z, y \rangle \;=\; z^* P_k A P_k y \;=\; \langle z, \mathcal{A}_k y \rangle$$

gilt. Damit besitzt die lineare Abbildung $\mathcal{A}_k$ nur reelle Eigenwerte, die wir wieder der Größe nach anordnen:

$$\lambda_0(\mathcal{A}_k) \;\geq\; \lambda_1(\mathcal{A}_k) \;\geq\; \cdots \;\geq\; \lambda_{k-1}(\mathcal{A}_k).$$

Wieder nach Satz 4.6 erhalten wir nun

$$\lambda_0(\mathcal{A}_k) \;=\; \sup_{x \in \mathcal{K}_k \setminus \{0\}} \frac{\langle x, \mathcal{A}_k x \rangle}{\langle x, x \rangle} \;=\; \sup_{x \in \mathcal{K}_k \setminus \{0\}} \frac{x^* A x}{x^* x} \;=\; \mu^{(k)},$$

da

$$\langle x, \mathcal{A}_k x \rangle \;=\; x^* P_k A x \;=\; x^* P_k^* A x \;=\; \langle P_k x \rangle^* A x \;=\; x^* A x$$

für alle $x \in \mathcal{K}_k$. Es liegt nun nahe für $j < k$ den Wert

$$\mu_j^k \;=\; \lambda_j(\mathcal{A}_k)$$

als Approximationen des Eigenwertes λ_j zu betrachten.

Um nun die Zahlen μ_j^k berechnen zu können, benötigen wir eine Orthonormalbasis $\{v_0, \ldots, v_{k-1}\}$ des Krylov-Raums $\mathcal{K}_k$ und die Matrix T_k, die die Abbildung $\mathcal{A}_k$ bezüglich dieser Basis repräsentiert. Die Matrixeinträge von T_k sind dann gegeben durch

$$(T_k)_{ij} \;=\; \langle v_i^*, \mathcal{A}_k v_j \rangle \;=\; v_i^* A v_j, \qquad i,j = 1, \ldots, k$$

also haben wir

$$T_k \;=\; V_k^* \cdot A \cdot V_k \qquad \text{mit} \qquad V_k \;=\; (v_0\, v_1\, \cdots\, v_{k-1}) \;\in\; \mathbb{K}^{n\times k}.$$

Damit gilt

$$\mu_j^k \;=\; \lambda_j(\mathcal{A}_k) \;=\; \lambda_j(T_k)$$

und für $k \ll n$ können wir die Eigenwerte von $T_k \in \mathbb{K}^{k\times k}$ mit wenig Aufwand mit dem QR-Verfahren berechnen.

Wir haben bereits beim cg-Verfahren 3.8 den Begriff des Krylov-Raumes verwendet. In der Tat lässt sich zeigen, dass sich die Einträge der Matrix T_k aus den Zwischenresultaten des cg-Verfahrens berechnen lassen. Insbesondere ergibt sich auch, dass T_k eine Tridiagonalmatrix ist. Wenn wir $A_k V_k = V_k T_k$ spaltenweise aufschreiben und die Tridiagonalgestalt von T_k nutzen, erhalten wir den im folgenden Satz angegeben Algorithmus zu Bestimmung von T_k bzw. V_k.

Satz 4.13. *Sei $A \in \mathbb{K}^{n\times n}$ hermitesch und $v_0 \in \mathbb{K}^n \setminus \{0\}$ Zufallsvektor mit $\|v_0\|_2 = 1$ und $v_{-1} = (0, \ldots, 0)^T$.*

Dann berechnen wir iterativ für $j = 0, \ldots, k-2$

$$\begin{aligned}
\alpha_j &= v_j^* A v_j, \\
r_j &= A v_j - \alpha_j v_j - \beta_j v_{j-1}, \\
\beta_{j+1} &= \|r_j\|_2, \\
v_{j+1} &= \frac{1}{\beta_{j+1}} r_j \qquad \text{falls} \qquad \beta_{j+1} \neq 0.
\end{aligned}$$

Gilt dabei $\beta_j \neq 0$ für $j = 1, \ldots, k-1$, so folgt

$$T_k = \begin{pmatrix} \alpha_0 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_1 & \beta_2 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \beta_{k-1} \\ 0 & \cdots & 0 & \beta_{k-1} & \alpha_{k-1} \end{pmatrix}$$

und die Vektoren $v_0, \ldots, v_{k-1}$ bilden eine Orthonormalbasis von $\mathcal{K}_k$.

Dabei ist zu beachten, dass das Verfahren je nach der Wahl von v_0 auch vorzeitig abbrechen kann, wenn v_0 selbst schon ein Eigenvektor von A ist bereits im ersten Schritt. Offen lassen wir an dieser Stelle den Beweis, dass $v_0, \ldots, v_{k-1}$ wirklich eine Orthonormalbasis von $\mathcal{K}_k$ ist. Dazu verweisen wir auf Hanke-Bourgeois (2006).

Abschließend erhalten wir noch die folgende Fehlerabschätzung an das Lanczos-Verfahren. Dabei verwenden wir die Bezeichnungen aus dem Satz zuvor.

Satz 4.14. *Sei (μ, w) ein Eigenpaar von T_k mit $\|w\|_2 = 1$ und w_k die letzte Komponente von w. Dann besitzt A einen Eigenwert λ mit*

$$|\lambda - \mu| \leq \beta_k |w_k|.$$

Der Vektor $x = V_k w$ ist dann eine Approximation an den zu λ gehörenden Eigenvektor.

4.6 Singulärwertzerlegung

In diesen Abschnitt wollen wir noch eine Variante der Eigenwertzerlegung $A = U \cdot D \cdot V^*$ einer nicht notwendigerweise quadratischen Matrix A kennenlernen, bei der von links und rechts mit verschiedenen unitären Matrizen U und V multipliziert werden kann. Es wird sich zeigen, dass eine derartige Zerlegung für jede Matrix existiert.

Definition 4.3. Die ***Singulärwertzerlegung*** einer Matrix $A \in \mathbb{K}^{m \times n}$ ist

eine Faktorisierung der Form $A = V \cdot \Sigma \cdot U^*$, wobei

$$\Sigma \;=\; \left(\begin{array}{c|c} \begin{array}{ccc} \sigma_0 & & \\ & \ddots & \\ & & \sigma_{r-1} \end{array} & 0 \in \mathbb{R}^{r\times(n-r)} \\ \hline 0 \in \mathbb{R}^{(m-r)\times r} & 0 \in \mathbb{R}^{(m-r)\times(n-r)} \end{array}\right)$$

mit $\sigma_0 \geq \sigma_1 \geq \ldots \geq \sigma_{r-1} > 0$ gelte und $U = (u_0 \cdots u_{n-1}) \in \mathbb{K}^{n\times n}$ sowie $V = (v_0 \ldots v_{m-1}) \in \mathbb{K}^{m\times m}$ unitären Matrizen seien.

Die Zahlen $\sigma_0, \ldots, \sigma_{r-1}$ heißen ***Singulärwerte*** von A und die Vektoren u_j und v_j rechte bzw. linke ***Singulärvektoren*** von A.

Lemma 4.15. *Sei $A \in \mathbb{K}^{m\times n}$ beliebig. Dann gilt:*

(1) *A besitzt eine Singulärwertzerlegung.*

(2) *Die Singulärwerte von A sind eindeutig bestimmt.*

(3) *Der Rang von A ist r.*

(4) *Ist $A = V \cdot \Sigma \cdot U^*$ eine Singulärwertzerlegung von A, so sind*

$$A^* \cdot A \;=\; U \cdot (\Sigma^*\Sigma) \cdot U^* \qquad \text{und} \qquad A \cdot A^* \;=\; V \cdot (\Sigma\Sigma^*) \cdot V^*$$

Eigenwertzerlegungen von A^*A bzw. AA^*. Insbesondere sind die Spalten von U Eigenvektoren von A^*A und die Spalten von V Eigenvektoren von AA^*. Weiter gilt für $j = 0, \ldots, r-1$

$$A \cdot u_j \;=\; \sigma_j \cdot v_j \qquad \text{und} \qquad A^* \cdot v_j \;=\; \sigma_j \cdot u_j.$$

Beweis. Wir beweisen die Behauptung der Reihe nach.

(1) Sei $A^*A \;=\; U \cdot \operatorname{diag}(\lambda_0, \ldots, \lambda_{n-1}) \cdot U^*$ eine Eigenwertzerlegung mit einer unitären Matrix $U = (u_1 \cdots u_n)$. Da A^*A positiv semidefinit ist, können wir die Eigenwertzerlegung so wählen, dass

$$\lambda_0 \;\geq\; \lambda_1 \;\geq\; \ldots \;\geq\; \lambda_{n-1} \;\geq\; 0$$

gilt. Sei $r \in \mathbb{N} \cup \{0\}$ die kleinste Zahl mit $\lambda_r = 0$, wobei wir $\lambda_n = 0$ vereinbaren. Dann setzen wir $\sigma_j = \sqrt{\lambda_j}$ und $v_j = A \cdot u_j/\sigma_j$ für $j = 0, \ldots, r-1$. Da weiter

$$v_j^* \cdot v_k \;=\; \frac{1}{\sigma_j\sigma_k} \cdot u_j^* \cdot A^*A \cdot u_k \;=\; \frac{\sigma_k^2}{\sigma_j\sigma_k} \cdot u_j^* \cdot u_k \;=\; \delta_{jk}$$

für $i, k \in \{0, \ldots, r-1\}$ gilt, können wir die Vektoren v_j zu einer Orthonormalbasis $\{v_0, \ldots, v_{m-1}\}$ von $\mathbb{K}^m$ ergänzen. Mit $V = (v_1, \ldots, v_{m-1})$ erhalten wir

$$V \cdot \Sigma \cdot U^* \cdot u_j \;=\; V \cdot \Sigma \cdot e_j \;=\; V \cdot \sigma_j \cdot e_j \;=\; \sigma_j \cdot v_j \;=\; A \cdot u_j$$

für $j = 0, \ldots, r-1$ und

$$V \cdot \Sigma \cdot U^* \cdot u_j \;=\; V \cdot \Sigma \cdot e_j \;=\; 0 \;=\; A \cdot u_j$$

für $j \geq r$. Da $u_1, \ldots, u_n$ aber eine Basis des $\mathbb{K}^n$ bilden, haben wir eine Singulärwertzerlegung $A = V \cdot \Sigma \cdot U^*$ gefunden.

(2) Zunächst sind die Zahlen $\sigma_0^2, \ldots, \sigma_{r-1}^2$ als Eigenwerte von A^*A eindeutig bestimmt. Wegen der Forderung $\sigma_j > 0$ sind damit aber auch die Singulärwerte selber eindeutig durch A bestimmt.

(3) Da U und V unitär sind, gilt

$$\operatorname{rang}(A) \;=\; \operatorname{rang}(V^* \cdot \Sigma \cdot U) \;=\; \operatorname{rang}(\Sigma) \;=\; r.$$

(4) Die erste Aussage folgt direkt aus der Unitarität von U und V. und die zweite Aussage gilt wegen

$$A \cdot u_j \;=\; V \cdot \Sigma U^* \cdot u_j \;=\; V \cdot \Sigma \cdot e_j \;=\; \sigma_j \cdot V \cdot e_j \;=\; \sigma_j \cdot v_j$$

für $j = 0, \ldots, r-1$. Der Fall $A^* v_j = \sigma_j u_j$ folgt analog. □

Der Beweis von Teil **(1)** zeigt also auch, wie wir aus einer Eigenwertzerlegung von A^*A eine Singulärwertzerlegung von A konstruieren können.

Kommen wir nun auf quadratische Matrizen zurück, so sagen in vielen Situationen die Singulärwerte mehr über die Matrix aus als ihre Eigenwerte. So sind zum Beispiel alle Eigenwerte einer oberen Dreiecksmatrix mit Nullen auf der Diagonalen 0, aber die euklidische Norm einer solchen Matrix kann beliebig groß werden. Für Singulärwerte erhalten wir jedoch das folgende Ergebnis.

Satz 4.16. *Sei $A = V \cdot \Sigma \cdot U^*$ eine Singulärwertzerlegung einer Matrix $A \in \mathbb{K}^{m \times n}$. Dann gilt*

$$\|A\|_2 \;=\; \sigma_0.$$

Ist insbesondere $A \in \mathbb{K}^{n \times n}$ quadratisch und regulär, so gilt

$$\|A^{-1}\|_2 \;=\; \frac{1}{\sigma_{n-1}} \qquad \text{und} \qquad \operatorname{cond}_2(A) \;=\; \frac{\sigma_0}{\sigma_{n-1}}.$$

Beweis. Da für alle $x \neq 0$

$$\|A \cdot x\|_2^2 \;=\; x^* \cdot A^* A \cdot x \;\leq\; \|x\|_2^2 \cdot \|A^* A\|_2$$

gilt, folgt $\|A\|_2^2 \leq \|A^* A\|_2$. Da weiter $A^* A$ hermitesch und positiv definit ist, ergibt sich $\|A^* A\|_2 = \lambda_0(A^* A)$. Ist nun x_0 ein Eigenvektor zum Eigenwert $\lambda_0(A^* A)$, so gilt andererseits

$$\|A \cdot x_0\|_2^2 \;=\; x_0^* \cdot A^* A \cdot x_0 \;=\; \lambda_0(A^* A) \cdot \|x_0\|_2^2,$$

also $\|A\|_2 \geq \lambda_0(A^* A)$. Insgesamt haben wir damit

$$\|A\|_2 \;=\; \sqrt{\lambda_0(A^* A)} \;=\; \sigma_0$$

gezeigt. Die zweiten Aussagen folgen sofort, da die Singulärwerte von A^{-1} gegeben sind durch $1/\sigma_{n-1}, \ldots, 1/\sigma_0$. □

4.7 Ausblick

Natürlich gibt es noch sehr viel mehr Verfahren zur numerischen Berechnung von Eigenwerten. Wie bereits hingewiesen, lassen sichvor allem beim QR-Verfahren noch sehr viele Spezialfälle und Erweiterungen angeben, welche die Effizienz der Verfahren zum Teil deutlich steigern.

Auch sind wir hier nur auf Eigenwerte einer Matrix eingegangen, natürlich lassen sich auch Eigenwerte von Operatoren numerisch berechnen. Zum Beispiel treten auch wichtige Eigenwertaufgaben bei gewöhnlichen Differentialgleichungen auf, siehe zum Beispiel Töring and Spellucci (1988).

5 Interpolation

Bei der Interpolation geht es darum, Funktionen aus einer bestimmten Klasse von Funktionen zu finden, die an einigen Stellen mit gegebenen Werten übereinstimmt. Dabei lassen sich Interpolationsaufgaben zu vielen Klassen von Funktionen angeben. Wir werden an dieser Stelle mit der Menge der Polynome zunächst den einfachsten Fall der Interpolation betrachten. Anschließend untersuchen wir die Splineinterpolation, was in Kapitel 8 bei Randwertproblemen noch wichtig sein wird.

5.1 Polynominterpolation

Zunächst wiederholen wir kurz die grundlegenden Definition und Aussagen zu Polynomen, die aber alle bekannt sein sollten.

Definition 5.1. Ein ***Polynom*** p ist eine Funktion der Form

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

mit $x \in \mathbb{K}$ und Koeffizienten $a_0, \ldots, a_n \in \mathbb{K}$. Ist $a_n \neq 0$, so ist n der ***Grad*** des Polynoms. Der Grad von $p(x) = 0$ wird als -1 festgesetzt.

Notation 5.2. Die Menge aller Polynome vom Grad kleiner oder gleich n bezeichnen wir mit Π_n.

Aus der linearen Algebra ist bekannt, dass Π_n mit komponentenweiser Addition und Skalarmultiplikation ein Vektorraum ist. Zudem wiederholen wir den Hauptsatz der Algebra.

Satz 5.1 (Hauptsatz der Algebra). *Sei $p(x) = a_n x^n + \ldots + a_1 x + a_0$ ein komplexes Polynom vom Grad n. Dann gibt es eindeutig bestimmte Zahlen $b_1, \ldots, b_n \in \mathbb{C}$ mit $p(x) \;=\; a_n \cdot (x - b_1) \cdot \ldots \cdot (x - b_n)$. Die Zahlen b_j sind damit die Nullstellen von p.*

Kommt der Faktor $(x - b_j)$ in $p(x)$ genau k-mal vor, so sagen wir die Nullstelle

b_j hat die ***Vielfachheit*** k.

Bemerkung 5.1. Sei b eine Nullstelle von p. Dann hat b genau dann die Vielfachheit k, wenn $p^{(j)}(b) = 0$ für $j = 0, \ldots, k-1$ gilt.

Satz 5.2. *Sei* $p(x) = a_n x^n + \ldots + a_1 x + a_0 \in \Pi_n$. *Hat* p *mehr als* n *Nullstellen, so verschwindet* p *identisch, also* $p(x) = 0$ *für alle* $x \in \mathbb{C}$.

Aus diesem Satz lässt sich direkt ableiten, dass die ***Monome***

$$M_k(x) \;:=\; x^k \;\in\; \Pi_k \qquad \text{für} \quad k \;=\; 0, \ldots, n$$

linear unabhängig sind. Da wir weiterhin durch Linearkombination der M_k jedes Polynom erzeugen können, ist

$$\{M_0(x), M_1(x), \ldots, M_n(x)\}$$

eine Basis von Π_n.

Mit der Wiederholung dieser Aussagen zu Polynomen können wir nun die ***Lagrange Interpolationsaufgabe*** angeben.

Notation 5.3. Gegeben seien $n+1$ paarweise verschiedene ***Stützstellen***

$$x_0, \ldots, x_n$$

und $n+1$ ***Stützwerte***

$$y_0, \ldots, y_n.$$

Gesucht ist ein Polynom $p \in \Pi_n$ mit

$$p(x_k) \;=\; y_k \qquad \text{für} \quad k = 0, \ldots, n. \tag{5.1}$$

Als erste Idee zur Lösung dieses Problems verwenden wir die Monome als Basis des Π_n und stellt das gesuchte Polynom dar durch

$$p(x) \;=\; \sum_{k=0}^{n} \alpha_k \cdot M_k(x)$$

dar. Die Bedingungen 5.1 führen zu folgendem Gleichungssystem mit den Unbekannten $\alpha_0, \ldots, \alpha_n$:

$$\sum_{k=0}^{n} \alpha_k M_k(x_j) \;=\; y_j \qquad \text{für} \quad j = 0, \ldots, n.$$

Die Koeffizienten-Matrix zu diesem Gleichungssystem wird gegeben durch die ***Vandermonde-Matrix***

$$A = \begin{pmatrix} 1 & x_0 & {x_0}^2 & \cdots & {x_0}^n \\ 1 & x_1 & {x_1}^2 & \cdots & {x_1}^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & {x_n}^2 & \cdots & {x_n}^n \end{pmatrix}.$$

Obwohl dieses Problem im Allgemeinen schlecht konditioniert und daher für praktischer Zwecke eher ungeeignet ist, ist es dennoch von theoretischem Interesse, wie der folgende Satz zeigt.

Satz 5.3. *Die Lagrange Interpolationsaufgabe ist für $n+1$ paarweise verschiedene Stützstellen $x_0, \ldots, x_n$ eindeutig lösbar und die Lösung ist gegeben durch*

$$L_n(x) = \sum_{k=0}^{n} y_k l_k(x)$$

mit

$$l_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j} \qquad \text{für} \qquad k = 0, \ldots, n.$$

Die Polynome $l_0, \ldots, l_n \in \Pi_n$ heißen **Lagrange-Polynome**.

Beweis. Per Konstruktion ist $L_n \in \Pi_n$ und es gilt

$$l_k(x_j) = \begin{cases} 1 & \text{für} \quad k = j \\ 0 & \text{für} \quad k \neq j \end{cases}$$

Damit erhalten wir

$$L_n(x_j) = \sum_{k=0}^{n} y_k l_k(x_j) = \sum_{k=0}^{n} y_k \delta_{k_j} = y_j$$

für $j = 0, \ldots, n$ und somit löst $L_n(x)$ die Interpolationsaufgabe.

Es bleibt noch die Eindeutigkeit zu zeigen. Seien dazu $p_1, p_2 \in \Pi_n$ zwei Polynome, die beide die Interpolationsaufgabe erfüllen. Dann gilt für ihre Differenz $p(x) = p_1(x) - p_2(x)$

$$p(x_j) = p_1(x_j) - p_2(x_j) = y_j - y_j = 0 \qquad \text{für} \qquad j = 0, \ldots, n$$

und somit hat p mindestens $n+1$ Nullstellen. Da aber $p \in \Pi_n$ gilt, folgt $p(x) = 0$ und damit $p_1(x) = p_2(x)$. □

Beispiel 5.1. *Gegeben seien die drei Stützstellen $x_0 = 0$, $x_1 = 1$ und $x_2 = 3$ mit den Stützwerten $y_0 = 1$, $y_1 = 3$ und $y_2 = 2$. Dann wird die Interpolationsaufgabe gelöst durch*

$$L_2(x) = \sum_{k=0}^{2} y_k l_k(x)$$

mit

$$\begin{aligned} l_0(x) &= \frac{(x-1)(x-3)}{(0-1)(0-3)} = \frac{1}{3} \cdot (x-1) \cdot (x-3) \\ l_1(x) &= \frac{(x-0)(x-3)}{(1-0)(1-3)} = -\frac{1}{2} \cdot x \cdot (x-3) \\ l_2(x) &= \frac{(x-0)(x-1)}{(3-0)(3-1)} = \frac{1}{6} \cdot x \cdot (x-1). \end{aligned}$$

Praktisch hat die Lagrange-Formel allerdings wenig Relevanz, da die Hinzunahme einer weiteren Stützstelle eine komplette Neuberechnung erfordert. Zudem ist das Gleichungssystem wie oben bereits erwähnt schlecht konditioniert. Daher untersuchen wir nun eine weitere Basis.

Bei den ***Newtonschen Interpolationsformeln*** verwenden wir als Basis des Π_n die Funktionen

$$h_k(x) = \prod_{i=0}^{k-1} (x - x_i) \qquad \text{für} \quad k = 0, \ldots, n,$$

wobei $x_0, x_1, \ldots, x_n$ wieder die Stützstellen sind.

Lemma 5.4. *Seien $x_0, \ldots, x_{n-1}$ paarweise verschiedene Stützstellen. Dann bilden die Newton-Polynome*

$$h_k(x) = \prod_{i=0}^{k-1} (x - x_i) \qquad \text{für} \quad k = 0, \ldots, n$$

eine Basis des Π_n.

Die Newton-Polynome $h_0(x)$ bis $h_2(x)$ haben also folgendes Aussehen:

$$\begin{aligned} h_0(x) &= 1, \\ h_1(x) &= (x - x_0), \\ h_2(x) &= (x - x_1)(x - x_0). \end{aligned}$$

Beweis. Sei

$$p(x) \;=\; \sum_{k=0}^{n} \alpha_k h_k(x) \;=\; \sum_{k=0}^{n} \alpha_k \prod_{i=0}^{k-1} (x - x_i) \;=\; 0$$

Insbesondere gilt dann

$$0 \;=\; p(x_0) \;=\; \alpha_0 h_0(x) \;=\; \alpha_0$$

und daraus folgern wir

$$0 \;=\; p(x_1) \;=\; \alpha_0 + \alpha_1(x_1 - x_0) \;=\; \alpha_1 \underbrace{(x_1 - x_0)}_{\neq 0},$$

also $\alpha_1 = 0$. Induktiv erhalten wir, dass alle Koeffizienten $\alpha_0, \ldots, \alpha_n$ Null sind. □

Um das Lagrange-Interpolationsproblem mit den Newton-Polynomen zu lösen, betrachtet wir also das Gleichungssystem

$$\sum_{k=0}^{n} \alpha_k h_k(x_j) \;=\; y_j \qquad \text{für} \quad j = 0, \ldots, n. \tag{5.2}$$

mit den Unbekannten $\alpha_0, \ldots, \alpha_n$. Da aber

$$\sum_{k=0}^{n} \alpha_k h_k(x_j) \;=\; \sum_{k=0}^{j} \alpha_k h_k(x_j)$$

gilt, erhalten wir die Koeffizienten-Matrix

$$A \;=\; \begin{pmatrix} h_0(x_0) & 0 & \cdots & \cdots & 0 \\ h_0(x_1) & h_1(x_1) & 0 & \cdots & 0 \\ h_0(x_2) & h_1(x_2) & h_2(x_2) & \cdots & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ h_0(x_n) & h_1(x_n) & \cdots & \cdots & h_n(x_n) \end{pmatrix}.$$

Es sei bemerkt, dass A eine untere Dreiecks-Matrix ist, die wegen $h_k(x_k) \neq 0$ für alle $k = 0, \ldots, n$ regulär ist. Wir können die gesuchten Koeffizienten also durch Vorwärtselimination bestimmen. Dies ergibt

$$\begin{aligned} \alpha_0 &= \frac{y_0}{h_0(x_0)} \;=\; \frac{y_0}{1} \;=\; y_0, \\ \alpha_1 &= \frac{1}{h_1(x_1)} \cdot (y_1 - \alpha_0 h_0(x_1)) \;=\; \frac{1}{x_1 - x_0} \cdot (y_1 - y_0), \\ \alpha_2 &= \frac{1}{h_2(x_2)} \cdot (y_2 - \alpha_1 h_1(x_2) - \alpha_0 h_0(x_2)) \\ &= \frac{1}{(x_2 - x_0)(x_2 - x_1)} \cdot \left(y_2 - \frac{1}{x_1 - x_0} \cdot (y_1 - y_0) - y_0 \right) \end{aligned}$$

und so weiter. Da diese Formeln aber recht mühsam zu bestimmen werden, gehen wir einen anderen Weg. Dazu definieren wir zunächst die ***Abschnittspolynome*** P_i^k.

Notation 5.4. Gegeben seien paarweise verschiedene Stützstellen x_0 bis x_n und Stützwerte y_0 bis y_n. Dann bezeichne $P_i^k \in \Pi_k$ das Polynom mit der Eigenschaft

$$P_i^k(x_j) \;=\; y_j \qquad \text{für} \quad i \le j \le i+k.$$

Insbesondere ist P_0^n das Interpolationspolynom zu allen Daten.

Wir bemerken, dass P_i^k nach Satz 5.3 eindeutig bestimmt ist. Untersuchen wir speziell die Polynome P_0^k, so erhalten wir nach Konstruktion

$$P_0^k(x) \;=\; \sum_{j=0}^{k} \alpha_j h_j(x) \;=\; \alpha_0 + \alpha_1(x-x_0) + \ldots + \alpha_k(x-x_0)\cdot\ldots\cdot(x-x_{k-1}).$$

Weiter gelten die folgenden Eigenschaften.

Lemma 5.5. *Es gilt:*

(1) $P_0^{k+1}(x) = P_0^k(x) + \alpha_{k+1} h_{k+1}(x)$ *für* $k = 0, \ldots, n-1$.

(2) α_k *ist der Koeffizient von* x^k *im Polynom* $P_0^k(x)$ *für* $k = 0, \ldots, n$.

Dies führt uns zur folgenden Definition.

Definition 5.5. Gegeben seien paarweise verschiedene Stützstellen x_0 bis x_n und Stützwerte y_0 bis y_n.

Dann definieren wir die ***dividierten Differenzen*** rekursiv durch

$$\begin{aligned} D_i^0 \;&:=\; y_i && \text{für} \quad i = 0, \ldots, n, \\ D_i^k \;&:=\; \tfrac{D_{i+1}^{k-1} - D_i^{k-1}}{x_{i+k} - x_i} && \text{für} \quad i = 0, \ldots, n-k \;\text{ und }\; k = 1, \ldots, n. \end{aligned}$$

Die Hauptaussage des folgenden Satzes liefert uns damit $\alpha_k = D_0^k$.

Satz 5.6. *Es gilt*

$$P_i^k(x) \;=\; D_i^0 + D_i^1(x-x_i) + \ldots + D_i^k(x-x_i)\cdot\ldots\cdot(x-x_{i+k-1}),$$

also $P_i^k(x_j) = y_j$ *für* $j = i, \ldots, k+i$. *Insbesondere gilt*

$$P_0^n(x) \;=\; \sum_{j=0}^{n} D_0^j h_j(x)$$

und damit ist $P_0^n(x)$ die Lösung der Lagrange Interpolationsaufgabe.

Da der etwas längliche Beweis keine neuen Erkenntnisse liefert, werden wir ihn hier nicht vorführen und es sei auf Schöbel (2006) verwiesen.

Wir wollen nun ein Schema angeben, mit dem wir D_i^k effizient berechnen können. Dazu ordnern wir die dividierten Differenzen nach dem folgenden Schema:

$$
\begin{array}{llccccc}
x_0 & y_0 = D_0^0 & \searrow & & & & \\
 & & & D_0^1 & \searrow & & \\
x_1 & y_1 = D_1^0 & \nearrow & & & D_0^2 & \searrow \\
 & & & D_1^1 & \nearrow & & D_0^3 \\
x_2 & y_2 = D_2^0 & \nearrow & & & D_1^2 & \nearrow \\
 & & & D_2^1 & \nearrow & & \\
x_3 & y_3 = D_3^0 & \nearrow & & & &
\end{array}
$$

Beispiel 5.2. *Verwenden wir auch hier das Beispiel mit den Stützstellen $x_0 = 0$, $x_1 = 1$ und $x_2 = 3$ und den Stützwerten $y_0 = 1$, $y_1 = 3$ und $y_2 = 2$, so erhalten wir*

$$
\begin{array}{llccccc}
0 & 1 = D_0^0 & \searrow & & & & \\
 & & & D_0^1 = 2 & \searrow & & \\
1 & 3 = D_1^0 & \nearrow & & & D_0^2 = \frac{-5/2}{3} = \frac{-5}{6} & \\
 & & & D_1^1 = \frac{-1}{2} & \nearrow & & \\
3 & 2 = D_2^0 & \nearrow & & & &
\end{array}
$$

und somit

$$
\begin{aligned}
P_0^2(x) &= 1 + 2 \cdot (x - x_0) - \frac{5}{6}(x - x_0)(x - x_1) \\
&= 1 + 2x - \frac{5}{6}x(x - 1).
\end{aligned}
$$

Bemerkung 5.2. Im Gegensatz zur Konstruktion des interpolierenden Polynoms über Lagrange-Polynome führt eine Hinzunahme einer weiteren Stützstelle nicht zur kompletten Neuberechnung des Schemas. Es müssen lediglich die neuen dividierten Differenzen D_{n+1}^0, D_n^1, ..., D_1^n, D_0^{n+1} berechnet werden.

Abschließend müssen wir uns noch mit dem Interpolationsfehler beschäftigen. Dazu sei im folgenden $f : [a, b] \to \mathbb{R}$ eine Funktion, welche die Stützwerte liefert. Mit $x_0, \ldots, x_n \in [a, b]$ gelte also $y_k = f(x_k)$ für $k = 0, \ldots, n$. Die Lösung der Interpolationsaufgabe zu den Stützstellen $x_0, \ldots, x_n \in [a, b]$ und den Stützwerten $f(x_0), \ldots, f(x_n)$ bezeichnen wir mit $L_n f \in \Pi_n$, es gelte also

$$(L_n f)(x_i) \;=\; f(x_i)$$

für $i = 0, \ldots, n$.

Den Operator

$$L_n : C[a, b] \to \Pi_n$$

nennen wir den ***Lagrange-Interpolationsoperator***. Von Interesse ist nun der ***Interpolationsfehler***

$$R_n f(x) \;:=\; f(x) - L_n f(x).$$

Dabei möchten wir die größtmögliche Differenz zwischen $f(x)$ und $L_n f(x)$ untersuchen.

Notation 5.6. Für eine Funktion $g : [a, b] \to \mathbb{R}$ bezeichnen wir mit

$$\|g\|_\infty \;:=\; \max\{|g(x)| \;:\; x \in [a, b]\}$$

den betragsmäßig größten Funktionswert aus dem Intervall $[a, b]$.

Satz 5.7. *Sei $f : [a, b] \to \mathbb{R}$ eine $(n+1)$-mal stetig differenzierbare Funktion. Dann hat*

$$R_n f(x) \;=\; f(x) - L_n f(x).$$

bei der Polynominterpolation an den $n + 1$ paarweise verschiedenen Stützstellen $x_0, \ldots, x_n \in [a, b]$ die Darstellung

$$(R_n f)(x) \;=\; \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (x - x_k),$$

dabei ist ξ ein von x abhängiger Wert aus $[a, b]$.

Beweis. Sei $x = x_j$ für ein $j \in \{0, 1, \ldots, n\}$. Dann 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 ein festes $x \in [a, b]$ mit $x \neq x_k$ für alle $k = 0, \ldots, n$ definieren wir die Funktion $g : [a, b] \to \mathbb{R}$ durch

$$g(y) \;=\; f(y) - (L_n f)(y) - h_{n+1}(y)\frac{f(x) - (L_n f)(x)}{h_{n+1}(x)}.$$

Diese Funktion ist $(n + 1)$-mal stetig differenzierbar und hat die $(n + 2)$ Nullstellen

$$x, x_0, \ldots, x_n.$$

Der Satz von Rolle besagt nun, dass es zu je zwei Nullstellen x_a und 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 $r = 0, 1, \ldots, n + 1$.

Somit hat auch $g^{(n+1)}$ eine Nullstelle auf $[a, b]$ und diese bezeichnen wir mit ξ. Es folgt

$$0 \;=\; g^{(n+1)}(\xi) \;=\; f^{(n+1)}(\xi) - (n + 1)!\frac{(R_n f)(x)}{h_{n+1}(x)},$$

denn mit $L_n f \in \Pi_n$ folgt $L_n f^{(n+1)} = 0$. Der Term $(n + 1)!$ ergibt sich, da $h_{n+1} \in \Pi_{n+1}$ und als höchsten Koeffizienten Eins hat. Damit folgt die Behauptung. □

Aus der Darstellung des Restglieds ergibt sich sofort folgende Abschätzung

Korollar 5.8. *Sei $f : [a, b] \to \mathbb{R}$ mindestens $(n+1)$-mal stetig differenzierbar und seien $x_0, \ldots, x_n$ paarweise verschiedene Stützstellen. Dann gilt*

$$\|R_n f\|_\infty \;=\; \|f - L_n f\|_\infty \;\leq\; \frac{1}{(n + 1)!}\|h_{n+1}\|_\infty \|f^{n+1}\|_\infty.$$

Mit $\|h_{n+1}\|_\infty \leq (b - a)^{n+1}$ folgt schließlich

$$\|R_n f\|_\infty \;=\; \|f - L_n f\|_\infty \;\leq\; \frac{(b - a)^{n+1}}{(n + 1)!}\|f^{(n+1)}\|_\infty.$$

Leider sind die Voraussetzungen normalerweise nicht erfüllt. Bei nur stetigen Funktionen gilt die Aussage des Satzes nicht.

Beispiel 5.3. *Sei $k \in \{0, \ldots, n\}$ und*

$$f(x) \;=\; \begin{cases} x \sin \frac{\pi}{x} & \text{für} \quad x \in (0,1] \\ 0 & \text{für} \quad x = 0 \end{cases} .$$

Mit $x_k = \frac{1}{k+1}$ ist wegen $f(x_k) = 0$ das Interpolationspolynom

$$L_n f(x) \;=\; 0$$

für alle $n \in \mathbb{N}$. Die Folge der Interpolationspolynome konvergiert also ausschließlich an den Stützstellen x_k gegen die Funktion f und der Fehler

$$\|R_n f\|_\infty \;=\; \|f - L_n f\|_\infty$$

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.

5.2 Spline-Interpolation

Wir haben 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 wie wir in Kapitel 8 sehen werden.

Die folgenden Inhalte in diesem Abschnitt sind zum großen Teil dem Skript Kress (2007) entnommen.

Definition 5.7. Sei $a = x_0 < x_1 < \ldots < x_n = b$ eine Unterteilung des Intervalls $[a, b]$ und $m \in \mathbb{N}$. Dann heißt eine Funktion

$$s : [a, b] \to \mathbb{R}$$

ein ***Spline*** vom Grad n, falls die folgenden beiden Bedingungen erfüllt sind:

(1) s ist $m - 1$ mal stetig differenzierbar auf $[a, b]$, also $s \in \mathcal{C}^{m-1}([a, b])$.

(2) Es gilt $s|_{[x_{j-1},x_j]} \in \Pi_m$ für $j = 1, \ldots, n$.

Die Menge aller Splines vom Grad m zu einer Unterteilung

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

mit $n+1$ Stützstellen wird mit $S_m^n([a,b])$ oder kurz S_m^n bezeichnet.

Für $m = 1$ bezeichnen wir die Splines als ***linear***, für $m = 2$ als ***quadratisch*** und für $m = 3$ als ***kubisch***.

Mit dieser Definition können wir die ***Spline Interpolationsaufgabe*** wie folgt angeben.

Notation 5.8. Gegeben sei mit $a = x_0 < x_1 < \ldots < x_n = b$ eine Unterteilung von $[a,b]$, eine Zahl $m \in \mathbb{N}$ sowie die $n+1$ ***Stützwerte*** $y_0, \ldots, y_n$. Gesucht ist ein Spline $s \in S_m^n$ mit

$$s(x_k) \;=\; y_k \qquad \text{für} \qquad k = 0, \ldots, n. \tag{5.3}$$

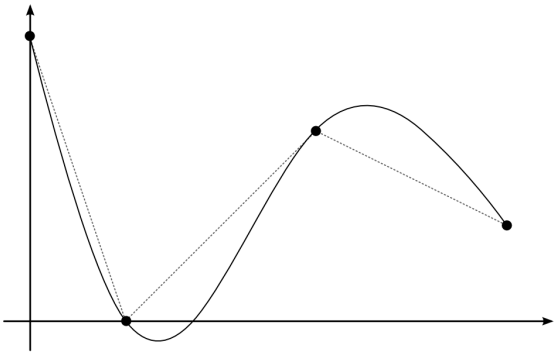


Abbildung 5.1: Beispiel einer Spline Interpolationsaufgabe: Linearer Spline (gestrichelt) und kubischer Spline (durchgezogen).

Lineare Splines sind stetige und stückweise lineare Funktionen. Die zugehörige Interpolationsaufgabe wird durch die lineare Verbindung der Punkte gelöst. Der wichtigste Fall sind kubische Splines, also Splines vom Grad 3 zunächst beschäftigen wir und mit der Lösbarkeit sowie der Eindeutigkeit der Spline Interpolationsaufgabe.

Satz 5.9. *S_m^n ist ein linearer Raum mit Dimension $n+m$.*

Beweis. Unter Verwendung der folgenden Bezeichnung

$$x_+^m \;:=\; \begin{cases} x^m & \text{für} \quad x \geq 0, \\ 0 & \text{für} \quad x < 0 \end{cases}$$

definieren wir die Funktionen

$$\begin{aligned} u_k(x) &:= (x-x_0)^k \qquad \text{für} \quad k=0,\ldots,m, \\ v_k(x) &:= (x-x_k)_+^m \qquad \text{für} \quad k=1,\ldots,n-1. \end{aligned}$$

Diese Funktionen sind linear unabhängig, denn aus

$$\sum_{k=0}^{m} \alpha_k u_k(x) + \sum_{k=1}^{n-1} \beta_k v_k(x) \;=\; 0$$

folgt insbesondere

$$\sum_{k=0}^{m} \alpha_k (x-x_0)^k k \;=\; 0$$

für alle $x \in [x_0, x_1]$. Damit ergibt sich sofort $\alpha_k = 0$ für $k = 0, \ldots, m$. Weiter folgt aus

$$\beta_1 (x-x_1)^m \;=\; 0$$

für $x \in [x_1, x_2]$ auch $\beta_1 = 0$ und fortfahrend $\beta_k = 0$ für $k = 1, \ldots, n-1$.

Wir wollen nun noch zeigen, dass die oben definierten Funktionen auch ein Erzeugendensystem und damit eine Basis von S_m^n darstellen. Dazu zeigen wir durch Induktion die Darstellung

$$s(x) \;=\; \sum_{k=0}^{m} \alpha_k (x-x_0)^k + \sum_{j=1}^{i-1} \beta_k (x-x_k)_+^m \tag{5.4}$$

für $x \in [x_0, x_i]$. Der Induktionsanfang $i = 1$ ist klar, da s auf $[x_0, x_1]$ aus Π_m ist. Wir nehmen an, dass eine Darstellung der Form (5.4) für ein $i \geq 1$ existiert. Dann beschreibt

$$p(x) \;:=\; s(x) - \sum_{k=0}^{m} (x-x_0)^k + \sum_{j=1}^{i-1} \beta_k (x-x_k)_+^m$$

auf dem Intervall $[x_i, x_{i+1}]$ ein Polynom aus Π_m. Da der Spline $m-1$ mal stetig differenzierbar ist, muss

$$p^{(j)}(x_i) \;=\; 0 \qquad \text{für} \quad j = 0, \ldots, m-1$$

gelten und dies bedeutet

$$p(x) \;=\; \beta_i (x-x_i)_+^m$$

auf $[x_i, x_{i+1}]$. Da nun aber $(x-x_i)_+^m = 0$ auf $[x_0, x_1]$ gilt, ist die Darstellung auch für $i+1$ gezeigt. □

Wir stellen die Basis von S_m^n noch einmal zusammen.

Notation 5.9. Die Funktionen

$$\begin{aligned} u_k(x) &:= (x - x_0)^k \qquad \text{für} \quad k = 0, \ldots, m, \\ v_k(x) &:= (x - x_k)_+^m \qquad \text{für} \quad k = 1, \ldots, n-1. \end{aligned}$$

sind die ***Kardinalsplines*** von S_m^n.

Bei der Spline Interpolationsaufgabe haben wir $n+1$ Stützstellen, der Raum S_m^n hat aber die Dimension $n+m$. Somit können wir für die übrigen $m-1$ *Freiheitsgrade* bei der Interpolation zusätzliche Bedinungen am Rand des Intervalls $[a, b]$ fordern. Da wir diese gleichmäßig auf beide Intervallenden verteilen wollen, betrachten wir hier nur ungerade m.

Satz 5.10. *Sei $m = 2l - 1$ mit $l \in \mathbb{N}$ sowie $f \in \mathcal{C}([a, b])$. Weiter sei $s \in S_m^n$ ein interpolierende Spline mit*

$$s(x_j) \;=\; f(x_j) \qquad \text{für} \quad j = 0, \ldots, n.$$

Zusätzlich erfülle $s(x)$ die $m-1$ Randbedingungen

$$s^{(j)}(a) \;=\; f^{(j)}(a) \qquad \text{und} \qquad s^{(j)}(b) \;=\; f^{(j)}(b)$$

für $j = 1, \ldots, l-1$. Dann gilt

$$\int_a^b (f^{(l)}(x) - s^{(l)}(x))^2 \,\mathrm{d}x \;=\; \int_a^b (f^{(l)}(x))^2 \,\mathrm{d}x - \int_a^b (s^{(l)}(x))^2 \,\mathrm{d}x$$

Der Beweis lässt sich durch zweifache partielle Integration unter Berücksichtigung der Randbedingungen führen, siehe Kress (2007).

Nun untersuchen wir die Eindeutigkeit der Spline Interpolation und benötigen dazu eine kleine Vorarbeit.

Lemma 5.11. *Unter den Voraussetzungen von Satz 5.10 sei $f(x) = 0$. Dann gilt $s(x) = 0$.*

Beweis. Für $f(x) = 0$ folgt nach Satz 5.10

$$\int_a^b (s^{(l)}(x))^2 \,\mathrm{d}x \;=\; 0.$$

Hieraus ergibt sich $s^{(l)}(x) = 0$ und daher $s \in \Pi_{l-1}$ auf $[a, b]$. Die Randbedingungen liefern nun $s(x) = 0$. □

Satz 5.12. *Sei $m = 2l - 1$ mit $l \geq 2$ sowie $a_1, b_1, \ldots, a_l, b_l \in \mathbb{R}$. Weiter sei $s \in S_m^n$ ein interpolierende Spline mit*

$$s(x_j) \;=\; y_j \qquad \text{für} \qquad j = 0, \ldots, n.$$

Zusätzlich erfülle s die $m - 1$ Randbedingungen

$$s^{(j)}(a) \;=\; a_j \qquad \text{und} \qquad s^{(j)}(b) \;=\; b_j$$

für $j = 1, \ldots, l - 1$. Dann ist s eindeutig bestimmt.

Beweis. Unter Verwendung der Kardinalsplines erhalten wir

$$s(x) \;=\; \sum_{k=0}^{m} \alpha_k u_k(x) + \sum_{j=1}^{i-1} \beta_k v_k(x).$$

Die Interpolationsaufgabe sowie die Randbedingungen sind damit genau dann erfüllt, wenn die $m+n$ Koeffizienten $\alpha_0, \ldots, \alpha_m, \beta_1, \ldots, \beta_{n-1}$ das System

$$\begin{aligned}
\sum_{k=0}^{m} \alpha_k u_k(x_j) + \sum_{j=1}^{i-1} \beta_k v_k(x_j) &= y_j \qquad \text{für} \qquad j = 0, \ldots, n, \\
\sum_{k=0}^{m} \alpha_k u_k^{(j)}(a) + \sum_{j=1}^{i-1} \beta_k v_k^{(j)}(a) &= a_j \qquad \text{für} \qquad j = 0, \ldots, l-1, \\
\sum_{k=0}^{m} \alpha_k u_k^{(j)}(b) + \sum_{j=1}^{i-1} \beta_k v_k^{(j)}(b) &= b_j \qquad \text{für} \qquad j = 0, \ldots, l-1
\end{aligned}$$

aus $m + n$ linearen Gleichungen lösen. Nach Lemma 5.11 besitzt die homogene Form dieses Gleichungssystems nur die triviale Lösung und daher ist das inhomogene Gleichungssystem eindeutig lösbar. □

Zur Berechnung der interpolierende Splines könnten wir die Kardinalsplines und das Gleichungssystem des vorherigen Satzes verwenden. Dieses System ist jedoch aufgrund der gloablen Struktur der Kardinalsplines schlecht konditioniert.

Daher verwenden wir hier die ***B-Splines***, welche einen lokalen Träger haben. Zur Vereinfachung betrachten wir hier nur eine äquidistante Zerlegungen, also

$$x_k \;=\; a + hk \qquad \text{für} \qquad j = 0, \ldots, n$$

mit $h = (b - a)/n$.

Definition 5.10. Die ***B-Splines*** werden rekursiv definiert durch

$$B_{m+1}(x) \;:=\; \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} B_m(y)\,\mathrm{d}y$$

mit

$$B_0(x) \;:=\; \begin{cases} 1 & \text{für} \quad |x| \le \frac{1}{2} \\ 0 & \text{für} \quad |x| > \frac{1}{2} \end{cases} .$$

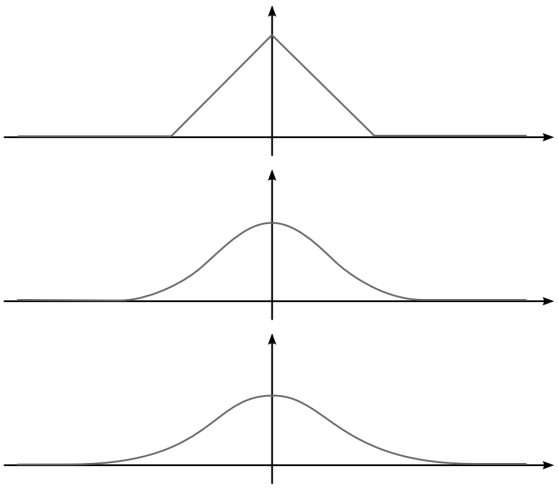


Abbildung 5.2: Die Funktionen $B_0(x)$, $B_1(x)$ und $B_2(x)$.

Per Induktion lassen sich leicht die folgenden Aussagen zeigen.

Lemma 5.13. *Für die B-Splines gelten die folgenden Aussagen.*

(1) $B_m(x) \ge 0$ *für alle* $x \in \mathbb{R}$ *und* $m \ge 0$.

(2) $B_m(x)$ *ist* $m-1$ *mal stetig differenzierbar für* $m \ge 1$.

(3) $B_m(x) = 0$ *für alle* $x \in [-m/2 - 1/2, m/2 + 1/2]$ *und* $m \ge 0$.

(4) $B_m(x)$ *reduzieren sich zu Polynomen aus* Π_m *für* $m \ge 0$ *auf den Teilintervallen* $[i, i+1]$ *für* m *ungerade und auf den Teilintervallen* $[i - 1/2, i + 1/2]$ *für* m *gerade,* $i \in \mathbb{Z}$.

Somit sind die $B_m(x)$ tatsächlich Splines vom Grad m. Durch elementare Rechnungen ergibt sich

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

$$B_2(x) = \frac{1}{2} \cdot \begin{cases} 2 - (|x| - \frac{1}{2})^2 - (|x| + \frac{1}{2})^2 & \text{für } |x| \leq \frac{1}{2} \\ (|x| - \frac{3}{2})^2 & \text{für } \frac{1}{2} \leq |x| \leq \frac{3}{2} \\ 0 & \text{für } |x| > \frac{3}{2} \end{cases},$$

$$B_3(x) = \frac{1}{6} \cdot \begin{cases} (2 - |x|)^3 - 4(1 - |x|)^3 & \text{für } |x| \leq 1 \\ (2 - |x|)^3 & \text{für } 1 \leq |x| \leq 2 \\ 0 & \text{für } |x| > 2 \end{cases}.$$

Satz 5.14. *Für $m \in \mathbb{N} \cup \{0\}$ sind die B-Splines $B(x-k)$ für $k = 0, \ldots, m$ linear unabhängig auf dem Intervall*

$$I_m := \left[\frac{m-1}{2}, \frac{m+1}{2}\right].$$

Beweis. Wieder führen wir eine vollständige Induktion über m durch. Der Induktionsanfang $m = 0$ ist trivial. Wir nehmen an, die Behauptung sei bewiesen für den Grad $m - 1$ mit $m \geq 1$. Sei nun

$$\sum_{k=0}^{m} \alpha_k B_m(x-k) = 0$$

für $x \in I_m$. Nach Definition der B-Splines erhalten wir durch Differentiation

$$\sum_{k=1}^{m} \alpha_k \cdot \left(B_{m-1}\left(x - k + \frac{1}{2}\right) - B_{m-1}\left(x - k - \frac{1}{2}\right)\right)$$

mit $x \in I_m$. Nutzen wir nun, dass die Träger der Funktionen

$$B_{m-1}\left(x + \frac{1}{2}\right) \qquad \text{und} \qquad B_{m-1}\left(x - m - \frac{1}{2}\right)$$

einen mit I_m leeren Durchschnitt haben, können wir die Summe umformulieren in

$$\sum_{k=1}^{m} (\alpha_k - \alpha_{k-1}) \cdot B_{m-1}\left(x - k + \frac{1}{2}\right)$$

für $x \in I_m$. Hieraus folgt $\alpha_k = \alpha_{k-1}$ für $k = 1, \ldots, m$ aufgrund der Induktionsannahme und daher $\alpha_k = \alpha$ für $k = 0, \ldots, m$. Es folgt

$$\alpha \cdot \sum_{k=0}^{m} B_m(x-k) = 0$$

für $x \in I_m$ und durch Integration über I_m ergibt sich

$$\alpha \cdot \int_{-\frac{m}{2}-\frac{1}{2}}^{\frac{m}{2}+\frac{1}{2}} B_m(x)\,\mathrm{d}x = 0.$$

Dies impliziert schließlich $\alpha = 0$, da die B_m nicht negativ sind. □

Diese Vorarbeit liefert uns nun eine Basis des S_m^n aus B-Splines.

Satz 5.15. *Sei $x_k = a + hk$ für $k = 0, \ldots, n$ mit $h = (b-a)/n$ eine äquidistante Zerlegung von $[a, b]$ mit $n \geq 2$ und sei $m = 2l - 1$ mit $l \in \mathbb{N}$. Dann bilden die B-Splines*

$$B_{m,k}(x) \;:=\; B_m\left(\frac{x - x_k}{h}\right)$$

für $k = -l + 1, \ldots, n + l - 1$ eine Basis von S_m^n.

Beweis. Offenbar gilt $B_{m,k} \in S_m^n$ und nach Satz 5.14 erhalten wir die lineare Unabhängigkeit auf $[a, b]$. Die Behauptung folgt dann aus Satz 5.9. □

Wir wollen nun zeigen, dass die $n + m$ B-Splines $B_{m,k}$ tatsächlich eine geeignete Basis von S_m^n sind, um die Spline Interpolationsaufgabe zu lösen.

Dazu betrachten wir den wichtigsten Fall $m = 3$ mit $l = 1$, also kubische Splines. Zunächst ergibt sich

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

Damit erfüllt der kubische Spline

$$s(x) \;=\; \sum_{k=-1}^{n+1} \alpha_k B_3\left(\frac{x - x_k}{h}\right)$$

die Spline Interpolationsaufgabe unter den Randbedingungen aus Satz 5.12 genau dann, wenn die $n + 3$ Koeffizienten $\alpha_{-1}, \ldots, \alpha_{n+1}$ das Gleichungssystem

$$\begin{aligned}
-\frac{1}{2}\alpha_{-1} + \frac{1}{2}\alpha_1 &= ha_1 \\
\frac{1}{6}\alpha_{j-1} + \frac{2}{3}\alpha_j + \frac{1}{6}\alpha_{j+1} &= y_j \qquad \text{für} \quad j = 0, \ldots, n, \\
-\frac{1}{2}\alpha_{n-1} + \frac{1}{2}\alpha_{n+1} &= hb_1
\end{aligned}$$

aus $n + 3$ linearen Gleichungen lösen. Es sei bemerkt, dass das Jacobi-Verfahren für jeden Startwert gegen die eindeutig Lösung dieses Gleichungssystems konvergiert.

Abschließend müssen wir uns noch mit dem Interpolationsfehler beschäftigen. Der folgende Satz liefert schließlich eine weitaus stärkere Aussage als bei der Polynominterpolation.

Satz 5.16. *Sei $f : [a, b] \to \mathbb{R}$ zweimal stetig differenzierbar und sei $s \in S_3^n$ der eindeutig bestimmte kubische Spline, der die Interpolationsaufgabe unter den Randbedingungen aus Satz 5.12 erfüllt. Dann gilt*

$$\|f - s\|_\infty \;\leq\; \frac{h^{3/2}}{2} \cdot \|f''\|_2 \qquad \text{und} \qquad \|f' - s'\|_\infty \;\leq\; h^{1/2} \cdot \|f''\|_2$$

mit $h = \max\{(x_j - x_{j-1}) \;:\; j = 1, \ldots, n\}$

Beweis. Zunächst definieren wir

$$r(x) \;=\; f(x) - s(x).$$

Damit hat $r(x)$ die $n + 1$ Nullstellen $x_0, \ldots, x_n$ und der Abstand zwischen zwei aufeinander folgenden Nullstellen von $r(x)$ ist kleiner oder gleich h. Nach dem Satz von Rolle hat die $r'(x)$ dann n Nullstellen mit einen Abstand kleiner oder gleich $2h$. Wählen wir nun ein $z \in [a, b]$ mit

$$|r'(z)| \;=\; \|r'\|_\infty,$$

dann hat die nächste gelegene Nullstelle ξ von $r'(x)$ einen Abstand

$$|\xi - z| \;\leq\; h.$$

Mit der Cauchy-Schwarz Ungleichung können wir nun abschätzen:

$$\|r'\|_\infty^2 \;=\; \left|\int_\xi^z r''(y)\,\mathrm{d}y\right|^2 \;\leq\; h \cdot \left|\int_\xi^z (r''(y))^2\,\mathrm{d}y\right| \;\leq\; h \cdot \int_a^b (r''(y))^2\,\mathrm{d}y.$$

Aus Satz 5.10 folgt damit

$$\|r'\|_\infty \;\leq\; h^{1/2} \cdot \|f''\|_2,$$

also die zweite Aussage. Nun wählen wir ein $x \in [a, b]$ mit

$$|r(x)| \;=\; \|r\|_\infty.$$

Dann hat die nächst gelegene Nullstelle ζ von $r(x)$ einen Abstand

$$|\zeta - x| \;\leq\; \frac{h}{2}.$$

Wieder können wir abschätzen:

$$\|r\|_\infty \;=\; \left|\int_\zeta^x r'(y)\,\mathrm{d}y\right| \;\leq\; \frac{h}{2} \cdot \|r'\|_\infty \;\leq\; \frac{h \cdot h^{1/2}}{2} \cdot \|f''\|_\infty$$

Damit ist auch die erste Aussage gezeigt. □

Der Satz besagt also auch, dass der interpolierende Spline $s(x)$ bei einer äquidistante Zerlegung von $[a, b]$ für $n \to \infty$ gegen die Funktion $f(x)$ konvergiert. Dies ist bei der Polynominterpolation nicht der Fall.

5.3 Ausblick

In diesem Kapitel haben wir uns mit zwei einfachen Interpolationsaufgaben beschäftigt, nämlich der Polynom- und der Splineinterpolation. Natürlich gibt es noch sehr viele weitere Interpolationsaufgaben. Sind bei der Polynominterpolation zu einigen Stützstellen auch Stützwerte der Ableitungen bekannt, so führt dies zur Hermite Interpolation, siehe Kress (1998). Dies lässt sich wiederum durch ein Schema aus dividierten Differenzen lösen. Weiterhin nicht betrachtet haben wir trigonometrische Interpolation, welche zur schnellen Fourier-Transformation führt, siehe Hohage (2005). Auch die Interpolation von Kurven mittels Bézier Kurven sowie die mehrdimensionale Interpolation haben wir komplett vernachlässigt, siehe und H. Wendland (2004).

6 Numerische Integration

Gerade in der Physik ist die numerische Auswertung von Integralen von großer Bedeutung, zum Beispiel bei der numerischen Lösung der Schrödinger Gleichung. Da sich komplizierte Funktionen oft nicht analytisch lösen lassen oder die exakte Lösung viel zu Aufwendig zu berechnen ist, sind numerische Verfahren mit geeigneten Fehlerabschätzungen oft hilfreich.

Unser Ziel in diesem Kapitel ist es, eine einfache Formel zur Berechnung von

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

zu finden. Eine Möglichkeit wäre die Annäherung durch Rechtecke:

$$\int_a^b f(x)\,\mathrm{d}x \;\approx\; \sum_{j=1}^{n} a_j f(x_j),$$

wobei a_j die Breite des jeweiligen Rechtecks und $f(x_j)$ die Höhe ist, siehe Abbildung 6.1.

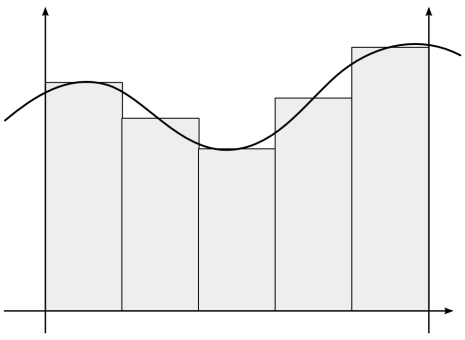


Abbildung 6.1: Numerische Integration durch Rechtecke.

Wir führen zunächst einige Notationen ein, bevor wir uns mit den Interpolationsquadraturen beschäftigen werden.

Notation 6.1. Sei $f \in \mathcal{C}([a,b])$ ein stetige Funktion. Eine Abbildung

$$Q : [a,b] \to \mathbb{R}$$

heißt ***Quadraturformel*** bezüglich der ***Quadraturstellen*** $x_0, \ldots, x_n \in [a,b]$, falls

$$Q(f) \;=\; \sum_{j=0}^{n} a_j f(x_j)$$

mit $a_0, \ldots, a_n \in \mathbb{R}$ gilt.

Wir werden versuchen Quadraturformeln Q zu finden, die Integrale

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

annähern, also $Q(f) \approx I(f)$.

Notation 6.2. Sei $\mathcal{F} \subset \mathcal{C}([a,b])$ ein endlich dimensionaler Unterraum von $\mathcal{C}([a,b])$. Eine Quadraturformel Q heißt ***exakt*** für $\mathcal{F}$, wenn

$$Q(f) \;=\; I(f)$$

für alle $f \in \mathcal{F}$ gilt.

Satz 6.1. *Sei $\mathcal{F} \subset \mathcal{C}([a,b])$ ein endlich dimensionaler Unterraum von $\mathcal{C}([a,b])$, sei $\{f_0, \ldots, f_m\}$ eine Basis von $\mathcal{F}$ und sei Q eine Quadraturformel. Gilt dann $Q(f_i) = I(f_i)$ für alle $i = 0, \ldots, m$, so ist Q exakt für $\mathcal{F}$.*

Beweis. Sei $f \in \mathcal{F}$. Dann kann f bezüglich der Basis $\{f_0, \ldots, f_m\}$ eine Darstellung

$$f(x) \;=\; \sum_{i=0}^{m} a_i f_i(x).$$

Nun gilt

$$\begin{aligned} Q(f) \;&=\; Q\left(\sum_{i=0}^{m} a_i f_i\right) \;=\; \sum_{i=0}^{m} a_i Q(f_i) \\ &=\; \sum_{i=0}^{m} a_i I(f_i) \\ &=\; I\left(\sum_{i=0}^{m} a_i f_i\right) \;=\; I(f) \end{aligned}$$

und damit ist der Satz gezeigt. □

6.1 Interpolationsquadraturen

Nach der allgemeinen Einleitung befassen wir uns nun mit Interpolationsquadraturen, dass heißt wir werden $\mathcal{F} = \Pi_n$ als Raum der Polynome vom Grad n annehmen.

Seien dazu Quadraturstellen $a \leq x_0 < \ldots < x_n \leq b$ gegeben. Die Idee ist es, das Integral von f durch das Integral des eindeutig bestimmten Interpolationspolynoms $(L_n f)(x)$ zu approximieren.

Definition 6.3. Eine Quadraturformel

$$Q_n(f) \;=\; \sum_{j=0}^{n} a_j f(x_j)$$

heißt ***Interpolationsquadratur*** der Ordnung n, falls für alle $f \in \mathcal{C}([a,b])$

$$Q_n(f) \;=\; \sum_{j=0}^{n} a_j f(x_j) \;=\; \int_a^b (L_n f)(x)dx \;=\; I(L_n f).$$

gilt. Dabei ist $L_n f \in \Pi_n$ das eindeutig bestimmte Interpolationspolynom zu f bezüglich der Stützstellen $a \leq x_0 < \ldots < x_n \leq b$.

Wir erinnern uns an die Lagrange Darstellung

$$(L_n f)(x) \;=\; \sum_{j=0}^{n} f(x_j) l_j(x)$$

mit den Lagrange-Polynomen

$$l_j(x) \;=\; \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}.$$

Unser Ziel ist es nun die Koeffizienten a_j von Interpolationsquadraturen der Ordnung n herzuleiten. Kennen wir diese, so können wir alle Polynome $p \in \Pi_n$ exakt integrieren. Erstaunlicherweise gilt auch die Umkehrung dieser Aussage.

Satz 6.2. *Eine Quadraturformel Q_n ist genau dann eine Interpolationsquadratur vom Grad n, wenn alle Polynome $p \in \Pi_n$ exakt integriert werden.*

Beweis. Zunächst sei

$$Q_n(f) \;=\; \int_a^b (L_n f)(x)\,\mathrm{d}x$$

eine Interpolationsquadratur der Ordnung n und sei $f \in \Pi_n$. Dann gilt $f(x) = (L_n f)(x)$ nach Satz 5.3, also folgt

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

und damit ist Q_n exakt für alle $f \in \Pi_n$.

Sei nun umgekehrt

$$Q(f) \;=\; \sum_{j=0}^{n} a_i f(x_j)$$

eine Quadraturformel, die $I(p) = Q(p)$ für alle $p \in \Pi_n$ erfüllt. Weiter sei $f \in \mathcal{C}([a,b])$. Dann ist $L_n f \in \Pi_n$ und es folgt

$$\begin{aligned} I(L_n f) \;&=\; Q(L_n f) \;=\; \sum_{j=0}^{n} a_j (L_n f)(x_j) \\ &=\; \sum_{j=0}^{n} a_j f(x_j) \;=\; Q(f), \end{aligned}$$

also ist $Q(f)$ eine Interpolationsquadratur der Ordnung n. □

Um schließlich die Koeffizienten a_j von Interpolationsquadraturen der Ordnung n zu berechnen, kann der folgende Satz verwendet werden.

Satz 6.3. *Gegeben seien Quadraturstellen $a \le x_0 < \ldots < x_n \le b$ und weiter sei*

$$h_{n+1}(x) \;=\; \prod_{j=0}^{n} (x - x_j).$$

Dann existiert genau eine Interpolationsquadratur der Ordnung n zu den Quadraturstellen $x_0, \ldots, x_n \in [a,b]$, die durch die Gewichte

$$a_j \;=\; \frac{1}{h'_{n+1}(x_j)} \cdot \int_a^b \frac{h_{n+1}(x)}{x - x_j}\,\mathrm{d}x$$

für $j = 0, \ldots, n$ gegeben ist.

Der Beweis hierzu kann in Schöbel (2007) gefunden werden. Wir werden uns im Folgenden jedoch nur mit äquidistanten Quadraturstellen

$$x_j \;=\; a + j \cdot h \qquad \text{für} \qquad j = 0, \ldots, n$$

mit $h = (b-a)/n$ befassen.

Notation 6.4. Die Interpolationsquadraturen der Ordnung n zu den äquidistanten Stützstellen $x_j = a + j \cdot h$ für $j = 0, \ldots, n$ mit $h = (b-a)/n$ heißen ***Newton-Côtes-Formeln***.

Natürlich können wir die Gewichte a_j der Newton-Côtes-Formeln nach Satz 6.3 bestimmen. Wir wollen nun aber zeigen, wie wir die a_j mit Hilfe von Satz 6.1 über die Lösung eines linearen Gleichungssystems sehr viel einfacher berechnen können.

Dazu wählen wir die Monom-Basis $\{1, x, x^2, \ldots, x^n\}$ von Π_n und fordern, dass diese exakt integriert werden, also

$$Q(p_k) \;=\; \sum_{j=0}^{n} a_j p_k(x_i) \;=\; \int_a^b p_k(x)\,\mathrm{d}x \;=\; I(p_k)$$

für $k = 0, \ldots, n$ mit $p_k(x) = x^k$. Das genaue Vorgehen sei an den folgenden Beispielen veranschaulicht.

Beispiel 6.1 (Trapez-Regel). *Für $n = 1$ und $[a, b] = [-1, 1]$ fordern wir, dass die Funktionen $p_0(x) = 1$ und $p_1(x) = x$ exakt integriert werden:*

$$\begin{aligned} Q(p_0) &= a_0 + a_1 = \int_{-1}^{1} 1\,\mathrm{d}x = 2, \\ Q(p_1) &= -a_0 + a_1 = \int_{-1}^{1} x\,\mathrm{d}x = 0. \end{aligned}$$

Dies führt uns zum Gleichungssystem

$$\begin{aligned} a_0 + a_1 &= 2, \\ -a_0 + a_1 &= 0 \end{aligned}$$

mit der eindeutigen Lösung $a_0 = a_1 = 1$. Die Newton-Côtes-Formel der Ordnung $n = 1$ zu $[-1, 1]$ ist damit

$$Q_1(f) \;=\; f(-1) + f(1).$$

Allgemein gilt für $[a, b]$

$$Q_1(f) \;=\; \frac{b-a}{2}(f(a) + f(b)).$$

Diese Formel wird auch **Trapez-Regel** *genannt.*

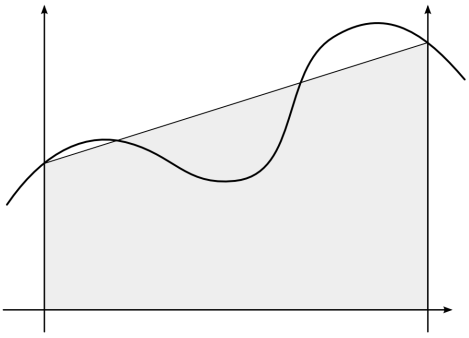


Abbildung 6.2: Veranschaulichung der Trapez-Regel.

Beispiel 6.2 (Simpson-Regel). *Für $n = 2$ und $[a, b] = [-1, 1]$ fordern wir, dass die Funktionen $p_0(x) = 1$, $p_1(x) = x$ und $p_2(x) = x^2$ exakt integriert werden:*

$$\begin{aligned} Q(p_0) &= a_0 + a_1 + a_2 = \int_{-1}^{1} 1 \,\mathrm{d}x = 2, \\ Q(p_1) &= -a_0 + a_2 = \int_{-1}^{1} x \,\mathrm{d}x = 0, \\ Q(p_2) &= a_0 + a_2 = \int_{-1}^{1} x^2 \,\mathrm{d}x = \frac{2}{3}. \end{aligned}$$

Dies führt uns zum Gleichungssystem

$$\begin{aligned} a_0 + a_1 + a_2 &= 2, \\ -a_0 + a_2 &= 0, \\ a_0 + a_2 &= \frac{2}{3} \end{aligned}$$

mit der eindeutigen Lösung $a_0 = \frac{1}{3}$, $a_1 = \frac{4}{3}$ und $a_2 = \frac{1}{3}$. Die Newton-Côtes-Formel der Ordnung $n = 2$ zu $[-1, 1]$ ist damit

$$Q_2(f) = \frac{1}{3} \cdot f(-1) + \frac{4}{3} \cdot f(0) + \frac{1}{2} \cdot f(1).$$

Allgemein gilt für $[a, b]$

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

Diese Formel wird auch **Simpson-Regel** *genannt.*

Tabelle 6.1 stellt die Gewichte der Newton-Côtes-Formeln bis $n = 5$ zusammen.

n	a_0	a_1	a_2	a_3	a_4	a_5	Bezeichnung
1	$\frac{1}{2}$	$\frac{1}{2}$					Trapez-Regel
2	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$				Simpson-Regel
3	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$			Newton-$\frac{3}{8}$-Regel
4	$\frac{7}{90}$	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$	$\frac{7}{90}$		1. Milne-Regel
5	$\frac{19}{288}$	$\frac{75}{288}$	$\frac{50}{288}$	$\frac{50}{288}$	$\frac{75}{288}$	$\frac{19}{288}$	2. Milne-Regel

Tabelle 6.1: Gewichte der Newton-Côtes-Formeln bis $n = 5$ für ein Intervall der Länge 1.

Wir wissen bereits, dass die Simpson-Regel alle Polynome vom Grad ≤ 2 exakt integriert. Es gilt jedoch noch mehr.

Lemma 6.4. *Die Simpson-Regel Q_2 ist exakt für alle Polynome $p \in \Pi_3$.*

Beweis. Nach Satz 6.1 ist eine Quadraturformel auf Π_3 exakt, wenn sie auf einer Basis von Π_n exakt ist. Wir wählen als Basis $\{p_0, \ldots, p_3\}$ mit

$$\begin{aligned} p_0(x) &= 1, \\ p_1(x) &= x - \frac{a+b}{2}, \\ p_2(x) &= \left(x - \frac{a+b}{2}\right)^2, \\ p_3(x) &= \left(x - \frac{a+b}{2}\right)^3. \end{aligned}$$

Wir wissen bereits, dass

$$Q(p_k) = I(p_k) \qquad \text{für} \qquad k = 0, 1, 2$$

gilt. Nun lässt sich aber auch einfach nachrechnen, dass

$$Q(p_3) = I(p_3)$$

gilt und damit werden alle Polynome aus Π_3 durch die Simpson-Regel exakt integriert. □

Analog lässt sich auch die folgende Verallgemeinerung beweisen.

Satz 6.5. *Die Newton-Côtes-Formel $Q_n(f)$ mit geradem n ist exakt für alle Polynome $p \in \Pi_{n+1}$.*

Leider tauchen bei den Newton-Côtes-Formeln ab $n \geq 8$ negative Gewichte auf, die unerwünschte Nebeneffekte haben:

(1) Es kann Auslöschung und damit numerischer Instabilität auftreten.

(2) Es lassen sich positive Funktionen $f \geq 0$ konstruieren mit $Q(f) < 0$.

Um diese Probleme zu umgehen, können zusammengesetzte Newton-Côtes-Formeln verwendet werden. Analog zur Spline-Interpolation zerlegen wir dazu das Integrationsintervall in m Teilintervalle und wendet auf jedes Teilintervall eine Quadraturformel niedriger Ordnung an.

Verwenden wir auf jedem Teilintervall die Trapez-Regel, so erhalten wir die ***zusammengesetzte Trapez-Regel***

$$\int_a^b f(x)\,\mathrm{d}x \;\approx\; T_h(f) \;:=\; h \cdot \left(\frac{1}{2} \cdot f(x_0) + \sum_{i=1}^{m-1} f(x_i) + \frac{1}{2} \cdot f(x_m) \right),$$

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

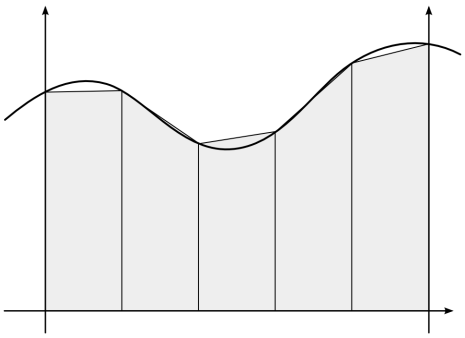


Abbildung 6.3: Veranschaulichung der zusammengesetzten Trapez-Regel.

Analog ergibt sich die ***zusammengesetzte Simpson-Regel***

$$\begin{aligned} \int_a^b f(x)\,\mathrm{d}x \;&\approx\; S_h(f) \\ &:=\; \frac{h}{3} \cdot \sum_{i=0}^{m/2-1} \left(\frac{1}{3} \cdot f(x_{2i}) + \frac{4}{3} \cdot f(x_{2i+1}) + \frac{1}{3} \cdot f(x_{2i+2}) \right) \end{aligned}$$

für ein gerades m.

Abschließend beschäftigen wir uns in diesem Abschnitt mit dem Fehler der numerischen Integration unter der Verwendung von Newton-Côtes-Formeln.

Satz 6.6. *Sei $f \in C^2([a,b])$. Dann erfüllt der Fehler der Trapez-Regel die Gleichung*

$$I(f) - Q_1(f) \;=\; -\frac{(b-a)^3}{12} \cdot f''(\xi)$$

für ein $\xi \in [a,b]$.

Bei der zusammengesetzten Trapez-Regel mit $h = (b-a)/m$ gilt die Fehlerabschätzung

$$|I(f) - T_h(f)| \;\leq\; \frac{b-a}{12} \cdot h^2 \cdot \|f''\|_\infty.$$

Beweis. Ist $L_1 f \in \Pi_1$ das Interpolationspolynom zu den Stützstellen $x_0 = a$ und $x_1 = b$, so gilt für den Fehler bei der Trapezregel

$$I(f) - Q_1(f) \;=\; \int_a^b (x-a)(x-b) \cdot \frac{f(x) - (L_1 f)(x)}{(x-a)(x-b)}\,\mathrm{d}x.$$

Da der erste Faktor des Integranden nicht positiv auf $[a,b]$ ist und da der zweite Faktor nach der Regel von l'Hopital stetig ist, folgt aus dem Mittelwertsatz der Integralrechnung, dass es eine Zwischenstelle $t \in [a,b]$ gibt mit

$$I(f) - Q_1(f) \;=\; \frac{f(t) - (L_1 f)(t)}{(t-a)(t-b)} \cdot \int_a^b (x-a)(x-b)\,\mathrm{d}x.$$

Da

$$\int_a^b (x-a)(x-b)\,\mathrm{d}x \;=\; -\frac{(b-a)^3}{6},$$

folgt die Fehlerabschätzung der Trapez-Regel aus der Fehlerdarstellung

$$\frac{f(t) - (L_1 f)(t)}{(t-a)(t-b)} \;=\; \frac{f''(\xi)}{2!}$$

für lineare Interpolation mit $\xi \in [a,b]$, siehe Satz 5.7.

Durch Summation dieser Fehlerabschätzungen erhalten wir direkt den Fehler der zusammengesetzten Trapez-Regel, da jedes Teilintervall durch

$$\frac{1}{12} \cdot \frac{(b-a)^3}{m^3} \cdot \|f''\|_\infty$$

beschränkt ist. □

Wieder unter der Verwendung von Satz 5.7 zur Polynominterpolation erhalten wir die folgende Aussage.

Satz 6.7. *Sei $f \in C^4([a,b])$. Dann erfüllt der Fehler der Simpson-Regel die Gleichung*

$$I(f) - Q_2(f) \;=\; -\frac{1}{90} \cdot \left(\frac{b-a}{2}\right)^5 \cdot f^{(4)}(\xi)$$

für ein $\xi \in [a,b]$.

Bei der zusammengesetzten Simpson-Regel mit $h = (b-a)/m$ gilt die Fehlerabschätzung

$$|I(f) - S_h(f)| \;\leq\; \frac{b-a}{180} \cdot h^4 \cdot \|f^{(4)}\|_\infty.$$

6.2 Gaußsche Quadraturformeln

Bislang haben wir zu gegebenen Quadraturstellen Interpolationsquadraturen der Ordnung n bestimmt, sodass alle Polynome aus Π_n exakt integriert wurden.

Bei den Gaußsche Quadraturformeln wollen wir nun neben den Gewichten $a_0, \ldots, a_n$ auch die Stützstellen $x_0, \ldots, x_n$ wählen. Wir erhoffen uns, dass damit auch Polynome höheren Grades exakt integriert werden.

Dazu ist es in verschiedenen Anwendungen günstig, den allgemeineren Fall von Quadraturformeln mit gemischten Integralen zu betrachten.

Definition 6.5. Sei $[a,b] \subset \mathbb{R}$ ein Intervall. Eine stetige und positive Funktion $\omega : (a,b) \to \mathbb{R}$ heißt ***positive Gewichtsfunktion***, wenn

$$\int_a^b \omega(x) x^k \,\mathrm{d}x$$

für alle $k \in \mathbb{N} \cup \{0\}$ existiert. Ist ω eine positive Gewichtsfunktion, so nennen wir

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

das ***gemischte Integral*** zur Funktion f.

Typische Beispiele für positive Gewichtsfunktionen sind die folgenden:

(1) Gauß-Legendre: $\omega(x) = 1$ auf $[a,b]$.

(2) Gauß-Tschebyscheff 1. Art: $\omega(x) = \frac{1}{\sqrt{1-x^2}}$ auf $x \in [-1,1]$.

(3) Gauß-Tschebyscheff 2. Art: $\omega(x) = \sqrt{1-x^2}$ auf $x \in [-1,1]$.

(4) Gauß-Laguerre: $\omega(x) = e^{-x}$ auf $[0,\infty)$.

(5) Gauß-Hermite: $\omega(x) = e^{-x^2}$ auf $(-\infty,\infty)$.

Notation 6.6. Eine Quadraturformel

$$Q_n(f) \;=\; \sum_{i=0}^{n} a_i f(x_i)$$

nennen wir ***Gaußsche Quadraturformel*** der Ordnung n wenn sie alle Polynome $p \in \Pi_{2n+1}$ exakt integriert.

Im Folgenden wollen wir uns damit beschäftigen, wie wir Gaußsche Quadraturformeln berechnen können. Zunächst ergibt sich das folgende Ergebnis.

Lemma 6.8. *Seien $x_0,\ldots,x_n \in [a,b]$ paarweise verschieden und sei L_nf das Interpolationspolynom bezüglich der Stützstellen $x_0,\ldots,x_n$ und einer Funktion f. Weiter sei ω eine positive Gewichtsfunktion. Dann ist*

$$Q_n^{\omega}(f) \;=\; \int_a^b \omega(x)(L_nf)(x)\,\mathrm{d}x$$

eine Quadraturformel zu den Quadraturstellen $x_0,\ldots,x_n$. Genauer gilt

$$Q_n^{\omega}(f) \;=\; \int_a^b \omega(x)(L_nf)(x)\,\mathrm{d}x \;=\; \sum_{j=0}^{n} a_j f(x_j)$$

mit

$$a_j \;=\; \int_a^b \omega(x) l_j(x)\,\mathrm{d}x.$$

Der folgende Satz zeigt nun wann eine Quadraturformel eine Gaußsche Quadraturformel ist.

Satz 6.9. *Seien $x_0,\ldots,x_n \in [a,b]$ paarweise verschieden und sei L_nf das Interpolationspolynom bezüglich der Stützstellen $x_0,\ldots,x_n$ und einer Funktion f. Weiter sei ω eine positive Gewichtsfunktion und wir definieren*

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

Dann ist

$$Q_n^{\omega}(f) \;=\; \int_a^b \omega(x)(L_nf)(x)\,\mathrm{d}x$$

genau dann eine Gaußsche Quadraturformel der Ordnung n, wenn

$$\int_a^b \omega(x) h_{n+1}(x) p(x)\,\mathrm{d}x \;=\; 0$$

für alle $p \in \Pi_n$ gilt.

Beweis. Zunächst sei $Q_n^\omega(f)$ eine Gaußsche Quadraturformel, es gelte also $Q_n^\omega(f) = I_\omega(f)$ für alle $f \in \Pi_{2n+1}$. Für ein $p \in \Pi_n$ ist dann $h_{n+1} \cdot p \in \Pi_{2n+1}$ und mit Lemma 6.8 erhalten wir

$$\int_a^b \omega(x) h_{n+1}(x) p(x) dx \;=\; Q_n^\omega(h_{n+1} \cdot p) \;=\; \sum_{j=0}^{n} a_j \underbrace{h_{n+1}(x_j)}_{=0} p(x_j) \;=\; 0$$

und somit gilt der der zweite Teil der Aussage.

Nun gelte umgekehrt

$$\int_a^b \omega(x) h_{n+1}(x) p(x)\,\mathrm{d}x \;=\; 0$$

für alle $p \in \Pi_n$ und es sei $q \in \Pi_{2n+1}$. Dann ist $L_n q \in \Pi_n$ und die Funktion $q - L_n q$ hat die Nullstellen $x_0, \ldots, x_n$. Nach dem Hauptsatz der Algebra gibt es also ein Polynom $s \in \Pi_n$ mit $q(x) - L_n q(x) = h_{n+1}(x) \cdot s(x)$. Damit gilt

$$\begin{aligned}\int_a^b \omega(x) q(x)\,\mathrm{d}x \;&=\; \int_a^b \omega(x) L_n q(x)\,\mathrm{d}x + \int_a^b \omega(x) h_{n+1}(x) s(x)\,\mathrm{d}x \\ &=\; \int_a^b \omega(x) L_n q(x)\,\mathrm{d}x \;=\; Q_n^\omega(q),\end{aligned}$$

also ist Q_n^ω eine Gaußsche Quadraturformel. □

Notation 6.7. Für $f, g \in C([a,b])$ definieren wir

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

Gilt $(f,g)_\omega = 0$, so bezeichnet man f und g als ω-***orthogonal***.

Bemerkung 6.1. Der Ausdruck $(f,g)_\omega$ existiert für alle Polynome f und g, sofern ω eine positive Gewichtsfunktion ist. Weiter lässt sich zeigen, dass $(f,g)_\omega$ ein Skalarprodukt ist.

Mit dieser Notation können wir Satz 6.9 also folgendermaßen umformulieren.

Korollar 6.10. *Seien $x_0, \ldots, x_n \in [a,b]$ paarweise verschieden und sei $L_n f$ das Interpolationspolynom bezüglich der Stützstellen $x_0, \ldots, x_n$ und einer Funktion f. Weiter sei ω eine positive Gewichtsfunktion und wir definieren*

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

Dann ist

$$Q_n^{\omega}(f) \;=\; \int_a^b \omega(x)(L_n f)(x)\,\mathrm{d}x$$

genau dann eine Gaußsche Quadraturformel der Ordnung n, wenn

$$(h_{n+1}, p)_{\omega} \;=\; 0$$

für alle $p \in \Pi_n$ gilt.

Satz 6.11. *Sei ω eine positive Gewichtsfunktion zum Intervall $[a,b]$. Dann existieren Polynome $p_k \in \Pi_k$ für alle $k \in \mathbb{N} \cup \{0\}$ mit den folgenden Eigenschaften:*

(1) *Für alle $n, m \in \mathbb{N} \cup \{0\}$ gilt*

$$(p_n, p_m)_{\omega} \;=\; \delta_{n,m} \;=\; \begin{cases} 1 & \text{falls} \quad n = m \\ 0 & \text{falls} \quad n \neq m \end{cases}.$$

(2) *Für alle $n \in \mathbb{N} \cup \{0\}$ sind die n Nullstellen von p_n reell und liegen in (a,b).*

Ein Beweis hierzu kann in Schöbel (2007) gefunden werden. Wir fassen alle Ergebnisse noch einmal zusammen.

Satz 6.12. *Sei $n \in \mathbb{N}$ und sei ω eine positive Gewichtsfunktion. Dann existiert eine Gaußsche Quadraturformel*

$$Q_n^{\omega}(f) \;=\; \sum_{j=0}^{n} a_j f(x_j),$$

wobei $x_0 < x_1 < \cdots < x_n \in (a,b)$ die Nullstellen des in Satz 6.11 konstruierten ω-orthogonalen Polynoms p_{n+1} sind und

$$a_j \;=\; \int_a^b \omega(x) l_j(x)\,\mathrm{d}x$$

gilt. Insbesondere ist damit $Q_n^{\omega}(p) = I_{\omega}(p)$ für alle $p \in \Pi_{2n+1}$.

Beweis. Weil $p_{n+1} \in \Pi_{n+1}$ ist, gilt $p_{n+1} = \alpha h_{n+1}$ mit $\alpha \neq 0$. Somit folgt

$$Q_n^\omega(f) \;=\; \int_a^b \omega(x)(L_n f)(x)\,\mathrm{d}x$$

und $Q_n^\omega(f)$ ist nach Korollar 6.10 eine Gaußsche Quadraturformel, sofern wir

$$(h_{n+1}, p) \;=\; 0$$

für alle $p \in \Pi_n$ zeigen können. Dies gilt aber, da

$$\begin{aligned}(h_{n+1}, p) \;&=\; \frac{1}{\alpha}(p_{n+1}, p) \;=\; \frac{1}{\alpha}\left(p_{n+1}, \sum_{i=0}^{n} \lambda_i p_i\right)\\ &=\; \frac{1}{\alpha}\sum_{i=0}^{n} \lambda_i (p_{n+1}, p_i) \;=\; 0\end{aligned}$$

mit geeigneten $\lambda_0, \ldots, \lambda_n \in \mathbb{R}$. □

Im Gegensatz zu den Newton-Côtes-Formeln gilt für die Gaußschen Quadraturformeln die folgende numerisch wertvolle Eigenschaft.

Lemma 6.13. *Alle Gewichte $a_0, \ldots, a_n$ der Gaußschen Quadraturformeln sind positiv.*

Beweis. Seien $x_0, \ldots, x_n$ die Quadraturstellen zu der Quadraturformel Q_n^ω. Nach Konstruktion sind dies die Nullstellen von p_{n+1} bezüglich der positive Gewichtsfunktion ω. Wir definieren wie gehabt

$$h_{n+1}(x) \;=\; \prod_{j=0}^{n}(x - x_j) \qquad \text{sowie} \qquad f_i(x) \;=\; \left(\frac{h_{n+1}(x)}{x - x_i}\right)^2$$

für $i = 0, \ldots, n$. Es ist also $f_i \in \Pi_{2n}$ und nach Satz 6.12 gilt

$$0 \;<\; \int_a^b \omega(x) f_i(x)\,\mathrm{d}x \;=\; Q_n^\omega(f_i) \;=\; \sum_{j=0}^{n} a_j f_i(x_j) \;=\; a_i f_i(x_i).$$

Mit $f_i(x_i) > 0$ folgt auch $a_i > 0$. □

Nach diesen bislang nur theoretischen Ergebnissen, wollen wir nun einige Beispiele diskutieren.

Beispiel 6.3 (Legendre-Polynome). *Wir betrachten das Intervall* $[-1,1]$ *und die positive Gewichtsfunktion* $\omega(x)=1$. *Dann werden die orthogonalen Polynome* $p_n \in \Pi_n$ *gegeben durch die* **Legendre-Polynome**

$$L_n(x) \;:=\; \frac{1}{2^n n!}\cdot\frac{\mathrm{d}^n}{\mathrm{d}x^n}(x^2-1)^n.$$

Speziell gilt damit

$$\begin{aligned}
L_0(x) &= 1,\\
L_1(x) &= x,\\
L_2(x) &= \frac{1}{2}\left(3x^2-1\right),\\
L_3(x) &= \frac{1}{2}\left(5x^3-3x\right),\\
L_4(x) &= \frac{1}{8}\left(35x^4-30x^2+3\right).
\end{aligned}$$

Nun lässt sich leicht nachrechnen, dass L_{k+1} *orthogonal zu allen* $p\in\Pi_k$ *ist. Dies zeigen wir am Beispiel von* L_2. *Sei dazu* $p(x)=ax+b\in\Pi_1$ *beliebig. Dann gilt*

$$\begin{aligned}
(L_2,p)_\omega &= \int_{-1}^{1} L_2(x)p(x)\,\mathrm{d}x \;=\; \int_{-1}^{1}(x^2-\tfrac{1}{3})(ax+b)\,\mathrm{d}x\\
&= \int_{-1}^{1} ax^3+bx^2-\tfrac{1}{3}ax-\tfrac{1}{3}b\,\mathrm{d}x\\
&= \left[\frac{a}{4}x^4+\frac{b}{3}x^3-\frac{a}{6}x^2-\frac{b}{3}x\right]_{-1}^{1}\\
&= \frac{a}{4}+\frac{b}{3}-\frac{a}{6}-\frac{b}{3}-\left(\frac{a}{4}-\frac{b}{3}-\frac{a}{6}+\frac{b}{3}\right) \;=\; 0.
\end{aligned}$$

Um nun die Quadraturstellen Q_n^ω *zu erhalten, müssen jeweils die Nullstellen von* $L_{n+1}(x)$ *bestimmt werden.*

Betrachten wir den Fall $n=1$. *Die Nullstellen von*

$$L_2(x) \;=\; x^2-\frac{1}{3}$$

sind $x_0=-\sqrt{\frac{1}{3}}$ *und* $x_1=\sqrt{\frac{1}{3}}$. *Wir erhalten somit die Quadraturformel*

$$Q_1^\omega(f) \;=\; a_0\cdot f\left(-\sqrt{\tfrac{1}{3}}\right)+a_1\cdot f\left(\sqrt{\tfrac{1}{3}}\right).$$

Die Gewichte a_0 *und* a_1 *lassen sich nun wieder über die Exaktheitsbedingung aus einem linearen Gleichungssystem bestimmen. Es gilt für* $p_0(x)=1$ *und*

$p_1(x) = x$

$$\begin{aligned} Q_1^\omega(p_0) &= a_0 + a_1 = \int_{-1}^{1} 1 \,\mathrm{d}x = 2, \\ Q_1^\omega(p_1) &= -a_0\sqrt{\tfrac{1}{3}} + a_1\sqrt{\tfrac{1}{3}} = \int_{-1}^{1} x \,\mathrm{d}x = 0. \end{aligned}$$

Die Lösung dieses Systems ist offenbar $a_0 = a_1 = 1$ *und somit folgt*

$$Q_1^\omega(f) = f\left(-\sqrt{\tfrac{1}{3}}\right) + f\left(\sqrt{\tfrac{1}{3}}\right).$$

Wir bemerken noch einmal, dass alle Polynome $p \in \Pi_{2n+1} = \Pi_3$ *auf dem Intervall* $[-1, 1]$ *exakt durch* $Q_1^\omega(p)$ *integriert werden.*

Beispiel 6.4 (Tschebyscheff-Polynome). *Weiterhin betrachten wir das Intervall* $[-1, 1]$*, nun aber die positive Gewichtsfunktion*

$$\omega(x) = \frac{1}{\sqrt{1 - x^2}}.$$

Die orthogonalen Polynome $p_n \in \Pi_n$ *werden gegeben durch die* **Tschebyscheff-Polynome**

$$T_n(x) := \cos(n \arccos(x)).$$

Die Additionstheoreme liefern die Rekursionsformel

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x) \end{aligned}$$

und somit gilt $T_n \in \Pi_n$*. Weiter lässt sich zeigen, dass*

$$(T_n, T_m)_\omega = \int_{-1}^{1} \frac{T_n(x)T_m(x)}{\sqrt{1 - x^2}} \,\mathrm{d}x = \begin{cases} \pi & \text{für} \quad n = m = 0 \\ \frac{\pi}{2} & \text{für} \quad n = m > 0 \\ 0 & \text{für} \quad n \neq m \end{cases}$$

gilt. Zudem werden die Nullstellen von $T_n(x)$ *gegeben durch*

$$x_i = \cos\left(\frac{2i+1}{2n}\pi\right) \qquad \text{für} \quad i = 0, \ldots, n-1.$$

Mit diesen Nullstellen erhalten wir die **Gauß-Tschebyscheff Quadratur**

$$I_\omega(f) = \int_{-1}^{1} \frac{f(x)}{\sqrt{1 - x^2}} \,\mathrm{d}x \approx \sum_{i=0}^{n-1} a_i f\left(\cos\left(\frac{2i+1}{2n}\pi\right)\right) = Q_{n-1}^\omega(f).$$

Die Gewichte ergeben sich aus den Exaktheitsbedingungen für T_m zu $a_i = \frac{\pi}{n}$ für $i = 0, \ldots, n-1$. Damit erhalten wir schließlich

$$Q^{\omega}_{n-1}(f) \;=\; \frac{\pi}{n} \cdot \sum_{i=0}^{n-1} f\left(\cos\left(\frac{2i+1}{2n}\pi\right)\right)$$

für $n \in \mathbb{N}$.

Nun untersuchen wir noch den Interpolationsfehler von Gauß-Quadraturen.

Satz 6.14. *Sei $f \in C^{2n+2}([a,b])$. Dann erfüllt der Fehler der Gauß-Quadraturformel der Ordnung n die Gleichung*

$$I_{\omega}(f) - Q^{\omega}_n(f) \;=\; \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \cdot \int_a^b \omega(x)(h_{n+1}(x))^2 \,\mathrm{d}x$$

für ein $\xi \in [a,b]$ mit

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

Beweis. Für den Beweis benötigen wir die Hermite Interpolation, welche wir nicht besprochen haben.

Sei $H_n f \in \Pi_{2n+1}$ das durch

$$(H_n f)(x_j) \;=\; f(x_j) \qquad \text{und} \qquad (H_n f)'(x_j) \;=\; f'(x_j)$$

für $j = 0, \ldots, n$ eindeutig bestimmte Hermite Interpolationspolynom. Dann gilt

$$Q^{\omega}_n(f) \;=\; \sum_{j=0}^{n} \alpha_j f(x_j) \;=\; \sum_{j=0}^{n} \alpha_j (H_n f)(x_j) \;=\; Q^{\omega}_n(H_n f) \;=\; I_{\omega}(H_n f)$$

und für den Fehler erhalten wir

$$\begin{aligned} I_{\omega}(f) - Q^{\omega}_n(f) \;&=\; \int_a^b \omega(x)\,(f(x) - (H_n f)(x))\,\mathrm{d}x \\ &=\; \int_a^b \omega(x) \cdot h_{n+1}(x)^2 \cdot \frac{(f(x) - (H_n f)(x))}{h_{n+1}(x)^2}\,\mathrm{d}x. \end{aligned}$$

Nach dem Mittelwertsatz der Integralrechnung gilt deshalb

$$I_{\omega}(f) - Q^{\omega}_n(f) \;=\; \frac{(f(t) - (H_n f)(t))}{h_{n+1}(t)^2} \cdot \int_a^b \omega(x) \cdot h_{n+1}(x)^2 \,\mathrm{d}x$$

für eine Zwischenstelle $t \in [a,b]$. Nun folgt die Behauptung aus der Darstellung des Interpolationsfehlers bei der Hermite Interpolation. □

6.3 Ausblick

In diesem Kapitel haben wir zunächst den einfachsten Fall der numerischen Integration behandelt, nämlich die Newton-Côtes-Formeln. Anschließend haben wir Gauß-Quadraturformeln untersucht und gesehen, dass Polynome sogar hohen Grades exakt integriert werden können.

Ein weiteres Verfahren zur numerischen Integration ist das Rombergverfahren, welches die zusammengesetzte Trapez-Regel verwendet. Weiterhin gar nicht untersucht haben wir die mehrdimensionale Integration.

7 Anfangswertprobleme

Bei Anfangswertprobleme handelt es sich um (zeitabhängige) Differentialgleichungen mit bekannten Startinformationen. Bei der numerischen Lösung von Anfangswertproblemen können wir so zum Beispiel die Bahn eines Satelliten vorhersagen, wenn wir dessen Position, Geschwindigkeit und Beschleunigung zum Zeitpunkt $t = 0$ kennen.

7.1 Notationen und Grundlagen

Bevor wir uns mit Lösungsverfahren von Anfangswertproblemen beschäftigen können, stellen wir in diesem Abschnitt zunächt grundlegende Notationen und Eigenschaften zusammen.

Wir werden in diesem Kapitel ausschließlich gewöhnliche, explizite Differentialgleichungen erster Ordnung untersuchen, die gegeben sind durch

$$x'(t) \;=\; f(t, x(t)) \qquad \text{mit} \qquad t \in I = [a, b]. \tag{7.1}$$

Dabei ist $x : I \to \mathbb{R}^d$ eine *gesuchte* und differenzierbare Funktion oder Kurve auf einem Intervall $I = [a, b]$ und $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ ist eine *gegebene* Funktion.

Notation 7.1. Wir werden die folgenden Notationen verwenden:

(1) Eine Differentialgleichung heißt ***gewöhnlich***, wenn die unbekannte Funktion x nur von einer reellen Variablen abhängt, also $x : I \to \mathbb{R}^d$ mit $I \subset \mathbb{R}$. Hängt x von mehreren Variablen ab, also $x : B \to \mathbb{R}^d$ mit $B \subset \mathbb{R}^s$ und $s > 1$, so liegt eine ***partielle*** vor.

(2) Eine Differentialgleichung hat die ***Ordnung*** k, falls nur Ableitungen von x bis zur Ordnung k vorkommen. Sie hat somit die Ordnung 1, falls nur die erste Ableitung von x vorkommt.

(3) Eine gewöhnliche Differentialgleichung der Form

$$x'(t) \;=\; f(x(t)),$$

bei der die rechte Seite nicht explizit von t abhängt, heißt ***autonom***.

(4) Wir nennen eine Differentialgleichung ***explizit***, falls der höchste Ableitungsterm isoliert auftaucht, anderenfalls ***implizit***.

Beispiel 7.1. *Wir stellen zunächst einige Beispiele zusammen.*

(1) $x'(t) = x(t)$ *ist eine gewöhnliche und explizite Differentialgleichung erster Ordnung.*

(2) *Durch*

$$x'(t) \;=\; f(t, x(t), x'(t))$$

mit einer gegebenen Funktion $f : \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ *wird eine gewöhnliche und implizite Differentialgleichung erster Ordnung gegeben.*

(3) *Durch*

$$x^{(k)}(t) \;=\; f(t, x(t), \ldots, x^{(k-1)}(t))$$

mit einer gesuchten Funktion $x : I \to \mathbb{R}^d$*, welche* k*-mal differenzierbar ist, wird eine gewöhnliche und explizite Differentialgleichung der Ordnung* k *gegeben.*

Wir werden uns m Wesentlichen mit expliziten und gewöhnlichen Differentialgleichungen beschäftigen.

Beispiel 7.2. *Gegeben sei die gewöhnliche, explizite und autonome Differentialgleichung*

$$x'(t) \;=\; (x_1'(t), x_2'(t)) \;=\; f(x_1'(t), x_2'(t)) \;=\; (-x_2(t), x_1(t))$$

mit $f(x_1, x_2) = (-x_2, x_1)$*. Eine Lösung dieser Differentialgleichung ist*

$$x(t) \;=\; \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix},$$

denn

$$\begin{aligned} x_1'(t) &= \cos'(t) = -\sin(t) = -x_2(t), \\ x_2'(t) &= \sin'(t) = \cos(t) = x_1(t). \end{aligned}$$

Es gibt aber noch weitere Lösungen, nämlich

$$\tilde{x}(t) \;=\; C \cdot \begin{pmatrix} \cos(t - t_0) \\ \sin(t - t_0) \end{pmatrix}$$

mit $C, t_0 \in \mathbb{R}$*.*

Anschaulich beschreibt $f(x_1, x_2) = (-x_2, x_1)$ *ein Vektorfeld, welches wir durch den Vektor* $(-x_2, x_1)$ *in jedem Punkt* (x_1, x_2) *skizzieren können. Jede Lösung der gegeben Differentialgleichung beschreibt dann eine Kurve im* $\mathbb{R}^2$*, zu der das Vektorfeld in jedem Punkt tangential ist.*

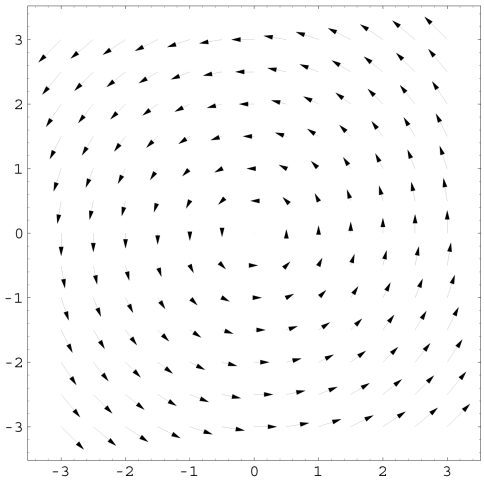


Abbildung 7.1: Vektorfeld zur gegebenen Differentialgleichung.

Das Beispiel zeigt also, dass die Lösung einer Differentialgleichung im Allgemeinen nicht eindeutig bestimmt ist. Um die Eindeutigkeit zu erhalten, müssen freie Variablen durch zusätzliche Bedingungen festgelegt werden.

Notation 7.2. Das ***Anfangswertproblem*** oder kurz ***AWP*** einer gewöhnlichen Differentialgleichung erster Ordnung wird gegeben durch

$$x'(t) \;=\; f(t, x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0 \;\in\; \mathbb{R}^d.$$

für ein $t_0 \in \mathbb{R}$. Die Gleichung $x(t) = x_0$ legt damit d Parameter fest.

Lemma 7.1. *Sei $x : I = [a, b] \to \mathbb{R}^d$ eine Lösung einer autonomen Differentialgleichung $x'(t) = f(x(t))$. Dann ist auch*

$$y : I \to \mathbb{R}^d \qquad \text{mit} \qquad y(t) \;=\; x(t - t_0)$$

für alle $t_0 \in \mathbb{R}$ eine Lösung der Differentialgleichung.

Beweis. Es gilt

$$y'(t) \;=\; x'(t - t_0) \;=\; f(x(t - t_0)) \;=\; f(y(t))$$

und damit folgt direkt die Behauptung. □

Oft beschreibt der Parameter t die Zeit. Die Aussage des Lemmas lautet dann, dass die Lösung einer autonomen Differentialgleichung unabhängig von der Startzeit ist, sie ist also invariant gegenüber Zeittransformationen.

Wir wollen nun zeigen, warum wir im Folgenden nur autonome Differentialgleichungen erster Ordnung betrachten werden.

Lemma 7.2. *Jede gewöhnliche und explizite Differentialgleichung der Ordnung k kann in eine äquivalente Differentialgleichung erster Ordnung transformiert werden.*

Beweis. Gegeben sei

$$y^{(k)}(t) \;=\; g(t, y(t), \ldots, y^{(k-1)}(t)) \qquad \text{mit} \qquad t \in I = [a, b]$$

und mit einer gesuchten k-mal differenzierbaren Funktion $y : I \to \mathbb{R}^d$. Dann definieren wir die Funktionen

$$x_j : I \to \mathbb{R}^d \qquad \text{mit} \qquad x_j(t) \;=\; y^{(j)}(t)$$

für $j = 0, \ldots, k-1$. Somit gilt

$$x_j'(t) \;=\; \left(y^{(j)}\right)'(t) \;=\; y^{(j+1)}(t) \;=\; x_{j+1}(t)$$

für $j = 0, \ldots, k-2$ sowie

$$\begin{aligned} x_{k-1}'(t) \;&=\; \left(y^{(k-1)}\right)'(t) \;=\; y^{(k)}(t) \\ &=\; g(t, y(t), \ldots, y^{(k-1)}(t)) \;=\; g(t, x_0(t), \ldots, x_{k-1}(t)). \end{aligned}$$

Damit erhalten wir das System

$$x'(t) \;=\; \begin{pmatrix} x_0'(t) \\ x_1'(t) \\ \vdots \\ x_{k-2}'(t) \\ x_{k-1}'(t) \end{pmatrix} \;=\; \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{k-1}(t) \\ g(t, x_0(t), \ldots, x_{k-1}(t)) \end{pmatrix} \;=\; f(t, x(t)).$$

Sei nun eine Lösung dieses Systems gegeben durch differenzierbare Funktionen x_j für $j = 0, \ldots, k-1$. Dann ist

$$y(t) \;=\; x_0(t)$$

k-mal differenzierbar, denn es gilt

$$y^{(j)}(t) \;=\; x_j(t)$$

für $j = 0, \ldots, k-1$ und alle x_j sind mindestens einmal differenzierbar. Weiterhin folgt

$$\begin{aligned} y^{(k)}(t) \;&=\; \left(y^{(k-1)}\right)'(t) \;=\; x_{k-1}'(t) \\ &=\; g(t, x_0(t), \ldots, x_{k-1}(t)) \;=\; g(t, y(t), \ldots, y^{(k-1)}(t)) \end{aligned}$$

und damit haben wir die urspünglich gegebene Differentialgleichung in ein System von Differentialgleichungen erster Ordnung überführt. □

Lemma 7.3. *Jedes Anfangswertproblem der Form $x'(t) = f(t, x(t))$ mit $x(t_0) = x_0$ lässt sich in ein äquivalentes autonomes Anfangswertproblem transformieren.*

Beweis. Wir definieren $s(t) = t$ und

$$y(t) = \begin{pmatrix} s(t) \\ x(t) \end{pmatrix}.$$

Dazu betrachten wir das autonome System

$$y'(t) = \begin{pmatrix} s'(t) \\ x'(t) \end{pmatrix} = \begin{pmatrix} 1 \\ f(y(t)) \end{pmatrix} \quad \text{mit} \quad y(t_0) = \begin{pmatrix} t_0 \\ x_0 \end{pmatrix}. \tag{7.2}$$

Wir wollen nun zeigen, dass beiden Differentialgleichungen äquivalent sind.

Zunächst sei x eine Lösung von $x'(t) = f(t, x(t))$ mit $x(t_0) = x_0$. Dann erhalten wir mit $s(t) = t$ direkt

$$\begin{aligned} y'(t) = \begin{pmatrix} s'(t) \\ x'(t) \end{pmatrix} &= \begin{pmatrix} 1 \\ f(t, x(t)) \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ f(s(t), x(t)) \end{pmatrix} = \begin{pmatrix} 1 \\ f(y(t)) \end{pmatrix} \end{aligned}$$

und mit dem zugehörigen Anfangswert $y(t_0) = (t_0, x(t_0)) = (t_0, x_0)$ eine Lösung von (7.2).

Nun sei $y(t) = (s(t), x(t))$ eine Lösung von (7.2). Dann gilt $s'(t) = 1$ und durch Integration ergibt sich $s(t) = t + C$ mit einer Integrationskonstanten C. Wegen des Anfangswertes $s(t_0) = t_0$ ist $C = 0$ und es folgt

$$x'(t) = f(y(t)) = f(s(t), x(t)) = f(t, x(t)).$$

Damit ist x eine Lösung von $x'(t) = f(t, x(t))$, die der Anfangsbedingung $x(t_0) = x_0$ genügt. □

Den Übergang eines Anfangswertproblemes nach Lemma 7.3 nennt man auch ***Autonomisierung*** des Anfangswertproblemes.

Bevor wir uns mit der Existenz und der Eindeutigkeit einer Lösung von Anfangswertproblemen befassen, wollen wir uns zwei weitere Anwendungsgebiete ansehen.

Beispiel 7.3 (Lotka-Volterra-Zyklus). *Wir betrachten 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*

$$\begin{aligned} x_J(t) &= \text{Größe der Jäger-Population zur Zeit } t, \\ x_B(t) &= \text{Größe der Beute-Population zur Zeit } t. \end{aligned}$$

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:

(1) *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.*

(2) *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 mit geeigneten Parametern $\gamma, \delta > 0$

$$x_J'(t) = \gamma x_J(t) x_B(t) - \delta x_J(t).$$

Die Interpretation dieser Gleichung ist wie folgt:

(1) *Die Jäger-Population wächst exponentiell mit Rate γ und proportional zur Beute-Population x_B.*

(2) *Die natürliche Sterberate ist (bei exponentiellem Wachstum) gegeben durch $x_J' = -\delta x_J$.*

Die Lösung dieses Systems von Differentialgleichungen führt zu periodischen Lösungen, die man auch **Lotka-Volterra-Zyklus** *nennt.*

Ein weiteres klassisches Anwendungsgebiet von Anfangswertproblemen in der Physik zeigt das folgende Beispiel.

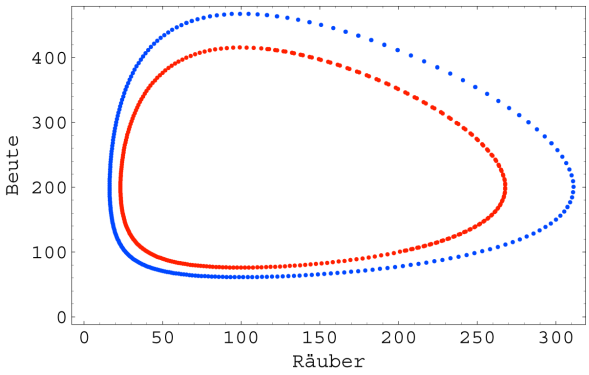


Abbildung 7.2: Beispiel eines Lotka-Volterra-Zyklus für unterschiedliche Startpopulationen.

Beispiel 7.4 (Bewegung eines Massepunktes). *Die Bewegung eines Massepunktes zur Zeit t am Ort x wird gegeben durch die Differentialgleichung*

$$m \cdot x''(t) \;=\; g(t,x),$$

dabei beschreibt $g(t,x)$ die Wirkung äußerer Kräfte.

Als Beispiel betrachten wir eine eingespannten Feder mit der Federkonstanten D und erhalten nach dem Hookeschen Gesetz $g(t,x) = -Dx$. Weiterhin sei der Anfangspunkt $x_0 = x(t_0)$ und die Anfangsgeschwindigkeit $v_0 = x_0' = x'(t_0)$ aus einem Experiment bekannt.

Das System kann dann in das folgende äquivalente System erster Ordnung überführt werden:

$$x'(t) \;=\; \begin{pmatrix} x_1'(t) \\ x_2'(t) \end{pmatrix} \;=\; \begin{pmatrix} x_2(t) \\ -\frac{k}{m} \cdot x_1(t) \end{pmatrix}$$

mit den Anfangsbedingungen

$$x(t_0) \;=\; \begin{pmatrix} x_1(t_0) \\ x_2(t_0) \end{pmatrix} \;=\; \begin{pmatrix} x_0 \\ v_0 \end{pmatrix}.$$

Dieses System von Differentialgleichungen erster Ordnung ist linear und autonom. Die Lösung wird gegeben durch

$$x(t) \;=\; x_1(t) \;=\; x_0 \cdot \cos\left(\sqrt{\tfrac{k}{m}}\, t\right) + v_0 \cdot \sin\left(\sqrt{\tfrac{k}{m}}\, t\right).$$

7.2 Existenz und Eindeutigkeit

In diesem Abschnitt betrachten wir mit

$$x'(t) \;=\; f(x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

weiterhin autonome Anfangswertprobleme erster Ordnung und wollen die Existenz und die Eindeutigkeit von Lösungen untersuchen. Dazu betrachten wir zunächst zwei Beispiele.

Beispiel 7.5. *Gegeben sei das Anfangswertproblem*

$$x'(t) \;=\; |x(t)|^{\alpha} \qquad \text{mit} \qquad x(0) \;=\; 0$$

für einen Parameter $\alpha \in (0,1)$*. Die Differentialgleichung hat die beiden Lösungen*

$$\tilde{x}(t) \;=\; 0 \qquad \text{und} \qquad x(t) \;=\; \begin{cases} ((1-\alpha)t)^{1/(1-\alpha)} & \text{für} \quad t \geq 0, \\ 0 & \text{für} \quad t < 0 \end{cases}$$

wie man leicht nachrechnet.

Dieses Beispiel zeigt also, dass die Lösung im Allgemeinen nicht eindeutig ist.

Beispiel 7.6. *Gegeben sei das Anfangswertproblem*

$$x'(t) \;=\; (x(t))^2 \qquad \text{mit} \qquad x(0) \;=\; 1.$$

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.*

Das zweite Beispiel zeigt damit, dass keine Lösung auf ganz I existieren muss.

Für alle folgenden Betrachtungen ist die nächste Aussage von zentraler Bedeutung.

Lemma 7.4. *Sei* $D \subseteq \mathbb{R}^{d+1}$ *offen, sei* $f : D \to \mathbb{R}^d$ *stetig, sei* $a \leq t_0 \leq b$ *und sei* $x : [a,b] \to \mathbb{R}^d$ *eine Funktion. Weiter gelte*

$$\{(t, x(t)) \;:\; t \in [a,b]\} \;\subset\; D.$$

Dann sind die folgenden Aussagen äquivalent:

(1) *$x(t)$ ist stetig differenzierbar und löst das Anfangswertproblem*

$$x'(t) \;=\; f(t, x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

für alle $t \in [a, b]$.

(2) *$x(t)$ ist stetig und erfüllt die* **Integralgleichung**

$$x(t) \;=\; x_0 + \int_{t_0}^{t} f(\tau, x(\tau))\,\mathrm{d}\tau \tag{7.3}$$

für alle $t \in [a, b]$.

Beweis. Zunächst sei $x(t)$ mit $x'(t) = f(t, x(t))$ und $x(t_0) = x_0$ eine stetig differenzierbare Lösung des Anfangswertproblemes. Nach dem Hauptsatz der Differential- und Integralrechnung gilt dann

$$x(t) \;=\; x(t_0) + \int_{t_0}^{t} x'(\tau)\,\mathrm{d}\tau \;=\; x_0 + \int_{t_0}^{t} f(\tau, x(\tau))\,\mathrm{d}\tau.$$

Nun gelte umgekehrt die Integralgleichung

$$x(t) \;=\; x_0 + \int_{t_0}^{t} f(\tau, x(\tau))\,\mathrm{d}\tau.$$

Da $f(t, x(t))$ und $x(t)$ beide stetig sind, ist

$$\int_{t_0}^{t} f(\tau, x(\tau))\,\mathrm{d}\tau$$

stetig nach t differenzierbar und somit auch $x(t)$. Die Ableitung von $x(t)$ wird gegeben durch

$$x'(t) \;=\; \frac{\mathrm{d}}{\mathrm{d}t} \int_{t_0}^{t} f(\tau, x(\tau))\,\mathrm{d}\tau \;=\; f(t, x(t))$$

nach dem Hauptsatz der Differential- und Integralrechnung. Weiter gilt

$$x(t_0) \;=\; x_0 + \int_{t_0}^{t_0} f(\tau, x(\tau))\,\mathrm{d}\tau \;=\; x_0,$$

also ist auch die Anfangsbedingung erfüllt. □

Der große Vorteil von Lemma 7.4 liegt darin, dass wir durch die Integralgleichung (7.3) eine Fixpunktgleichung in der unbekannten Funktion x gefunden haben.

Sei $\mathcal{C}([a,b],\mathbb{R}^d)$ der (unendlich-dimensionale) Raum aller stetigen Funktionen, die von $I=[a,b]$ in den $\mathbb{R}^d$ abbilden. Dann definieren wir den Operator F, den wir auf Funktionen $x : I \to \mathbb{R}^d$ aus $\mathcal{C}([a,b],\mathbb{R}^d)$ anwenden wollen, durch

$$(F(x))(t) \;:=\; x_0 + \int_{t_0}^{t} f(\tau, x(\tau))\,\mathrm{d}\tau.$$

Wir können die Integralgleichung (7.3) dann schreiben als

$$x(t) \;=\; (F(x))(t).$$

Unsere gesuchte Lösung x kann also als Lösung einer Fixpunktgleichung in einem unendlich-dimensionalen Raum aufgefasst werden. Wir wollen darauf nun den Banachschen Fixpunktsatz anwenden, welcher nach Kapitel 2 ja auch in unendlich-dimensionalen Banachräumen gilt.

Bevor wir auf dieser Idee basierend das Hauptergebnis dieses Abschnitts angeben, fassen wir noch einige Schreibweisen zusammen.

Notation 7.3. Sei $D \subset \mathbb{R}^{d+1}$ und $f : D \to \mathbb{R}^d$. Dann heißt f ***lokal Lipschitzstetig*** bezüglich der letzten d Variablen, falls zu jedem $(t_0,x_0) \in D$ eine Umgebung $U = U(t_0,x_0) \subset D$ und eine eine Konstante $L = L(t_0,x_0)$ gibt mit

$$\|f(t,x) - f(t,y)\|_2 \;\le\; L \cdot \|x-y\|_2$$

für alle $(t,x),(t,y) \in U$. Dabei ist $\|\cdot\|_2$ die Euklidische Norm.

f heißt ***global Lipschitzstetig*** bezüglich der letzten d Variablen, falls es eine Konstante L gibt mit

$$\|f(t,x) - f(t,y)\|_2 \;\le\; L \cdot \|x-y\|_2$$

für alle $x,y \in \mathbb{R}^d$.

Damit formulieren wir nun das Hauptergebnis dieses Abschnitts.

Satz 7.5 (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 lokal Lipschitzstetig. Dann existiert zu jedem $(t_0,x_0) \in D$ eine Umgebung I von t_0, auf der das Anfangswertproblem*

$$x'(t) \;=\; f(t,x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

eindeutig lösbar ist.

Beweis. Wir wollen an dieser Stelle nur die Idee des Beweises wiedergeben, ausführlich kann dieser in Schöbel (2007) oder Hohage (2006) nachgelesen werden.

Wie bereits beschrieben, wollen wir den Banachschen Fixpunktsatz anwenden. Dazu geben wir nun die Voraussetzungen an.

Zunächst lässt sich zu jedem $(t_0, x_0) \in D$ ein Intervall $I(t_0) = [t_0 - \alpha, t_0 + \alpha]$ mit einem geeigneten $\alpha > 0$ finden, auf welchem wir die lokale Existenz und Eindeutigkeit der Lösung zeigen wollen.

Dazu verwenden wir den Raum $X := \mathcal{C}(I, \mathbb{R}^d)$ aller stetigen Funktionen, die von I in den $\mathbb{R}^d$ abbilden. Als Norm verwenden wir

$$\|x\|_B \;:=\; \sup_{t \in I} \exp(-2L \cdot |t - t_0|) \cdot \|x(t)\|_2$$

für alle $x \in X$, wobei L die Lipschitskonstante der lokalen Lipschitzstetigkeit ist. Als Teilmenge U von X verwenden wir

$$U \;:=\; \left\{ x \in X \;:\; \sup_{t \in I} \|x(t) - x_0\| \leq \beta \right\}$$

mit einem geeigneten $\beta > 0$. Als Abbildung von U nach X verwenden wir natürlich den Opterator $F : U \to X$ mit

$$(F(x))(t) \;:=\; x_0 + \int_{t_0}^{t} f(\tau, x(\tau))\, \mathrm{d}\tau.$$

Nun lassen sich alle Voraussetzungen des Banachschen Fixpunktsatzes zeigen:

(1) $(X, \| \cdot \|_B)$ ist tatsächlich ein Banachraum.

(2) Die Menge U ist abgeschlossen.

(3) Es gilt $F(x) \in U$ für alle $x \in U$ und damit ist F eine Selbstabbildung.

(4) F ist eine Kontraktion mit Kontraktionsfaktor $1/2$.

Der Banachsche Fixpunktsatz liefert damit die Aussage, dass das Anfangswertprobleme eine eindeutige Lösung in U hat. Um allgemein eine eindeutige Lösung zu garantieren, muss nun noch gezeigt werden, dass F keinen Fixpunkt in $X \setminus U$ besitzen kann. □

Ganz analog unter der Verwendung des Banachschen Fixpunktsatzes lässt sich auch eine globale Existenz und Eindeutigkeit zeigen.

Satz 7.6. *Sei $I \subset \mathbb{R}$ ein Intervall, sei $D = I \times \mathbb{R}^d$ und sei $f : D \to \mathbb{R}^d$ stetig und bezüglich der letzten d Variablen global Lipschitzstetig. Dann existiert zu jedem $(t_0, x_0) \in D$ eine eindeutige Lösung des Anfangswertproblems*

$$x'(t) \;=\; f(t, x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

und $x : I \to \mathbb{R}^d$.

Der Banachsche Fixpunktsatz liefert damit 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 sich durch die ***Picard-Iteration*** auf Anfangswertprobleme anwenden:

$$x^{(n+1)}(t) \;:=\; x^{(n)}(t_0) + \int_{t_0}^{t} f(\tau, x^{(n)}(\tau))\,\mathrm{d}\tau.$$

Als Startwert kann zum Beispiel $x^{(0)}(t) := x_0$ gewählt werden.

Das resultierende Verfahren ist allerdings durch die dazu nötige numerische Auswertung der zahlreich auftretenden Integrale ineffizient und wird in der Praxis fast nicht verwendet.

Wir untersuchen nun noch einmal das Beispiel 7.6 von zuvor unter Berücksichtigung des Satzes von Picard-Lindelöf.

Beispiel 7.7. *Gegeben sei wieder das Anfangswertproblem*

$$x'(t) \;=\; f(t, x(t)) \;=\; (x(t))^2 \qquad \text{mit} \qquad x(0) \;=\; 1$$

aus Beispiel 7.6. Wir hatten bereits gesehen, das diese Problem keine globale Lösung für $t > 0$ *besitzt. Mit* $f(t, x) = x^2$ *erhalten wir*

$$\|f(t, x) - f(t, y)\|_2 \;=\; |x^2 - y^2| \;=\; |x + y| \cdot |x - y| \;\leq\; L \cdot |x - y|$$

für alle $x, y \in I$*, falls* $L \geq |x + y|$ *für alle* $x, y \in I$.

Das ist auf jedem beschränkten Intervall erfüllt, nicht jedoch auf $I = [0, \infty)$ *oder auf* $I = \mathbb{R}$*. Das Anfangswertproblem erfüllt daher die Voraussetzungen von Satz 7.5, aber nicht die von Satz 7.6.*

Wir untersuchen nun noch eine weitere Aussage zur globalen Eindeutigkeit.

Satz 7.7 (Globale Eindeutigeit). *Sind die Voraussetzungen von Satz 7.5 erfüllt und sind* $x(t)$ *und* $y(t)$ *Lösungen des Anfangswertproblemes*

$$x'(t) \;=\; f(t, x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

auf einem beliebigen Intervall I *mit* $t_0 \in I$*, so gilt* $x(t) = y(t)$ *für alle* $t \in I$.

Beweis. Sei $I = [a, b]$, sei $t_0 \in I$ und seien $x(t)$ und $y(t)$ zwei Lösungen des gegebenen Anfangswertproblemes. Wir wählen $I' \subset I$ als das längste Intervall mit $x(t) = y(t)$ für alle $t \in I'$ und möchten nun zeigen, dass dann $I = I'$ gilt.

Angenommen, dies ist gilt nicht der Fall, dann sei $I' = [a', b'] \subset I$ mit $I' \neq I$. Sei ohne Beschränkung der Allgemeinheit $b' < b$ und betrachte dazu das neue Anfangswertproblem

$$z'(t) \;=\; f(t, z(t)) \qquad \text{mit} \qquad z(b') \;=\; x(b'). \tag{7.4}$$

Nach dem Satz 7.5 von Picard-Lindelöf existiert nun eine Umgebung $U = (b' - \alpha, b' + \alpha)$ mit $\alpha > 0$, auf der das neue Anfangswertproblem (7.4) eindeutig lösbar ist. Weil x und y nun aber beides Lösungen von (7.4) sind, folgt $x(t) = y(t)$ für alle $t \in U$. Das ist ein aber Widerspruch zur Maximalität von I'. □

Abschließend geben wir noch ein Kriterium an, anhand dessen sich die geforderte Lipschitz-Bedingung des Satzes von Picard-Lindelöf nachweisen lassen.

Lemma 7.8. *Sei $f : I \times \mathbb{R}^d \to \mathbb{R}^d$ bezüglich seiner letzten d Variablen stetig partiell differenzierbar. Dann ist f stetig und bezüglich der letzten d Variablen lokal Lipschitzstetig für alle $(t_0, x_0) \in I \times \mathbb{R}^d$.*

Beweis. Der Beweis ist ähnlich zum Beweis von Lemma 2.10.

Da f bezüglich seiner letzten d stetig partiell differenzierbar ist, existiert die Ableitung

$$D_x f(t, x) : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^d$$

und es gilt

$$L \;=\; \sup_{(t,x) \in U(t_0, x_0)} \|D_x f(t, x)\|_2 \;<\; \infty$$

für eine kompakte Umgebung $U = U(t_0, x_0)$, die zusätzlich noch konvex sei. Mit

$$g(\xi) \;:=\; f(t, x + \xi(y - x))$$

erhalten wir dann

$$\begin{aligned} \|f(t, y) - f(t, x)\|_2 \;&=\; \|g(1) - g(0)\|_2 \;= \left\| \int_0^1 g'(\tau)\, \mathrm{d}\tau \right\| \\ &=\; \left\| \int_0^1 D_x f(t, x + \tau(y - x)) \cdot (y - x)\, \mathrm{d}\tau \right\|_2 \\ &\leq\; \int_0^1 \|D_x f(t, x + \tau(y - x))\|_2 \cdot \|y - x\|_2\, \mathrm{d}\tau \end{aligned}$$

$$\leq \int_0^1 L \cdot \|y - x\|_2 \, \mathrm{d}\tau \;=\; L \cdot \|y - x\|_2$$

mit $L < \infty$, da U kompakt ist. □

7.3 Evolutionen

Wir werden in diesem Abschnitt den Satz von Picard-Lindelöf verwenden, um den Begriff der Evolution zu definieren und zu untersuchen. Dieses Konzept wird für die danach folgenden numerischen Lösungsverfahren sehr nützlich sein.

Definition 7.4. Sei $D \subset \mathbb{R}^{d+1}$ offen und sei $f : D \to \mathbb{R}^d$ stetig und lokal Lipschitzstetig bezüglich der letzten d Variablen. Weiter seien $t_0, t \in I$ und $|t - t_0|$ hinreichend klein.

Dann definieren wir die zweiparametrige Funktion

$$\Phi^{t,t_0} : \mathbb{R}^d \to \mathbb{R}^d \qquad \text{mit} \qquad \Phi^{t,t_0}(x_0) \;=\; x(t),$$

wobei $x(t)$ die nach dem Satz von Picard-Lindelöf lokal eindeutige Lösung des Anfangswertproblemes

$$x'(t) \;=\; f(t, x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

ist. Man nennt Φ die ***Evolution*** der Differentialgleichung $x'(t) = f(t, x(t))$.

Damit bildet Φ^{t,t_0} den Wert der Lösung $x(t)$ zur Zeit t_0 auf den Wert der gleichen Lösung zur Zeit t ab.

Beispiel 7.8. *Gegeben sei noch einmal das Anfangswertproblem*

$$x'(t) \;=\; f(t, x(t)) \;=\; (x(t))^2 \qquad \text{mit} \qquad x(0) \;=\; x_0$$

aus Beispiel 7.6. Die lokal eindeutige Lösung zu (t_0, x_0) mit $t_0 = 0$ und $x_0 > 0$ wird dann gegeben durch

$$x(t) \;=\; \frac{x_0}{1 - t \cdot x_0}$$

für $t < 1/x_0$. Für die Evolution gilt entsprechend im Fall $t > 0$

$$\Phi^{t,0}(x_0) \;=\; \frac{x_0}{1 - t \cdot x_0}$$

für $x_0 < 1/t$.

Lemma 7.9. *Sei $D \subset \mathbb{R}^{d+1}$ offen und sei $f : D \to \mathbb{R}^d$ stetig und lokal Lipschitzstetig bezüglich der letzten d Variablen.*

Dann besitzt die Evolution Φ der Differentialgleichung $x'(t) = f(t, x(t))$ die folgenden Eigenschaften für alle $(t_0, x_0) \in D$ und $|t_1 - t_0|$, $|t_2 - t_0|$ und $|t - t_0|$ hinreichend klein:

(1) $\Phi^{t_0,t_0}(x_0) = x_0$,

(2) $\frac{\partial}{\partial \tau} \Phi^{t+\tau,t}(x_0)\Big|_{\tau=0} = f(t, x_0)$,

(3) $\Phi^{t_2,t_0}(x_0) = \Phi^{t_2,t_1}(\Phi^{t_1,t_0}(x_0))$.

Weiter ist Φ durch diese drei Bedingungen eindeutig charakterisiert.

Beweis. Zunächst zeigen wir alle drei Punkte der Reihe nach.

Nach Definition der Evolution gilt **(1)**, da $\Phi^{t_0,t_0}(x_0) = x(t_0) = x_0$.

Um **(2)** zu zeigen, seien x_0 und t fest. Weiter sei $x(t)$ die Lösung des Anfangswertproblemes zum Startwert (t, x_0). Dann definieren wir

$$g(\tau) \;:=\; \Phi^{t+\tau,t}(x_0) \;=\; x(t+\tau)$$

und damit gilt

$$\frac{\partial}{\partial \tau} \Phi^{t+\tau,t}(x_0) \;=\; g'(\tau) \;=\; x'(t+\tau) \;=\; f(t+\tau, x(t+\tau)).$$

Schließlich folgt somit

$$\frac{\partial}{\partial \tau} \Phi^{t+\tau,t}(x_0)\Big|_{\tau=0} \;=\; g'(0) \;=\; f(t, x(t)) \;=\; f(t, x_0).$$

Nun sei $x(t)$ eine Lösung des Anfangswertproblemes

$$x'(t) \;=\; f(t, x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

und damit $\Phi^{t,t_0}(x_0) = x(t)$ für alle t nahe genug an t_0. Damit folgt

$$\Phi^{t_2,t_1}(\Phi^{t_1,t_0}(x_0)) \;=\; \Phi^{t_2,t_1}(x(t_1)) \;=\; x(t_2) \;=\; \Phi^{t_2,t_0}(x_0),$$

wobei die vorletzte Gleichheit gilt, da für $t_2 - t_0$ hinreichend klein $x(t)$ auch eine Lösung des Anfangswertproblemes

$$y'(t) \;=\; f(t, y(t)) \qquad \text{mit} \qquad y(t_1) \;=\; x(t_1)$$

ist. Damit ist auch **(3)** gezeigt.

Es bleibt noch die Eindeutigkeit zu zeigen. Dazu sei $\Psi^{t,t_0} : \mathbb{R}^d \to \mathbb{R}^d$ eine weitere Funktion, die ebenfalls alle drei Bedingungen erfüllt. Weiter sei (t_0, x_0) beliebig und wir definieren

$$x(t) \;:=\; \Psi^{t,t_0}(x_0).$$

Dann gilt

$$\begin{aligned} x'(t) &= \frac{\partial}{\partial\tau}\Psi^{t+\tau,t_0}(x_0)\Big|_{\tau=0} = \frac{\partial}{\partial\tau}\left(\Psi^{t+\tau,t}\left(\Psi^{t,t_0}(x_0)\right)\right)\Big|_{\tau=0} \\ &= f(t,\Psi^{t,t_0}(x_0)) \;=\; f(t,x(t)) \end{aligned}$$

und zudem ist $x(t_0) = \Psi^{t_0,t_0}(x_0) = x_0$. Damit ist

$$x(t) \;=\; \Phi^{t,t_0}(x_0)$$

nach dem Satz von Picard-Lindelöf die lokal eindeutige Lösung des Anfangswertproblemes

$$x'(t) \;=\; f(t,x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

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. □

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*, also nur für Werte $t \geq t_0$.

Definition 7.5. Sei $D \subset \mathbb{R}^d$, sei $t_0 \in \mathbb{R}$ und die Funktion $f : [t_0, \infty) \times D \to \mathbb{R}^d$ sei ***einseitig*** Lipschitzstetig mit Konstante $L^+ \in \mathbb{R}$, es gelte also

$$(x - y)^T \cdot (f(t,x) - f(t,y)) \;\leq\; L^+ \cdot \|x - y\|_2^2$$

für alle $x, y \in D$ und alle $t \in [t_0, \infty)$.

Ist diese Bedingung für ein $L^+ \leq 0$ erfüllt, so heißt die zugehörige Differentialgleichung $x'(t) = f(t, x(t))$ ***dissipativ***.

Es sei bemerkt, dass aus der globalen Lipschitzstetigkeit auf $[t_0, \infty)$ nach der Cauchy-Schwarz Ungleichung auch die einseitige Lipschitz-Bedingung folgt. Die Umkehrung gilt aber nicht, wie das folgende Beispiel zeigt.

Beispiel 7.9. *Die Funktion $f(t,x) = -x$ erfüllt die einseitige Lipschitz-Bedingung mit $L^+ = -1$, denn*

$$(x-y)\cdot(f(t,x)-f(t,y)) \;=\; (x-y)\cdot(y-x) \;=\; -(x-y)^2 \;=\; -1\cdot\|x-y\|_2^2.$$

Dagegen ergibt die globale Lipschitz-Bedingung

$$|f(t,x)-f(t,y)| \;=\; |x-y| \;\leq\; L\cdot|y-x|$$

und gilt damit nur für $L \geq 1$.

Satz 7.10. *Ist $f : [0,\infty)\times D \to \mathbb{R}^d$ einseitig Lipschitzstetig mit Konstante L^+, so gilt für die Evolution Φ von $x'(t) = f(t,x(t))$*

$$\|\Phi^{t,t_0}(x_0) - \Phi^{t,t_0}(y_0)\|_2 \;\leq\; \mathrm{e}^{L^+(t-t_0)}\cdot\|x_0-y_0\|_2.$$

Für dissipative Systeme gilt insbesondere

$$\|\Phi^{t,t_0}(x_0) - \Phi^{t,t_0}(y_0)\|_2 \;\leq\; \|x_0-y_0\|_2.$$

7.4 Euler-Verfahren

Wir haben bereits gesehen, dass die Lösung einer Differentialgleichung bei geeigneten Eingangsdaten immer existiert. Trotzdem ist die Lösung im Allgemeinen selbst bei skalaren Differentialgleichungen mit $d = 1$ nicht in geschlossener Form darstellbar.

Die Grundidee der numerischen Lösung von Anfangswertproblemen ist es daher die Lösung $x(t)$ näherungsweise an diskreten Punkten zu ermitteln. Gesucht sind also Näherungswerte von $x(t)$ für $t \in \Delta = \{t_0, t_1, \ldots, t_N\}$ mit $t_0 < t_1 < \ldots < t_N$.

Notation 7.6. Gegeben seien reelle Zahlen $t_0 < t_1 < \ldots < t_N = T$. Dann nennen wir

$$\Delta \;:=\; \{t_0, t_1, \ldots, t_N\}$$

ein ***Gitter*** auf $[t_0, T]$. Die Werte $T_j = t_{j+1} - t_j$ für $j = 0, \ldots, N-1$ heißen ***Schrittweite*** und die ***Feinheit*** eines Gitters ist

$$\tau_\Delta \;:=\; \max_{j=0,\ldots,N-1} T_j.$$

Gesucht ist dann eine ***Gitterfunktion***

$$x_\Delta : \Delta \to \mathbb{R}^d,$$

welche die Lösung von $x'(t) = f(t, x(t))$ mit $x'(t_0) = x_0$ auf dem Gitter möglichst gut approximiert. Vereinfacht schreiben wir $x_j := x_\Delta(t_j) \in \mathbb{R}^d$ für $j = 0, \ldots, N$.

Im Folgenden werden wir die exakte Evolution Φ der Differentialgleichung $x'(t) = f(t, x(t))$ mit $x'(t_0) = x_0$ durch eine ***diskrete Evolution*** Ψ ersetzen.

Notation 7.7. Gegeben sei eine Differentialgleichung $x'(t) = f(t, x(t))$ mit $x'(t_0) = x_0$ und ein Gitter $\Delta = \{t_0, \ldots, t_N\}$. Dann approximieren wir die exakte Evolution

$$x(t_{j+1}) \;=\; \Phi^{t_{j+1},t_j}(x(t_j)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

durch die diskrete Evolution

$$x_\Delta(t_{j+1}) \;=\; \Psi^{t_{j+1},t_j}(x_\Delta(t_j)) \qquad \text{mit} \qquad x_\Delta(t_0) \;=\; x_0.$$

Eine diskrete Evolution zu einem Gitter Δ liefert uns damit ein Verfahren zur diskreten Approximation der exakten Lösung $x(t)$ durch die Gitterfunktion $x_\Delta(t)$. Ein solches Verfahren heißt ***Einschrittverfahren***, da nur $x_\Delta(t_j)$ in die Berechnung von $x_\Delta(t_{j+1})$ eingeht.

Für alle folgenden Verfahren werden wir stets ein äquidistantes Gitter Δ mit fester Schrittweite τ verwenden, also

$$\Delta \;=\; \{t_0, t_0 + \tau, \ldots, t_0 + N \cdot \tau\}.$$

Um nun eine diskreten Evolution herzuleiten, nutzen wir die Integraldarstellung

$$x(t_0 + \tau) \;=\; x_0 + \int_{t_0}^{t_0+\tau} f(t, x(t))\,\mathrm{d}t \tag{7.5}$$

aus Lemma 7.4. Damit können wir nun das Euler-Verfahren motivieren.

Die Idee besteht darin, dass wir das Integral in Gleichung (7.5) durch die Rechteck-Regel mit Funktionsauswertung am linken Randpunkt approximieren. Somit erhalten wir

$$\begin{aligned} x(t_1) \;&=\; x(t_0 + \tau) \;=\; x_0 + \int_{t_0}^{t_0+\tau} f(t, x(t))\,\mathrm{d}t \\ &\approx\; x_0 + \tau \cdot f(t_0, x(t_0)) \;=\; x_0 + \tau \cdot f(t_0, x_0). \end{aligned}$$

Daher wählen wir für unsere Gitterfunktion

$$x_1 \;=\; x_\Delta(t_1) \;=\; x_0 + \tau \cdot f(t_0, x_0).$$

Ist nun $x(t_0) \approx x_1$ approximativ bekannt, erhalten wir analog durch die Rechteck-Regel am linken Randpunkt

$$x(t_2) \;=\; x_1 + \int_{t_1}^{t_1+\tau} f(t, x(t))\,\mathrm{d}t \;\approx\; x_1 + \tau \cdot f(t_1, x_1) \;=\; x_\Delta(t_2) \;=\; x_2.$$

Dies führt uns zur Rekursion

$$x_{j+1} \;=\; x_\Delta(t_{j+1}) \;=\; x_j + \tau \cdot f(t_j, x_j) \qquad \text{mit} \qquad x_0 \;=\; x(t_0).$$

Die diskrete Evolution des Euler-Verfahrens ist damit

$$\Psi^{t+\tau,t}(x) \;=\; x + \tau \cdot f(t, x).$$

Dieses Ergebniss fassen wir noch einmal zusammen.

Korollar 7.11 (Explizites Euler-Verfahren). *Gegeben sei eine Differentialgleichung*

$$x'(t) \;=\; f(t, x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

sowie ein äquidistantes Gitter Δ mit Schrittweite τ. Dann liefert uns das **explizite Euler-Verfahren** *die Gitterfunktion*

$$x_{j+1} \;=\; x_\Delta(t_{j+1}) \;=\; \Psi^{t_j+\tau,t_j}(x_j) \;=\; x_j + \tau \cdot f(t_j, x_j).$$

Wir wollen nun noch ein Beispiel zum expliziten Euler-Verfahren diskutieren.

Beispiel 7.10. *Gegeben sei die Differentialgleichung*

$$x'(t) \;=\; f(t, x(t)) \;=\; \lambda \cdot x(t) \qquad \text{mit} \qquad x(0) \;=\; 1$$

mit $t_0 = 0$. Dann ergibt das explizite Euler-Verfahren

$$x_{j+1} \;=\; x_j + \tau \cdot \lambda \cdot x_j \qquad \text{mit} \qquad x_0 \;\; x(0) \;=\; 1.$$

Eine weitere Version des Euler-Verfahrens ergibt sich, wenn wir das Integral in Gleichung (7.5) durch die Rechteck-Regel mit Funktionsauswertung am rechten Randpunkt approximieren. Analog erhalten wir die Rekursion

$$x_{j+1} \;=\; x_\Delta(t_{j+1}) \;=\; x_j + \tau \cdot f(t_j, x_{j+1}) \qquad \text{mit} \qquad x_0 \;=\; x(t_0).$$

Dieses Verfahren wird als implizites Euler-Verfahren bezeichnet, das wir in jedem Schritt zur Bestimmung von x_{j+1} ein nicht lineares Gleichungssystem mit d Unbekannten und d Gleichungen lösen müssen.

Korollar 7.12 (Implizites Euler-Verfahren). *Gegeben sei eine Differentialgleichung*

$$x'(t) \;=\; f(t, x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

sowie ein äquidistantes Gitter Δ mit Schrittweite τ. Dann liefert uns das **implizite Euler-Verfahren** *die Gitterfunktion*

$$x_{j+1} \;=\; x_\Delta(t_{j+1}) \;=\; x_j + \tau \cdot f(t_j, x_{j+1}).$$

Auch zum impliziten Euler-Verfahren diskutieren wir noch einmal das Beispiel von zuvor.

Beispiel 7.11. *Gegeben sei wieder die Differentialgleichung*

$$x'(t) \;=\; f(t, x(t)) \;=\; \lambda \cdot x(t) \qquad \text{mit} \qquad x(0) \;=\; 1$$

mit $t_0 = 0$. Dann ergibt das implizite Euler-Verfahren

$$x_{j+1} \;=\; x_j + \tau \cdot \lambda \cdot x_{j+1} \qquad \text{mit} \qquad x_0 \;\; x(0) \;=\; 1.$$

Dies lässt sich auflösen zu

$$x_{j+1} \;=\; \frac{x_j}{1 - \tau \cdot \lambda} \qquad \text{mit} \qquad x_0 \;\; x(0) \;=\; 1.$$

7.5 Konsistenz und Konvergenz

Nachdem wir mit dem Euler-Verfahren erste Beispiele zur numerischen Lösung von Anfangswertproblemen kennengelernt haben, wollen wir nun das Konvergenzverhalten theoretisch untersuchen. Dazu fordern wir zunächst die ersten beiden der drei Eigenschaften einer exakten Evolution Φ aus Lemma 7.9 auch für die diskrete Evolution Ψ.

Definition 7.8. Eine diskrete Evolution Ψ heißt ***konsistent*** zur Differentialgleichung $x'(t) = f(t, x(t))$ mit $f : D \to \mathbb{R}^n$, wenn für alle $(t_0, x_0) \in D$ die folgenden beiden Bedingungen erfüllt sind:

(1) $\Psi^{t_0, t_0}(x_0) = x_0$,

(2) $\frac{\partial}{\partial \tau} \Psi^{t_0+\tau, t_0}(x_0)\Big|_{\tau=0} = f(t_0, x_0)$.

Wir wollen nun Kriterien angeben, wann eine diskrete Evolution konsistent ist.

Lemma 7.13. *Die diskrete Evolution $\Psi^{t_0+\tau,t_0}(x_0)$ sei für alle $(t_0, x_0) \in D$ und hinreichend kleines τ nach τ differenzierbar. Dann sind die folgenden Aussagen äquivalent:*

(1) *Ψ ist konsistent.*

(2) *Es gibt eine bezüglich τ stetige Verfahrensfunktion $\phi = \phi(t_0, x_0, \tau)$ mit den Eigenschaften*

$$\begin{aligned} \Psi^{t_0+\tau,t_0}(x_0) &= x_0 + \tau \cdot \phi(t_0, x_0, \tau), && (7.6)\\ \phi(t_0, x_0, 0) &= f(t_0, x_0). && (7.7) \end{aligned}$$

(3) *Es gilt*

$$\lim_{\tau\to 0} \frac{1}{\tau} \left\| \Psi^{t_0+\tau,t_0}(x_0) - \Phi^{t_0+\tau,t_0}(x_0) \right\| = 0. \qquad (7.8)$$

Beweis. Zunächst zeigen wir, dass **(2)** aus **(1)** folgt, Ψ sei also konsistent. Dann definieren wir

$$\phi(t_0, x_0, \tau) := \begin{cases} \frac{1}{\tau}\left(\Psi^{t_0+\tau,t_0}(x_0) - x_0\right) & \text{für } \tau \neq 0 \\ f(t_0, x_0) & \text{für } \tau = 0 \end{cases}.$$

Dann sind die Gleichungen (7.6) und (7.7) direkt erfüllt und es muss nur die Stetigkeit von ϕ gezeigt werden. Dazu betrachten wir

$$\begin{aligned} \lim_{\tau\to 0} \frac{1}{\tau}\left(\Psi^{t_0+\tau,t_0}(x_0) - x_0\right) &= \lim_{\tau\to 0} \frac{\Psi^{t_0+\tau,t_0}(x_0) - \Psi^{t_0,t_0}(x_0)}{\tau} \\ &= \frac{\partial}{\partial\tau}\Psi^{t_0+\tau,t_0}(x_0)\Big|_{\tau=0} \\ &= f(t_0, x_0), \end{aligned}$$

also ist ϕ stetig.

Nun zeigen wir, dass **(3)** aus **(2)** folgt, dazu sei ϕ eine Verfahrensfunktion, die die Gleichungen (7.6) und (7.7) erfüllt. Dann gilt

$$\begin{aligned} & \lim_{\tau\to 0} \frac{1}{\tau} \|\Psi^{t_0+\tau,t_0}(x_0) - \Phi^{t_0+\tau,t_0}(x_0)\| \\ = & \lim_{\tau\to 0} \left\| \frac{\Psi^{t_0+\tau,t_0}(x_0) - x_0}{\tau} - \frac{\Phi^{t_0+\tau,t_0}(x_0) - x_0}{\tau} \right\| \\ = & \|\phi(t_0, x_0, 0) - f(t_0, x_0)\| = 0 \end{aligned}$$

und dies war zu zeigen.

Schließlich wollen wir noch zeigen, dass **(1)** aus **(3)** foglt und daher sei Gleichung (7.8) erfüllt. Die Taylorentwicklung bis zum linearen Term liefert dann

$$\Phi^{t_0+\tau,t_0}(x_0) = x_0 + \tau f(t_0, x_0) + \mathcal{O}(\tau).$$

Weiter ist Ψ nach Voraussetzung für hinreichend kleines τ differenzierbar bezüglich τ. Dies ergibt

$$\Psi^{t_0+\tau,t_0}(x_0) \;=\; \Psi^{t_0,t_0}(x_0) + \tau \frac{\partial}{\partial \tau} \Psi^{t_0+\tau,t_0}(x_0)\Big|_{\tau=0} + O(\tau).$$

Nach Gleichung (7.8) sind aber die linken Seiten der letzten beiden Gleichungen für $\tau \to 0$ gleich und daher muss dies auch für die rechten Seiten für $\tau \to 0$. Durch Koeffizientenvergleich folgen dann genau die beiden Bedingungen

(1) $\Psi^{t_0,t_0}(x_0) = x_0$,

(2) $\frac{\partial}{\partial \tau} \Psi^{t_0+\tau,t_0}(x_0)\Big|_{\tau=0} = f(t_0, x_0)$.

und damit ist Ψ nach Definition konsistent. □

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

$$\max_{t_j \in \Delta} \|x_\Delta(t_j) - x(t_j)\| \;=\; \max_{t_j \in \Delta} \|x_j - x(t_j)\|,$$

der möglichst klein sein soll.

Notation 7.9. Ein Einschrittverfahren heißt ***konvergent***, falls

$$\lim_{\tau \to 0} \sup_{\Delta \text{ mit} \tau_\Delta = \tau} \max_{t_j \in \Delta} \|x_\Delta(t) - x(t)\| \;=\; 0$$

gilt. Dabei ist τ_Δ die Feinheit des Gitters Δ und das Supremum ist als Supremum über allen möglichen nicht notwendigerweise äquidistanten Gittern mit Feinheit τ zu verstehen.

Der folgende Satz zeigt nun, dass aus der Konsistenz unter einer zusätzlichen Stabilitätsannahme die Konvergenz von Einschrittverfahren folgt. Dabei müssen wir die Konsistenzbedingung allerdings verstärken und zusätzlich fordern, dass die Konsistenzbedingung gleichmäßig erfüllt ist, also für alle $x(t)$ auf der Lösungskurve.

Satz 7.14. *Gegeben sei eine diskrete Evolution Ψ, die in einer Umgebung U der* **Trajektorie**

$$\{(t, x(t)) \;:\; t \in [t_0, T]\}$$

der exakten Lösung $x(t)$ definiert ist und zusätzlich die folgenden Bedingungen erfüllt.

Stabilitätsbedingung. *Es gibt Konstanten $L_\Psi \geq 0$ und $\tau_0 > 0$ mit*

$$\|\Psi^{t+\tau,t}(x_1) - \Psi^{t+\tau,t}(x_2)\| \;\leq\; \mathrm{e}^{L_\Psi \tau} \cdot \|x_1 - x_2\|$$

für alle $(t, x_1), (t, x_2) \in U$ und alle $0 \leq \tau \leq \tau_0$.

Konsistenzbedingung. *Es gibt eine monoton wachsende Fehlerfunktion* $\mathrm{err} : [0, \tau_0] \to [0, \infty)$ *mit*

$$\lim_{\tau \to 0} \mathrm{err}(\tau) \;=\; 0$$

und mit

$$\|\Phi^{t+\tau,t}(x(t)) - \Psi^{t+\tau,t}(x(t))\| \;\leq\; \tau \cdot \mathrm{err}(\tau)$$

für alle $t \in [0, T]$.

Dann gibt es ein $\tau_1 \in [0, \tau_0]$, sodass für jedes Gitter $\Delta = \{t_0, \ldots, t_N\}$ auf $[t_0, T]$ mit Feinheit $\tau_\Delta \leq \tau_1$ die Gitterfunktion x_Δ durch die diskrete Evolution

$$x_{j+1} \;=\; x_\Delta(t_{j+1}) \;=\; \Psi^{t_{j+1},t_j}(x_\Delta(t_j)) \qquad \text{mit} \qquad x_\Delta(t_0) \;=\; x_0$$

wohldefiniert ist. Der Fehler genügt dabei der Abschätzung

$$\|x_j - x(t_j)\| \;\leq\; r(\tau_\Delta) \;:=\; \begin{cases} \mathrm{err}(\tau_\Delta) \cdot \frac{1}{L_\Psi} \cdot \left(\mathrm{e}^{L_\Psi (t_j - t_0)} - 1\right) & \text{für} \quad L_\Psi > 0 \\ \mathrm{err}(\tau_\Delta) \cdot (t_j - t_0) & \text{für} \quad L_\Psi = 0 \end{cases}$$

für alle $t_j \in \Delta$.

Der Satz besagt also, dass wir unter gewissen Voraussetzungen aus der Stabilitäts- und Konsistenzbedingung die Konvergenz folgern können:

"Stabilität + Konvergenz = Konvergenz".

Der Beweis kann zum Beispiel in Schöbel (2007) nachgelesen werden.

Abschließend wollen wir noch die Begriffe der Konsistenz- und Konvergenzordnung einführen.

Definition 7.10. Eine diskrete Evolution Ψ zu einer Differentialgleichung $x'(t) = f(t, x(t))$ mit $f : D \to \mathbb{R}^d$ hat die ***Konsistenzordnung*** $p > 0$, falls es für jede kompakte Teilmenge $K \subset D$ eine Konstante $C > 0$ mit

$$\|\Psi^{t+\tau,t}(x) - \Phi^{t+\tau,t}(x)\| \;\leq\; C \cdot \tau^{p+1}$$

für alle $(t,x) \in K$ und alle hinreichend kleinen $\tau \geq 0$.

Ein Einschrittverfahren besitzt die Konsistenzordnung $p > 0$, falls für jede rechte Seite $f : D \to \mathbb{R}^d$ mit $f \in C^\infty(D)$ die zugeordnete diskrete Evolution Ψ die Konsistenzordnung p besitzt.

Definition 7.11. Ein Einschrittverfahren besitzt die ***Konvergenzordnung*** $p > 0$, falls für jede Lösung $x : [t_0, T] \to \mathbb{R}^d$ eines Anfangswertproblemes $x'(t) = f(t, x(t))$ mit $x(t_0) = x_0$ und rechter Seite $f : D \to \mathbb{R}^d$ mit $f \in \mathcal{C}^\infty(D)$ der globale Fehler, der durch das Verfahren bestimmten Gitterfunktion x_Δ auf einem Gitter Δ mit hinreichend kleiner Feinheit τ_Δ die Abschätzung

$$\max_{t_j \in \Delta} \|x_\Delta(t_j) - x(t_j)\| \;\leq\; C' \cdot {\tau_\Delta}^p$$

erfüllt. Dabei ist C' nicht von Δ abhängig.

Lemma 7.15. *Besitzt ein Einschrittverfahren die Konsistenzordnung p und erfüllt es die Stabilitätsbedingung aus Satz 7.14, dann hat das Einschrittverfahren die Konvergenzordnung p.*

Beweis. Sei $f : D \to \mathbb{R}^d$ mit $f \in \mathcal{C}^\infty(D)$ beliebig. Da das gegebene Verfahren die Konsistenzordnung p hat, gilt für die diskrete Evolution Ψ

$$\|\Psi^{t+\tau,t}(x) - \Phi^{t+\tau,t}(x)\| \;\leq\; C \cdot \tau^{p+1}.$$

Die Funktion $\mathrm{err}(\tau) := C \cdot \tau^p$ erfüllt dann offenbar die Konsistenzbedingung aus Satz 7.14 und die Anwendung des Satzes liefert

$$\max_{t_j \in \Delta} \|x_\Delta(t_j) - x(t_j)\| \;\leq\; C' \cdot \tau_\Delta^p$$

mit

$$C' \;=\; \begin{cases} C \cdot \frac{1}{L_\Psi} \cdot \left(\mathrm{e}^{L_\Psi (T-t_0)} - 1\right) & \text{für} \quad L_\Psi > 0 \\ C \cdot (T - t_0) & \text{für} \quad L_\Psi = 0 \end{cases}.$$

Damit konnte die Bedingung aus Definition 7.11 nachgewiesn werden. □

Nach dieser theoretischen Vorarbeit können wir nun eine Aussage zum Euler-Verfahren treffen.

Satz 7.16. *Die diskrete Evolution des expliziten Euler-Verfahrens hat für stetig differenzierbare Funktionen $f : D \to \mathbb{R}^d$ die Konsistenzordnung $p = 1$.*

Einen Verallgemeinerung dieses Satzes werden wir im nächsten Abschnitt bei den Ordnungsbedingen von Runge-Kutta-Verfahren beweisen.

7.6 Runge-Kutta-Verfahren

Analog zum Euler-Verfahren wollen wir nun weitere diskrete Evolutionen bzw. Einschrittverfahren mit höheren Konsistenzordnungen herleiten. Dazu werden wir im folgenden wieder stets ein äquidistantes Gitter Δ mit fester Schrittweite τ verwenden, also

$$\Delta \;=\; \{t_0, t_0+\tau, \ldots, t_0+N\cdot\tau\}.$$

Weiter erinnern wir uns daran, dass das Euler-Verfahren durch die numerischen Integration motiviert wurde. Speziell haben wir die Integraldarstellung

$$x(t_0+\tau) \;=\; x_0 + \int_{t_0}^{t_0+\tau} f(t,x(t))\,\mathrm{d}t \tag{7.9}$$

verwendet und das Integral am linken (explizites Euler-Verfahren) bzw. am rechten Randpunkt (implizites Euler-Verfahren) durch die Rechteck-Regel approximiert. Die Idee der Runge-Kutta-Verfahren besteht nun darin Quadraturformel höherer Ordnungen zu verwenden, um damit auch höhere Konsistenzordnungen zu erhalten. Dies veranschaulichen wir am speziellen Verfahren von Runge.

Das Verfahren von Runge verwendet die Mittelpunkt zur Auswertung des Integrals in der Integraldarstellung (7.9). Somit erhalten wir

$$x(t_1) \;=\; x(t_0+\tau) \;=\; x_0+\int_{t_0}^{t_0+\tau} f(t,x(t))\,\mathrm{d}t \;\approx\; x_0+\tau\cdot f(t_0+\tfrac{\tau}{2}, x(t_0+\tfrac{\tau}{2})).$$

Das Problem besteht nun darin, dass auch $x(t_0+\frac{\tau}{2})$ nicht bekannt ist. Wir nutzen daher das explizite Euler-Verfahren für dessen Bestimmung:

$$x(t_0+\tfrac{\tau}{2}) \;=\; x(t_0)+\tfrac{\tau}{2}\cdot f(t_0,x(t_0)) \;=\; x_0+\tfrac{\tau}{2}\cdot f(t_0,x_0).$$

Dies liefert uns

$$x(t_1) \;\approx\; x_0+\tau\cdot f(t_0+\tfrac{\tau}{2}, x(t_0+\tfrac{\tau}{2})) \;\approx\; x_0+\tau\cdot f(t_0+\tfrac{\tau}{2}, x_0+\tfrac{\tau}{2}\cdot f(t_0,x_0))$$

und für unsere Gitterfunktion bedeutet dies

$$x_1 \;=\; x_\Delta(t_1) \;=\; x_0+\tau\cdot f(t_0+\tfrac{\tau}{2}, x_0+\tfrac{\tau}{2}\cdot f(t_0,x_0)).$$

Dies führt uns zur Rekursion

$$x_{j+1} \;=\; x_\Delta(t_{j+1}) \;=\; x_j+\tau\cdot f(t_j+\tfrac{\tau}{2}, x_j+\tfrac{\tau}{2}\cdot f(t_j,x_j))$$

mit $x_0 = x(t_0)$. Die diskrete Evolution des Verfahrens von Runge ist damit

$$\Psi^{t+\tau,t}(x) \;=\; x+\tau\cdot f(t+\tfrac{\tau}{2}, x+\tfrac{\tau}{2}\cdot f(t,x)).$$

Dieses Ergebniss fassen wir noch einmal zusammen.

Korollar 7.17 (Verfahren von Runge). *Gegeben sei eine Differentialgleichung*

$$x'(t) \;=\; f(t, x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

sowie ein äquidistantes Gitter Δ mit Schrittweite τ. Dann liefert uns das **Verfahren von Runge** *die Gitterfunktion*

$$x_{j+1} \;=\; x_\Delta(t_{j+1}) \;=\; \Psi^{t_j+\tau,t_j}(x_j) \;=\; x_j + \tau \cdot f(t_j + \tfrac{\tau}{2}, x_j + \tfrac{\tau}{2} \cdot f(t_j, x_j)).$$

Wir werden später sehen, dass dieses Verfahren die Konsistenzordnung $p = 2$ hat.

Algorithmisch lässt sich die diskrete Evolution

$$\Psi^{t+\tau,t}(x) \;=\; x + \tau \cdot f(t + \tfrac{\tau}{2}, x + \tfrac{\tau}{2} \cdot f(t, x)).$$

des Verfahrens von Runge auch schreiben als

$$\begin{aligned} k_1 &= f(t, x), \\ k_2 &= f(t + \tfrac{\tau}{2}, x + \tfrac{\tau}{2} \cdot k_1), \\ \Psi^{t+\tau,t}(x) &= x + \tau \cdot k_2. \end{aligned}$$

Genau dies wollen wir nun verallgemeinern zum Runge-Kutta-Verfahren.

Notation 7.12. Ein ***explizites Runge-Kutta-Verfahren*** wird gegeben durch

$$k_i \;=\; k_i(t, x, \tau) \;=\; f\left(t + c_i\tau,\; x + \tau \cdot \sum_{j=1}^{i-1} a_{ij} k_j\right)$$

für $i = 1, \ldots, s$ und

$$\Psi^{t+\tau,t}(x) \;=\; x + \tau \cdot \sum_{j=1}^{s} b_j k_j.$$

Dazu verwenden wir die Notationen

$$A \;=\; \begin{pmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ a_{31} & a_{32} & 0 & & \\ \vdots & & \ddots & \ddots & \\ a_{s1} & a_{s2} & \cdots & a_{s,s-1} & 0 \end{pmatrix} \;\in\; \mathbb{R}^{s\times s},$$

$$b \;=\; \begin{pmatrix} b_1 \\ \vdots \\ b_s \end{pmatrix} \;\in\; \mathbb{R}^s \qquad \text{und} \qquad c \;=\; \begin{pmatrix} c_1 \\ \vdots \\ c_s \end{pmatrix} \;\in\; \mathbb{R}^s.$$

Weiter heißt s die ***Stufenzahl*** des Runge-Kutta-Verfahrens und k_i die i-te ***Stufe*** des Verfahrens. In Kurzform schreiben wir ein Runge-Kutta-Verfahen (A, b, c) im ***Butcher-Schema***

$$(A,b,c) \;=\; \begin{array}{c|ccccc} c_1 & 0 & & & & \\ c_2 & a_{21} & 0 & & & \\ c_3 & a_{31} & a_{32} & 0 & & \\ \vdots & \vdots & & \ddots & \ddots & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & b_2 & \dots & b_{s-1} & b_s \end{array}.$$

Beispiel 7.12. *Das explizite Euler-Verfahren lässt sich durch das Butcher-Schema*

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

beschreiben, denn damit erhalten wir

$$\begin{aligned} k_1 &= f(t + c_1\tau,\ x) = f(t,x), \\ \Psi^{t+\tau,t} &= x + \tau b_1 k_1 = x + \tau k_1 = x + \tau \cdot f(t,x). \end{aligned}$$

Analog liefert das Verfahren von Runge das Schema

$$\begin{array}{c|cc} 0 & 0 & \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}.$$

Das klassische Runge-Kutta-Verfahren mit der Stufenzahl $s = 4$ wird gegeben durch

$$\begin{array}{c|cccc} 0 & 0 & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}.$$

Ausgeschrieben bedeutet dies

$$\begin{aligned} k_1 &= f(t,x), \\ k_2 &= f(t + \tfrac{1}{2}\tau,\ x + \tfrac{1}{2}\tau k_1), \\ k_3 &= f(t + \tfrac{1}{2}\tau,\ x + \tfrac{1}{2}\tau k_2), \\ k_4 &= f(t + \tau,\ x + \tau k_3), \\ \Psi^{t+\tau,t}(x) &= x + \tau(\tfrac{1}{6}k_1 + \tfrac{1}{3}k_2 + \tfrac{1}{3}k_3 + \tfrac{1}{6}k_4). \end{aligned}$$

Wir werden später sehen, dass das klassische Runge-Kutta-Verfahren die Konsistenzordnung $p = 4$ hat.

Beispiel 7.13. *In diesem Beispiel betrachten wir den Lotka-Volterra-Zyklus, der durch die Differentialgleichung*

$$\begin{aligned} x_1'(t) &= 2x_1(t) - 0.01x_1(t)x_2(t), \\ x_2'(t) &= -x_2(t) + 0.01x_1(t)x_2(t). \end{aligned}$$

gegeben wird, siehe auch Beispiel 7.3. Wir wollen nun das Euler-Verfahren mit dem Verfahren von Runge vergleichen.

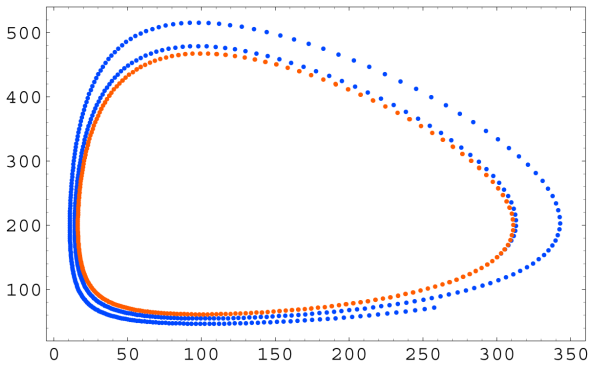


Abbildung 7.3: Diskrete Lösungen eines Lotka-Volterra-Zyklus für identische Startpopulationen.

Mit der gegeben Differentialgleichung haben wir $f : \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R}^2$ *mit*

$$f(t, x_1, x_2) = (2x_1 - 0.01x_1x_2, \quad -x_2 + 0.01x_1x_2).$$

Für beide Verfahren verwenden wir den Startwert $x(0) = (300, 150)$ *sowie die Schrittweite* $\tau = 0.02$. *Abbildung 7.3 zeigt die diskreten Lösungen zum Verfahren von Runge (orange) und zum Euler-Verfahren (blau) für die ersten* $N = 500$ *Schritte. Während das Verfahren von Runge die exakte Lösung bereits sehr gut approximiert, ist ein deutliches Divergenzverhalten beim Euler-Verfahren zu erkennen.*

Mit dem allgemeinen Runge-Kutta-Verfahren können wir nun beliebig viele Einschrittverfahren zur Lösung von Anfangswertproblemen angeben. In den folgenden Aussagen werden wir untersuchen wie die Parameter a_{ij}, b_i und c_i gewählt werden müssen, damit wir konsistente Verfahren erhalten. Zudem untersuchen wir die Invarianz gegenüber Autonomisierung sowie Konsistenzordnungen.

Lemma 7.18. *Ein Runge-Kutta-Verfahren* (A, b, c) *ist genau dann konsistent*

für alle $f : D \to \mathbb{R}^d$ mit $f \in \mathcal{C}(D)$, falls

$$\sum_{j=1}^{s} b_j \;=\; 1$$

gilt.

Beweis. Zum Beweis verwenden wir **(2)** aus Lemma 7.13, wir haben also

$$\begin{aligned} \Psi^{t+\tau,t}(x) &= x + \tau \cdot \phi(t,x,\tau), \\ \phi(t,x,0) &= f(t,x) \end{aligned}$$

zu zeigen. Dazu definieren wir

$$\phi(t,x,\tau) \;=\; \sum_{j=1}^{s} b_j k_j(t,x,\tau)$$

und erhalten damit direkt

$$\Psi^{t+\tau,t}(x) \;=\; x + \tau \cdot \sum_{j=1}^{s} b_j k_j(t,x,\tau) \;=\; x + \tau \cdot \phi(t,x,\tau).$$

Weiterhin gilt $k_j(t,x,0) = f(t,x)$ für alle $j = 1,\ldots,s$, also auch

$$\phi(t,x,0) \;=\; \sum_{j=1}^{s} b_j k_j(t,x,0) \;=\; f(t,x) \cdot \sum_{j=1}^{s} b_j.$$

Damit ist die zweite Bedingung genau dann erfüllt, wenn

$$\sum_{j=1}^{s} b_j \;=\; 1$$

gilt. □

Lemma 7.19. *Besitzt ein Runge-Kutta-Verfahren mit Stufenzahl s für alle $f : D \to \mathbb{R}^d$ mit $f \in \mathcal{C}^\infty(D)$ die Konsistenzordnung p, so gilt $p \le s$.*

Beweis. Zum Beweis verwenden wir das Beispiel

$$x'(t) \;=\; f(x(t)) \;=\; x(t) \qquad \text{mit} \qquad x(0) \;=\; 1.$$

Die exakte Lösung ist offenbar

$$x(\tau) \;=\; \Phi^{\tau,0}(1) \;=\; \mathrm{e}^{\tau} \;=\; 1 + \tau + \tfrac{1}{2!}\tau^2 + \cdots + \tfrac{1}{p!}\tau^p + \mathcal{O}(\tau^{p+1}).$$

Für die Konsistenzordnung wollen wir $\Phi^{t+\tau,t}(x)$ und $\Psi^{t+\tau,t}(x)$ vergleichen mit $t = 0$ sowie $x = 1$. Die exakte Evolution Φ kennen wir schon. Um auch $\Psi^{t+\tau,t}(x)$ zu verstehen wollen wir zeigen, dass $k_j = k_j(0,1,\tau)$ in τ ein Polynom in Π_{j-1} ist für alle $j = 1,\dots,s$. Dazu führen wir eine vollständige Induktion über j durch.

Für $j = 1$ gilt offenbar

$$k(0,1,\tau) \;=\; f(0 + c_1\tau, 1) \;=\; 1 \;\in\; \Pi_0,$$

da diese Funktion konstant in τ ist. Wir nehmen an die Induktionsvoraussetzung gelte für ein j und wir wollen nun auf $j+1$ schließen. Dazu berechnen wir

$$\begin{aligned} k_{j+1}(0,1,\tau) &= f\left(0 + c_j\tau,\; 1 + \tau \cdot \sum_{l=1}^{j} a_{jl} k_l(0,1,\tau)\right) \\ &= 1 + \tau \cdot \sum_{l=1}^{j} a_{jl} k_l(0,1,\tau). \end{aligned}$$

Nun gilt nach Induktionsvoraussetzung $k_l(0,1,\tau) \in \Pi_{j-1}$ für $l = 1,\dots,j$ und somit ergibt sich

$$k_{j+1}(0,1,\tau) \;\in\; \Pi_j.$$

Damit haben wir gezeigt, dass $\Psi^{\tau,0}(1) \in \Pi(s)$ gilt. Damit können höchstens die ersten s Potenzen in der Reihenentwicklung von $\Phi^{\tau,0}(1)$ durch $\Psi^{\tau,0}(1)$ ausgeglichen werden und wir können maximal erreichen

$$\|\Psi^{\tau,0}(1) - \Phi^{\tau,0}(1)\| \;\le\; C \cdot \tau^{s+1}.$$

Folglich kann die Konsistenzordnung höchstens s sein. □

Wie wir in Lemma 7.3 bereits gezeigt haben, lässt sich jedes Anfangswertproblem der Form

$$x'(t) \;=\; f(t, x(t)) \qquad \text{mit} \qquad x(t_0) \;=\; x_0$$

in ein äquivalentes autonomes Anfangswertproblem transformieren, nämlich in

$$\hat{x}'(t) \;=\; \begin{pmatrix} 1 \\ f(\hat{x}(t)) \end{pmatrix} \qquad \text{mit} \qquad \hat{x}(t_0) \;=\; \hat{x}_0 \;=\; \begin{pmatrix} t_0 \\ x_0 \end{pmatrix}.$$

Nun sei Φ die exakte Evolution zur Differentialgleichung $x'(t) = f(t, x(t))$ und $\hat{\Phi}$ die zu $\hat{x}'(t) = (1, f(y(t)))$. Dann kann die Äquivalenz der beiden Anfangswertprobleme auch geschrieben werden als

$$\begin{pmatrix} t+\tau \\ \Phi^{t+\tau,t}(x) \end{pmatrix} \;=\; \hat{\Phi}^{t+\tau,t} \begin{pmatrix} t \\ x \end{pmatrix}.$$

Diese Eigenschaft soll nun auch auf die diskreten Evolutionen Ψ und $\hat{\Psi}$ übertragen werden, also

$$\begin{pmatrix} t+\tau \\ \Psi^{t+\tau,t}(x) \end{pmatrix} = \hat{\Psi}^{t+\tau,t} \begin{pmatrix} t \\ x \end{pmatrix}. \tag{7.10}$$

Dies bedeutet also, dass das Einschrittverfahren gegeben durch Ψ die gleiche diskrete Lösung liefert wie das durch $\hat{\Psi}$ gegebene Einschrittverfahren. Wir sagen dann auch, dass das Einschrittverfahren ***invariant gegenüber Autonomisierung*** ist.

Diese Eigenschaft soll speziell auch für Runge-Kutta-Verfahren gelten.

Lemma 7.20. *Ein explizites Runge-Kutta-Verfahren ist genau dann invariant gegen Autonomisierung, wenn es konsistent ist und es*

$$c_i = \sum_{j=1}^{i-1} a_{ij} \qquad \text{für} \qquad j = 1, \ldots, s$$

erfüllt.

Beweis. Sei

$$\hat{x}'(t) = \hat{f}(\hat{x}(t)) = \begin{pmatrix} 1 \\ f(t, x(t)) \end{pmatrix}$$

die autonomisierte Differentialgleichung und sei $\hat{\Psi}$ die zugehörige diskrete Evolution mit $\hat{x}(t) = (t, x(t))$. Weiter bezeichnen wir mit

$$\hat{K}_i = \begin{pmatrix} \hat{l}_i \\ \hat{k}_i \end{pmatrix} \qquad \text{für} \qquad i = 1, \ldots, s$$

die s Stufen von $\hat{\Psi}$. Dann gilt

$$\begin{aligned} \hat{K}_i &= \hat{f}\left(t + c_i\tau,\ \hat{x} + \tau \cdot \sum_{j=1}^{i-1} a_{ij}\hat{K}_j\right) = \hat{f}\left(\hat{x} + \tau \cdot \sum_{j=1}^{i-1} a_{ij}\hat{K}_j\right) \\ &= \hat{f}\left(\begin{pmatrix} t \\ x \end{pmatrix} + \tau \cdot \sum_{j=1}^{i-1} a_{ij} \begin{pmatrix} \hat{l}_j \\ \hat{k}_j \end{pmatrix}\right) \\ &= \begin{pmatrix} 1 \\ f\left(t + \tau \cdot \sum_{j=1}^{i-1} a_{ij}\hat{l}_j,\ x + \tau \cdot \sum_{j=1}^{i-1} \hat{k}_j\right) \end{pmatrix} \end{aligned}$$

für $i = 1, \ldots, s$. Dies bedeutet

$$\hat{l}_i = 1 \qquad \text{und} \qquad \hat{k}_i = f\left(t + \tau \cdot \sum_{j=1}^{i-1} a_{ij},\ x + \tau \cdot \sum_{j=1}^{i-1} \hat{k}_j\right)$$

für $i = 1, \ldots, s$. Für das Runge-Kutta-Verfahren gilt für die diskrete Evolution weiterhin

$$\hat{\Psi}^{t+\tau,t}\begin{pmatrix} t \\ x \end{pmatrix} = \begin{pmatrix} t \\ x \end{pmatrix} + \tau \cdot \sum_{j=1}^{s} b_j \begin{pmatrix} \hat{t}_j \\ \hat{k}_j \end{pmatrix} = \begin{pmatrix} t + \tau \cdot \sum_{j=1}^{s} b_j \\ x + \tau \cdot \sum_{j=1}^{s} b_j \hat{k}_j \end{pmatrix}.$$

Damit haben wir bis hierhin die gegen Autonomisierung invariante diskrete Evolution $\hat{\Psi}$ durch die gegebenen Parameter a_{ij} und b_i der eigentlichen Evolution Ψ ausgedrückt. Nun können wir Gleichung (7.10) anwenden und einen Koeffizientenvergleich durchführen. Das Verfahren Ψ ist also genau dann invariant gegenüber Autonomisierung, wenn

$$\begin{pmatrix} t + \tau \\ \Psi^{t+\tau,t}(x) \end{pmatrix} = \hat{\Psi}^{t+\tau,t}\begin{pmatrix} t \\ x \end{pmatrix}$$

gilt. Dies bedeutet aber gerade

$$\begin{aligned} & t + \tau = t + \tau \cdot \sum_{j=1}^{s} b_j \quad \text{und} \quad \Psi^{t+\tau,t}(x) = x + \tau \cdot \sum_{j=1}^{s} b_j \hat{k}_j \\ \Leftrightarrow \quad & \sum_{j=1}^{s} b_j = 1 \quad \text{und} \quad x + \tau \cdot \sum_{j=1}^{s} b_j k_j = x + \tau \cdot \sum_{j=1}^{s} b_j \hat{k}_j \\ \Leftrightarrow \quad & \Psi \text{ ist konsistent} \quad \text{und} \quad k_i = \hat{k}_i \text{ für alle } i = 1, \ldots, s. \end{aligned}$$

Letzteres ist genau dann der Fall, wenn

$$f\left(t + c_i\tau,\ x + \tau \cdot \sum_{j=1}^{i-1} a_{ij} k_j\right) = f\left(t + \tau \cdot \sum_{j=1}^{i-1} a_{ij},\ x + \tau \cdot \sum_{j=1}^{i-1} a_{ij} \hat{k}_j\right)$$

und damit wenn $c_i = \sum_{j=1}^{i-1} a_{ij}$ für $i = 1, \ldots, s$. □

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 c ja von der Matrix A abhängig ist.

Folgende Bedingungen an die Koeffizienten eines Runge-Kutta-Verfahrens haben wir bisher erarbeitet:

(1) Das Verfahren ist genau dann konsistent, wenn

$$\sum_{i=1}^{s} b_i \;=\; 1$$

gilt.

(2) Das Verfahren ist genau dann invariant gegen Autonomisierung, wenn es konsistent ist und

$$c_i \;=\; \sum_{j=1}^{2} a_{ij}$$

gilt für alle $i = 1, \ldots, s$.

Im folgenden wollen wir weitere Bedingungen an die Parameter (A, b) von gegen Autonomisierung invariante Runge-Kutta-Verfahren stellen, sodass wir eine höhere Konsistenzordnung erhalten. Diese Bedingungen werden als ***Ordnungsbedingungen*** bezeichnet.

Satz 7.21 (Ordnungsbedingungen). *Ein gegen Autonomisierung invariantes Runge-Kutta-Verfahren besitzt für jede Differentialgleichung*

$$x'(t) \;=\; f(x(t))$$

mit $f : D \to \mathbb{R}^d$ und $f \in \mathcal{C}^p(D)$ die folgenden Konsistenzordnung genau dann, falls die folgenden Bedingungen erfüllt sind.

(1) *Konsistenzordnung $p = 1$ genau dann, wenn*

$$\sum_{i=1}^{s} b_i \;=\; 1 \tag{7.11}$$

gilt.

(2) *Konsistenzordnung $p = 2$ genau dann, wenn zusätzlich*

$$\sum_{i=1}^{s} b_i c_i \;=\; \frac{1}{2} \tag{7.12}$$

gilt.

(3) *Konsistenzordnung $p = 3$ genau dann, wenn zusätzlich*

$$\sum_{i=1}^{s} b_i c_i^2 \;=\; \frac{1}{3}, \tag{7.13}$$

$$\sum_{i,j=1}^{s} b_i a_{ij} c_j \;=\; \frac{1}{6} \tag{7.14}$$

gilt.

(4) *Konsistenzordnung $p = 4$ genau dann, wenn zusätzlich*

$$\begin{aligned}
\sum_{i=1}^{s} b_i c_i^3 &= \frac{1}{4}, \\
\sum_{i,j=1}^{s} b_i c_i a_{ij} c_j &= \frac{1}{8}, \\
\sum_{i,j=1}^{s} b_i a_{ij} c_j^2 &= \frac{1}{12}, \\
\sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k &= \frac{1}{24}
\end{aligned}$$

gilt.

Beweis. Sei $\Phi^\tau(x)$ die exakte und $\Psi^\tau(x)$ die disktre Evolution, beide seien invariant gegen Autonomisierung. Wir leiten die Ordnungsbedingungen zur Konsistenz bis zur Ordnung $p = 3$ in drei Schritten her.

Schritt 1 (Taylorentwicklung der exakten Evolution)

Wir führen eine Taylorentwicklung von

$$g(\tau) \;:=\; \Phi^\tau(x) \tag{7.15}$$

um $\tau = 0$ durch. Zunächst gilt

$$\Phi^0(x) \;=\; x \qquad \text{und} \qquad \frac{\mathrm{d}}{\mathrm{d}\tau}\Phi^\tau(x) \;=\; f(\Phi^\tau(x)),$$

damit erhalten wir

$$\begin{aligned}
& \Phi^\tau(x) \\
= \;& \Phi^0(x) + \tau \left(\frac{\mathrm{d}}{\mathrm{d}\tau}\Phi^\tau(x)\Big|_{\tau=0} \right) + \frac{\tau^2}{2} \left(\frac{\mathrm{d}^2}{\mathrm{d}\tau^2}\Phi^\tau(x)\Big|_{\tau=0} \right) \\
& + \frac{\tau^3}{6} \left(\frac{\mathrm{d}^3}{\mathrm{d}\tau^3}\Phi^\tau(x)\Big|_{\tau=0} \right) + \mathcal{O}(\tau^4) \\
= \;& x + \tau f(x) + \frac{\tau^2}{2} f'(x) f(x) \\
& + \frac{\tau^3}{6} \big(f''(x) f(x) f(x) + f'(x) f'(x) f(x) \big) + \mathcal{O}(\tau^4) \\
= \;& x + \tau f(x) + \frac{\tau^2}{2} f'(x) f(x) \\
& + \frac{\tau^3}{6} f''(x) f(x) f(x) + \frac{\tau^3}{6} f'(x) f'(x) f(x) + \mathcal{O}(\tau^4),
\end{aligned} \tag{7.16}$$

denn mit den unter (7.15) festgehaltenen Bemerkungen gilt

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\tau}\Phi^\tau(x)\Big|_{\tau=0} &= f(\Phi^\tau(x))\Big|_{\tau=0} = f(\Phi^0(x)) = f(x),\\
\frac{\mathrm{d}^2}{\mathrm{d}\tau^2}\Phi^\tau(x)\Big|_{\tau=0} &= \frac{\mathrm{d}}{\mathrm{d}\tau}f(\Phi^\tau(x))\Big|_{\tau=0}\\
&= f'(\Phi^\tau(x))\cdot\frac{\mathrm{d}}{\mathrm{d}\tau}\Phi^\tau(x)\Big|_{\tau=0}\\
&= f'(\Phi^\tau(x))\cdot f(\Phi^\tau(x))\Big|_{\tau=0}\\
&= f'(\Phi^0(x))\cdot f(\Phi^0(x)) = f'(x)f(x)\\
\frac{\mathrm{d}^3}{\mathrm{d}\tau^3}\Phi^\tau(x)\Big|_{\tau=0} &= \frac{\mathrm{d}^2}{\mathrm{d}\tau^2}f(\Phi^\tau(x))\Big|_{\tau=0}\\
&= \frac{\mathrm{d}}{\mathrm{d}\tau}\left(\frac{\mathrm{d}}{\mathrm{d}\tau}f(\Phi^\tau(x))\right)\Big|_{\tau=0}\\
&= \frac{\mathrm{d}}{\mathrm{d}\tau}\left(f'(\Phi^\tau(x))\cdot f(\Phi^\tau(x))\right)\Big|_{\tau=0}\\
&= f''(\Phi^\tau(x))\cdot\frac{\mathrm{d}}{\mathrm{d}\tau}\Phi^\tau(x)\cdot f(\Phi^\tau(x))\\
&\quad + f'(\Phi^\tau(x))\cdot\frac{\mathrm{d}}{\mathrm{d}\tau}f(\Phi^\tau(x))\Big|_{\tau=0}\\
&= f''(\Phi^\tau(x))\cdot f(\Phi^\tau(x))\cdot f(\Phi^\tau(x))\\
&\quad + f'(\Phi^\tau(x))\cdot f'(\Phi^\tau(x))\cdot\frac{\mathrm{d}}{\mathrm{d}\tau}\Phi^\tau(x)\Big|_{\tau=0}\\
&= f''(\Phi^0(x))\cdot f(\Phi^0(x))\cdot f(\Phi^0(x))\\
&\quad + f'(\Phi^0(x))\cdot f'(\Phi^0(x))\cdot f(\Phi^0(x))\\
&= f''(x)\cdot f(x)\cdot f(x) + f'(x)\cdot f'(x)\cdot f(x).
\end{aligned}$$

Schritt 2 (Taylorentwicklung der diskreten Evolution)

Zur Berechnung der Taylorentwicklung der diskreten Evolution betrachten wir zunächst

$$h(\tau) := k_i = f\left(x+\tau\sum_{j=1}^{s}a_{ij}k_j\right) \tag{7.17}$$

und führen eine Taylorentwicklung von $h(\tau)$ um $\tau = 0$ bis zur ersten Ordnung durch, also

$$k_i = f(x) + \mathcal{O}(\tau).$$

Dies setzten wir nun wieder in (7.17) ein:

$$\begin{aligned} k_i &= f\left(x + \tau \sum_{j=1}^{s} a_{ij} k_j\right) = f\left(x + \tau \sum_{j=1}^{s} a_{ij}(f(x) + \mathcal{O}(\tau))\right) \\ &= f\left(x + \tau c_i f(x) + \mathcal{O}(\tau^2)\right) \end{aligned}$$

Dabei haben wir verwendet, dass $\Psi^\tau(x)$ autonomisierungsinvariant, dass also

$$c_i = \sum_{j=1}^{s} a_{ij}$$

gilt. Nun führen wir abermals eine Taylorentwicklung um $\tau = 0$ durch:

$$\begin{aligned} k_i &= f\left(x + \tau c_i f(x) + \mathcal{O}(\tau^2)\right) \\ &= f(x) + \tau \left(\frac{\mathrm{d}}{\mathrm{d}\tau} f(x + \tau c_i f(x) + \mathcal{O}(\tau^2))\Big|_{\tau=0}\right) + \mathcal{O}(\tau^2) \\ &= f(x) + \tau \left(f'(x + \tau c_i f(x) + \mathcal{O}(\tau^2)) \cdot (c_i f(x) + \mathcal{O}(\tau))\Big|_{\tau=0}\right) + \mathcal{O}(\tau^2) \\ &= f(x) + \tau \left(f'(x) \cdot c_i f(x)\right) + \mathcal{O}(\tau^2) \\ &= f(x) + \tau c_i f'(x) f(x) + \mathcal{O}(\tau^2). \end{aligned}$$

Dieses Ergebnis setzen wir noch einmal in (7.17) ein und führen wieder eine Taylorentwicklung um $\tau = 0$ durch. Mit

$$m_i(\tau) = f\left(x + \tau c_i f(x) + \tau^2 \sum_{j=1}^{s} a_{ij} c_j f'(x) f(x) + \mathcal{O}(\tau^3)\right)$$

erhalten wir die folgende etwas abgekürzte Rechnung:

$$\begin{aligned} k_i &= f\left(x + \tau \sum_{j=1}^{s} a_{ij} \left((f(x) + \tau c_j f'(x) f(x) + \mathcal{O}(\tau^2)\right)\right) \\ &= f\left(x + \tau c_i f(x) + \tau^2 \sum_{j=1}^{s} a_{ij} c_j f'(x) f(x) + \mathcal{O}(\tau^3)\right) = m_i(\tau) \\ &= m_i(0) + \tau \left(\frac{\mathrm{d}}{\mathrm{d}\tau} m_i(\tau)\Big|_{\tau=0}\right) + \frac{\tau^2}{2} \left(\frac{\mathrm{d}^2}{\mathrm{d}\tau^2} m_i(\tau)\Big|_{\tau=0}\right) + \mathcal{O}(\tau^3) \\ &= f(x) + \tau c_i f'(x) f(x) \\ &\quad + \frac{\tau^2}{2} \left(c_i^2 f''(x) f(x) f(x) + 2 \sum_{j=1}^{s} a_{ij} c_i f'(x) f'(x) f(x)\right) + \mathcal{O}(\tau^3) \\ &=: n_i(\tau) \end{aligned}$$

Dieses Ergebnis setzen wir nun in die diskrete Evolution ein und erhalten

$$\begin{aligned}\Psi^\tau(x) &= x + \tau \sum_{i=1}^{s} b_j k_j = x + \tau \sum_{i=1}^{s} b_j n_i(\tau) \\ &= x + \tau \sum_{i=1}^{s} b_i f(x) + \tau^2 \sum_{i=1}^{s} b_i c_i f'(x) f(x) \\ &\quad + \frac{\tau^3}{2} \sum_{i=1}^{s} b_i c_i^2 f''(x) f(x) f(x) \\ &\quad + \frac{\tau^3}{2} \sum_{i=1}^{s} b_i 2 \sum_{j=1}^{s} a_{ij} c_i f'(x) f'(x) f(x) + \mathcal{O}(\tau^4) \qquad (7.18)\end{aligned}$$

Schritt 3 (Koeffizientenvergleich)

Nun führen wir einen Koeffizientenvergleich der exakten und der diskreten Evolution durch, siehe (7.16) und (7.18).

Die diskrete Evolution $\Psi^\tau(x)$ besitzt damit die Konsistensordnung $p = 1$, wenn

$$\sum_{i=1}^{s} b_i = 1$$

gilt. $\Psi^\tau(x)$ besitzt die Konsistensordnung $p = 2$, wenn zusätzlich

$$\sum_{i=1}^{s} b_i c_i = \frac{1}{2}$$

gilt. Und schließlich hat $\Psi^\tau(x)$ die Konsistensordnung $p = 3$, wenn zusätzlich

$$\frac{1}{2} \sum_{i=1}^{s} b_i c_i^2 = \frac{1}{6} \qquad \Leftrightarrow \qquad \sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}$$

sowie auch noch

$$\frac{1}{2} \sum_{i=1}^{s} b_i 2 \sum_{j=1}^{s} a_{ij} c_i = \sum_{i,j=1}^{s} b_i a_{ij} c_i = \frac{1}{6}$$

gilt. □

Damit lassen sich die Konsistenzordnung des Euler-Verfahrens, des Verfahrens von Runge und des klassischen Runge-Kutta-Verfahrens sofort nachweisen. Wir betrachten noch einige weitere Beispiele.

Beispiel 7.14 (Konsistenzordnung 1). *Um alle gegen Autonomisierung invarianten Runge-Kutta-Verfahren der Konsistenzordnung $p = 1$ zu bestimmen, nutzen wir das Butcher-Schema*

$$\begin{array}{c|c} c_1 & 0 \\ \hline & b_1 \end{array}.$$

Damit erhalten wir sofort $c_1 = 0$ und $b_1 = 1$. Somit ist das explizite Euler-Verfahren das einzige gegen Autonomisierung invarianten Runge-Kutta-Verfahren der Konsistenzordnung $p = 1$.

Beispiel 7.15 (Konsistenzordnung 2). *Um alle gegen Autonomisierung invarianten Runge-Kutta-Verfahren der Konsistenzordnung $p = 2$ zu bestimmen, nutzen wir das Butcher-Schema*

$$\begin{array}{c|cc} c_1 & 0 & \\ c_2 & a_{21} & 0 \\ \hline & b_1 & b_2 \end{array}.$$

Wir erhalten sofort $c_1 = 0$ und $c_2 = a_{21}$. Als Variablen verbleiben also nur noch a_{21}, b_1 und b_2, wobei wir zur Konsistenzordnung $p = 2$ aber weitere Bedingungen haben, nämlich

$$\begin{aligned} b_1 + b_2 &= 1, \\ b_1 c_1 + b_2 c_2 &= b_2 c_2 = \frac{1}{2}. \end{aligned}$$

Aus dem Gleichungssystem ergibt sich für $b_2 \neq 0$

$$\begin{aligned} b_1 &= 1 - b_2, \\ c_2 &= \frac{1}{2b_2}. \end{aligned}$$

Somit sind alle gegen Autonomisierung invarianten Runge-Kutta-Verfahren der Konsistenzordnung $p = 2$ von der Form

$$\begin{array}{c|cc} 0 & 0 & \\ \frac{1}{2\lambda} & \frac{1}{2\lambda} & 0 \\ \hline & 1-\lambda & \lambda \end{array}$$

mit $\lambda \in \mathbb{R} \setminus \{0\}$. Für $\lambda = 1$ erhalten wir das Verfahren von Runge.

Es sei bemerkt, dass die Anzahl der Ordnungsbedingen für höhere Ordnungen sehr schnell wächst. Für $p = 10$ erhalten wir bereits 1 205 Bedinungen und für $p = 20$ sogar 20 247 374.

Zur Konvergenz benötigen wir noch folgendes Satz, dessen Beweis Lube (2005b) entnommen werden kann:

Satz 7.22. *Sei die Funktion f der autonomen Gleichung $x' = f(x)$ global Lipschitz- stetig, d.h.*

$$\|f(x) - f(y)\| \leq L\|x - y\|, \qquad x, y \in D.$$

Dann genügt die diskrete Evolution eines gegen Autonomisierung invarianten RK-Verfahrens der Stabilitätsbedingung aus Satz 7.14.

Abschließend untersuchen wir noch ein weiteres praktisches Beispiel.

Beispiel 7.16 (Satellitenbahn). *Wir betrachten die Bahn eines Satelliten, der sich im Gravitationsfeld zwischen Erde und Mond bewegt. Zur Vereinfachung machen wir die folgenden Annahmen:*

(1) *Der Abstand L zwischen Erde und Mond sei mit $L = 384\,000\,km$ konstant.*

(2) *Die Masse des Satelliten ist gegenüber der Masse von Erde bzw. Mond vernachlässigbar.*

(3) *Erde, Mond und Satellit bewegen sich in einer Ebene.*

Wir wählen ein mitrotierendes, baryzentrisches Koordinatensystem mit Längeneinheit L, in dem sich die Erde im Punkt $(-\mu, 0)$ und Mond im Punkt $(\hat{\mu}, 0)$ zum Zeitpunkt $t = 0$ befinden. Dabei ist $\mu = 0,012\,277\,471$ das Verhältnis der Mondmasse zur Masse des Gesamtsystems und $\hat{\mu} = 1 - \mu$. Die Bahn des Satelliten $x(t) = (x_1(t), x_2(t)) \in \mathbb{R}^2$ wird nun durch die Differentialgleichungen

$$\begin{aligned} x_1''(t) &= x_1(t) + 2x_2'(t) - \hat{\mu} \cdot \frac{x_1(t) + \mu}{((x_1(t) + \mu)^2 + (x_2(t))^2)^{3/2}} \\ &\qquad - \mu \cdot \frac{x_1(t) - \hat{\mu}}{((x_1(t) + \hat{\mu})^2 + (x_2(t))^2)^{3/2}}, \\ x_2''(t) &= x_2(t) - 2x_1'(t) - \hat{\mu} \cdot \frac{x_2(t)}{((x_1(t) + \mu)^2 + (x_2(t))^2)^{3/2}} \\ &\qquad - \mu \cdot \frac{x_2(t)}{((x_1(t) + \hat{\mu})^2 + (x_2(t))^2)^{3/2}}, \end{aligned}$$

beschrieben. Dabei stehen die Terme (x_1, x_2) und $(2x_2'(t), -2x_1'(t))$ für die Zentrifugal- bzw. Corioliskraft. Als Zeiteinheit haben wir das $1/2\pi$-fache der Umlaufzeit des Mondes um die Erde um ihren gemeinsamen Schwerpunkt gewählt, also ungefährt einen Monat. Die Anfangswerte seien

$$x_1(0) = 0,994, \quad x_1'(0) = 0, \quad x_2(0) = 0, \quad x_2'(0) = -2,001\,585\,106$$

und wir betrachten das Zeitintervall $[0, 18]$.

Um dieses System numerisch zu Lösen, müssen wir es zunächst nach Lemma 7.2 in ein äquivalentes Differentialgleichungssystem erster Ordnung transformiert. Abbildung 7.4 zeigt die numerische Lösung des gegebenen Anfangswertproblems mit dem klassischen Runge-Kutta-Verfahren zur Schrittweite $\tau = 0,001$ *und* $N = 18\,000$ *Schritte.*

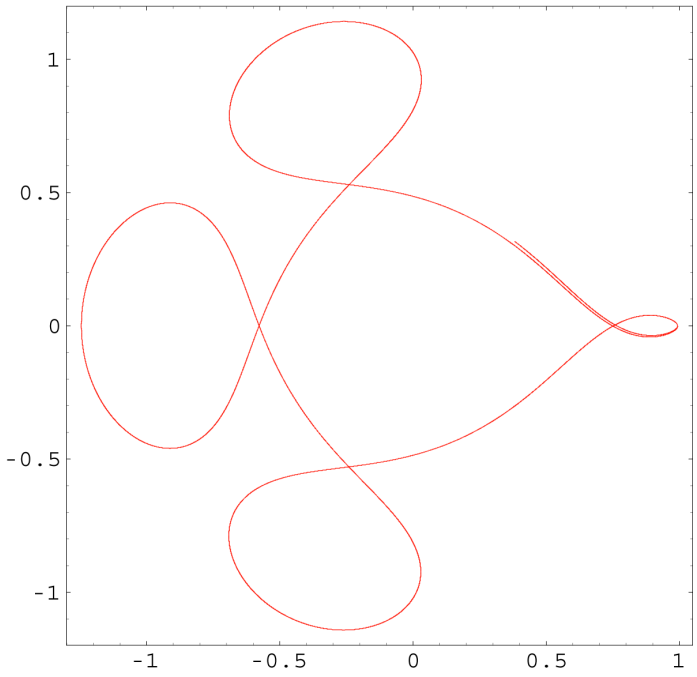


Abbildung 7.4: Diskrete Lösung der Satellitenbahn.

Es sei bemerkt, dass bei diesem Beispiel Verfahren mit einer Konsistenzordnung von $p < 4$ *bereits sehr viel früher stark von der exakten Lösung abweichen.*

7.7 Adaptive Schrittweitensteuerung

Ziel dieses Abschnitts ist die vollautomatische Erzeugung problemangepasster Gitter Δ zur Lösung von Anfangswertproblemen. Dabei sollte einerseits der globale Fehler $\sup_{t\in\Delta} \|x_\Delta(t) - x(t)\|$ möglichst klein sein und andererseits sollte Δ aus Effizienzgründen möglichst wenig Gitterpunkte enthalten. Schließlich sollte der Aufwand zur Erzeugung des Gitters im Verhältnis zum Aufwand zur Lösung des Anfangswertproblems auf einem vorgegebenen Gitter nicht übermäßig groß sein.

Angenommen, wir haben bereits eine numerische Lösung x_Δ bis zu einem

Zeitpunkt t_j berechnet und möchten den nächsten Gitterpunkt t_{j+1} geeignet wählen. Den ***globalen Diskretisierungsfehler***

$$\epsilon_\Delta(t_{j+1}) \;:=\; x_\Delta(t_{j+1}) - x(t_{j+1})$$

können wir folgendermaßen zerlegen:

$$\begin{aligned} & \epsilon_\Delta(t_{j+1}) \\ = \; & \underbrace{\Psi^{t_{j+1},t_j}(x_\Delta(t_j)) - \Phi^{t_{j+1},t_j}(x_\Delta(t_j))}_{=:\epsilon_{j+1}} + \underbrace{\Phi^{t_{j+1},t_j}(x_\Delta(t_j)) - \Phi^{t_{j+1},t_j}(x(t_j))}_{=:p_{j+1}}. \end{aligned}$$

Er zerfällt also in zwei Anteile, den ***lokalen Diskretisierungsfehler*** (Konsistenzfehler) ϵ_{j+1} und den ***Propagationsfehler*** p_{j+1}. Den letzten Anteil könnten wir nur dadurch reduzieren, indem wir die gesamte Rechnung bis zum Zeitpunkt t_j neu aufrollen. Deshalb beschränkt man sich in der Regel darauf, den lokalen Diskretisierungsfehler zu reduzieren, indem man zum Beispiel verlangt, dass in jedem Schritt für einen vorgegebenen Toleranzwert TOL die Ungleichung

$$\|\epsilon_{j+1}\| \;\leq\; \mathrm{TOL} \tag{7.19}$$

gilt.

Wie wir gesehen hatten, lässt sich der globale Diskretisierungsfehler im Prinzip durch den lokalen Diskretisierungsfehler beschränken, allerdings ist die Fehlerkonstante im allgemeinen nicht explizit bekannt. Im allgemeinen wird man aber selbst den lokalen Diskretisierungsfehler ϵ_{j+1} nicht exakt kennen, sondern wird auf eine Schätzung $[\epsilon_{j+1}] \approx \epsilon_{j+1}$ angewiesen sein. Praktische Methoden zur Fehlerschätzung werden wir später kennenlernen. Die implementierbare Ersatzforderung lautet dann

$$\|[\epsilon_{j+1}]\| \;\leq\; \mathrm{TOL}. \tag{7.20}$$

Wir unterscheiden nun zwei Fälle:

(1) Die Ungleichung (7.20) ist verletzt. Dann verwerfen wir die in diesem Schritt berechnete Näherung $\Psi^{t_j+\tau_j,t_j} x_\Delta(t_j)$ und berechnen eine ***optimierte Schrittweite*** τ_j^*, mit der wir die Näherung $\Psi^{t_j+\tau_j^*,t_j} x_\Delta(t_j)$ berechnen. τ_j^* sollte so gewählt werden, dass die gegebene Fehlerschranke weder deutlich unterschritten wird (Effizienz) noch überschritten wird (Verlässlichkeit):

$$\|[\epsilon_{j+1}^*]\| \;\approx\; \mathrm{TOL}. \tag{7.21}$$

(2) Die Ungleichung (7.20) ist erfüllt. In diesem Fall wird die in diesem Schritt berechnete Näherung $\Psi^{t_j+\tau_j,t_j} x_\Delta(t_j)$ akzeptiert.

Es verbleibt die Berechnung einer optimalen Schrittweite τ_j^* zu diskutieren. Mit diesem wollen wir zum einen im ersten Fall unsere berechnete Lösung verbessern. Wir können sie jedoch zusätzlich als Schrittweite für den darauffolgenden Schritt verwenden. Dazu nehmen wir an, dass der lokale Fehlerschätzer sich im Limes $\tau \to 0$ wie

$$\|[\epsilon_{j+1}]\| \;\approx\; c(t_j)\tau_j^{p+1} + O(\tau_j^{p+2}) \;\approx\; c(t_j)\tau_j^{p+1}$$

verhält, wobei $c(t_{j+1})$ eine (unbekannte) Fehlerkonstante ist. Ein derartiges Verhalten lässt sich in der Tat unter vernünftigen Voraussetzungen nachweisen. Entsprechend gilt

$$\mathrm{TOL} \;\approx\; \|[\epsilon_{j+1}^*]\| \;\approx\; c(t_j)(\tau_j^*)^{p+1}.$$

Durch Division kürzt sich der unbekannte Faktor $c(t_j)$ heraus:

$$\frac{\mathrm{TOL}}{\|[\epsilon_{j+1}]\|} \;\approx\; \frac{(\tau_j^*)^{p+1}}{\tau_j^{p+1}}.$$

Wir lösen dies nach τ_j^* auf und fügen zur Sicherheit einen Faktor $\rho < 1$ ein:

$$\tau_j^* \;=\; \rho \sqrt[p+1]{\frac{\mathrm{TOL}}{\|[\epsilon_{j+1}]\|}}\;\tau_j. \tag{7.22}$$

Da $\|[\epsilon_{j+1}]\|$ verschwinden kann, ist es notwendig, eine sogenannte ***Hochschaltbeschränkung*** einzuführen. Wir verlangen, dass $|\tau_j^*| \le q\tau_j$ mit einem vorgegebenen Faktor $q > 1$ und/oder $|\tau_j^*| \le \tau_{\max}$ mit einer vorgegebenen maximalen Schrittweite $\tau_{\max} > 0$. Außerdem müssen wir sicherstellen, dass wir nicht über das Ziel hinausschießen, dass also bei einem Vorschlag τ_{j+1}^* für die Schrittweite im $(j+1)$-ten Schritt $t_{j+1} + \tau_{j+1}^* \le T$ gilt.

Wir erhalten somit den Grundalgorithmus 7.1 mit adaptiver Schrittweitensteuerung.

Noch offen ist die Frage, wie ein effizienter und zuverlässiger Fehlerschätzer implementiert werden kann. Eine Variante wäre, eine diskrete Evolution mit unterschiedlichen Schrittweiten zu verwenden. Wählt man z.B. einmal die Schrittweite τ und einmal $\frac{\tau}{2}$, so könnte man sich von letzterer Rechnung eine höhere Genauigkeit erwarten. Die Differenz der berechneten Näherungen

$$\Psi^{t+\tau,t}(x) \qquad \text{und} \qquad \Psi^{t+\,\tau,\,t+\frac{\tau}{2}}\left(\Psi^{t+\,\frac{\tau}{2},\,t}(x)\right)$$

Initialisierung: Diskrete Evolution Ψ der Ordnung p, Fehlerschätzer, Toleranz TOL > 0, Startschrittweite $0 < \tau_0 \leq T - t_0$, Hochschaltfaktor $q > 1$, Sicherheitsfaktor $\rho \in (0, 1)$, maximale Schrittweite τ_{max} $j := 0$;

$\Delta := \{t_0\}$;
$x_\Delta(t_0) := x_0$;
while $(t_j < T)$ **do**
    $t := t_j + \tau_j$;
    $x := \Psi^{t,t_j} x_\Delta(t_j)$;
    berechne Fehlerschätzer $\|[\epsilon_{j+1}]\|$;
    $\tau := \min(q\tau_j, \tau_{\max}, \rho\tau_j \sqrt[p+1]{\frac{\text{TOL}}{\|[\epsilon_{j+1}]\|}})$;
    if $(\|[\epsilon_{j+1}]\| > \text{TOL})$ ***// Schritt wird verworfen***
        $\tau_j := \tau$;
    else ***// Schritt wird akzeptiert***
        $t_{j+1} := t$;
        $\Delta := \Delta \cup \{t_{j+1}\}$;
        $x_\Delta(t_{j+1}) := x$;
        $\tau_{j+1} := \min(\tau, T - t_{j+1})$;
        $j := j + 1$;
    end
end

Algorithmus 7.1: Adaptiver Grundalgorithmus zur Lösung von $x'(t) = f(t, x), x(t_0) = x_0$ auf dem Intervall $[t_0, T]$.

kann dann als Fehlerschätzer verwendet werden.

Gebräuchlicher ist die Idee, zwei verschiedene Verfahren unterschiedlicher Ordnung mit jeweils der gleichen Schrittweite zu verwenden. Seien dazu zwei diskrete Evolutionen Ψ und $\hat{\Psi}$ mit den Diskretisierungsfehlern

$$\begin{aligned} \epsilon &= \Psi^{t+\tau,t}(x) - \Phi^{t+\tau,t}(x), \\ \hat{\epsilon} &= \hat{\Psi}^{t+\tau,t}(x) - \Phi^{t+\tau,t}(x) \end{aligned}$$

gegeben. Wir nehmen an, dass $\hat{\Psi}$ die genauere der beiden Evolutionen ist und dass

$$\theta := \frac{\|\hat{\epsilon}\|}{\|\epsilon\|} < 1 \tag{7.23}$$

gilt. Als Schätzung von ϵ erhalten wir

$$[\epsilon] := \Psi^{t+\tau,t}x - \hat{\Psi}^{t+\tau,t}x.$$

Also gilt $[\epsilon] = \epsilon - \hat{\epsilon}$ und

$$\|[\epsilon] - \epsilon\| = \|\hat{\epsilon}\| = \theta\|\epsilon\|.$$

Mit Hilfe der Dreiecksungleichung folgt

$$\begin{aligned} \|[\epsilon]\| - \|\epsilon\| &\leq \theta\|\epsilon\|, \\ -\|[\epsilon]\| + \|\epsilon\| &\leq \theta\|\epsilon\| \end{aligned}$$

und somit

$$(1-\theta)\|\epsilon\| \leq \|[\epsilon]\| \leq (1+\theta)\|\epsilon\|. \tag{7.24}$$

Der Fehler wird also unter der Annahme (7.23) weder grob überschätzt noch grob unterschätzt. Ist $\hat{\Psi}$ sogar von höherer Ordnung als Ψ, so gilt $\theta \to 0$ für $\tau \to 0$ und deshalb

$$\|[\epsilon]\| \overset{\tau\to 0}{\longrightarrow} \|\epsilon\|.$$

In diesem Fall nennen wir den Fehlerschätzer ***asymptotisch exakt***.

Wir stehen nun allerdings vor einem Dilemma: Wenn wir schon die genauere Approximation $\hat{\Psi}^{t+\tau,t}x$ an $\Phi^{t+\tau,t}x$ ausrechnen, möchten wir mit diesem Wert auch weiterrechnen! Dies wird heute auch üblicherweise so gemacht. Die Fehlertoleranzbedingung wird damit (für $\theta \leq \frac{1}{2}$) typerischerweise übererfüllt:

$$\|\hat{\epsilon}\| = \theta\|\epsilon\| \leq \frac{\theta}{1-\theta}\|[\epsilon]\| \leq \|[\epsilon]\| \approx \text{TOL}.$$

Wir sind also auf der sicheren Seite. Andererseits wird damit das Konzept der Fehlerschätzung aufgegeben. Da aber der lokale Fehler im allgemeinen

ohnehin wenig Informationen über den globalen Fehler liefert, ist dieses Argument nicht so gewichtig.

Wir berechnen also eigentlich ein *optimales* Gitter für das ungenauere Verfahren Ψ. Dieses ist dann aber im allgemeinen auch ein gutes Gitter für das genauere Verfahren $\hat{\Psi}$.

Notation 7.13. Mit $RKp(q)$ bezeichnen wir ein adaptives Runge-Kutta-Verfahren, bei dem wir mit einer Evolution der Ordnung p weiterrechnen und eine Evolution der Ordnung q zur Fehlerschätzung, bzw. Schrittweitensteuerung benutzen.

Zum Beispiel bedeutet RK5(4) in dieser Notation, dass $\hat{\Psi}$ die Ordnung 5 und Ψ die Ordnung 4 besitzt. Dies ist etwa mit Matlab-Integrator `ode45` der Fall.

7.8 Eingebettete Runge-Kutta-Verfahren

Um die Anzahl der Funktionsauswertung von f zu reduzieren, suchen wir Paare von diskreten Evolutionen $\hat{\Psi}$ und Ψ, die von Butcher-Schemata $(A, \hat{b})$ und (A, b) mit der gleichen Runge-Kutta-Matrix A erzeugt werden. Derartige Paare heißen ***eingebettete Runge-Kutta-Verfahren*** und werden durch ein erweitertes Butcher-Schema

$$\begin{array}{c|c} c & A \\ \hline & \hat{b}^\top \\ \hline & b^\top \end{array}$$

dargestellt.

Als Beispiel suchen wir ein eingebettetes Runge-Kutta-Verfahren vom Typ RK4(3), bei dem die genauere Evolution $\hat{\Psi}$ durch das Standard-Runge-Kutta-Verfahren der Ordnung 4 gegeben ist:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \qquad \hat{b} = \begin{pmatrix} \frac{1}{6} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{6} \end{pmatrix}.$$

Für b erhalten wir analog zu Satz 7.21 die Ordnungsbedingungen

$$\begin{aligned} b_1 + b_2 + b_3 + b_4 &= 1, \\ \frac{1}{2}b_2 + \frac{1}{2}b_3 + b_4 &= \frac{1}{2}, \end{aligned}$$

$$\begin{aligned} \frac{1}{4}b_2 + \frac{1}{4}b_3 + b_4 &= \frac{1}{3}, \\ \frac{1}{4}b_3 + \frac{1}{2}b_4 &= \frac{1}{6}. \end{aligned}$$

Diese besitzen aber die eindeutige Lösung $b = \hat{b}$. Es ist als $\Psi = \hat{\Psi}$, und dies ist kein sinnvolles eingebettetes Runge-Kutta-Verfahren.

Aus diesen Betrachtungen folgt, dass wir, um von $\hat{\Psi}$ verschiendenes Runge-Kutta-Verfahren Ψ der Ordnung 3 zu konstruieren, das die Stufen $k_1, \ldots, k_4$ von $\hat{\Psi}$ besitzt, zusätzliche Stufen von Ψ einführen müssen. Dies erscheint aber unökonomisch, da wir mit mehr Stufen weniger Genauigkeit erreichen. Einen Ausweg bietet der ***Fehlbergtrick***: Wir können als zusätzliche Stufe die erste Stufe des nächsten Schrittes zu benutzen. Da wir diese Stufe so und so berechnen müssen, erhalten wir formal ein 5-stufiges, effektiv jedoch ein 4-stufiges Verfahren.

Allgemein sind bei einem s-stufigen Runge-Kutta-Verfahren $(A, \hat{b})$ die s-te Stufe k_s und die erste Stufe k_1^* des folgenden Schritts gegeben durch

$$\begin{aligned} k_1^* &= f(t + \tau, x + \tau \sum_{j=1}^{s} \hat{b}_j k_j), \\ k_s &= f(t + c_s\tau, x + \tau \sum_{j=1}^{s-1} a_{sj} k_j). \end{aligned}$$

Aus der Forderung $k_s = k_1^*$ bei einem FSAL-Verfahren ergibt sich

$$c_s = 1, \quad \hat{b}_s = 0, \quad a_{sj} = \hat{b}_j \qquad \text{für} \quad j = 1, \ldots, s-1.$$

Für das obige Beispiel gelangen wir damit zu dem Ansatz

$$\begin{array}{c|ccccc} 0 & & & & & \\ \frac{1}{2} & \frac{1}{2} & & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & & \\ 1 & 0 & 0 & 1 & & \\ 1 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & \\ \hline & b_1 & b_2 & b_3 & b_4 & b_5 \end{array} .$$

Dies führt auf die Ordnungsbedingungen

$$\begin{aligned} b_1 + b_2 + b_3 + b_4 + b_5 &= 1, \\ \frac{1}{2}b_2 + \frac{1}{2}b_3 + b_4 + b_5 &= \frac{1}{2}, \end{aligned}$$

$$\begin{aligned} \frac{1}{4}b_2 + \frac{1}{4}b_3 + b_4 + b_5 &= \frac{1}{3}, \\ \frac{1}{4}b_3 + \frac{1}{2}b_4 + \frac{1}{2}b_5 &= \frac{1}{6}. \end{aligned}$$

Da in diesem Gleichungssystem die Rollen von b_4 und b_5 vertauschbar sind, muss mit

$$\hat{b}^\top = \left(\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6}, 0\right)$$

auch

$$b^\top = \left(\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, 0, \frac{1}{6}\right)$$

eine Lösung sein. Der Fehlerschätzer dieses effektiv 4-stufigen Verfahrens vom Typ RK4(3) ist gegeben durch

$$[\epsilon] = \frac{1}{6}(k_4 - k_1^*).$$

Beispiel 7.17. *Ein eingebettetes Runge-Kutta Verfahren vom Typ RK3(2) ist gegeben durch*

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{3}{4} & 0 & \frac{3}{4} & & \\ 1 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & \\ \hline & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & 0 \\ \hline & \frac{7}{24} & \frac{1}{4} & \frac{1}{3} & \frac{1}{8} \end{array}.$$

Dieses Verfahren wird im Matlab-Solver `ode23` *benutzt.*

7.9 Implizite Runge-Kutta-Verfahren

In Abschnitt 7.4 haben wir neben dem expliziten Euler-Verfahren bereits auch das implizite Euler-Verfahren eingeführt. An diesem Abschnitt wollen wir abschließend auch noch Runge-Kutta-Verfahren auf implizite Verfahren verallgemeinern. Dazu veranschaulichen wir zunächst mögliche Vorteile gegenüber expliziten Verfahren an einem Beispiel.

Beispiel 7.18. *Wie unter Abschnitt 7.4 betrachten wir die Differentialgleichung*

$$x'(t) = f(t, x(t)) = \lambda \cdot x(t) \qquad \text{mit} \qquad x(0) = 1$$

mit $t_0 = 0$. *Das explizite Euler-Verfahren ergab*

$$x_{j+1} = x_j + \tau \cdot \lambda \cdot x_j \qquad \text{mit} \qquad x_0 \;\; x(0) = 1$$

und das implizite Euler-Verfahren lieferte

$$x_{j+1} \;=\; \frac{x_j}{1-\tau\cdot\lambda} \qquad \text{mit} \qquad x_0 \;\; x(0) \;=\; 1.$$

In Abbildung 7.5 ist die exakte Lösung $x(t)=\exp(\lambda t)$ *sowie sind die beiden diskreten Lösungen für* $\lambda=-85$ *dargestellt. Dabei wurde beim expliziten Euler-Verfahren* $\tau=0,025$ *und beim impliziten Euler-Verfahren* $\tau=0,1$ *gewählt.*

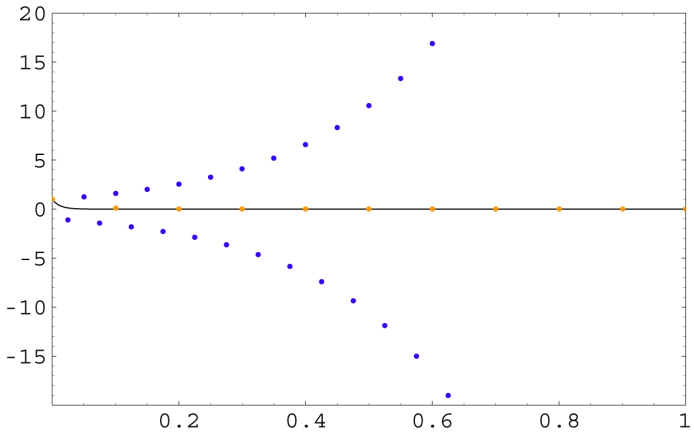


Abbildung 7.5: Vergleich von explizitem und implizitem Euler-Verfahren.

Es ist deutlich zu erkennen, dass das implizite Euler-Verfahren die exakte Lösung im Rahmen der Genauigkeit gut approximiert während die Lösung des expliziten Euler-Verfahrens selbst bei deutlich kleinerer Schrittweite stark oszilliert und damit eine unbrauchbare Lösung liefert.

Wir wollen nun klären, wieso diese Effekte auftreten. Dazu bemerken wir, dass das explizite Euler-Verfahren mit $x(0)=1$ *auch dargestellt werden kann als*

$$x_{j+1} \;=\; (1+\lambda\cdot\tau)^j$$

und das implizit Euler-Verfahren als

$$x_{j+1} \;=\; \left(\frac{1}{1-\lambda\cdot\tau}\right)^j.$$

Wir wissen, dass die exakte Lösung für $\lambda<0$ *und* $t\to\infty$ *gegen 0 konvergiert. Das explizite Euler-Verfahren konvergiert für* $\lambda<0$ *nun genau dann gegen 0, wenn*

$$|1+\lambda\cdot\tau| \;<\; 1$$

gilt. Dies wiederrum ist nur dann der Fall, wenn $\tau<2/|\lambda|$ *gilt. Das implizite Euler-Verfahren hingegen konvergiert für alle* $\tau>0$ *gegen 0, denn für alle*

$\lambda < 0$ und $\tau > 0$ gilt

$$\left| \frac{1}{1 - \lambda \cdot \tau} \right| \; < \; 1.$$

Es sei bemerkt, dass dieser Effekt nicht nur beim Euler-Verfahren, sondern bei allen Runge-Kutta-Verfahren auftreten kann.

Lemma 7.23. *Sei $\Psi_\lambda^\tau(x)$ die diskrete Evolution eines konsistenten und autonomisierungsinvarianten expliziten Runge-Kutta-Verfahrens zur Differentialgleichung $x'(t) = \lambda \cdot x(t)$ mit $x(0) = 1$. Dann gilt für alle Schrittweiten $\tau > 0$*

$$\lim_{\lambda \to -\infty} |\Psi_\lambda^\tau(1)| \; \to \; \infty.$$

Beweis. Wir verwenden die Tatsache, dass $p(\lambda) = \Psi_\lambda^\tau(1)$ ein Polynom in λ vom Grad $u \leq s$ ist. Da der Grad von $p(\lambda)$ aufgrund der Konsistenz mindestens Eins ist, folgt direkt die Behauptung. □

Wir erinnern uns daran, dass die exakte Evolution einer Differentialgleichungen die Stabilitätsbedingung

$$\|\Phi^{t,t_0}(x_0) - \Phi^{t,t_0}(y_0)\|_2 \; \leq \; \mathrm{e}^{L_+(t-t_0)} \cdot \|x_0 - y_0\|_2$$

erfüllt, siehe Satz 7.10. Dabei ist L_+ die einseitige Lipschitzkonstante von f. Für explizite Runge-Kutta-Verfahren *erbt* die diskrete Evolution Ψ diese Stabilitätseigenschaften, aber nur mit der Konstanten $L_\Psi = \gamma L$, wobei L die Lipschitzkonstante von f ist.

Diese Konstante geht exponentiell in die Fehlerabschätzung aus Satz 7.14 ein:

$$\|x_j - x(t)\| \; \leq \; r(\tau_\Delta) \; = \; \begin{cases} \mathrm{err}(\tau_\Delta) \cdot \frac{1}{L_\Psi} \cdot \left(\mathrm{e}^{L_\Psi(t-t_0)} - 1\right) & \text{für} \quad L_\Psi > 0 \\ \mathrm{err}(\tau_\Delta) \cdot (t - t_0) & \text{für} \quad L_\Psi = 0 \end{cases}$$

Daher wäre es gut, wenn $L_\Psi \approx L_+$ gilt. Das ist aber bei expliziten Runge-Kutta-Verfahren nicht gegeben, falls $L_+ \ll L$. Solche Differentialgleichungen heißten ***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_+ einfließt. Es sei bemerkt, dass

steife Differentialgleichungen in der Praxis sehr häufig auftreten. Sie können aber meistens gut mit impliziten Runge-Kutta-Verfahren gelöst werden.

Wir wollen nun die Bezeichungen von impliziten Runge-Kutta-Verfahren einführen.

Notation 7.14. Ein ***implizites Runge-Kutta-Verfahren*** wird gegeben durch

$$k_i \;=\; k_i(t,x,\tau) \;=\; f\left(t+c_i\tau,\; x+\tau\cdot\sum_{j=1}^{s} a_{ij}k_j\right)$$

für $i=1,\ldots,s$ und

$$\Psi^{t+\tau,t}(x) \;=\; x+\tau\cdot\sum_{j=1}^{s} b_j k_j.$$

Dazu verwenden wir die Notationen

$$A \;=\; \begin{pmatrix} a_{11} & \ldots & a_{1s} \\ \vdots & & \vdots \\ a_{s1} & \ldots & a_{ss} \end{pmatrix} \;\in\; \mathbb{R}^{s\times s},$$

$$b \;=\; \begin{pmatrix} b_1 \\ \vdots \\ b_s \end{pmatrix} \;\in\; \mathbb{R}^s \qquad \text{und} \qquad c \;=\; \begin{pmatrix} c_1 \\ \vdots \\ c_s \end{pmatrix} \;\in\; \mathbb{R}^s.$$

In Kurzform schreiben wir ein Runge-Kutta-Verfahen (A,b,c) im Butcher-Schema

$$(A,b,c) \;=\; \begin{array}{c|ccc} c_1 & a_{11} & \ldots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \ldots & a_{ss} \\ \hline & b_1 & \ldots & b_s \end{array}.$$

Die folgende Notation beschreibt einige Spezialfälle.

Notation 7.15. Gegeben sei ein implizites Runge-Kutta-Verfahren durch das Butcher-Schema (A,b,c).

(1) Gilt $a_{ij}=0$ für $i\le j$, so liegt ein explizites Runge-Kutta-Verfahren vor.

(2) Gilt $a_{ij}=0$ für $i<j$, so liegt ein ***diagonal-implizites*** Runge-Kutta-Verfahren vor. Gilt speziell $a_{ii}=\theta$ für $i=1,\ldots,s$, so liegt ein ***SDIRK***-Verfahren vor.

(3) Gilt $a_{ij} \neq j$ für ein $j > i$, so liegt ein ***voll-implizites*** Runge-Kutta-Verfahren vor.

Beispiel 7.19. *Das implizite Euler-Verfahren lässt sich durch das Butcher-Schema*

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

beschreiben. Das **Mittelpunktsverfahren** *wird gegeben durch das Butcher-Schema*

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

und hat die Konsistenzordnung $p = 2$.

Bei der Implementation von impliziten Runge-Kutta-Verfahren sind in jedem Schritt die k_i durch Lösen von

$$k_i(t,x,\tau) \;=\; f\left(t + c_i\tau,\; x + \tau \sum_{j=1}^{s} a_{ij} k_j(t,x,\tau)\right) \qquad \text{für} \quad i = 1,\ldots,s$$

zu ermitteln. Leider funktionieren Fixpunktiterationen nur mit Schrittweitenbeschränkungen, oft werden daher Newton-Verfahren oder Varianten davon verwendet.

Im folgenden stellen wir kurz die wichtigsten Aussagen zu impliziten Runge-Kutta-Verfahren zusammen.

Satz 7.24. *Es gelten sinngemäß die Bedingungen für Konsistenz und Invarianz gegen Autonomisierung sowie die Ordnungsbedingungen auch für implizite Runge-Kutta-Verfahren.*

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 $p \in \Pi_s^n$ mit $p(x_1,\ldots,x_n)$ zu approximieren. Dieses soll das Anfangswertproblem an vorgegebenen Stützstellen lösen. Als Stützstellen definieren wir die Kollokationspunkte $t_0 + c_i\tau$ für $i = 1,\ldots,s$. Weiter soll

$$\begin{aligned} p'(t_0 + c_i\tau) &= f(t_0 + c_i\tau,\; p(t_0 + c_i\tau)) \qquad \text{für} \quad i = 1,\ldots,s, & (7.25)\\ p(t_0) &= x_0. & (7.26) \end{aligned}$$

gelten.

Lemma 7.25. *Seien für $0 \leq c_1 < \ldots < c_s \leq 1$ die Bedingungen (7.25) und (7.26) eindeutig lösbar. Dann wird durch die diskrete Evolution*

$$\Psi^{t_0+\tau,t}(x_0) \;=\; p(t_0+\tau)$$

ein implizites Runge-Kutta-Verfahren definiert, das durch die Parameter

$$\begin{aligned} a_{ij} &= \int_0^{c_i} L_j(\tau)\,\mathrm{d}\tau \qquad \text{für} \quad i,j = 1,\ldots,s \\ b_i &= \int_0^{1} L_i(\tau)\,\mathrm{d}\tau \qquad \text{für} \quad i = 1,\ldots,s \end{aligned}$$

gegeben ist. Dabei sind $L_i(\tau)$ für $i = 1,\ldots,s$ die Langrange-Polynome zu den Stützstellen $c_1,\ldots,c_s$.

Lemma 7.26. *Ein durch Kollokation erzeugtes 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 sehr viel tiefliegenderes Ergebnis.

Satz 7.27. *Für gegebene Parameter $c_1,\ldots,c_s$ sei die Quadraturformel*

$$\int_0^1 g(t)\,\mathrm{d}t \;\approx\; \sum_{i=1}^{s} b_i g(c_i)$$

exakt für alle Polynome in Π_{p-1} mit $p \geq s$. Dann hat das zu $c_1,\ldots,c_s$ gehörende durch Kollokation erzeugte Runge-Kutta-Verfahren die Konsistenzordnung p.

Eine ausführlichere Beschreibung von Kollokationsverfahren kann zum Beispiel in Hohage (2006) nachgelesen werden.

7.10 Ausblick

In diesem Kapitel haben wir zunächst die wichtigsten Grundlagen von Anfangswertproblemen diskutiert und die Existenz und Eindeutigkeit von Lösungen diskutiert. Anschließend haben wir mit dem Runge-Kutte-Verfahren eine der wichtigsten Lösungsmethoden ausführlich vorgestellt.

Aufgrund der großen Anzahl von Anwendungsgebieten der Anfangswertprobleme gibt es hier auch sehr viele weitere Theorien wie zum Beispiel weitere

Begriffe der Stabilität. Weiter sei auch bemerkt, dass wir nur Anfangswertprobleme betrachtet haben, bei denen die Differentialgleichung jeweils nur von einer Größe abhingen, nämlich von der Zeit t. Weiterhin haben wir auch nur Einschrittverfahren untersucht und Mehrschrittverfahren komplett vernachlässigt. Weitere Ergebnisse zu Mehrschrittverfahren sowie zu steifen Differentialgleichungen sind in Töring and Spellucci (1990) zu finden.

8 Randwertprobleme

In diesem Kapitel werden wir grundlegende Verfahren zur numerischen Lösung von Randwertproblemen einführen. Dabei beschränken wir uns wieder nur auf gewöhnliche Differentialgleichungen und auch hier untersuchen wir besonders einige Spezialfälle wie lineare Probleme. Zunächst werden wir Notationen einführen, einige Beispiele geben und auf allgemeine Existenz und Eindeutigkeitsaussagen hinweisen. Anschließend diskutieren wir kurz Schießverfahren, die Grundidee der Methode der finiten Differenzen und schließlich betrachten wir die Methode der finiten Elemente. Dabei erläutern wir das Verfahren nur anhand symmetrischer und linearer gewöhnlicher Differentialgleichungen zweiter Ordnung und weisen darauf hin, dass sich das Verfahren auf viele weitere Klassen von Differentialgleichungen erweitern lässt.

8.1 Notationen und Grundlagen

Nachdem wir uns bislang mit Anfangswertproblemen bei gewöhnlichen Differentialgleichungen beschäftigt haben, gehen wir nun zu Randwertproblemen über. Dazu beschränken wir uns auf gewöhnlichen Differentialgleichungen zweiter Ordnung, alles andere würde hier zu weit führen.

Während wir bei Anfangswertproblemen eine Lösung $x(t)$ gesucht haben, die eine Differentialgleichung der Form

$$x''(t) \;=\; f(t, x(t), x'(t))$$

unter den *Anfangsbedingungen* $x(t_0) = \alpha$ und $x'(t_0) = \beta$ an der *einen* Stelle t_0 erfüllt, wollen wir nun eine Lösung $u(x)$ von

$$u''(x) \;=\; f(x, u(x), u'(x))$$

finden, die bestimmten *Randbedingungen* and *zwei* Stellen a und b erfüllt.

Gesucht ist in unserem Falle damit eine Funktion $u : [a, b] \to \mathbb{R}$, die die gewöhnliche Differentialgleichung zweiter Ordnung

$$u''(x) \;=\; f(x, u(x), u'(x))$$

unter den beiden Randbedingungen

$$g_i(a, b, u(a), u(b), u'(a), u'(b)) \;=\; 0 \qquad \text{für} \quad i = 1, 2$$

erfüllt.

Beispiel 8.1. *Auf eine an den Stellen a und b eingespannte Saite wirke eine Federkraft der Dichte c und eine äußere Last der Dichte h. Dabei nehmen wir an, dass die Federkraft linear ist. Weiter sei die Saitenspannung gegeben durch T und $u(x)$ bezeichne die vertikale Auslenkung der Saite an der Stelle $x \in [a, b]$. Dann gehorcht u der Differentialgleichung*

$$-Tu''(x) + c(x)u(x) \;=\; h(x)$$

für $x \in (a, b)$ mit den Randbedingungen $u(a) = u(b) = 0$.

Zur Festlegung einiger Bezeichnungen und Vereinfachungen führen wir die folgenden beiden Definitionen ein.

Definition 8.1. Eine Differentialgleichung zweiter Ordnung heißt ***quasilinear***, falls

$$u'' \;=\; f(x, u, u') \;=\; B(x, u)u' + C(x, u)$$

gilt. Sie heißt ***semilinear***, falls

$$u'' \;=\; f(x, u, u') \;=\; b(x)u' + C(x, u)$$

und ***linear***, falls

$$u'' \;=\; f(x, u, u') \;=\; b(x)u' + c(x)u - h(x)$$

gilt.

In praktischen Anwendungen reicht es zudem oft aus nur spezielle, einfache Randbedingungen zu betrachten.

Definition 8.2. Die Randbedingungen einer gewöhnlichen Differentialgleichung zweiter Ordnung heißen ***entkoppelt***, wenn

$$g_1(a, u(a), u'(a)) \;=\; 0 \qquad \text{und} \qquad g_2(b, u(b), u'(b)) \;=\; 0$$

gilt.

Weiter heißen lineare und entkoppelte Randbedingungen der Form

$$\begin{aligned} u(a) \;&=\; \alpha \qquad \text{und} \qquad u(b) \;=\; \beta, \\ u'(a) \;&=\; \alpha \qquad \text{und} \qquad u'(b) \;=\; \beta, \\ cu(a) + u'(a) \;&=\; \alpha \qquad \text{und} \qquad du(b) + u'(b) \;=\; \beta \end{aligned}$$

Randbedingungen erster Art (***Dirichlet Randbedingungen***), Randbedingungen zweiter Art (***Neumann Randbedingungen***) bzw. Randbedingungen dritter Art (***Robin Randbedingungen***).

Im allgemeinen Fall nennen wir die Randbedingungen ***gemischt***, wenn auf $x = a$ und $x = b$ unterschiedliche Typen von Randbedingungen gestellt werden.

Bei allen weiteren Betrachtungen werden wir nur lineare Randwertprobleme zweiter Ordnung mit Randbedingungen erster Art untersuchen, also Probleme der Form

$$(Lu)(x) \;:=\; -u''(x) + b(x)u'(x) + c(x)u(x) \;=\; h(x) \tag{8.1}$$

mit $x \in (a, b)$ unter den Randbedingungen

$$u(a) \;=\; \alpha \qquad \text{und} \qquad u(b) \;=\; \beta.$$

Lemma 8.1. *Jedes lineare Randwertproblem erster Art lässt sich überführen in ein lineare Randwertproblem erster Art mit* **homogenen** *Randbedingungen, also mit* $\alpha = \beta = 0$.

Beweis. Der Behauptung folgt direkt, wenn die Transformation

$$u(x) \;=\; v(x) \;+\; \alpha \cdot \frac{x-b}{a-b} \;+\; \beta \cdot \frac{x-a}{b-a}$$

angewandt wird. □

Lemma 8.2. *Jedes lineare Randwertproblem erster Art lässt sich überführen in ein lineare Randwertproblem erster Art mit* $[a, b] = [0, 1]$.

Beweis. Hierzu muss lediglich die Transformation $u = (b - a)\xi$ angewandt werden. □

Durch diese beiden Aussagen reicht es, wenn wir uns im folgenden auf die Spezialfälle $\alpha = \beta = 0$ und $[a, b] = [0, 1]$ beziehen. Zunächst betrachten wir aber noch ein Beispiel.

Beispiel 8.2. *Wir betrachten einen isothermen Strömungsreaktor mit kontinuierlicher Zufuhr bzw. Abfuhr der Reaktionsmasse bzw. des Reaktionspro-*

duktes. Die Konzentrationsverteilung $c(\tau, \xi_1, \xi_2, \xi_3)$ im Reaktor ergibt sich nun aus der Stoffbilanzgleichung

$$\frac{\partial c}{\partial t} \;=\; -\sum_{i=1}^{3} \frac{\partial}{\partial \xi_i}(w_i c) + \sum_{i=1}^{3} \frac{\partial}{\partial \xi_i}\left(D \frac{\partial c}{\partial \xi_i}\right) + r(c).$$

Dabei ist $w = (w_1, w_2, w_3)$ das Geschwindigkeitsfeld, D die Diffusionskonstante und $r(c)$ der Reaktionsterm.

Zur Vereinfachung des Problems nehmen wir an, dass der Reaktor zeitunabhängig betrieben wird, dass die Diffusionskonstante zeitunabhängig ist und dass $(w_1, w_2, w_3) = (v, 0, 0)$ gilt. Zuden werden wir nur die Änderung der Konzentration in ξ_1 Richtung betrachten, sodass wir schließlich die gewöhnliche Differentialgleichung zweiter Ordnung

$$-D \frac{\mathrm{d}^2 c}{\mathrm{d}\xi} + v \frac{\mathrm{d}c}{\mathrm{d}\xi} + r(c) \;=\; 0$$

für $0 < \xi < L$ erhalten.

Durch Entdimensionierung mittels $x = \xi/L$, $u(x) = c(x)/c_0$ mit der Anfangskonzentration c_0 und $P = vL/D$ erhalten wir

$$-\frac{1}{P} u''(x) + u'(x) + R(u) \;=\; 0$$

für $0 < x < 1$. Zudem können wir die Randbedingungen

$$u(0) - \frac{1}{P} u'(0) \;=\; 1 \qquad \text{und} \qquad u'(1) \;=\; 0$$

festlegen. Damit haben wir ein Beispiel einer semilinearen Differentialgleichung zweiter Ordnung mit gemischten Randbedingungen.

Im folgenden wollen wir die Lösbarkeit von linearen Randwertproblem erster Art für den symmetrischen Fall, d.h. mit $b(x) = 0$, untersuchen. Auch hierzu beginnen wir mit einem Beispiel.

Beispiel 8.3. *Die allgemeine Lösung der Schwingungsgleichung*

$$-u''(x) - u(x) \;=\; 0 \qquad \text{für} \qquad x \in (a, b)$$

hat die Form

$$u(x) \;=\; c \cdot \cos(x) + d \cdot \sin(x).$$

Die beiden Konstanten c und d sind so zu bestimmen, dass jeweils die Randbedingungen $u(a) = \alpha$ und $u(b) = \beta$ erfüllt sind. Daraus erhalten wir das lineare Gleichungssystem

$$\cos(a) \cdot c + \sin(a) \cdot d \;=\; \alpha \qquad \text{und} \qquad \cos(b) \cdot c + \sin(b) \cdot d \;=\; \beta.$$

Wir erkennen, dass dieses Gleichungssystem in Abhängigkeit von a und b sowie α und β entweder gar keine, eine oder unendlich viele Lösungen hat.

Schon an diesem einfachen Beispiel sehen wir, dass es bei Randwertproblemen kein allgemeines Ergebnis analog zum Satz von Picard-Lindelöf bei Anfangswertproblemen gibt. Trotzdem erhalten wir für eine Spezialfälle eine eindeutige Lösung, wie der folgende Satz zeigt.

Satz 8.3. *Wir betrachten die Differentialgleichung*

$$-u''(x) + c(x)u(x) \;=\; h(x)$$

mit $x \in (0,1)$ unter den Randbedingungen $u(0) = u(1) = 0$. Dabei seien $c, h \in \mathcal{C}([0,1])$ und $c(x) \geq 0$ für alle $x \in [0,1]$.

Dann hat das gegebene (symmetrische) Randwertproblem genau eine Lösung.

Beweis. Zunächst beschäftigen wir uns mit der Eindeutigkeit und nehmen an u_1 und u_2 seien zwei Lösungen. Dann löst die Funktion $u = u_1 - u_2$ das Randwertproblem

$$-u''(x) + c(x)u(x) \;=\; 0 \qquad \text{mit} \qquad x \in (0,1)$$

unter den Randbedingungen $u(0) = u(1) = 0$. Multiplizieren wir diese Differentialgleichung nun mit der Funktion u, Integrieren über $[0,1]$ und führen eine partielle Integration mit $u''u$ durch, so erhalten wir

$$0 \;=\; \int_0^1 \left[-u''(x) + c(x) \cdot u(x)\right] \mathrm{d}x \;=\; \int_0^1 \left[(u'(x))^2 + c(x) \cdot (u(x))^2\right] \mathrm{d}x$$

unter Beachtung der Randbedingungen. Mit $c(x) \geq 0$ und $u \in \mathcal{C}([0,1])$ folgt damit aber $u = 0$ und somit die Eindeutigkeit der Lösung.

Zu Existenz betrachten wir die allgemeine Lösung

$$u(x) \;=\; \alpha_1 u_1(x) + \alpha_2 u_2(x) + \tilde{u}(x)$$

des gegebenen Problems. Dabei bilden u_1 und u_2 ein Fundamentalsystem aus zwei linear unabhängigen Lösungen der homogenen Differentialgleichung, d.h. mit $h = 0$. $\tilde{u}$ sei eine beliebige Lösung von (8.1). Die Aussage lässt sich

nun mit dem Satz von Picard-Lindelöf zeigen, siehe Kapitel 7. Zur Erfüllung der Randwertbedingungen entsteht das lineare Gleichungssystem

$$\begin{aligned} u_1(0)\alpha_1 + u_2(0)\alpha_2 &= \alpha - \tilde{u}(0), \\ u_1(1)\alpha_1 + u_2(1)\alpha_2 &= \beta - \tilde{u}(1). \end{aligned}$$

Dieses System ist eindeutig lösbar. Sind nämlich α_1 und α_2 Lösung des zugehörigen homogenen Systems, dann wäre $u = \alpha_1 u_1 + \alpha_2 u_2$ eine Lösung des entsprechenden homogenen Randwertproblems und damit $u = 0$ durch die bereits bewiesene Eindeutigkeitsaussage. Da u_1 und u_2 aber linear unabhängig sind, folgt $\alpha_1 = \alpha_2 = 0$. Damit haben wir aber eine Lösung gefunden. □

8.2 Schießverfahren

In diesem Abschnitt werden wir bereits bekannte Methoden zur Lösung von Anfangswertproblemen verwenden, um Randwertprobleme zu lösen. Die wesentliche Idee besteht darin, fehlende Anfangswerte durch Parameter zu ersetzen und damit das Anfangswertproblem zu lösen. Wir schiessen also zunächst ins Blaue hinein und schauen uns dann den Fehler in den Randwerten an. Um diesen zu minimieren, d.h. den zugefügten Parameter in den Anfangswerten zu optimieren, werden wir das schon bekannte Newton-Verfahren verwenden.

Der Vorteil dieser Methoden liegt in den Voraussetzungen an das Randwertproblem, welches wir der Einfachheit halber folgendermaßen annehmen:

$$u''(x) = f(x, u, u'), \quad x \in (0,1), \quad u(0) = a \quad \text{und} \quad u(1) = b. \tag{8.2}$$

Die Funktion f kann im Gegensatz zu den später dargestellten Verfahren nichtlinear sein. Wir nehmen an, dass es eine Lösung u dieses Randwertproblems gibt. Dann löst u auch das Anfangswertproblem

$$v''(x) = f(x, v, v'), \quad x \in (0,1), \quad v(0) = a \quad \text{und} \quad v'(0) = \alpha, \tag{8.3}$$

wobei $\alpha = u'(0)$ genau die *richtige* Ableitung an der Stelle 0 sein soll. Da diese jedoch nicht bekannt ist, wählen wir zunächst α beliebig und erhalten so eine Lösung v_α, welche von der Wahl von α abhängt.

Unter der Voraussetzung, dass das Anfangswertproblem für alle α eindeutig lösbar ist, erhalten wir eine Funktion $F : \mathbb{R} \to \mathbb{R}$, welche α auf den Fehler von v_α im Randwert an der Stelle 1 abbildet:

$$F(\alpha) := v_\alpha(1) - b. \tag{8.4}$$

Die Berechnung dieser Funktion erfordert für jedes α die Lösung eines Anfangswertproblems! Mit dieser Formulierung ist es uns jedoch möglich, das Randwertproblem umzuformulieren: Eine Lösung v_α des Anfangswertproblems (8.3) löst das Randwertproblem (8.2) genau dann, wenn $F(\hat{\alpha}) = 0$ ist.

Wir haben das Randwertproblem also auf ein Anfangswertproblem kombiniert mit einer Nullstellensuche zurückgeführt. Letztere können wir theoretisch mit jedem beliebigen Verfahren zur Nullstellensuche lösen. Wir werden hier das bereits aus Abschnitt 2.4 bekannte Newton-Verfahren verwenden. Dazu benötigen wir jedoch die Ableitung der Funktion F nach α, welche durch

$$F'(\alpha) \;=\; \partial_\alpha v_\alpha(1)$$

gegeben ist. Dazu sei bemerkt, dass v_α die Lösung des Anfangswertproblems (8.3) ist, indem α als einer der Anfangswerte vorkommt. Um diese Ableitung nach α zu berechnen, differenzieren wir die Differentialgleichung

$$\partial_\alpha v''_\alpha \;=\; \partial_\alpha f(x, v_\alpha, v'_\alpha) \;=\; f_{v_\alpha}(x, v_\alpha, v'_\alpha)\; \partial_\alpha v_\alpha + f_{v'_\alpha}(x, v_\alpha, v'_\alpha)\; \partial_\alpha v'_\alpha.$$

f_{v_α} und $f_{v'_\alpha}$ sind die partiellen Ableitungen von f in Richtung der zweiten bzw. dritten Variablen.

Wir definieren nun $w_\alpha := \partial_\alpha v_\alpha$ und setzen voraus, dass wir in voriger Gleichung die Differentiationen nach α und nach x vertauschen dürfen. Damit erhalten wir ein Anfangswertproblem für w_α:

$$w''_\alpha \;=\; f_{v_\alpha}(x, v_\alpha, v'_\alpha)\; w_\alpha + f_{v'_\alpha}(x, v_\alpha, v'_\alpha)\; w'_\alpha, \qquad w_\alpha(0) \;=\; 0, \qquad w'_\alpha(0) \;=\; 1. \tag{8.5}$$

Die Anfangswerte entstehen direkt durch Differentiation der Anfangswerte von v_α nach α. Eine exakte Herleitung dieses Anfangswertproblems findet sich zum Beispiel in Hanke-Bourgeois (2006).

Wir erhalten auf diese Weise die benötigte Ableitung $F'(\alpha) = w_\alpha(1)$. Auch für diese muss für jedes α ein Anfangswertproblem gelöst werden, welches sogar noch von v_α abhängt. Es ergibt sich somit ein gekoppeltes Anfangswertproblem zur Berechnung von (v_α, w_α):

$$\begin{aligned}
v''_\alpha(x) &= f(x, v_\alpha, v'_\alpha), \\
w''_\alpha &= f_{v_\alpha}(x, v_\alpha, v'_\alpha)\; w_\alpha + f_{v'_\alpha}(x, v_\alpha, v'_\alpha)\; w'_\alpha, \\
v_\alpha(0) &= a, \quad v'_\alpha(0) \;=\; \alpha, \\
w_\alpha(0) &= 0, \quad w'_\alpha(0) \;=\; 1.
\end{aligned}$$

Zusammen mit dem Newton-Verfahren zur Nullstellensuche ergibt sich daraus Algorithmus 8.1.

Input: f, f_u, $f_{u'}$, a, b, Startwert α_0

`for` $k = 0, 1, \ldots$ `do`

Löse das Anfangswertproblem für $(v_{\alpha_k}, w_{\alpha_k})$

Berechne $\alpha_{k+1} = \alpha_k - \dfrac{v_{\alpha_k}(1) - b}{w_{\alpha_k}(1)}$

`end for`.

Output: v_α ist Approximation an die Lösung des Randwertproblems (8.2) und α ist Approximation an die Ableitung an der Stelle 0.

Algorithmus 8.1: Einfaches Schießverfahren mit Newton-Iterationen.

In der bislang beschriebenen Form ist das Schießverfahren nicht besonders stabil. So wird es keineswegs für jedes α immer eine Lösung des Anfangsproblems geben, welches das gesamte Intervall umfasst. Der Satz von Picard-Lindelhöf 7.5 garantiert nur unter den geforderten Voraussetzungen die Lösung auf einem möglicherweise sehr kleinen Intervall! Auch aus anderen Gründen kann das Verfahren instabil werden. Daher wird es häufig durch Mehrzielmethoden verbessert, welche jedoch auch nur bedingt brauchbar sind. Z.B. erscheint eine Verallgemeinerung auf partielle Differentialgleichungen, welche häufig bei Randwertproblemen auftauchen, schwierig.

Dennoch haben Schießverfahren einen wichtigen Vorteil: Sie verwenden numerische Verfahren zu Lösung von Anfangswertproblemen, welche zum einen gut funktionieren, vor allem aber keine besonderen Voraussetzungen an f benötigen. Gefordert wird in der Regel nur hinreichende Regularität. Für die nachfolgenden Verfahren muss f in der Regel linear sein, was eine schwerwiegende Einschränkung darstellt.

8.3 Methode der finiten Differenzen

Wir betrachten nun normierte lineare Randwertprobleme mit homogenen Randbedingungen erster Art, also

$$-u''(x) + b(x)u'(x) + c(x)u(x) \;=\; f(x) \qquad \text{mit} \qquad x \in (0,1)$$

unter den Randbedingungen $u(0) = u(1) = 0$. Dabei schreiben wir $f(x)$ statt $h(x)$ wie zuvor verwendet, um eine Verwechselung mit der späteren Schrittweite h zu vermeiden.

Bei der Methode der finiten Differenzen ersetzen wir die Ableitungen durch Differenzenquotienten und leiten ein Gleichungssystem zur her, welches die gesuchte Lösung u an vorgegebenen Knotenpunkten approximiert.

Dazu betrachten wir für ein festes $n \in \mathbb{N}$ die äquidistante Zerlegung

$$\Delta \;=\; \{x_i = ih \;:\; i = 0, \ldots, n+1\} \qquad \text{mit} \qquad h \;=\; \frac{1}{n+1}.$$

Zur Approximation der ersten Ableitung $u'(x_i)$ an einem $x_i \in \{x_1, \ldots, x_n\}$ betrachten wir drei Varianten:

(1) Den ***Vorwärtsdifferenzen Quotienten***

$$u'(x_i) \;\approx\; D^+u(x_i) \;=\; \frac{u(x_{i+1}) - u(x_i)}{h}.$$

(2) Den ***Rückwärtsdifferenzen Quotienten***

$$u'(x_i) \;\approx\; D^-u(x_i) \;=\; \frac{u(x_i) - u(x_{i-1})}{h}.$$

(3) Den ***Zentraldifferenzen Quotienten***

$$u'(x_i) \;\approx\; D^0u(x_i) \;=\; \frac{u(x_{i+1}) - u(x_{i-1})}{2h}.$$

Zur Approximation der zweiten Ableitung $x''(x_i)$ an einer Stelle $x_i \in \Delta$ betrachten wir nur den ***Zentraldifferenzen Quotienten*** 2. Ordnung, also

$$u''(x_i) \;\approx\; D^+D^-u(x_i) \;=\; \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2}.$$

Mit u_i für $i = 1, \ldots, n$ bezeichnen wir die Näherungswerte an die gesuchte exakte Lösung u an den Stellen x_i, es soll also gelten $u_i \approx u(x_i)$. Nutzen wir diese Notation und ersetzen nun die ersten und zweiten Ableitungen durch ihre Zentraldifferenzen Quotienten, so erhalten wir

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + b(x_i)\frac{u_{i+1} - u_{i-1}}{2h} + c(x_i)u_i \;=\; f(x_i).$$

Mit den Bezeichnungen

$$b_i \;=\; b(x_i), \qquad c_i \;=\; c(x_i) \qquad \text{und} \qquad f_i \;=\; f(x_i)$$

für $i = 1, \ldots, n$ erhalten wir damit das Gleichungssystem

$$\frac{1}{h^2} \cdot \left(-\left(1 + \frac{b_i h}{2}\right) u_{i-1} + (2 + c_i h^2)u_i - \left(1 - \frac{b_i h}{2}\right) u_{i+1} \right) \;=\; f_i.$$

Die Randbedingungen liefern uns sofort $u_0 = u_{n+1} = 0$. Mit der Matrix

$$A \;=\; \frac{1}{h^2} \cdot \operatorname{tridiag}\left(-\left(1 + \frac{b_i h}{2}\right),\; (2 + c_i h^2),\; -\left(1 - \frac{b_i h}{2}\right) \right) \qquad (8.6)$$

sowie den Vektoren $v = (u_1, \ldots, u_n)$ und $g = (f_1, \ldots, f_n)$ erhalten wir schließlich das lineare Gleichungssystem

$$A \cdot v = g. \tag{8.7}$$

Das durch dieses Gleichungssystem gegebene Lösungsverfahren heißt ***Methode der finiten Differenzen***.

Bevor wir uns mit Konvergenzaussagen beschäftigen, wollen wir kurz die Lösbarkeit dieses Gleichungssystems diskutieren.

Satz 8.4. *Gegeben sei das Randwertproblem*

$$-u''(x) + b(x)u'(x) + c(x)u(x) = f(x) \qquad \text{mit} \qquad x \in (0,1)$$

unter den Randbedingungen $u(0) = u(1) = 0$. *Weiter gelte*

$$c_i \geq 0 \qquad \text{und} \qquad \left|\frac{b_i h}{2}\right| \leq 1$$

für $i = 1, \ldots, n$. *Dann hat das Gleichungssystem* $Av = g$ *zur Methode der finiten Differenzen genau eine Lösung.*

Beweis. Sei $A = (a_{i,j})_{1 \leq i,j \leq n}$. Mit den gegebenen Voraussetzungen haben wir

$$|a_{ii}| = |2 + c_i h^2| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| = \left|1 + \frac{b_i h}{2}\right| + \left|1 - \frac{b_i h}{2}\right| = 2$$

für $i = 1, \ldots, n$. Dies heißt aber, dass die Matrix A schwach diagnoaldominant ist.

Ferner ist A als Tridiagonalmatrix irreduzibel und dies impliziert die Invertierbarkeit von A. □

Da es sich bei der Matrix A um eine Tridiagonalmatrix handelt, lässt sich mit dem Verfahren aus Abschnitt 3.6 die Methode der finiten Differenzen in linearer Zeit in n durchführen.

Für die Stabilitäts- und Konvergenzanalyse bezeichnen wir mit Rw die Einschränkung einer Funktion $w \in \mathcal{C}([0,1])$ auf die Gitterpunkte $\{x_1, \ldots, x_n\}$ und mit L den Differentialoperator des Randwertproblems. Weiter sei u eine

exakte und v eine durch die Methode der finiten Differenzen approximierte Lösung. Dann gilt für den Diskretisierungsfehler $Ru - v$ gerade

$$A \cdot (Ru - v) \;=\; ARu - Av \;=\; ARu - g \;=\; ARu - RLu.$$

Dabei bezeichnen wir den letzten Term als ***Defekt***. Im folgenden werden wir stets die Maximumsnorm verwenden und definieren

$$\|v\|_{\infty,\Delta} \;:=\; \max_{i=1,\ldots,n} |v_i| \qquad \text{für} \qquad v = (v_1, \ldots, v_n).$$

Der Index Δ soll dabei verdeutlichen, dass das Argument der Norm im folgenden von der Zerlegung Δ abhängt.

Definition 8.3. Eine finite Differenzen Methode heißt ***konsistent*** in der Maximumsnorm, falls

$$\lim_{n\to\infty} \|ARu - RLu\|_{\infty,\Delta} \;=\; 0$$

gilt. Weiter hat eine finite Differenzen Methode die ***Konsistenzordnung*** p, falls es eine von h unabhängige Konstante $C > 0$ gibt mit

$$\|ARu - RLu\|_{\infty,\Delta} \;\leq\; C \cdot h^p.$$

Diese Begriffe haben wir bereits für Anfangswertprobleme verwendet. Dabei beschreibt die Konsistenz wie gut der Differentialoperator durch die Diskretisierung approximiert wird.

Definition 8.4. Eine finite Differenzen Methode heißt ***stabil*** in der Maximumsnorm, falls für die Lösung v des Gleichungssystems $Av = g$ die Existenz einer von h unabhängige Konstanten $D > 0$ folgt mit

$$\|v\|_{\infty,\Delta} \;=\; \|A^{-1} \cdot g\|_{\infty,\Delta} \;\leq\; D \cdot \|g\|_{\infty,\Delta}.$$

Schließlich definieren wir die Konvergenz.

Definition 8.5. Eine finite Differenzen Methode heißt ***konvergent*** in der Maximumsnorm, falls

$$\lim_{n\to\infty} \|Ru - v\|_{\infty,\Delta} \;=\; 0$$

gilt. Weiter hat eine finite Differenzen Methode die ***Konvergenzordnung*** p, falls es eine von h unabhängige Konstante $M > 0$ gibt mit

$$\|Ru - v\|_{\infty,\Delta} \;\leq\; M \cdot h^p.$$

Natürlich wollen wir diese Begriffe nun anhand der speziellen finiten Differenzen Methode mit den Zentraldifferenzen Quotientenaus den Gleichungen (8.6) und (8.7) analysieren. Dazu untersuchen wir zunächst ausführlich den Konsistenzfehler.

Lemma 8.5. *Sei $u \in \mathcal{C}^4([0,1])$. Dann gilt*

$$(D^0 u)(x) \;=\; u'(x) + h^2 R \qquad \text{mit} \qquad |R| \;\leq\; \frac{1}{6}\|u^{(3)}\|_2$$

sowie

$$(D^+ D^- u)(x) \;=\; u''(x) + h^2 R \qquad \text{mit} \qquad |R| \;\leq\; \frac{1}{12}\|u^{(4)}\|_2.$$

Beweis. Aus der Taylorentwicklung von u an einer Stelle $x \in (0,1)$ folgt

$$\begin{aligned} u(x \pm h) &= u(x) \pm h u'(x) + h^2 \frac{u''(x)}{2} \pm h^3 R_3^{\pm}, \\ u(x \pm h) &= u(x) \pm h u'(x) + h^2 \frac{u''(x)}{2} \pm h^2 \frac{u^{(3)}(x)}{6} + h^4 R_4^{\pm} \end{aligned}$$

mit den Fehlerdarstellungen

$$\begin{aligned} R_3^{\pm} &= \frac{1}{h^3} \cdot \int_x^{x \pm h} \big(u''(\xi) - u''(x)\big) \cdot (x \pm h - \xi)\, \mathrm{d}\xi, \\ R_4^{\pm} &= \frac{1}{h^4} \cdot \int_x^{x \pm h} \Big(u^{(3)}(\xi) - u^{(3)}(x)\Big) \cdot \frac{(x \pm h - \xi)^2}{2}\, \mathrm{d}\xi. \end{aligned}$$

Somit folgt die erste Behauptung durch einfaches Nachrechnen aus

$$(D^0 u)(x) \;=\; \frac{u(x+h) - u(x-h)}{2h} \;=\; u'(x) + h^2 (R_3^+ - R_2^-)$$

und analog zeigt man auch die zweite Behauptung. □

Mit dieser kleinen Vorarbeit erhalten wir nun eine Aussage über die Konsistensordnung.

Satz 8.6. *Sei $u \in \mathcal{C}^4([0,1])$ eine Lösung des Randwertproblems*

$$-u''(x) + b(x)u'(x) + c(x)u(x) \;=\; f(x) \qquad \text{mit} \qquad x \in (0,1)$$

unter den Randbedingungen $u(0) = u(1) = 0$. Dann hat die zugehörige finite Differenzen Methode aus (8.6) und (8.7) die Konsistenzordnung 2.

Beweis. Unter Beachtung der eingeführten Bezeichnungen erhalten wir

$$\begin{aligned} & (ARu - RLu)(x_i) \\ = \; & [D^+D^-u(x_i) + b_i D^0 u(x_i) + c_i u(x_i)] - [-u''(x_i) + b_i u'(x_i) + c_i u(x_i)]. \end{aligned}$$

Lemma 8.5 liefert uns nun direkt

$$|(ARu - RLu)(x)| \;\leq\; \frac{1}{12}h^2\|u^{(4)}\|_2 + \frac{1}{6}h^2\|b\|\|u^{(3)}\|_2$$

und die Behauptung folgt aus Maximumbildung über den Gitterpunkten $x_1, \ldots, x_n$. □

Unter den Voraussetzung der Sätze 8.4 und 8.6 lässt sich etwas aufwendiger auch zeigen, dass die finite Differenzen Methode aus (8.6) und (8.7) die Konvergenzordnung 2 hat. Hierzu sei aber auf Lube (2005b) verwiesen.

8.4 Methode der finiten Elemente

Abschließend wollen wir noch die Methode der finiten Elemente vorstellen, die vorallem bei der Numerik von Randwertproblemen bei partiellen Differentialgleichungen viele Anwendungen findet. Wir diskutieren an dieser Stelle selbstverständlich nur den einfacher zu behandelnde Fall einer linearen gewöhnlichen Differentialgleichung mit homogenen Randbedingungen.

Dazu benötigen wir einige Vorarbeit, welche sich auf ganz allgemeine lineare Räume verallgemeinern lässt.

Definition 8.6. Seien $c, f \in \mathcal{C}([0,1])$ mit $c \geq 0$. Weiter definieren wir für $u, v \in \mathcal{C}([0,1])$

$$\begin{aligned} B(u,v) &= \int_0^1 (u'(x)v'(x) + c(x)u(x)v(x))\,\mathrm{d}x, \\ l(u) &= \int_0^1 f(x)u(x)\,\mathrm{d}x. \end{aligned}$$

Zudem sei

$$J(u) \;=\; B(u,u) - 2 \cdot l(u)$$

und

$$U \;=\; \{u \in \mathcal{C}^1([0,1]) \;:\; u(0) = u(1) = 0\}.$$

Dann heißt die Aufgabe J auf U zu minimieren, also ein $u \in U$ zu finden mit

$$J(u) \;\leq\; J(v) \qquad \text{für alle} \qquad v \in U,$$

das ***Variationsproblem*** zum Randwertproblem

$$-u''(x) + c(x)u(x) \;=\; f(x) \qquad \text{mit} \qquad x \in (0,1)$$

unter den Randbedingungen $u(0) = u(1) = 0$.

Satz 8.7. *Die Funktion $u \in U$ ist genau dann Lösung des Variationsproblems, wenn sie die* **Variationsgleichung**

$$B(u,v) \;=\; l(v) \qquad \text{für alle} \qquad v \in U$$

erfüllt.

Beweis. Zunächst gilt für alle $u, v \in U$ die Beziehung

$$J(u+v) - J(u) \;=\; 2 \cdot (B(u,v) - l(v)) + B(v,v). \tag{8.8}$$

Nun sei u eine Lösung des Variationsproblems und wir nehmen an es gibt ein $v \in U$ mit

$$B(u,v) \;\neq\; l(v).$$

Dann erhalten wir mit (8.8) für $w = \alpha v$ mit $\alpha \in \mathbb{R}$ die Gleichung

$$J(u+\alpha v) - J(u) \;=\; 2\alpha \cdot (B(u,v) - l(v)) + \alpha^2 \cdot B(v,v).$$

Für

$$\alpha \;=\; -\frac{B(u,v) - l(v)}{B(v,v)}.$$

erhalten wir damit aber $J(u+\alpha v) < J(u)$, was ein Widerspruch ist.

Nun sei umgekehrt die Variationsgleichung $B(u,v) = l(v)$ für alle $v \in U$ erfüllt. Dann folgt aber mit (8.8) sowie mit $v = u + w$ sofort $J(u) \leq J(v)$ für alle $v \in U$. □

Der folgende Satz liefert schließlich die entscheidene Verbindung zwischen Variations- und Randwertproblemen.

Satz 8.8. *Die Funktion $u \in U$ ist genau dann Lösung des Variationsproblems, wenn sie das Randwertproblem*

$$-u''(x) + c(x)u(x) \;=\; f(x) \qquad \text{mit} \qquad x \in (0,1)$$

unter den Randbedingungen $u(0) = u(1) = 0$ löst.

Beweis. Zunächst sei u eine Lösung des Variationsproblems. Weiter definieren wir

$$r(x) \;=\; \int_0^x (c(\xi)u(\xi) - f(\xi))\,\mathrm{d}\xi$$

für $x \in [0,1]$. Offenbar gilt $r \in \mathcal{C}^1([0,1])$ mit $r'(x) = c(x)u(x) - f(x)$. Partielle Integration der Variationsgleichung ergibt

$$\int_0^1 (u'(x) - r(x)v(x)\,\mathrm{d}x \;=\; 0$$

für alle $v \in U$. Wir setzen nun

$$q \;=\; \int_0^1 (u'(x) - r(x))\,\mathrm{d}x$$

sowie

$$v_0(x) \;=\; \int_0^x (u'(\xi) - r(\xi) - q)\,\mathrm{d}x$$

für $x \in [0,1]$. Dann ist $v_0 \in U$ und es gilt

$$\int_0^1 (u'(x) - r(x) - q)^2\,\mathrm{d}x \;=\; \int_0^1 (u'(x) - r(x) - q)v_0'(x)\,\mathrm{d}x$$
$$= \int_0^1 (u'(x) - r(x))v_0'(x)\,\mathrm{d}x \;-\; q \cdot \int_0^1 v_0'(x)\,\mathrm{d}x \;=\; 0.$$

Daher gilt aber $u'(x) = r(x) + q$ und da $r \in \mathcal{C}^1([0,1])$ folgt $u \in \mathcal{C}^2([0,1])$ mit

$$u''(x) \;=\; r'(x) \;=\; c(x)u(x) - f(x).$$

Sei nun u eine Lösung des Randwertproblems. Dann liefert uns partielle Integration direkt

$$B(u,v) - l(v) \;=\; \int_0^1 (-u''(x) + c(x)u(x) - f(x))v(x)\,\mathrm{d}x \;=\; 0$$

für alle $v \in U$, d.h. die Variationsgleichung ist erfüllt und mit Satz 8.7 folgt die Behauptung. □

Ausgehend von der Formulierung als Variationsproblem lässt sich auch ein Existenz- und Eindeutigkeitsbeweis für das zugehörige Randwertproblem führen, was wir hier aber nicht vorführen wollen. Benötigt wird dazu der Hilbertraum mit der durch B erklärten Norm.

Wir stellen nun das ***Ritz-Galerkin Verfahren*** vor. Die Grundidee dabei ist die Zielfunktion J auf einem endlich dimensionalen Unterraum von U zu minimieren.

Satz 8.9. *Sei U_n ein endlich dimensionaler Unterraum von U. Dann gibt es genau ein $u_n \in U_n$, welches J auf U_n minimiert.*

Sei weiter $\{\varphi_1, \ldots, \varphi_m\}$ eine Basis von U_n, dann sind die Koeffizienten in

$$u_n(x) \;=\; \sum_{k=1}^{m} \alpha_k \varphi_k(x)$$

eindeutig bestimmt als Lösung der **Ritz-Galerkin Gleichungen**

$$\sum_{k=1}^{m} \alpha_k B(\varphi_i, \varphi_k) \;=\; l(\varphi_i) \qquad \text{für} \quad i = 1, \ldots, m.$$

Beweis. Analog zu Satz 8.8 gilt $J(u_n) \leq J(v)$ für alle $v \in U_n$ genau dann, wenn

$$B(u_n, v) \;=\; l(v) \qquad \text{für alle} \quad v \in U_n.$$

Dies ist aber äquivalent zu den Ritz-Galerkin Gleichungen. Wegen

$$\sum_{i,k=1}^{m} \alpha_i \alpha_k B(\varphi_i, \varphi_k) \;=\; B(v, v) \;>\; 0 \qquad \text{mit} \qquad v(x) \;=\; \sum_{k=1}^{m} \alpha_k \varphi_k(x) \;\neq\; 0$$

ist die Matrix $B(\varphi_i, \varphi_k)$ dieses Gleichungssystems positiv definit und daher regulär. □

Wir werden uns nun mit der Frage beschäftigen, welche Unterräume U_n gewählt werden sollten und das Ritz-Galerkin Verfahren mit linearen Splines vorführen.

Zunächst bemerken wir, dass Unterräume aus Polynomen ungeeignet sind, da sie zu schlecht konditionierten Gleichungssystemen führen. Geeigneter sind dager Splines, hier vor allem die Räume der linearen und kubischen Splines.

Für den Fall von linearen Splines wählen wir die äquidistante Zerlegung

$$\Delta \;=\; \{x_i = ih \;:\; i = 0, \ldots, n+1\} \qquad \text{mit} \qquad h \;=\; \frac{1}{n+1}.$$

Damit besteht U_n aus allen auf $[0,1]$ stetigen Funktionen u mit $u(0) = u(1) = 0$, die auf jedem der Intervalle $[x_i, x_{i+1}]$ für $i = 0, \ldots, n$ affine linear sind.

Als Basis von U_n wählen wir die ***Dachfunktionen***

$$\varphi_k(x) \;=\; \begin{cases} \frac{1}{h}(x - x_{k-1}) & \text{für} \quad x \in [x_{k-1}, x_k] \\ \frac{1}{h}(x_{k+1} - x) & \text{für} \quad x \in [x_k, x_{k+1}] \\ 0 & \text{sonst} \end{cases} \qquad \text{für} \quad k = 1, \ldots, n.$$

Für jedes $u \in U_n$ gilt damit für $\alpha_k = u(x_k)$ die Darstellung

$$u(x) \;=\; \sum_{k=1}^{n} \alpha_k \varphi_k(x).$$

Wir bemerken, dass

$$B(\varphi_i, \varphi_k) \;=\; \int_0^1 (\varphi_i'(x)\varphi_k'(x) + c(x)\varphi_i(x)\varphi_k(x))\,\mathrm{d}x \;=\; 0$$

für $(x_{i-1}, x_{i+1}) \cap (x_{k-1}, x_{k+1}) = \emptyset$, falls also $|i-k| > 2$ gilt. Somit ist $B(\varphi_i, \varphi_k)$ eine Tridiagonalmatrix. Weiter berechnen wir

$$B(\varphi_i, \varphi_i) \;=\; \frac{2}{h} + \frac{1}{h^2} \cdot \left(\int_{x_{i_1}}^{x_i} c(x)(x - x_{i-1})^2\,\mathrm{d}x \;+\; \int_{x_i}^{x_{i+1}} c(x)(x_{i+1} - x)^2\,\mathrm{d}x \right)$$

und

$$B(\varphi_i, \varphi_{i+1}) \;=\; -\frac{1}{h} + \frac{1}{h^2} \cdot \int_{x_i}^{x_{i+1}} c(x)(x_{i+1} - x)(x - x_i)\,\mathrm{d}x$$

sowie

$$l(\varphi_i) \;=\; \frac{1}{h} \cdot \left(\int_{x_{i-1}}^{x_i} f(x)(x - x_{i-1})\,\mathrm{d}x \;+\; \int_{x_i}^{x_{i+1}} f(x)(x_{i+1} - x)\,\mathrm{d}x \right).$$

Dabei werden die Koeffizienten im Allgemeinen durch numerische Integration berechnet. Für jedes Teilintervall, also für jedes ***finite Element***, sollten sich die Koeffizienten nach der gleichen Vorschrift ergeben. Definieren wir wie bei den finiten Differenzen

$$c_i \;=\; c(x_i) \qquad \text{und} \qquad f_i \;=\; f(x_i)$$

für $i = 1, \ldots, n$ und approximieren die Funktionen c und f durch lineare Splines, so folgt

$$\begin{aligned} B(\varphi_i, \varphi_i) &= \frac{2}{h} + \frac{h}{12} \cdot (c_{i-1} + 6c_i + c_{i+1}), \\ B(\varphi_i, \varphi_{i+1}) &= -\frac{1}{h} + \frac{h}{12} \cdot (c_i + c_{i+1}), \\ l(\varphi_i) &= \frac{h}{6} \cdot (f_{i-1} + 4f_i + f_{i+1}). \end{aligned}$$

Mit diesen Näherungen haben wir analog zur Methode der finiten Differenzen ein lineares Gleichungssystem mit einer Tridiagonalmatrix zu lösen,

nämlich das Gleichungssystem

$$\begin{aligned} & \frac{1}{h^2} \cdot \left[-\alpha_{i-1} \left(1 - \frac{h^2}{12}(c_{i-1} + c_i) \right) + \alpha_i \left(2 + \frac{h^2}{12}(c_{i-1} + 6c_i + c_{i+1}) \right) \right. \\ & \quad \left. -\alpha_{i+1} \left(1 - \frac{h^2}{12}(c_i + c_{i+1}) \right) \right] \\ = \; & \frac{1}{6}(f_{i-1} + 4f_i + f_{i+1}) \end{aligned}$$

für $i = 1, \ldots, n$. Dabei gelte $\alpha_0 = \alpha_{n+1} = 0$. Mit der Lösung $(\alpha_1, \ldots, \alpha_n)$ dieses Gleichungssystems erhalten wir

$$u_n(x) \;=\; \sum_{k=1}^{n} \alpha_k \varphi_k(x) \;\approx\; u(x).$$

als Approximation an die gesuchte exakte Lösung $u(x)$.

Abschließend wollten wir noch kurz eine Fehlerabschätzung ohne Beweis angeben. Der interessierte Leser sei hier auf Kress (2008) verwiesen.

Satz 8.10. *Sei $u \in \mathcal{C}^2([0,1])$ eine Lösung des Randwertproblems*

$$-u''(x) + c(x)u(x) \;=\; f(x) \qquad \text{mit} \qquad x \in (0,1)$$

unter den Randbedingungen $u(0) = u(1) = 0$. Weiter sei u_n die zughörige Lösung des Ritz-Galerkin Verfahrens mit linearen Splines.

Dann gibt es eine von h unabhängige Konstante $C > 0$ mit

$$\|u - u_n\|_2 \;\leq\; C \cdot \|u''\|_2 \cdot h^2.$$

Dabei verwenden wir die Norm

$$\|u\|_2 \;=\; \left(\int_0^1 |u(x)|^2 \, \mathrm{d}x \right)^{1/2}.$$

8.5 Ausblick

Wie bereits mehrmals hingewiesen, haben wir hier nur die einfachsten und grundlegenden Ideen zur numerischen Lösungen von Randwertproblemen eingeführt. Dabei lässt sich die Methode der finiten Differenzen auch auf partielle Differentialgleichungen übertragen, sodass sich explizite und implizite Differenzenverfahren herleiten lassen.

Neben der Methode der finiten Differenzen lässt sich vor allem auch die Methode der finiten Elemente auf viele weitere Klassen von Randwertproblemen verallgemeinern, zum Beispiel auf mehrdimensionale Probleme oder sogar auf partielle Differentialgleichungen. Damit findet die Methode der finiten Elemente sehr viele Anwendungen in modernen Problemen und es hat sich eine große Zahl von kommerziellen und freien Solvern entwickelt, welche genau diese Methode verwenden.

Zudem sind wir nur wenig auf Schießverfahren zur Lösung von gewöhnlichen Differentialgleichungen eingegangen. Eine Einführung zu all diesen Bereichen kann in Töring and Spellucci (1990) gefunden werden.

Literaturverzeichnis

G. Bärwolff, 2007. *Numerik für Ingenieure, Physiker und Informatiker*. Spektrum Akademischer Verlag, Heidelberg, 1. Auflage.

M. Hanke-Bourgeois, 2006. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. Vieweg und Teubner, 2. Auflage.

T. Hohage, 2005. *Numerische Mathematik I*. Skript zur Vorlesung im Wintersemester 2005/06 an der Universität Göttingen.

T. Hohage, 2006. *Numerische Mathematik II*. Skript zur Vorlesung im Sommersemester 2006 an der Universität Göttingen.

R. Kress, 1998. *Numerical Analysis (Graduate Texts in Mathematics)*. Springer, New York, 1. Auflage.

R. Kress, 2007. *Numerische Mathematik I*. Skript zur Vorlesung im Wintersemester 2007/08 an der Universität Göttingen.

R. Kress, 2008. *Numerische Mathematik II*. Skript zur Vorlesung im Sommersemester 2008 an der Universität Göttingen.

G. Lube, 2005a. *Numerische Mathematik I*. Skript zur Vorlesung im Wintersemester 2004/05 an der Universität Göttingen.

G. Lube, 2005b. *Numerische Mathematik II*. Skript zur Vorlesung im Sommersemester 2005 an der Universität Göttingen.

A. Schöbel, 2006. *Numerik I*. Skript zur Vorlesung im Wintersemester 2006/07 an der Universität Göttingen.

A. Schöbel, 2007. *Numerik II*. Skript zur Vorlesung im Sommersemester 2007 an der Universität Göttingen.

W. Töring, P. Spellucci, 1988. *Numerische Mathematik für Ingenieure und Physiker. Band 1: Numerische Methoden der Algebra*. Springer, Berlin, 2. Auflage.

W. Töring, P. Spellucci, 1990. *Numerische Mathematik für Ingenieure und Physiker. Band 2: Numerische Methoden der Analysis.* Springer, Berlin, 2. Auflage.

R. Schaback und H. Wendland, 2004. *Numerische Mathematik.* Springer, Berlin, 5. Auflage.

Stichwortverzeichnis

ω-orthogonal, 121
äquivalente Normen, 16

A-konjugiert, 67
Abschnittspolynome, 96
Anfangswertproblem, 130
Aufwand, 37
Auslöschung, 7
autonome Differentialgleichung, 128
Autonomisierung, 132
AWP, 130

B-Splines, 105
Banachraum, 21
beschränkt, 17
Butcher-Schema, 154

Cauchy-Folge, 21
CG-Verfahren, 66
charakteristisches Polynom, 71
Cholesky-Zerlegung, 54

Dachfunktionen, 196
Defekt, 191
diagonalisierbar, 79
Differentialgleichung
    gewöhnliche, 128
Dirichlet Randbedingungen, 183
dissipativ, 143
divergent, 16
dividierte Differenzen, 96
Dreiecksmatrix, 37

Eigenvektor, 20, 71
Eigenwert, 20, 71
Einheitskreis, 15
Einschrittverfahren, 145
    konvergent, 149
Einzelschrittverfahren, 62
Energienorm, 65
entkoppelte Randbedingungen, 182
Euler-Verfahren, 146
    explizites, 146
    implizites, 147
Evolution, 141
    diskrete, 145
explizite Differentialgleichung, 129

Fehlbergtrick, 173
Feinheit eines Gitters, 144
finites Element, 197
Fixpunkt, 12
Fixpunktgleichung, 12
Frozen Newton-Verfahren, 28

Gauß-Matrix, 40
Gauß-Seidel-Verfahren, 62
Gauß-Verfahren, 40
Gaußsche Quadraturformel, 120
gedämpftes Newton-Verfahren, 28
gemischte Randbedingungen, 183
gemischtes Integral, 119
Gershgorin-Kreise, 78
Gesamtschrittverfahren, 60
Gitter, 144
Gitterfunktion, 144
Givens-Rotationen, 84
globaler Diskretisierungsfehler, 168
Grad eines Polynoms, 91
Grenzwert, 16

Hauptachsentransformation, 73
Hauptminor, 42

hermitesch, 10
Hessenberg-Matrix, 84
Hochschaltbeschränkung, 169
homogene Randbedingungen, 183
homogenes Gleichungssystem, 35
Householder-Matrizen, 50

implizite Differentialgleichung, 129
Interpolationsfehler, 98
Interpolationsquadratur, 112
inverse Vektoriteration, 81

Jacobi-Matrix, 23
Jacobi-Verfahren, 60

Kardinalsplines, 103
Kern einer Matrix, 36
Kollokationsverfahren, 178
Kondition, 46, 47
konjugierten Gradienten, 66
konsistent, 147, 191
Konsistenzordnung, 150, 191
Kontraktion, 21
konvergent, 16, 191
Konvergenzordnung, 29, 151, 191
konvexe Menge, 10
Krylov-Raum, 69

Lagrange-Interpolation, 92
Lagrange-Interpolationsoperator, 98
Lagrange-Polynome, 93
Lanczos-Verfahren, 84
Landau-Symbol, 37
LDL-Zerlegung, 53
Legendre-Polynome, 124
linear, 182
linear unabhängig, 35
lineare Gleichungssysteme, 35
Lipschitzstetig, 137
    einseitig, 143
    global, 137
    lokal, 137
Literaturverzeichnis, 200
lokaler Diskretisierungsfehler, 168
Lotka-Volterra-Zyklus, 133

LU-Zerlegung, 38

Matrix, 34
Mises, Verfahren von, 78
Mittelpunktsverfahren, 178
Monome, 92

Neumann Randbedingungen, 183
Newton-Côtes-Formeln, 114
Newton-Verfahren, 26, 28
Newtonsche Interpolationsformel, 94
nichtlineares Gleichungssystem, 11
Norm, 15
normierte Dreiecksmatrizen, 37
normierter Raum, 15

obere Dreiecksmatrix, 37
Ordnung einer Differentialgleichung, 128
Ordnungsbedingungen, 160
orthogonal, 10

partielle Differentialgleichung, 128
Permutation, 45
Permutationsmatrix, 45
Picard-Iteration, 139
Pivotelement, 45
Pivotisierung, 45
Polynom, 91
positiv definit, 10
positiv semi-definit, 10
positive Gewichtsfunktion, 119
Propagationsfehler, 168

QR-Zerlegung, 49
quadratische Matrizen, 34
Quadraturformel, 111
Quadraturstellen, 111
Quasi Newton-Verfahren, 28
quasilinear, 182

Rückwärtsdifferenzen Quotient, 189
Rückwärtselimination, 37
Rang einer Matrix, 35
Rayleigh-Quotienten, 75

Rayleigh-Quotienten-Iteration, 81
regulär, 36
Residuum, 66
Ritz-Galerkin Gleichungen, 196
Ritz-Galerkin Verfahren, 195
Robin Randbedingungen, 183
Runge-Kutta-Verfahren, 153
    diagonal-implizites, 177
    eingebettete, 172
    explizites, 153
    implizites, 177
    voll-implizites, 178

Sassenfeld-Kriterium, 62
Satellitenbahn, 166
Schießverfahren, 186
schlecht gestellt, 8
Schrittweite, 66
Schur-Zerlegung, 73
SDIRK-Verfahren, 177
semilinear, 182
Simpson-Regel, 115
    zusammengesetzte, 117
singulär, 36
Singulärvektoren, 88
Singulärwerte, 88
Singulärwertzerlegung, 87
Spaltenpivotisierung, 45
Spaltenrang, 35
Spaltensummennorm, 19
Spektralnorm, 20
Spektralradius, 20
Spline, 100
Störungslemma, 47
Stützstellen, 92
Stützwerte, 92
stabil, 46, 191
starkes Spaltensummenkriterium, 62
starkes Zeilensummenkriterium, 62
steife Differentialgleichungen, 176
Stufenzahl, 154
Suchrichtung, 66
sukzessiven Approximation, 13
symmetrisch, 10

Trajektorie, 149
Trapez-Regel, 114
    zusammengesetzte, 117
Tschebyscheff-Polynome, 125

unitär, 10
untere Dreiecksmatrix, 37

Vandermonde-Matrix, 93
Variationsgleichung, 194
Variationsproblem, 194
Verfahren
    Lanczos, 84
Verfahren von
    Mises, 78
    Wieland, 81
Verfahren von Runge, 153
Vielfachheit einer Nullstelle, 92
vollständiger Raum, 21
Vorwärtsdifferenzen Quotient, 189
Vorwärtselimination, 37

Wellengleichung, 73
wesentliche Rechenoperationen, 37
Wieland, Verfahren von, 81

Zeilenrang, 35
Zeilensummennorm, 19
Zentraldifferenzen Quotient, 189