MATTHIAS GERDTS

NUMERISCHE MATHEMATIK II

Universität Würzburg
SoSe 2010

Addresse des Authors:

Matthias Gerdts

Institut für Mathematik
Universität Würzburg
Am Hubland
97074 Würzburg

E-Mail: gerdts@mathematik.uni-wuerzburg.de
WWW: www.mathematik.uni-wuerzburg.de/∼gerdts

Vorläufige Version: 31. Juli 2010

Inhaltsverzeichnis

Lecture plan

Vorlesungen:

Date	Hours	Pages
21.04.2010	13:30-15:00	VL, 1–11
22.04.2010	08:00-09:45	VL, 11–15
28.04.2010	13:30-15:00	VL 15–20
29.04.2010	08:00-09:45	ÜB
05.05.2010	13:30-15:00	VL, 20–24
06.05.2010	08:00-09:45	VL, 25–31
12.05.2010	13:30-15:00	VL, 31–37
13.05.2010	08:00-09:45	Himmelfahrt
19.05.2010	13:30-15:00	VL, 37–42,45
20.05.2010	08:00-09:45	ÜB
26.05.2010	13:30-15:00	VL, 45-46, 43-44, 51-53
27.05.2010	08:30-09:45	VL, 47–51
02.06.2010	13:30-15:00	ÜB
03.06.2010	08:00-09:45	Fronleichnam
09.06.2010	13:30-15:00	VL, 55-65
10.06.2010	08:00-09:45	VL, 66-73
16.06.2010	13:30-15:00	VL, 73-77
17.06.2010	08:00-09:45	VL,78-84
23.06.2010	13:30-15:00	VL, 84-88
24.06.2010	08:00-09:45	ÜB
30.06.2010	13:30-15:00	VL, 89–95,101-102
01.07.2010	08:00-09:45	VL, 95–102
07.07.2010	13:30-15:00	VL, 102–107, 109
08.07.2010	08:00-09:45	ÜB
14.07.2010	13:30-15:00	112– 119 VL
15.07.2010	08:00-09:45	ausgefallen
21.07.2010	13:30-15:00	VL, 108, 120–122
22.07.2010	08:00-09:45	ÜB

Kapitel 1

Eigenwertprobleme

Ziel dieses Kapitels ist die numerische Bestimmung von einem oder mehreren Eigenwerten einer Matrix $A \in \mathbb{C}^{n\times n}$, sowie ggf. die Berechnung zugehöriger Eigenvektoren.

Definition 1.0.1

Sei $A \in \mathbb{C}^{n\times n}$.

(i) Die Zahl $\lambda \in \mathbb{C}$ heißt **Eigenwert der Matrix** *A, wenn*

$$Ax = \lambda x \quad \text{für ein } x \neq 0$$

gilt. Ein Vektor $0 \neq x \in \mathbb{C}$ mit $Ax = \lambda x$ heißt **Eigenvektor zum Eigenwert** *λ.*

(ii) Die Menge aller Eigenwerte von A heißt **Spektrum von** *A.*

(iii) Der **Spektralradius von** *A ist definiert als*

$$\rho(A) := \max\{|\lambda| \mid \lambda \text{ ist Eigenwert von } A\}.$$

Eigenwerte spielen in vielen Anwendungen eine wichtige Rolle.

Beispiel 1.0.2 (Stabilität von Differentialgleichungen)

Sei $x(t)$ eine auf $[t_0, \infty)$ definierte Lösung des Anfangswertproblems

$$x'(t) = f(t, x(t)), \quad x(t_0) = x_0.$$

x heißt **stabil**, *wenn zu jedem $\varepsilon > 0$ ein $\delta > 0$ existiert, so daß alle Lösungen $y(t)$ des Anfangswertproblems mit*

$$\|y(t_0) - x(t_0)\| < \delta$$

für alle $t \geq t_0$ existieren und

$$\|y(t) - x(t)\| < \varepsilon$$

für alle $t \geq t_0$ erfüllen. Die Lösung $x(t)$ heißt **asymptotisch stabil**, *wenn sie stabil ist und wenn $\delta > 0$ existiert, so daß für alle Lösungen $y(t)$ mit $\|y(t_0) - x(t_0)\| < \delta$ gilt*

$$\lim_{t\to\infty} \|y(t) - x(t)\| = 0.$$

Eine Lösung $x(t)$ heißt **instabil**, *wenn sie nicht stabil ist.*

Für Differentialgleichungen der Form

$$x'(t) = Ax(t) + g(t, x(t))$$

mit einer konstanten Matrix $A \in \mathbb{R}^{n\times n}$ und einer stetigen Funktion g, die gleichmäßig in $t \in [t_0, \infty)$ die Bedingung

$$\lim_{\|x\|\to 0} \frac{\|g(t,x)\|}{\|x\|} = 0$$

erfüllt, zeigt Walter [Wal90, Satz VII, S. 215], daß $x(t) \equiv 0$ asymptotisch stabil ist, falls

$$Re(\lambda_j) < 0$$

für alle Eigenwerte λ_j von A gilt. Gibt es einen Eigenwert λ_j mit $Re(\lambda_j) > 0$, so ist die Lösung instabil. Für den Fall $Re(\lambda_j) = 0$ kann nichts ausgesagt werden.

Übungsaufgabe: *Untersuche*

$$\begin{pmatrix} x_1'(t) \\ x_2'(t) \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

im Hinblick auf Stabilität in Abhängigkeit von $a, b, c, d \in \mathbb{R}$.

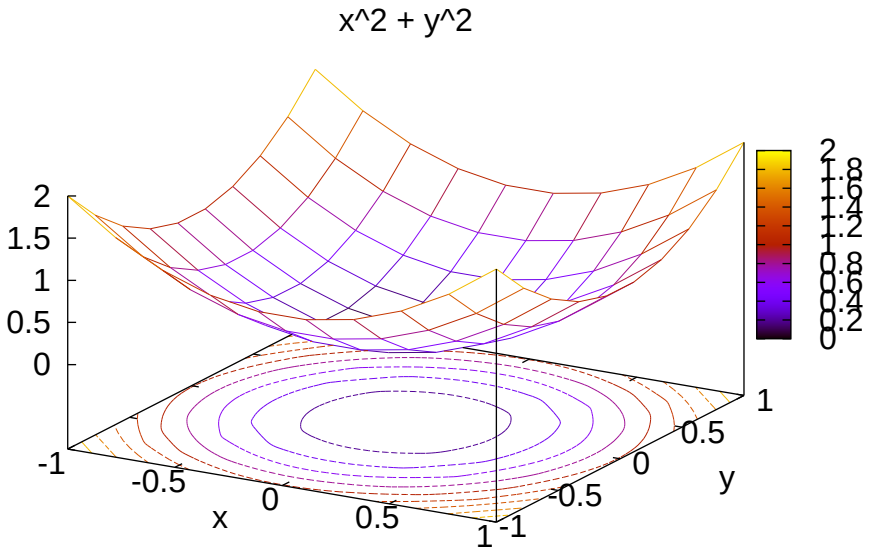
Beispiel 1.0.3 (Konvexität und Eigenwerte)

Man kann zeigen, daß eine zweimal stetig differenzierbare Funktion $f : \mathbb{R}^n \to \mathbb{R}$ genau dann konvex ist, wenn sämtliche Eigenwerte der Hessematrix $H_f(\hat{x}) := \nabla^2 f(\hat{x})$ nicht negativ sind.

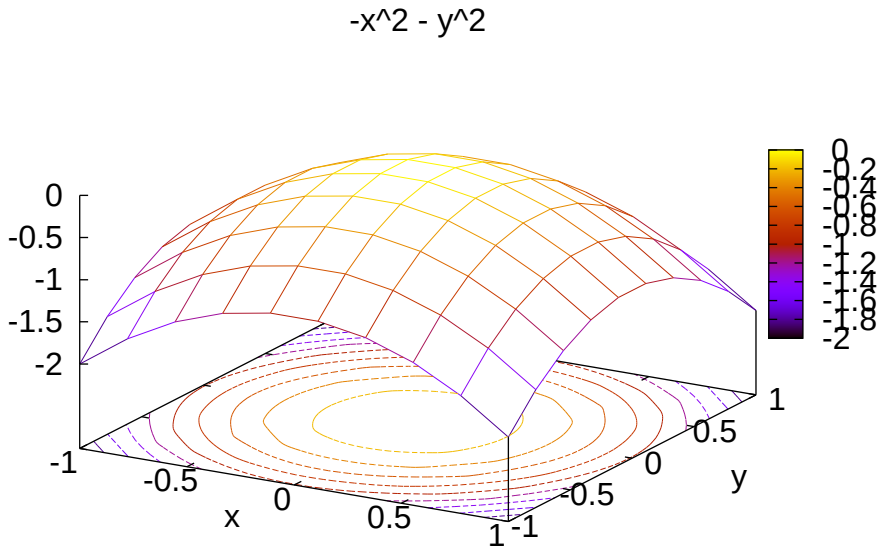
Umgekehrt ist eine Funktion konkav genau dann, wenn sämtliche Eigenwerte der Hessematrix $H_f(\hat{x}) := \nabla^2 f(\hat{x})$ nicht positiv sind.

Wir veranschaulichen mögliche Fälle im $\mathbb{R}^2$:

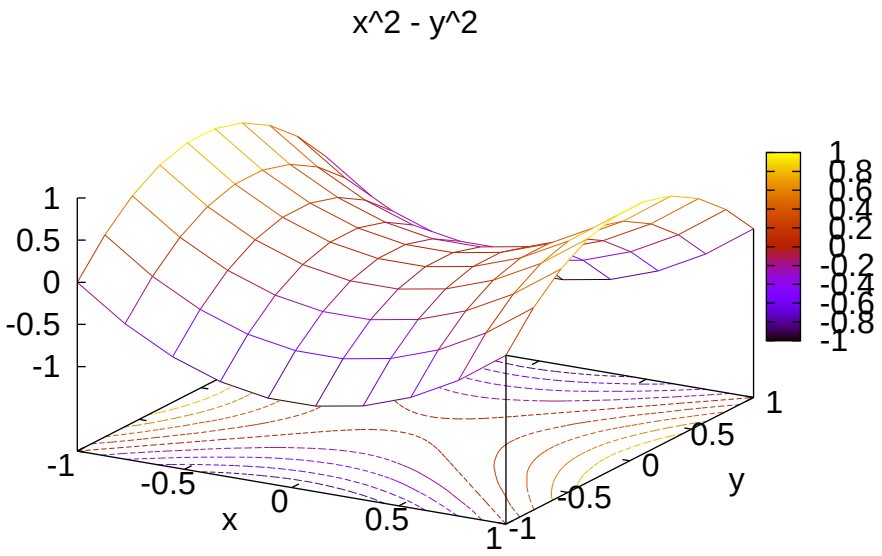
- $f(x, y) = x^2 + y^2$, $H_f(\hat{x}) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ *positiv definit:*



- $f(x,y) = -x^2 - y^2$, $H_f(\hat{x}) = \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix}$ *negativ definit:*



- $f(x,y) = x^2 - y^2$, $H_f(\hat{x}) = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$ *indefinit:*



Beispiel 1.0.4 **(Eigenwerte und Optimierung)**

Gesucht ist ein lokales Minimum (bzw. Maximum) der zweimal stetig differenzierbaren Funktion $f : \mathbb{R}^n \to \mathbb{R}$. Hat man mit einem geeigneten Verfahren einen Kandidaten $\hat{x} \in \mathbb{R}^n$ mit $\nabla f(\hat{x}) = 0$ für ein lokales Minimum (bzw. Maximum) bestimmt, so gilt es zu entscheiden, ob dieser Kandidat tatsächlich ein lokales Minimum (bzw. Maximum) ist. Ein hinreichendes Kriterium für lokale Minimalität (bzw. Maximalität) lautet wie folgt:

Sei $\hat{x}$ ein Punkt mit $\nabla f(\hat{x}) = 0$. Ist die Hessematrix $H_f(\hat{x}) := \nabla^2 f(\hat{x})$ positiv (bzw. negativ) definit, so ist $\hat{x}$ ein lokales Minimum (bzw. Maximum) von f.

Darüber hinaus kann gezeigt werden, daß die folgende notwendige Bedingung gilt:

Sei $\hat{x}$ ein lokales Minimum (bzw. Maximum) von f. Dann ist die Hessematrix $H_f(\hat{x})$ positiv (bzw. negativ) semidefinit.

Daraus ergeben sich folgende Aussagen für $\hat{x}$ mit $\nabla f(\hat{x}) = 0$:

- *Sind sämtliche Eigenwerte von $H_f(\hat{x})$ positiv (bzw. negativ), so ist $\hat{x}$ lokales Minimum (bzw. Maximum) von f.*
- *Hat $H_f(\hat{x})$ mindestens einen negativen (bzw. positiven) Eigenwert, so ist $\hat{x}$ kein lokales Minimum (bzw. Maximum) von f.*

Wir illustrieren die Aussagen für die Funktion

$$f(x, y) := y^2(x - 1) + x^2(x + 1)$$

und berechnen den Gradienten

$$\nabla f(x,y) = \begin{pmatrix} y^2 + 3x^2 + 2x \\ 2y(x-1) \end{pmatrix}$$

und die Hessematrix

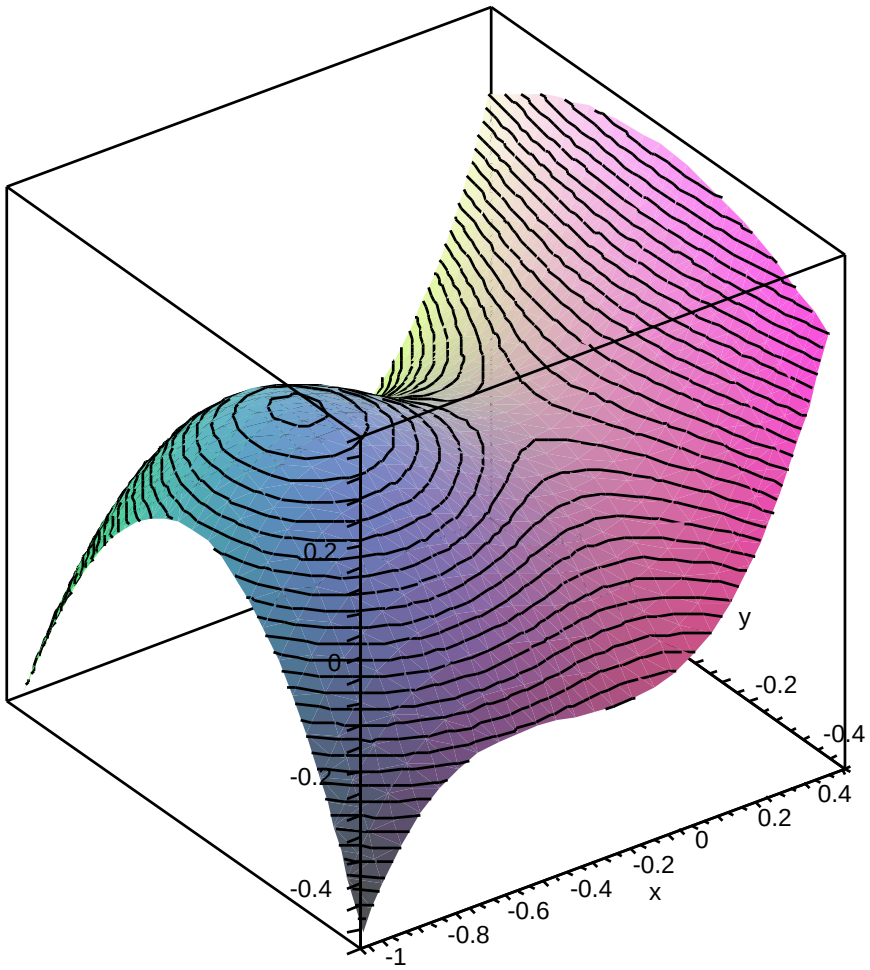
$$\nabla^2 f(x,y) = \begin{pmatrix} 6x+2 & 2y \\ 2y & 2(x-1) \end{pmatrix}.$$

Aus $\nabla f(x,y) = 0$ *ergeben sich somit die stationären Punkte* $(x^0,y^0) = (0,0)$ *und* $(x^1,y^1) = (-2/3, 0)$.

Für die zugehörigen Hesse-Matrizen erhält man

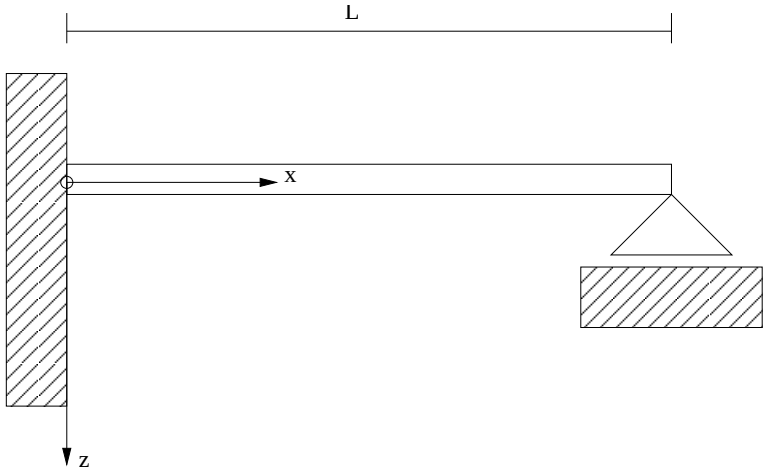
$$\nabla^2 f(x^0,y^0) = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix} \textit{ indefinit } \Rightarrow \begin{pmatrix} x^0 \\ y^0 \end{pmatrix} \textit{ Sattelpunkt}$$

$$\nabla^2 f(x^1,y^1) = \begin{pmatrix} -2 & 0 \\ 0 & -\frac{10}{3} \end{pmatrix} \textit{ negativ definit } \Rightarrow \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} \textit{ striktes lokales Maximum}$$



Beispiel 1.0.5 **(Eigenwerte und Schwingungen)**

Eigenwertprobleme in einer allgemeineren Form treten auch in der Mechanik im Zusammenhang mit Schwingungen auf. Betrachte einen elastischen Balken der Länge L, der am linken und rechten Rand eingespannt ist und schwingen kann, siehe Abbildung.



Ein Eigenwertproblem für diesen Balken lautet wie folgt. Finde eine **Eigenfrequenz** ω *mit*

$$\begin{aligned} z''''(x) &= \omega^2 \frac{\rho A}{EI} z(x), \\ z(0) &= 0, \\ z'(0) &= 0, \\ z(L) &= 0, \\ z''(L) &= 0, \end{aligned}$$

wobei die Massenbelegung ρA und die Biegesteifigkeit EI materialabhängige Konstanten sind. Die zugehörigen „Eigenvektoren“ z sind Lösungen des Randwertproblems und somit Funktionen. Sie heißen **Eigenschwingungen**. *Interessiert ist man hierbei insbesondere an nichttrivialen Lösungen $z \not\equiv 0$.*

Dass die Untersuchung von Eigenschwingungen und Eigenfrequenzen in Bezug auf Resonanz wichtig bei der Konstruktion von z.B. Brücken ist, zeigt der Zusammensturz der Tacoma Bridge, bei der Wind zur einer Anregung geführt hat, was letztendlich in einem Aufschaukeln der Anregung endete und zum Einsturz der Brücke führte. Details finden sich auf der WWW-Seite

http://de.wikipedia.org/wiki/Tacoma-Narrows-Br%C3%BCcke

Aus der Definition des Eigenwerts ergibt sich, daß λ genau dann Eigenwert von A ist, wenn $A - \lambda I$ singulär ist, was wiederum genau dann der Fall ist, wenn

$$\det(A - \lambda I) = 0$$

gilt. Die Eigenwerte einer Matrix A sind also durch die Nullstellen des sogenannten charakteristischen Polynoms φ_A gegeben.

Definition 1.0.6 **(charakteristisches Polynom)**
Sei $A \in \mathbb{C}^{n\times n}$. Die Funktion

$$\varphi_A(\mu) = \det(A - \mu I)$$

heißt **charakteristisches Polynom von** A.

Beispiel 1.0.7
Das charakteristische Polynom der rechten oberen Dreiecksmatrix

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ & \ddots & \vdots \\ & & a_{nn} \end{pmatrix}$$

lautet

$$\varphi_A(\mu) = (a_{11} - \mu)(a_{22} - \mu)\cdots(a_{nn} - \mu).$$

Die Eigenwerte von A sind also gerade die Diagonalelemente a_{ii}, $i = 1, \ldots, n$, von A. Eine analoge Aussage gilt für linke untere Dreiecksmatrizen.

Mit Hilfe des Determinantenentwicklungssatzes läßt sich zeigen, daß φ_A ein Polynom n-ten Grades der Form

$$\varphi_A(\mu) = (-1)^n \left(\mu^n + \alpha_{n-1}\mu^{n-1} + \ldots + \alpha_0\right)$$

ist. Sind $\lambda_i \in \mathbb{C}$, $i = 1, \ldots, k$, die verschiedenen Nullstellen von φ_A mit Vielfachheit σ_i, $i = 1, \ldots, k$, so läßt sich φ_A in der Form

$$\varphi_A(\mu) = (-1)^n(\mu - \lambda_1)^{\sigma_1}(\mu - \lambda_2)^{\sigma_2}\cdots(\mu - \lambda_k)^{\sigma_k}$$

darstellen. Die Zahl σ_i wird auch als **algebraische Vielfachheit des Eigenwerts** λ_i bezeichnet und wir bezeichnen mit σ diejenige Funktion, die einem Eigenwert λ_i dessen algebraische Vielfachheit zuordnet, d.h. $\sigma(\lambda_i) = \sigma_i$.
Die Menge der Eigenvektoren (zzgl. des Nullvektors)

$$L(\lambda) := \{x \in \mathbb{C}^n \mid (A - \lambda I)x = 0\} = Kern(A - \lambda I)$$

bildet einen linearen Teilraum des $\mathbb{C}^n$ mit Dimension $n - Rang(A - \lambda I)$. Insbesondere ist mit x und y auch jede Linearkombination $c_1 x + c_2 y \neq 0$ mit $c_1, c_2 \in \mathbb{C}$ ein Eigenvektor zum Eigenwert λ. Die Zahl $\varrho(\lambda) = \dim(L(\lambda)) = n - Rang(A - \lambda I)$ heißt **geometrische Vielfachheit des Eigenwerts** λ und gibt die maximale Anzahl linear unabhängiger Eigenvektoren zum Eigenwert λ an.
I.a. sind die algebraische und geometrische Vielfachheit zum selben Eigenwert verschieden.

Beispiel 1.0.8

Betrachte die Matrix (Jordankästchen)

$$J(\lambda) = \begin{pmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{pmatrix} \in \mathbb{C}^{n \times n}.$$

Das charakteristische Polynom lautet $\varphi_A(\mu) = (\lambda - \mu)^n$ und $\mu = \lambda$ ist der einzige Eigenwert mit algebraischer Vielfachheit $\sigma(\lambda) = n$. Der Rang von $J(\lambda) - \lambda I$ ist jedoch $n - 1$, so daß die geometrische Vielfachheit $\varrho(\lambda) = n - (n - 1) = 1$ beträgt.

Satz 1.0.9

Zwischen algebraischer und geometrischer Vielfachheit eines Eigenwerts λ besteht folgender Zusammenhang:

$$1 \leq \varrho(\lambda) \leq \sigma(\lambda) \leq n.$$

Beweis: Die Grenzen 1 und n sind klar. Sei nun $\varrho = \varrho(\lambda)$. Seien x_i, $i = 1, \ldots, \varrho$, linear unabhängige Eigenvektoren zum Eigenwert λ mit $Ax_i = \lambda x_i$, $i = 1, \ldots, \varrho$. Ergänze die Vektoren x_i, $i = 1, \ldots, \varrho$ durch $n - \varrho$ weitere linear unabhängige Vektoren $x_i \in \mathbb{C}^n$, $i = \varrho + 1, \ldots, n$, zu einer Basis. Dann ist die Matrix $T = (x_1, \ldots, x_n)$ invertierbar. Für $i = 1, \ldots, \varrho$ gilt

$$T^{-1} A \underbrace{T e_i}_{= x_i} = T^{-1} A x_i = \lambda T^{-1} x_i = \lambda e_i.$$

Daraus folgt die Darstellung

$$T^{-1} A T = \left(\begin{array}{c|c} \lambda I & B \\ \hline 0 & C \end{array}\right)$$

mit Matrizen $B \in \mathbb{C}^{\varrho \times (n-\varrho)}$ und $C \in \mathbb{C}^{(n-\varrho)\times(n-\varrho)}$. Weiter folgt

$$\varphi(\mu) = \det(A - \mu I) = \det(T^{-1} A T - \mu I) = (\lambda - \mu)^{\varrho} \cdot \det(C - \mu I).$$

Somit ist λ mindestens ϱ-fache Nullstelle von φ. □

Im Beweis haben wir ausgenutzt, daß ähnliche Matrizen dieselben Eigenwerte besitzen.

Definition 1.0.10

(i) *Zwei Matrizen $A, B \in \mathbb{C}^{n\times n}$ heißen* **ähnlich**, *wenn es eine reguläre Matrix $T \in \mathbb{C}^{n\times n}$ gibt mit*

$$B = T^{-1} \cdot A \cdot T.$$

(ii) *A heißt* **diagonalisierbar**, *wenn A ähnlich ist zu einer Diagonalmatrix.*

(iii) *A heißt* **normal**, *wenn $A \cdot A^* = A^* \cdot A$ gilt, wobei $A^* := \bar{A}^\top$.*

(iv) *A heißt* **hermitesch**, *wenn $A^* = A$ gilt.*

(v) *A heißt* **unitär**, *wenn $A^* = A^{-1}$ gilt.*

Satz 1.0.11

Seien $A, B \in \mathbb{C}^{n\times n}$ gegeben.

(a) *Ist λ Eigenwert von A, so ist λ auch Eigenwert von $A^\top$ und $\bar{\lambda}$ ist Eigenwert von A^*.*

(b) *Seien A und B ähnlich mit $B = T^{-1}AT$. Dann besitzen A und B dieselben Eigenwerte.*

Ist x Eigenvektor von A zum Eigenwert λ, so ist $T^{-1}x$ Eigenvektor von B zum Eigenwert λ.

Ist y Eigenvektor von B zum Eigenwert λ, so ist Ty Eigenvektor von A zum Eigenwert λ.

Beweis:

(a) Folgt aus

$$\begin{aligned} \det(A-\lambda I) &= \det((A-\lambda I)^\top) = \det(A^\top - \lambda I), \\ \det(A^* - \bar{\lambda} I) &= \det((A-\lambda I)^*) = \det \overline{(A-\lambda I)^\top} = \overline{\det(A-\lambda I)}. \end{aligned}$$

(b) Seien A, B ähnlich, d.h. es gibt T invertierbar mit $B = T^{-1} \cdot A \cdot T$. Sei λ Eigenwert von A, d.h. $Ax = \lambda x$ für ein $x \neq 0$. Dann gilt für $y = T^{-1}x$:

$$By = BT^{-1}x = T^{-1}ATT^{-1}x = T^{-1}Ax = \lambda T^{-1}x = \lambda y.$$

Umgekehrt gilt für einen Eigenwert λ von B mit zugehörigem Eigenvektor y und $x = Ty$:

$$Ax = ATy = TBT^{-1}Ty = TBy = \lambda Ty = \lambda x.$$

□

Im folgenden überlegen wir uns, welche Matrizen diagonalisierbar sind und beginnen mit der Schur'schen Normalform.

Satz 1.0.12 (Schur'sche Normalform)
Sei $A \in \mathbb{C}^{n\times n}$. Dann gibt es eine unitäre Matrix T mit

$$T^{-1}AT = \Lambda + R := \begin{pmatrix} \lambda_1 & r_{12} & \dots & r_{1n} \\ & \ddots & \ddots & \vdots \\ & & \lambda_{n-1} & r_{n-1,n} \\ & & & \lambda_n \end{pmatrix}$$

Insbesondere enthält $\Lambda := diag(\lambda_1, \dots, \lambda_n)$ die Eigenwerte von A.

Beweis: Wir zeigen die Aussage per Induktion nach n.
Für $n = 1$ ist die Aussage offenbar richtig.
Sei die Aussage richtig für Matrizen der Dimension $n-1$.
Sei λ_1 Eigenwert von A zum Eigenvektor $x \neq 0$.
Wir konstruieren nun eine unitäre Householder-Matrix $H = I - \frac{2}{v^*v} v \cdot v^*$, um x auf ein Vielfaches des ersten Einheitsvektors $e_1 = (1, 0, \dots, 0)^\top \in \mathbb{R}^n$ abzubilden.
Mit $v = \frac{1}{\|x\|_2} x + \frac{x_1}{|x_1|} e_1$, wobei x_1 die erste Komponente von x bezeichnet und die Konvention $0/0 = 1$ im Fall $x_1 = 0$ beachtet wird, folgt

$$Hx = ke_1 \qquad \text{mit} \qquad k = -\frac{x_1}{|x_1|}\|x\|_2.$$

Wegen $H^{-1} = H = H^*$ gilt dann

$$H^{-1}AHe^1 = HA\frac{1}{k}x = \frac{1}{k}H\lambda_1 x = \lambda_1 e^1.$$

Daraus folgt die Darstellung

$$H^{-1}AH = \left(\begin{array}{c|c} \lambda_1 & a \\ \hline 0 & A_1 \end{array}\right) \qquad \text{mit} \qquad a^* \in \mathbb{C}^{n-1},\ A_1 \in \mathbb{C}^{(n-1)\times(n-1)}.$$

Nach Induktionsannahme existiert eine unitäre Matrix T_1, die A_1 auf obere Dreiecksgestalt transformiert. Definiere die unitäre Matrix

$$T = H\left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & T_1 \end{array}\right).$$

Dann ist

$$\begin{aligned} T^{-1}AT &= \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & T_1^{-1} \end{array}\right) H^{-1}AH \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & T_1 \end{array}\right) \\ &= \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & T_1^{-1} \end{array}\right) \left(\begin{array}{c|c} \lambda_1 & a \\ \hline 0 & A_1 \end{array}\right) \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & T_1 \end{array}\right) \\ &= \left(\begin{array}{c|c} \lambda_1 & a \\ \hline 0 & T_1^{-1}A_1 \end{array}\right) \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & T_1 \end{array}\right) \\ &= \left(\begin{array}{c|c} \lambda_1 & aT_1 \\ \hline 0 & T_1^{-1}A_1T_1 \end{array}\right) \end{aligned}$$

obere Dreiecksmatrix, welche nach Satz 1.0.11 (b) dieselben Eigenwerte wie A besitzt. □

Mit Hilfe der Schur'schen Normalform zeigen wir

Satz 1.0.13

(a) A ist genau dann normal, wenn es eine unitäre Matrix $T \in \mathbb{C}^{n\times n}$ gibt mit

$$T^{-1} \cdot A \cdot T = \Lambda := diag(\lambda_1, \ldots, \lambda_n) \in \mathbb{C}^{n\times n}.$$

(b) A ist genau dann hermitesch, wenn es eine unitäre Matrix $T \in \mathbb{C}^{n\times n}$ gibt mit

$$T^{-1} \cdot A \cdot T = \Lambda := diag(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^{n\times n}.$$

Beweis:

(a) Es gebe eine unitäre Matrix T mit $T^{-1} \cdot A \cdot T = \Lambda \in \mathbb{C}^{n\times n}$. Dann gilt

$$\begin{aligned} A^*A &= (T\Lambda T^{-1})^*T\Lambda T^{-1} = T\bar{\Lambda}T^{-1}T\Lambda T^{-1} = T\bar{\Lambda}\Lambda T^{-1} \\ AA^* &= T\Lambda T^{-1}(T\Lambda T^{-1})^* = T\Lambda T^{-1}T\bar{\Lambda}T^{-1} = T\Lambda\bar{\Lambda}T^{-1} \end{aligned}$$

Wegen $\bar{\Lambda}\Lambda = \Lambda\bar{\Lambda}$ folgt Gleichheit, also ist A normal.

Sei nun A normal. Nach der Schur'schen Normalform gibt es eine unitäre Matrix T mit $T^{-1}AT = \Lambda + R$. Da A normal ist, folgt

$$\begin{aligned} (\Lambda + R)^*(\Lambda + R) &= T^{-1}A^*TT^{-1}AT \\ &= T^{-1}A^*AT \\ &= T^{-1}AA^*T \\ &= T^{-1}ATT^{-1}A^*T \\ &= (\Lambda + R)(\Lambda + R)^*. \end{aligned}$$

Für das (1,1)-Element von $(\Lambda + R)^*(\Lambda + R) = (\Lambda + R)(\Lambda + R)^*$ ergibt sich

$$\bar{\lambda}_1\lambda_1 = |\lambda_1|^2 = |\lambda_1|^2 + \sum_{j=2}^{n} |r_{1j}|^2,$$

woraus $r_{1,j} = 0$ für alle $2 \leq j \leq n$ folgt. Durch sukzessiven Vergleich der Einträge an den Positionen (2,2), (3,3), ... zeigt man zeilenweise, daß alle Einträge von R verschwinden.

(b) Es gebe eine unitäre Matrix T mit $T^{-1} \cdot A \cdot T = \Lambda \in \mathbb{R}^{n\times n}$. Dann gilt

$$A^* = (T\Lambda T^{-1})^* = (T^{-1})^*\Lambda^* T^* = T\Lambda T^{-1} = A.$$

Sei nun A hermitesch. Dann ist A auch normal. Die Behauptung folgt mit (a) aus

$$T \cdot \Lambda \cdot T^{-1} = A = A^* = (T^{-1})^* \cdot \bar{\Lambda} \cdot T^* = T \cdot \bar{\Lambda} \cdot T^{-1}.$$

Folglich gilt $\Lambda = \bar{\Lambda}$ und somit $\Lambda \in \mathbb{R}^{n\times n}$.

□

Die Konsequenz aus Satz 1.0.13 ist, daß normale (bzw. hermitesche) Matrizen ein System von n linear unabhängigen, orthogonalen Eigenvektoren besitzen. Diese sind gerade durch die Spaltenvektoren t^i, $i = 1, \ldots, n$, von T gegeben, da aus der Diagonalisierbarkeit die Beziehung

$$A \cdot T = T \cdot \Lambda \qquad \Leftrightarrow \qquad At^i = \lambda_i t^i, \quad i = 1, \ldots, n,$$

folgt. Die Eigenwerte λ_i sind für normale Matrizen i.a. komplexwertig und für hermitesche Matrizen nach Satz 1.0.13 (b) sogar **stets reellwertig**.
Es gibt auch Matrizen, die nicht diagonalisierbar sind, z.B.

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

Immerhin kann man auch für solche Matrizen noch eine weitere Normalform erreichen. Es gilt:

Satz 1.0.14 (Jordan'sche Normalform)
Sei $A \in \mathbb{C}^{n\times n}$ und $\lambda_1, \ldots, \lambda_k$ ihre verschiedenen Eigenwerte. λ_i sei eine σ_i-fache Nullstelle des charakteristischen Polynoms mit der geometrischen Vielfachheit ϱ_i, $i = 1, \ldots, k$. Zu jedem λ_i existieren dann ϱ_i eindeutig bestimmte Zahlen

$$\nu_n^{(i)} \geq \nu_{n-1}^{(i)} \geq \ldots \geq \nu_{n-\varrho_i+1}^{(i)} \in \mathbb{N}$$

mit

$$\sum_{j=n-\varrho_i+1}^{n} \nu_j^{(i)} = \sigma_i, \qquad i = 1, \dots, k,$$

sowie eine invertierbare Matrix T mit $J = T^{-1}AT$, wobei

$$J = \left(\begin{array}{ccc|c|cccc} J_{\nu_n^{(1)}}(\lambda_1) & & & & & & & \\ & \ddots & & & & & & \\ & & J_{\nu_{n-\varrho_1+1}^{(1)}}(\lambda_1) & & & & & \\ \hline & & & \ddots & & & & \\ \hline & & & & J_{\nu_n^{(k)}}(\lambda_k) & & & \\ & & & & & \ddots & & \\ & & & & & & J_{\nu_{n-\varrho_k+1}^{(k)}}(\lambda_k) \end{array}\right)$$

die **Jordanmatrix** *und*

$$J_{\nu_j^{(i)}}(\lambda_i) = \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix} \in \mathbb{C}^{\nu_j^{(i)} \times \nu_j^{(i)}}$$

die **Jordankästchen** *bezeichnen.*

Beweis: Lineare Algebra Vorlesung □

Aus der Jordan'schen Normalform kann man insbesondere ablesen, daß A genau dann diagonalisierbar ist, wenn geometrische und algebraische Vielfachheit für jeden Eigenwert gleich sind.

Bemerkung 1.0.15

Für die Spalten $t^1, \dots, t^{\nu_n^{(1)}}$ von T folgt aus $T^{-1}(A - \lambda_1 I)T = J - \lambda_1 I$ sofort

$$(A - \lambda_1 I)t^m = t^{m-1}, \qquad m = \nu_n^{(1)}, \nu_n^{(1)} - 1, \dots, 2,$$

und

$$(A - \lambda_1 I)t^1 = 0.$$

Die Spalten $t^{\nu_n^{(1)}}, \dots, t^1$ bilden eine **Hauptvektorkette** *zum Eigenwert λ_1, wobei t^1 ein Eigenvektor zum Eigenwert λ_1 ist. Die Vektoren $t^{\nu_n^{(1)}}, \dots, t^2$ heißen* **Hauptvektoren**. *Analoge Bezeichnungen gelten für die übrigen Jordankästchen, so daß die Spalten von T aus Eigen- und Hauptvektoren von A bestehen.*

Die Schur'sche und Jordan'sche Normalform sind primär von theoretischem Interesse, da man die Eigenwerte zur Bestimmung der Normalformen bereits kennen muß. Zur Berechnung von Eigenwerten müssen andere Verfahren verwendet werden. Ein Ansatz besteht

darin, solange Ähnlichkeitstransformation von A durchzuführen bis eine ähnliche Matrix mit einfacher Struktur entsteht, für die die Eigenwerte bestimmt werden können.
In diesem Kapitel werden folgende Fragestellungen untersucht:

- Abschätzung von Eigenwerten
- Berechnung des betragsmäßig größten oder kleinsten Eigenwerts einer Matrix (ohne zugehörige Eigenvektoren)
- Berechnung aller Eigenwerte einer Matrix (ohne zugehörige Eigenvektoren)
- Berechnung des betragsmäßig größten oder kleinsten Eigenwerts einer Matrix und eines zugehörigen Eigenvektors
- Berechnung aller Eigenwerte einer Matrix und zugehörige Eigenvektoren

1.1 Eigenwertabschätzungen

Ziel dieses Abschnitts ist die Abschätzung von Eigenwerten allein mit Hilfe der Matrix $A \in \mathbb{C}^{n\times n}$. Eine erste Abschätzung erhalten wir durch jede Matrixnorm, wie der folgende Satz zeigt.

Satz 1.1.1

Es bezeichne $\rho(A)$ den Spektralradius von A. Dann gilt $|\lambda| \le \rho(A) \le \|A\|_M$ für jede Matrixnorm $\|\cdot\|_M$, die von einer Vektornorm $\|\cdot\|_V$ induziert wird, und jeden Eigenvektor λ von A.

Beweis: Sei λ Eigenwert von A und $z \neq 0$ zugehöriger Eigenvektor. Die Behauptung folgt aus

$$\|A\|_M = \sup_{\|x\|_V \neq 0} \frac{\|Ax\|_V}{\|x\|_V} \ge \frac{|\lambda|\|z\|_V}{\|z\|_V} = |\lambda|.$$

□

Mit Hilfe einer Matrixnorm kann man also den betragsmäßig größten Eigenwert einer Matrix nach oben abschätzen. Eine differenziertere Abschätzung liefert folgender Satz.

Satz 1.1.2 (Bauer und Fike)

Für eine beliebige Matrix $B \in \mathbb{C}^{n\times n}$ gilt für alle Eigenwerte λ von $A \in \mathbb{C}^{n\times n}$, die nicht gleichzeitig Eigenwert von B sind, die Abschätzung

$$1 \le \|(\lambda I - B)^{-1}(A - B)\|_M \le \|(\lambda I - B)^{-1}\|_M \cdot \|A - B\|_M,$$

wobei $\|\cdot\|_M$ eine durch eine Vektornorm $\|\cdot\|_V$ induzierte Matrixnorm sei.

Beweis: Sei x Eigenvektor zum Eigenwert λ von A, wobei λ kein Eigenwert von B sei. Dann ist $\lambda I - B$ invertierbar und aus der Identität

$$(A - B)x = (\lambda I - B)x$$

folgt

$$(\lambda I - B)^{-1}(A - B)x = x$$

und weiter

$$\|x\|_V \quad \leq \quad \|(\lambda I - B)^{-1}(A - B)\|_M \cdot \|x\|_V \leq \|(\lambda I - B)^{-1}\|_M \cdot \|A - B\|_M \cdot \|x\|_V.$$

Division durch $\|x\|_V \neq 0$ liefert die Behauptung. □

Satz 1.1.3 (Gerschgorin)
Sei $A \in \mathbb{C}^{n\times n}$ gegeben. Definiere Radien r_i gemäß

$$r_i = \sum_{j=1, j\neq i}^{n} |a_{ij}|, \quad i = 1, \ldots, n,$$

*und Kreisscheiben (**Gerschgorin-Kreise**) gemäß*

$$K_i := \{z \in \mathbb{C} \mid |z - a_{ii}| \leq r_i\}, \quad i = 1, \ldots, n.$$

Dann gelten folgende Aussagen.

(a) Für jeden Eigenwert λ von A gilt

$$\lambda \in \bigcup_{i=1}^{n} K_i.$$

(b) Sei U_1 die Vereinigung von m dieser Kreissscheiben und sei U_2 die Vereinigung der verbleibenden $n-m$ Kreissscheiben. Gilt $U_1 \cap U_2 = \emptyset$, so enthält U_1 m Eigenwerte von A und U_2 $n-m$ Eigenwerte von A (Eigenwerte und Kreisscheiben sind entsprechend ihrer Vielfachheit zu zählen.).

Beweis:

(a) Ist $\lambda = a_{ii}$ für ein $1 \leq i \leq n$, so ist die Behauptung offenbar richtig. Andernfalls wähle in Satz 1.1.2 $B = diag(a_{11}, \ldots, a_{nn})$. Dann folgt für die Zeilensummennorm

$$1 \leq \|(\lambda I - B)^{-1}(A - B)\|_\infty = \max_{1\leq i\leq n} \frac{1}{|\lambda - a_{ii}|} \sum_{j=1, j\neq i}^{n} |a_{ij}| = \max_{1\leq i\leq n} \frac{r_i}{|\lambda - a_{ii}|}.$$

Dann gibt es einen Index $1 \leq i_o \leq n$ mit $|\lambda - a_{i_0 i_0}| \leq r_{i_0}$ und somit ist $\lambda \in K_{i_0} \subseteq \bigcup_{i=1}^{n} K_i$.

(b) Seien $D = diag(a_{11}, \ldots, a_{nn})$ und $A =: D + R$. Definiere für $t \in [0,1]$ $A_t := D + tR$. Für $t = 1$ gilt $A_1 = A$ und A_1 besitzt dieselben Eigenwerte wie A.

Für $t = 0$ gilt $A_0 = D$ und A_0 besitzt die Eigenwerte a_{ii}, $i = 1, \ldots, n$, wovon genau m in U_1 und $n - m$ in U_2 liegen, falls $U_1 \cap U_2 = \emptyset$ gilt.

Nach (a) liegen die Eigenwerte von A_t für $0 < t \leq 1$ wegen

$$\{z \in \mathbb{C} \mid |z - a_{ii}| \leq tr_i\} \subseteq K_i, \qquad i = 1, \ldots, n,$$

ebenfalls entweder in U_1 oder U_2.

Da die Nullstellen des charakteristischen Polynoms stetig von den Einträgen in der Matrix abhängen, hängen die Eigenwerte von A_t stetig von t ab. Damit lassen sich für $j = 1, \ldots, n$ stetige Abbildungen $\lambda_j : [0,1] \to \mathbb{C}$ definieren, wobei $\lambda_j(t)$ Eigenwert von A_t für $0 \leq t \leq 1$ ist.

O.B.d.A. seien $\lambda_i(0) := a_{ii} \in U_1$ für $i = 1, \ldots, m$. Für $j \in \{1, \ldots, m\}$ seien

$$\begin{aligned} B_1 &= \{t \in [0,1] \mid \lambda_j(t) \in U_1\}, \\ B_2 &= \{t \in [0,1] \mid \lambda_j(t) \notin U_1\} = \{t \in [0,1] \mid \lambda_j(t) \in U_2\}. \end{aligned}$$

Dann folgt $B_1 \neq \emptyset$, $B_1 = \bar{B}_1$, $B_2 = \bar{B}_2$, $B_1 \cap B_2 = \emptyset$ und $B_1 \cup B_2 = [0,1]$, da U_1 und U_2 abgeschlossen und die Abbildungen $\lambda_j(\cdot)$ stetig sind (das Urbild einer abgeschlossenen Menge unter einer stetigen Abbildung ist abgeschlossen). Da $[0,1]$ zusammenhängend ist, muß $B_2 = \emptyset$ gelten. Also ist $\lambda_j(1) \in U_1$ für $j = 1, \ldots, m$. Analog zeigt man, daß $\lambda_j(1) \in U_2$ für $j = m+1, \ldots, n$ gilt.

□

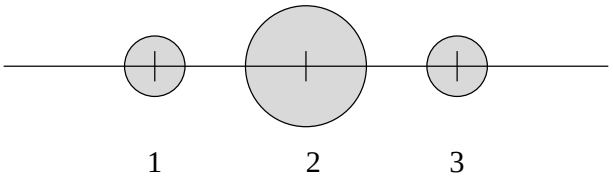
Mit Hilfe des Satzes von Gerschgorin erhält man insbesondere dann einen guten Überblick über die Lage der Eigenwerte, wenn sämtliche Kreisscheiben paarweise disjunkt sind, da dann in jeder Kreisscheibe genau ein Eigenwert liegt. Darüber hinaus können durch Anwendung des Satzes auf $A^\top$ (A und $A^\top$ besitzen dieselben Eigenwerte!) mitunter bessere Abschätzungen für einige Eigenwerte erreicht werden. Ein Ansatz zum Erreichen besserer Abschätzungen besteht darin, eine Ähnlichkeitstransformation $D^{-1}AD$ mit einer geeignet gewählten Diagonalmatrix D vor Anwendung des Satzes von Gerschgorin durchzuführen. D sollte idealerweise so gewählt werden, daß die Radien kleiner werden, was häufig aber nur für einige Radien erreicht werden kann, wobei sich die übrigen vergrößern.

Beispiel 1.1.4

Betrachte

$$A = \begin{pmatrix} 1 & 0.1 & -0.1 \\ 0 & 2 & 0.4 \\ -0.2 & 0 & 3 \end{pmatrix}.$$

Die Gerschgorin-Kreise

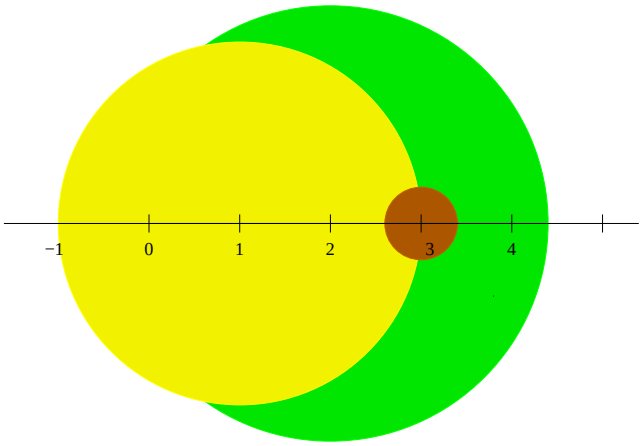


$$\begin{aligned} K_1 &= \{z \in \mathbb{C} \mid |z-1| \leq 0.2\}, \\ K_2 &= \{z \in \mathbb{C} \mid |z-2| \leq 0.4\}, \\ K_3 &= \{z \in \mathbb{C} \mid |z-3| \leq 0.2\} \end{aligned}$$

sind paarweise disjunkt, d.h. in jedem Kreis liegt genau ein Eigenwert von A.
Für die symmetrische Matrix

$$B = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 2 & 0.4 \\ 0 & 0.4 & 3 \end{pmatrix}.$$

überschneiden sich die Gerschgorin-Kreise



$$\begin{aligned} K_1 &= \{z \in \mathbb{C} \mid |z-1| \leq 2\}, \\ K_2 &= \{z \in \mathbb{C} \mid |z-2| \leq 2.4\}, \\ K_3 &= \{z \in \mathbb{C} \mid |z-3| \leq 0.4\}, \end{aligned}$$

so daß man aus dem Satz von Gerschgorin nur die Abschätzung

$$-1 \leq \lambda \leq 4.4$$

für die Eigenwerte λ von B erhält. Beachte, daß die Eigenwerte von B reell sind, da B symmetrisch ist.

Eine weitere Abschätzung basiert auf dem Wertebereich von A.

Definition 1.1.5 **(Wertebereich)**

Die Menge

$$W(A) = \left\{ \frac{x^*Ax}{x^*x} \;\middle|\; x \neq 0 \right\}$$

der sogenannten Rayleigh-Quotienten heißt **Wertebereich von** A.

Für jeden Eigenwert λ von A, gilt offenbar $\lambda \in W(A)$. Darüber hinaus gilt

Satz 1.1.6

(a) Sei $A \in \mathbb{C}^{n\times n}$ normal. Dann ist $W(A)$ die konvexe Hülle der Eigenwerte von A.

(b) Sei $A \in \mathbb{C}^{n\times n}$ hermitesch. Dann gilt $W(A) = [\lambda_1, \lambda_n]$, wobei die reellen (!) Eigenwerte von A geordnet seien gemäß $\lambda_1 \leq \ldots \leq \lambda_{n-1} \leq \lambda_n$. Insbesondere gilt

$$\lambda_1 = \min_{0\neq x\in\mathbb{C}^n} \frac{x^*Ax}{x^*x}, \qquad \lambda_n = \max_{0\neq x\in\mathbb{C}^n} \frac{x^*Ax}{x^*x}. \tag{1.1}$$

Beweis: Teil (b) folgt sofort aus (a). Wir zeigen daher nur (a). Ist A normal, so ist A Satz 1.0.13 diagonalisierbar mit $T^*AT = \Lambda = diag(\lambda_1, \ldots, \lambda_n)$, wobei Λ die Eigenwerte von A enthält und T unitär ist. Es folgt

$$\begin{aligned}
W(A) &= \left\{ \frac{x^*Ax}{x^*x} \;\middle|\; x \neq 0 \right\} \\
&= \left\{ \frac{x^*T\Lambda T^*x}{x^*TT^*x} \;\middle|\; x \neq 0 \right\} \\
&= \left\{ \frac{y^*\Lambda y}{y^*y} \;\middle|\; y \neq 0 \right\} \\
&= \left\{ \frac{\sum_{j=1}^n \lambda_j |y_j|^2}{\sum_{j=1}^n |y_j|^2} \;\middle|\; y \neq 0 \right\} \\
&= \left\{ \sum_{j=1}^n \lambda_j \alpha_j \;\middle|\; \alpha_j \geq 0,\ j = 1, \ldots, n,\ \sum_{j=1}^n \alpha_j = 1 \right\}.
\end{aligned}$$

□

Satz 1.1.6 stellt einen Zusammenhang mit der Optimierung her. Demnach können der kleinste und größte Eigenwert einer hermiteschen Matrix im Prinzip durch Lösen der Optimierungsprobleme (1.1) bestimmt werden. Der folgende Satz erweitert dieses Prinzip auf sämtliche Eigenwerte.

Satz 1.1.7 **(Rayleigh'sches Variationsprinzip)**

Sei $A \in \mathbb{C}^{n\times n}$ hermitesch mit den Eigenwerten $\lambda_1 \leq \ldots \leq \lambda_n$ und zugehörigen orthonor-

malen Eigenvektoren $v^1, \dots, v^n$. Dann gilt für $j = 1, \dots, n$:

$$\begin{aligned} \lambda_j &= \min \left\{ \frac{x^* A x}{x^* x} \;\middle|\; x \neq 0,\ x^* v^k = 0,\ k = 1, \dots, j-1 \right\} \\ &= \max \left\{ \frac{x^* A x}{x^* x} \;\middle|\; x \neq 0,\ x^* v^k = 0,\ k = j+1, \dots, n \right\}. \end{aligned}$$

Beweis: Seien j und $0 \neq x$ fest gewählt mit $x^* v^k = 0$, $k = 1, \dots, j-1$.
x läßt sich als Linearkombination der Eigenvektoren darstellen:

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

Es folgt

$$\frac{x^* A x}{x^* x} = \frac{x^* \sum_{k=1}^{n} \alpha_k \lambda_k v^k}{x^* \sum_{k=1}^{n} \alpha_k v^k} = \frac{\sum_{k=j}^{n} |\alpha_k|^2 \lambda_k}{\sum_{k=j}^{n} |\alpha_k|^2} \geq \lambda_j.$$

Für $x = v^j$ folgt speziell $(v^j)^* v^k = 0$ für $k = 1, \dots, j-1$, sowie

$$\frac{(v^j)^* A v^j}{(v^j)^* v^j} = \lambda_j.$$

Dies zeigt die erste Gleichung. Die zweite läßt sich analog zeigen. □

Das folgende Variationsprinzip kommt ohne die Verwendung der Eigenvektoren aus.

Satz 1.1.8 (Ritz-Poincaré'sches Variationsprinzip)
Sei $A \in \mathbb{C}^{n \times n}$ hermitesch mit den Eigenwerten $\lambda_1 \leq \dots \leq \lambda_n$. Dann gilt für $j = 1, \dots, n$:

$$\begin{aligned} \lambda_j &= \min_{V \subseteq \mathbb{C}^n, \dim V = j} \max_{0 \neq x \in V} \frac{x^* A x}{x^* x} \\ &= \max_{V \subseteq \mathbb{C}^n, \dim V = n-j+1} \min_{0 \neq x \in V} \frac{x^* A x}{x^* x}. \end{aligned}$$

Beweis: Sei $v^1, \dots, v^n$ ein Orthonormalsystem aus Eigenvektoren zu $\lambda_1, \dots, \lambda_n$. Sei $\hat{V}$ ein j-dimensionaler Unterraum des $\mathbb{C}^n$ mit Basis $y^1, \dots, y^j$. Nach dem Projektionssatz (vgl. Numerik I) existiert für $k = 1, \dots, j$ zu y^k ein $\hat{y}^k \in span\{v^1, \dots, v^{j-1}\}$ mit

$$\langle y^k - \hat{y}^k, v \rangle = 0 \qquad \forall v \in span\{v^1, \dots, v^{j-1}\}.$$

Wegen $\dim(span\{v^1, \dots, v^{j-1}\}) = j-1$ gibt es $0 \neq (\alpha_1, \dots, \alpha_j)^\top \in \mathbb{C}^j$ mit

$$\sum_{k=1}^{j} \alpha_k \hat{y}^k = 0.$$

Definiere

$$\hat{x} = \sum_{k=1}^{j} \alpha_k y^k \in \hat{V}.$$

Da nicht alle α_k verschwinden und y^k, $k = 1, \dots, j$, Basis ist, gilt $\hat{x} \neq 0$ und

$$\langle \hat{x}, v \rangle = 0 \qquad \forall v \in span\{v^1, \dots, v^{j-1}\}.$$

Nach Satz 1.1.7 ist

$$\lambda_j \leq \frac{\hat{x}^* A \hat{x}}{\hat{x}^* \hat{x}} \leq \max_{0 \neq x \in \hat{V}} \frac{x^* A x}{x^* x}.$$

Da $\hat{V}$ j-dimensional ist, ansonsten aber beliebig gewählt war, folgt

$$\lambda_j \leq \min_{V \subseteq \mathbb{C}^n, \dim V = j} \max_{0 \neq x \in V} \frac{x^* A x}{x^* x}. \tag{1.2}$$

Wähle nun speziell $\hat{V} = \{v^1, \dots, v^j\}$. Für jedes

$$0 \neq x = \sum_{k=1}^{j} \alpha_k v^k \in \hat{V}$$

gilt dann

$$\frac{x^* A x}{x^* x} = \frac{\sum_{k=1}^{j} |\alpha_k|^2 \lambda_k}{\sum_{k=1}^{j} |\alpha_k|^2} \leq \lambda_j.$$

Insbesondere für $x = v^j$ folgt

$$\frac{(v^j)^* A v^j}{(v^j)^* v^j} = \lambda_j.$$

Also gilt

$$\lambda_j = \max_{0 \neq x \in \hat{V}} \frac{x^* A x}{x^* x} \geq \min_{V \subseteq \mathbb{C}^n, \dim V = j} \max_{0 \neq x \in V} \frac{x^* A x}{x^* x}.$$

Zusammen mit (1.2) folgt die erste Behauptung. Die zweite läßt sich analog zeigen. □

Beispiel 1.1.9 (Übungsaufgabe)

Sei A hermitesch und λ_{min} der kleinste und λ_{max} der größte Eigenwert von A. Zeige: Für alle x gilt

$$\lambda_{min} \|x\|_2^2 \leq x^\top A x \leq \lambda_{max} \|x\|_2^2.$$

Wann gilt Gleichheit?

1.2 Kondition des Eigenwertproblems

Mit Hilfe des Satzes von Bauer und Fike läßt sich auch eine Aussage über die Kondition des Eigenwertproblems treffen, wobei B als gestörte Matrix von A interpretiert wird. Insbesondere interessieren wir uns für die Abhängigkeit eines Eigenwerts von Störungen in A. Dazu sei $A \in \mathbb{C}^{n\times n}$ gegeben und $B := A + \Delta A$ eine gestörte Matrix. Sei $A + \Delta A$ diagonalisierbar gemäß

$$A + \Delta A = T \cdot \Lambda \cdot T^{-1}, \qquad \Lambda = diag(\lambda_1, \dots, \lambda_n),$$

wobei λ_i, $i = 1, \dots, n$, die Eigenwerte von $A + \Delta A$ bezeichnen.
Wählt man in Satz 1.1.2 eine Vektornorm $\|\cdot\|_V$, so daß die induzierte Matrixnorm $\|\cdot\|_M$

$$D = diag(d_1, \dots, d_n) \qquad \Rightarrow \qquad \|D\|_M = \max_{1\le i\le n} |d_i| \tag{1.3}$$

erfüllt, also etwa die euklidische Norm $\|\cdot\|_2$ oder die Maximumsnorm $\|\cdot\|_\infty$, so folgt unter der Annahme, daß λ kein Eigenwert von $A + \Delta A$ ist,

$$\begin{aligned}
\|(\lambda I - (A + \Delta A))^{-1}\|_M &= \|T(\lambda I - \Lambda)^{-1}T^{-1}\|_M \\
&\le \|(\lambda I - \Lambda)^{-1}\|_M \cdot \|T\|_M \cdot \|T^{-1}\|_M \\
&= \max_{i=1,\dots,n} \frac{1}{|\lambda - \lambda_i|}\kappa_M(T) \\
&= \frac{1}{\min\limits_{i=1,\dots,n} |\lambda - \lambda_i|}\kappa_M(T).
\end{aligned}$$

Zusammen mit Satz 1.1.2 haben wir damit bewiesen:

Satz 1.2.1 **(Kondition des Eigenwertproblems)**
Seien $\|\cdot\|_M$ eine durch eine Vektornorm induzierte Matrixnorm mit (1.3), $A \in \mathbb{C}^{n\times n}$ und $A + \Delta A$ diagonalisierbar mit $A + \Delta A = T \cdot \Lambda \cdot T^{-1}$, $\Lambda = diag(\lambda_1, \dots, \lambda_n)$.
Dann gibt es zu jedem Eigenwert λ von A einen Eigenwert λ_i von $A + \Delta A$ mit

$$|\lambda - \lambda_i| \le \kappa_M(T)\|\Delta A\|_M.$$

Die Fehlerverstärkung hängt also maßgeblich von der Kondition der Matrix T ab. Die Spalten von T sind aber gerade die Eigenvektoren von $A + \Delta A$. Für normale Matrizen kann T nach Satz 1.0.13 unitär gewählt werden, was $\kappa_2(T) = 1$ impliziert. Damit folgt

Satz 1.2.2 **(Kondition des Eigenwertproblems für normale Matrizen)**
Ist $A + \Delta A$ normal, so gilt unter den Voraussetzungen des Satzes 1.2.1 sogar

$$|\lambda - \lambda_i| \le \|\Delta A\|_2.$$

Insbesondere ist das Eigenwertproblem für normale Matrizen stets gut konditioniert.

Für allgemeine Matrizen bekommt man nur die stetige Abhängigkeit von Eigenwerten von den Störungen. Der relative Fehler kann dabei beliebig schlecht werden.

Beispiel 1.2.3

Jedem Polynom vom Grad n der Form

$$p(\mu) = \mu^n + \alpha_{n-1}\mu^{n-1} + \ldots + \alpha_1\mu + \alpha_0$$

kann die sogenannte **Frobenius-Begleitmatrix**

$$F = \begin{pmatrix} 0 & \cdots & 0 & -\alpha_0 \\ 1 & \ddots & \vdots & \vdots \\ & \ddots & 0 & -\alpha_{n-2} \\ & & 1 & -\alpha_{n-1} \end{pmatrix}$$

zugeordnet werden. Durch Entwicklung nach der letzten Spalte zeigt man

$$p(\mu) = (-1)^n \det(F - \mu I),$$

so daß p bis auf den Faktor $(-1)^n$ mit dem charakteristischen Polynom von F übereinstimmt und insbesondere dieselben Nullstellen besitzt.
Betrachte nun für $a \neq 0$ und $\varepsilon > 0$ die Polynome

$$p_0(\mu) = (\mu - a)^n, \qquad p_\varepsilon(\mu) = (\mu - a)^n - \varepsilon,$$

welche die Nullstellen $\lambda_0 = a$ bzw.

$$\lambda_k = a + \sqrt[n]{\varepsilon}\exp(i2\pi k/n), \qquad k = 1, \ldots, n,$$

besitzen. Die zugehörigen Frobenius-Begleitmatrizen F_0 bzw. F_ε unterscheiden sich nur minimal:

$$\Delta F = F_\varepsilon - F_o = \begin{pmatrix} 0 & \cdots & 0 & \varepsilon \\ 0 & \ddots & \vdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}, \qquad \|\Delta F\|_2 = \varepsilon.$$

Für die Differenz der zugehörigen Eigenwerte gilt jedoch

$$|\lambda_k - \lambda_o| = \sqrt[n]{\varepsilon}, \qquad k = 1, \ldots, n.$$

Die Eigenwerte hängen immer noch stetig von der Störung ab, für großes n ist diese Konvergenz jedoch sehr langsam. Betrachtet man den relativen Fehler, so folgt

$$\frac{|\lambda_k - \lambda_o|}{|\lambda_0|} = \frac{\sqrt[n]{\varepsilon}}{|\lambda_0|} = C_\varepsilon \frac{\|\Delta F\|_2}{\|F_0\|_2}, \qquad C_\varepsilon = \frac{\|F_0\|_2}{|a|}\frac{\sqrt[n]{\varepsilon}}{\varepsilon}.$$

Für $\varepsilon \to 0$ gilt $C_\varepsilon \to \infty$. Das Eigenwertproblem ist also i.a. nicht gut konditioniert.

1.3 Singulärwertzerlegung und Pseudoinverse

Wir interessieren uns hier für sogenannte Minimalnormlösungen des Ausgleichsproblems

$$\min_{x\in\mathbb{C}^n} \quad \frac{1}{2}\|Ax-b\|_2^2$$

mit $A \in \mathbb{C}^{m\times n}$ und $b \in \mathbb{C}^m$. In Numerik I haben wir gesehen, daß das Ausgleichsproblem genau dann eine eindeutig bestimmte Lösung besitzt, wenn $Rang(A) = n$ gilt. Die Lösung des Ausgleichsproblems ist charakterisiert durch die Gauss'schen Normalgleichungen

$$A^*A\hat{x} = A^*b.$$

Im Falle der eindeutigen Lösbarkeit ist die Lösung gegeben durch

$$\hat{x} = (A^*A)^{-1}A^*b.$$

Ist die Rangbedingung hingegen nicht erfüllt, so gibt es i.a. mehrere Lösungen des Ausgleichsproblems, die wir in der Menge

$$M := \{x \in \mathbb{C}^n \mid \|Ax-b\|_2 = \hat{f}\}$$

mit

$$\hat{f} := \inf_{x\in\mathbb{C}^n} \|Ax-b\|_2$$

zusammenfassen. Im folgenden sind wir an solchen Lösungen mit minimaler euklidischer Norm interessiert, d.h. an sogenannten Minimalnormlösungen.

Definition 1.3.1 **(Minimalnormlösung)**
$\hat{x} \in M$ *heißt* **Minimalnormlösung** *des Ausgleichsproblems, wenn*

$$\|\hat{x}\|_2 \le \|x\|_2 \qquad \forall x \in M$$

gilt.

Die Funktion $\|\cdot\|_2^2$ ist streng konvex, da die Hessematrix gerade $2I$ beträgt, und die Menge M ist konvex, denn für $x, y \in M$ und $0 \le \lambda \le 1$ folgt

$$\|A(\lambda x + (1-\lambda)y) - b\|_2 \le \lambda\|Ax-b\|_2 + (1-\lambda)\|Ay-b\|_2 = \hat{f}.$$

Nach Definition von $\hat{f}$ muß Gleichheit gelten, was die Konvexität von M impliziert. Die Bestimmung einer Minimalnormlösung ist also gleichbedeutend mit der Lösung des konvexen Optimierungsproblems

$$\min_{x\in\mathbb{C}^n} \qquad \|x\|_2^2 \qquad \text{unter} \qquad x \in M.$$

Da $\|\cdot\|_2^2$ streng konvex ist, ist die Minimalnormlösung eindeutig.

Unser Ziel ist es, die Minimalnormlösung $\hat{x}$ des Ausgleichsproblems mit Hilfe der sogenannten Pseudoinversen von A als

$$\hat{x} = A^+ b$$

darzustellen. Diese Gleichung besitzt formale Ähnlichkeit mit der Lösung eines linearen Gleichungssystems $x = A^{-1}b$, falls A invertierbar ist. Die Pseudoinverse von A spielt also eine zur Inversen analoge Rolle im Falle unter- oder überbestimmter Gleichungssysteme. Formal definieren wir die Pseudoinverse einer Matrix A wie folgt.

Definition 1.3.2 **(Pseudoinverse (Moore-Penrose Inverse))**
Sei $A \in \mathbb{C}^{m\times n}$ gegeben. Die Abbildung $A^+ : \mathbb{C}^m \to \mathbb{C}^n$, die jedem $b \in \mathbb{C}^m$ die Minimalnormlösung $\hat{x} \in M$ gemäß $\hat{x} = A^+ b$ zuordnet, heißt **Pseudoinverse (oder Moore-Penrose Inverse) von** *A.*

Beispiel 1.3.3
Betrachte das lineare Ausgleichsproblem

$$\min_{x\in\mathbb{C}^n} \quad \frac{1}{2}\|Ax - b\|_2^2$$

mit

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, \qquad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Gauss'sche Normalgleichung: $A^*Ax = A^*b$ bzw.

$$\begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 + b_2 \\ 0 \end{pmatrix}.$$

Die allgemeine Lösung der Normalgleichung lautet also $x_1 = (b_1 + b_2)/2$, $x_2 \in \mathbb{R}$. Die Minimalnormlösung ist folglich gegeben durch

$$\begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} \frac{b_1+b_2}{2} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}.$$

Damit lautet die Pseudoinverse von A

$$A^+ = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Wir werden die Pseudoinverse mit Hilfe der Singulärwertzerlegung von A berechnen. Dazu bemerken wir, daß die Matrix A^*A hermitesch und positiv semidefinit ist und somit nichtnegative reelle Eigenwerte besitzt.

Definition 1.3.4 **(singuläre Werte)**
Seien $A \in \mathbb{C}^{m\times n}$ und $\lambda_i \geq 0$, $i = 1,\dots,n$, die Eigenwerte von A^*A. Dann heißen die nichtnegativen Zahlen

$$\sigma_i = \sqrt{\lambda_i}, \qquad i = 1,\dots,n,$$

die **singulären Werte von** *A.*

Nach Teil (b) in Satz 1.0.13 ist A^*A diagonalisierbar und es gibt eine unitäre Matrix $T \in \mathbb{C}^{n\times n}$ mit

$$A^*AT = T\Lambda, \tag{1.4}$$

wobei die Diagonalmatrix $\Lambda = diag(\lambda_1,\dots,\lambda_n) \geq 0$ die nichtnegativen Eigenwerte von A^*A enthält.
O.B.d.A. seien die Eigenwerte und die Spalten von T so sortiert, daß

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0, \quad \text{und} \quad \sigma_{r+1} = \dots = \sigma_n = 0$$

für $r = Rang(A) = Rang(A^*A)$ gilt.
Aus (1.4) folgt mit $AT = (At^1,\dots,At^n)$ die Beziehung

$$AA^*AT = AT\Lambda. \tag{1.5}$$

Damit ist jeder Vektor

$$s^i = \frac{1}{\sigma_i}At^i, \qquad i = 1,\dots,r,$$

Eigenvektor von AA^*. Für $1 \leq i,k \leq r$ sind diese Vektoren wegen

$$(s^i)^*s^k = \frac{1}{\sigma_i\sigma_k}(t^i)^*A^*At^k = \frac{\sigma_k}{\sigma_i}(t^i)^*t^k = \begin{cases} 1, & \text{falls } k = i, \\ 0, & \text{falls } k \neq i. \end{cases}$$

orthonormal. Diese Vektoren lassen sich durch $m-r$ passend gewählte Vektoren $s^{r+1},\dots,s^m$ aus dem Kern von A^* zu einer unitären $m\times m$-Matrix $S = (s^1,\dots,s^m)$ ergänzen, so daß

$$AA^*S = SD$$

mit $D = diag(\mu_1,\dots,\mu_m)$ gilt, wobei D die Diagonalmatrix mit den Eigenwerten $\mu_1 \geq 0,\dots,\mu_m \geq 0$ von AA^* bezeichnet.
Aus (1.5) und der obigen Herleitung folgt

$$\mu_i = \lambda_i, \quad i = 1,\dots,r, \quad \mu_i = 0, \quad i = r+1,\dots,m.$$

Mit diesen Matrizen gilt nun

$$\begin{aligned} S^*AT &= \left(\frac{1}{\sigma_1}At^1,\dots,\frac{1}{\sigma_r}At^r,s^{r+1},\dots,s^m\right)^* A\left(t^1,\dots,t^n\right) \\ &= \left(\begin{array}{ccc|ccc} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ \hline & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{array}\right) =: \Sigma \in \mathbb{R}^{m\times n}. \end{aligned}$$

Damit haben wir gezeigt:

Satz 1.3.5 (Singulärwertzerlegung)

Sei $A \in \mathbb{C}^{m\times n}$ beliebig. Dann gibt es unitäre Matrizen $T \in \mathbb{C}^{n\times n}$ und $S \in \mathbb{C}^{m\times m}$ mit

$$S\Sigma T^* = A,$$

wobei $\Sigma \in \mathbb{R}^{m\times n}$ Diagonalmatrix mit den Singulärwerten von A ist.

Beobachtungen:

- Die Spalten von T bilden ein Orthonormalsystem aus Eigenvektoren von A^*A. Die Spalten von S bilden ein Orthonormalsystem aus Eigenvektoren von AA^*.
- σ_1 ist die Spektralnorm von A.
- Falls $Rang(A) = n$ ist, so ist A^*A invertierbar und $r = n$ sowie $\sigma_n > 0$.
- Ist $m = n$ und A invertierbar, so ist $1/\sigma_n$ gleich der Spektralnorm von A^{-1}, da die Eigenwerte von $(A^*A)^{-1} = A^{-1}(A^*)^{-1}$ durch $1/\lambda_i$, $i = 1,\dots,n$, gegeben sind und $\rho(A^{-1}(A^*)^{-1}) = \rho((A^{-1})^*A^{-1})$ gilt. Insbesondere ist die Kondition von A bzgl. der Spektralnorm gegeben durch σ_1/σ_n.

Der folgende Satz stellt den Zusammenhang zwischen Singulärwertzerlegung und Pseudoinverse her.

Satz 1.3.6

Sei $A \in \mathbb{C}^{m\times n}$ vom Rang r mit der Singulärwertzerlegung

$$A = S\Sigma T^*, \qquad \Sigma = \left(\begin{array}{c|c} D & 0 \\ \hline 0 & 0 \end{array}\right) \in \mathbb{R}^{m\times n}, \qquad D = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{pmatrix} \in \mathbb{R}^{r\times r},$$

wobei $\sigma_i > 0$, $i = 1, \dots, r$, die positiven Singulärwerte von A seien. Dann ist die Pseudoinverse von Σ gegeben durch

$$\Sigma^+ = \left(\begin{array}{c|c} D^{-1} & 0 \\ \hline 0 & 0 \end{array}\right) \in \mathbb{R}^{n\times m}$$

und die von A durch

$$A^+ = T\Sigma^+ S^*.$$

Beweis:

(i) Wir betrachten das Ausgleichsproblem

$$\min_{x\in\mathbb{C}^n} \quad \frac{1}{2}\|\Sigma x - b\|_2^2.$$

Ausnutzung der speziellen Struktur von Σ mit entsprechender Partitionierung $x = (x_1, x_2)^\top$ und $b = (b_1, b_2)^\top$ führt auf

$$\|\Sigma x - b\|_2^2 = \|Dx_1 - b_1\|_2^2 + \|b_2\|_2^2.$$

Damit ist jedes $\hat{x}$ mit $\hat{x}_1 = D^{-1}b_1$ Lösung des Ausgleichsproblems. Die Minimalnormlösung ergibt sich für $\hat{x}_2 = 0$, insgesamt gilt also $\hat{x} = \Sigma^+ b$.

(ii) Wir betrachten das Ausgleichsproblem

$$\min_{x\in\mathbb{C}^n} \quad \frac{1}{2}\|Ax - b\|_2^2.$$

Ausnutzen der Singulärwertzerlegung und der Unitarität von S und T liefert

$$\|Ax - b\|_2^2 = \|S^*(ATT^*x - b)\|_2^2 = \|\Sigma y - c\|_2^2$$

mit $y = T^*x$ und $c = S^*b$. Für jedes y gibt es ein eindeutiges x mit $y = T^*x$ und umgekehrt. Weiter gilt $\|x\|_2 = \|Ty\|_2 = \|y\|_2$.

Nach (i) ist $\hat{y} = \Sigma^+ c$ Minimalnormlösung. Für $\hat{x} := T\hat{y}$ folgt $\|\hat{x}\|_2 = \|\hat{y}\|_2$ und

$$\|\hat{x}\|_2 = \|\hat{y}\|_2 \le \|y\|_2 = \|x\|_2 \quad \forall x = Ty, y \in \mathbb{C}^n,$$

sowie

$$\|A\hat{x} - b\|_2^2 = \|\Sigma\hat{y} - c\|_2^2 \le \|\Sigma y - c\|_2^2 = \|Ax - b\|_2^2 \quad \forall x = Ty, y \in \mathbb{C}^n.$$

Damit ist $\hat{x}$ Minimalnormlösung mit

$$\hat{x} = T\hat{y} = T\Sigma^+ S^* b$$

und $A^+ = T\Sigma^+ S^*$.

□

Man kann zeigen, daß die Pseudoinverse A^+ von A folgende Eigenschaften besitzt:

$$\begin{aligned} A^+A &= (A^+A)^* \\ AA^+ &= (AA^+)^* \\ AA^+A &= A \\ A^+AA^+ &= A^+ \end{aligned}$$

Durch diese vier Bedingungen wird die Pseudoinverse sogar eindeutig festgelegt. Beachte, daß die Inverse Matrix A^{-1} alle Bedingungen erfüllt.

Beispiel 1.3.7

Wir betrachten wiederum Beispiel 1.3.3 und möchten die Pseudoinverse von

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

mittels Singulärwertzerlegung bestimmen. Die Eigenwerte und Eigenvektoren von

$$A^*A = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$$

lauten $\lambda_1 = 2, \lambda_2 = 0$, $t^1 = (1,0)^\top$, $t^2 = (0,1)^\top$, die Singulärwerte lauten $\sigma_1 = \sqrt{2}$, $\sigma_2 = 0$. Der Vektor $s^1 = \frac{1}{\sigma_1}At^1$ berechnet sich zu $s^1 = (1/\sqrt{2}, 1/\sqrt{2}, 0)^\top$ und kann durch $s^2 = (1/\sqrt{2}, -1/\sqrt{2}, 0)^\top$ und $s^3 = (0,0,1)^\top$ aus dem Kern von A^* zu einer Orthonormalbasis des $\mathbb{R}^3$ ergänzt werden (s^2 und s^3 sind hierbei Eigenvektoren zum Eigenwert 0 von AA^*). Damit lautet die Singulärwertzerlegung von A

$$A = S\Sigma T^* = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{2} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

und die Pseudoinverse ist gegeben durch

$$A^+ = T\Sigma^+S^* = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

1.4 Vektoriteration

Wir untersuchen einfache Iterationsverfahren zur Approximation des betragsmäßig größten bzw. des betragsmäßig kleinsten Eigenwerts, sowie eines zugehörigen Eigenvektors.

1.4.1 Potenzmethode (von Mises-Verfahren)

Ist man nur am betragsgrößten Eigenwert mit einem zugehörigen Eigenvektor interessiert, kann die **Potenzmethode von von Mises** verwendet werden.

Algorithmus 1.4.1 (Potenzmethode (von Mises-Verfahren))

(0) Gegeben seien $A \in \mathbb{C}^{n\times n}$, $tol \geq 0$ und ein geeigneter Startvektor $x^{(0)} \in \mathbb{C}^n$, $x^{(0)} \neq 0$. Setze $k = 0$.

(1) Berechne (soweit möglich)

$$x^{(k+1)} = \frac{1}{\|Ax^{(k)}\|} Ax^{(k)}$$

und

$$\lambda^{(k)} = \left(\frac{(Ax^{(k)})_1}{x_1^{(k)}}, \ldots, \frac{(Ax^{(k)})_n}{x_n^{(k)}} \right)^\top$$

(2) Falls $\|Ax^{(k)} - \lambda_j^{(k)} x^{(k)}\|_2 \leq tol$ für ein $1 \leq j \leq n$, STOP. Ansonsten setze $k := k+1$ und gehe zu (1).

Beispiel 1.4.2

Betrachte die symmetrische Matrix

$$A = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}.$$

A besitzt die Eigenwerte 2, 6 sowie den zweifachen Eigenwert 4. Im folgenden wird $x^{(k)}$, $k = 0, 1, \ldots$, durch die Potenzmethode berechnet. Für jede Iterierte wird der Vektor $\lambda^{(k)} = \left(\frac{(Ax^{(k)})_1}{x_1^{(k)}}, \ldots, \frac{(Ax^{(k)})_n}{x_n^{(k)}} \right)^\top$ berechnet.

Für den Startwert $x^{(0)} = (1, 0, 0, 0)^\top$ liefert die Potenzmethode folgenden Output:

```
iter= 1 : x=[9.428090e-01 -2.357023e-01 -2.357023e-01 0.000000e+00]  lambda=[4.000000e+00  -Inf        -Inf        NaN        ]
iter= 2 : x=[8.429272e-01 -3.746343e-01 -3.746343e-01 9.365858e-02]  lambda=[4.500000e+00 8.000000e+00 8.000000e+00 Inf        ]
iter= 3 : x=[7.510676e-01 -4.438127e-01 -4.438127e-01 2.048366e-01]  lambda=[4.888889e+00 6.500000e+00 6.500000e+00 12.00000e+00]
iter= 4 : x=[6.777076e-01 -4.755843e-01 -4.755843e-01 2.972402e-01]  lambda=[5.181818e+00 6.153846e+00 6.153846e+00 8.333333e+00]
iter= 5 : x=[6.230275e-01 -4.895216e-01 -4.895216e-01 3.641070e-01]  lambda=[5.403509e+00 6.050000e+00 6.050000e+00 7.200000e+00]
iter= 6 : x=[5.839930e-01 -4.955092e-01 -4.955092e-01 4.097480e-01]  lambda=[5.571429e+00 6.016529e+00 6.016529e+00 6.688889e+00]
iter= 7 : x=[5.568520e-01 -4.980681e-01 -4.980681e-01 4.401956e-01]  lambda=[5.696970e+00 6.005495e+00 6.005495e+00 6.418605e+00]
iter= 8 : x=[5.382758e-01 -4.991644e-01 -4.991644e-01 4.603574e-01]  lambda=[5.788871e+00 6.001830e+00 6.001830e+00 6.262940e+00]
iter= 9 : x=[5.256821e-01 -4.996366e-01 -4.996366e-01 4.736927e-01]  lambda=[5.854679e+00 6.000610e+00 6.000610e+00 6.168595e+00]
iter=10 : x=[5.171945e-01 -4.998412e-01 -4.998412e-01 4.825219e-01]  lambda=[5.900908e+00 6.000203e+00 6.000203e+00 6.109539e+00]
iter=11 : x=[5.114955e-01 -4.999304e-01 -4.999304e-01 4.883765e-01]  lambda=[5.932895e+00 6.000068e+00 6.000068e+00 6.071787e+00]
iter=12 : x=[5.076781e-01 -4.999694e-01 -4.999694e-01 4.922644e-01]  lambda=[5.954779e+00 6.000023e+00 6.000023e+00 6.047315e+00]
iter=13 : x=[5.051252e-01 -4.999865e-01 -4.999865e-01 4.948490e-01]  lambda=[5.969631e+00 6.000008e+00 6.000008e+00 6.031304e+00]
iter=14 : x=[5.034197e-01 -4.999940e-01 -4.999940e-01 4.965688e-01]  lambda=[5.979654e+00 6.000003e+00 6.000003e+00 6.020764e+00]
iter=15 : x=[5.022811e-01 -4.999974e-01 -4.999974e-01 4.977138e-01]  lambda=[5.986390e+00 6.000001e+00 6.000001e+00 6.013796e+00]
iter=16 : x=[5.015213e-01 -4.999988e-01 -4.999988e-01 4.984764e-01]  lambda=[5.990907e+00 6.000000e+00 6.000000e+00 6.009176e+00]
iter=17 : x=[5.010144e-01 -4.999995e-01 -4.999995e-01 4.989845e-01]  lambda=[5.993929e+00 6.000000e+00 6.000000e+00 6.006108e+00]
iter=18 : x=[5.006764e-01 -4.999998e-01 -4.999998e-01 4.993231e-01]  lambda=[5.995948e+00 6.000000e+00 6.000000e+00 6.004068e+00]
iter=19 : x=[5.004510e-01 -4.999999e-01 -4.999999e-01 4.995488e-01]  lambda=[5.997297e+00 6.000000e+00 6.000000e+00 6.002710e+00]
iter=20 : x=[5.003007e-01 -5.000000e-01 -5.000000e-01 4.996992e-01]  lambda=[5.998197e+00 6.000000e+00 6.000000e+00 6.001806e+00]
```

Die Quotienten $\frac{(Ax^{(k)})_j}{x_j^{(k)}}$ *konvergieren für* $j = 1, \ldots, 4$ *offenbar gegen den betragsgrößten Eigenwert* 6. *Die Folge* $x^{(k)}$ *konvergiert gegen einen Eigenvektor zum Eigenwert* 6. *Für den Startwert* $x^{(0)} = (1, 1, 1, 1)^\top$ *liefert die Potenzmethode folgenden Output:*

```
iter= 1 : x=[5.000000e-01 5.000000e-01 5.000000e-01 5.000000e-01]  lambda=[2 2 2 2]
iter= 2 : x=[5.000000e-01 5.000000e-01 5.000000e-01 5.000000e-01]  lambda=[2 2 2 2]
iter= 3 : x=[5.000000e-01 5.000000e-01 5.000000e-01 5.000000e-01]  lambda=[2 2 2 2]
iter= 4 : x=[5.000000e-01 5.000000e-01 5.000000e-01 5.000000e-01]  lambda=[2 2 2 2]
iter= 5 : x=[5.000000e-01 5.000000e-01 5.000000e-01 5.000000e-01]  lambda=[2 2 2 2]
```

Hier konvergiert das Verfahren gegen den betragskleinsten Eigenwert 2. *Beachte, daß* $x^{(0)}$ *gerade ein Eigenvektor zum Eigenwert* 2 *ist.*

Wir untersuchen Konvergenzeigenschaften der Potenzmethode und treffen folgende Annahmen:

Annahme 1.4.3

(i) A besitze n linear unabhängige Eigenvektoren $v^1, \ldots, v^n$ *zu den Eigenwerten* $\lambda_1, \ldots, \lambda_n$.

(ii) Für die Eigenwerte $\lambda_1, \ldots, \lambda_n$ *gelte*

$$\lambda_1 = \lambda_2 = \ldots = \lambda_r \quad \textit{und} \quad |\lambda_r| > |\lambda_{r+1}| \geq \ldots \geq |\lambda_n|,$$

d.h. der betragsgrößte Eigenwert ist separiert.

(iii) Nach (i) kann jeder Startvektor $x^{(0)}$ *dargestellt werden als*

$$x^{(0)} = \sum_{j=1}^{n} \alpha_j v^j$$

mit eindeutigen Koeffizienten $\alpha_j \in \mathbb{C}$, $j = 1, \ldots, n$.

Wir setzen voraus, daß die zum betragsgrößten Eigenwert gehörenden Eigenvektoren $v^1, \ldots, v^r$ *in der Darstellung von* $x^{(0)}$ *tatsächlich vertreten sind, d.h. es gelte*

$$z := \sum_{j=1}^{r} \alpha_j v^j \neq 0.$$

Beachte, daß Annahme 1.4.3, (iii) im zweiten Teil von Beispiel 1.4.2 nicht erfüllt war.

Hilfsatz 1.4.4 (Wohldefiniertheit)

Unter Annahme 1.4.3 ist Algorithmus 1.4.1 wohldefiniert, d.h. es gilt $\|x^{(k)}\| \neq 0$ *für alle* $k = 0, 1, \ldots$.

Beweis: Zunächst bemerken wir, daß die Folge $x^{(k)}$ auch als

$$x^{(k)} = \frac{1}{\|A^k x^{(0)}\|} A^k x^{(0)}$$

geschrieben werden kann. Für $k \in \mathbb{N}$ gilt

$$A^k x^{(0)} = \sum_{j=1}^{r} \alpha_j A^k v^j + \sum_{j=r+1}^{n} \alpha_j A^k v^j = \sum_{j=1}^{r} \alpha_j \lambda_1^k v^j + \sum_{j=r+1}^{n} \alpha_j \lambda_j^k v^j.$$

Angenommen, es gilt $A^k x^{(0)} = 0$. Auf Grund der linearen Unabhängigkeit der Eigenvektoren folgt dann $\alpha_j \lambda_1^k = 0$ für $j = 1, \dots, r$ und $\alpha_j \lambda_j^k = 0$ für $j = r+1, \dots, n$. Gemäß Annahme 1.4.3 (ii) ist $\lambda_1 \neq 0$, so daß $\alpha_j = 0$ für $j = 1, \dots, r$ gelten muß. Dies steht jedoch im Widerspruch zur Annahme $z \neq 0$ in Teil (iii) von Annahme 1.4.3. □

Da λ_1 der betragsgrößte Eigenwert ist, folgt

$$\lim_{k\to\infty} \frac{1}{\lambda_1^k} A^k x^{(0)} = \lim_{k\to\infty} \left(\sum_{j=1}^{r} \alpha_j v^j + \sum_{j=r+1}^{n} \alpha_j \left(\frac{\lambda_j}{\lambda_1} \right)^k v^j \right) = \sum_{j=1}^{r} \alpha_j v^j = z.$$

Für Komponenten $z_j \neq 0$ folgt weiter

$$\lim_{k\to\infty} \frac{(Ax^{(k)})_j}{x_j^{(k)}} = \lim_{k\to\infty} \frac{(A^{k+1}x^{(0)})_j \lambda_1^k \lambda_1}{(A^k x^{(0)})_j \lambda_1^{k+1}} = \frac{z_j \lambda_1}{z_j} = \lambda_1$$

und schließlich

$$\begin{aligned} x^{(k)} &= \frac{1}{\|A^k x^{(0)}\|} A^k x^{(0)} \\ &= \frac{\sum_{j=1}^{r} \alpha_j \lambda_1^k v^j + \sum_{j=r+1}^{n} \alpha_j \lambda_j^k v^j}{\left\| \sum_{j=1}^{r} \alpha_j \lambda_1^k v^j + \sum_{j=r+1}^{n} \alpha_j \lambda_j^k v^j \right\|} \\ &= \frac{\lambda_1^k \left(z + \sum_{j=r+1}^{n} \alpha_j \left(\frac{\lambda_j}{\lambda_1} \right)^k v^j \right)}{|\lambda_1|^k \left\| z + \sum_{j=r+1}^{n} \alpha_j \left(\frac{\lambda_j}{\lambda_1} \right)^k v^j \right\|}. \end{aligned}$$

Da $\lambda_1^k / |\lambda_1^k|$ für alle k auf dem kompakten komplexen Einheitskreis liegt, gibt es eine konvergente Teilfolge mit

$$\lim_{\ell\to\infty} \frac{\lambda_1^{k_\ell}}{|\lambda_1^{k_\ell}|} = \gamma, \quad |\gamma| = 1,$$

und

$$\hat{x} = \lim_{\ell\to\infty} x^{(k_\ell)} = \gamma \frac{z}{\|z\|}.$$

Wegen

$$A\hat{x} = \frac{\gamma}{\|z\|} \sum_{j=1}^{r} \alpha_j A v^j = \lambda_1 \frac{\gamma}{\|z\|} \sum_{j=1}^{r} \alpha_j v^j = \lambda_1 \frac{\gamma}{\|z\|} z = \lambda_1 \hat{x}$$

ist $\hat{x}$ Eigenvektor zum Eigenwert λ_1. Zusammenfassend haben wir gezeigt:

Satz 1.4.5 (Konvergenz)
Unter Annahme 1.4.3 gilt für die Potenzmethode

$$\lim_{k\to\infty} \frac{(Ax^{(k)})_j}{x_j^{(k)}} = \lambda_1 \quad \text{für alle } j \text{ mit } z_j \neq 0$$

und

$$\lim_{\ell\to\infty} x^{(k_\ell)} = \gamma \frac{z}{\|z\|} = \hat{x}$$

für eine Teilfolge $\{k_\ell\}_{\ell\in\mathbb{N}}$ mit $|\gamma| = 1$. $\hat{x}$ ist dabei ein Eigenvektor zum Eigenwert λ_1.

Der Umstand, daß die Konvergenz gegen einen Eigenvektor nur für eine Teilfolge $(k_\ell)_{\ell\in\mathbb{N}}$ gilt, zeigt sich bei reeller Rechnung dadurch, daß die Folge $x^{(k)}$ mit $k \in \mathbb{N}$ für $\lambda_1 \in \mathbb{R}$, $\lambda_1 < 0$ alternieren kann, d.h. man erhält ab einer gewissen Iterationszahl k näherungsweise abwechselnd den Vektor $x^{(k)}$ und $-x^{(k)}$.
Aus theoretischer Sicht ist die Annahme $z \neq 0$ kritisch, da man a priori nicht weiß, ob der Startwert $x^{(0)}$ die Bedingung $z \neq 0$ erfüllt. In der Praxis erweist sich dies allerdings in der Regel nicht als problematisch, da diese Bedingung durch den Rundungsfehlereinfluß i.a. numerisch erfüllt wird.

Bemerkung 1.4.6 (Rayleigh-Quotient)
Ist $x \neq 0$ ein (approximativer) Eigenvektor zum Eigenwert λ, d.h. $Ax \approx \lambda x$, so folgt durch Multiplikation von links mit x^*,

$$\lambda \approx \frac{x^*Ax}{x^*x}.$$

Der Quotient auf der rechten Seite heißt **Rayleigh-Quotient**. *Da $x^{(k)}$ in Algorithmus 1.4.1 auf eins normiert ist, kann eine Näherung des Eigenwerts auch durch $\lambda^{(k)} := x^*Ax$ bestimmt werden.*

1.4.2 Inverse Iteration von Wielandt

Kennt man eine gute Näherung $\tilde{\lambda}$ für einen Eigenwert λ_j der Matrix $A \in \mathbb{C}^{n\times n}$, so kann man mit der **Inversen Iteration von Wielandt** den Eigenwert λ_j berechnen. Dabei führen wir eine **Spektralverschiebung** für die Matrix A durch, indem die Matrix $A - \tilde{\lambda}I$ betrachtet wird.

Algorithmus 1.4.7 (Inverse Iteration von Wielandt)

(0) Gegeben seien $A \in \mathbb{C}^{n\times n}$, tol ≥ 0, $\tilde{\lambda} \in \mathbb{C}$ und ein geeigneter Startvektor $x^{(0)} \in \mathbb{C}^n$, $x^{(0)} \neq 0$. Setze $k = 0$.

(1) Berechne (soweit möglich)

$$\begin{aligned} y^{(k+1)} &= (A-\tilde{\lambda}I)^{-1}\cdot x^{(k)}, \\ x^{(k+1)} &= \frac{1}{\|y^{(k+1)}\|}y^{(k+1)} \end{aligned}$$

und

$$\lambda^{(k)} = \left(\tilde{\lambda}+\frac{x_1^{(k)}}{y_1^{(k+1)}},\ldots,\tilde{\lambda}+\frac{x_n^{(k)}}{y_n^{(k+1)}}\right)^\top$$

(2) Falls $\|Ax^{(k)}-\lambda_j^{(k)}x^{(k)}\|_2 \le tol$ für ein $1\le j\le n$, STOP. Ansonsten setze $k:=k+1$ und gehe zu (1).

Zunächst bemerken wir, daß die Inverse von $(A-\tilde{\lambda}I)$ existiert, falls $\tilde{\lambda}$ kein Eigenwert von A ist. In der Praxis wird die Inverse $(A-\tilde{\lambda}I)^{-1}$ nicht explizit berechnet, sondern $y^{(k+1)}$ wird durch Lösen des linearen Gleichungssystems $(A-\tilde{\lambda}I)y^{(k+1)} = x^{(k)}$ berechnet. Da die Matrix $A-\tilde{\lambda}I$ sich in den Iterationen nicht ändert, muß zu Beginn nur einmalig eine LR- (oder QR-) Zerlegung berechnet werden, so daß pro Iteration nur eine Vorwärts- und Rückwärtssubstitution durchgeführt werden muß.
Ein Vergleich mit der Potenzmethode zeigt, daß die inverse Iteration von Wielandt nichts anderes als die Potenzmethode angewendet auf die Matrix $(A-\tilde{\lambda}I)^{-1}$ ist. Wir untersuchen dazu die Eigenwerte der Matrix $A-\tilde{\lambda}I$ bzw. die Eigenwerte einer regulären Matrix B und ihrer Inversen:

- Sei λ ein Eigenwert von A, d.h. $Ax=\lambda x$. Subtraktion von $\tilde{\lambda}x$ auf beiden Seiten liefert
$$(A-\tilde{\lambda}I)x = (\lambda-\tilde{\lambda})x,$$
d.h. $\lambda-\tilde{\lambda}$ ist Eigenwert von $A-\tilde{\lambda}I$. Die Eigenwerte von $A-\tilde{\lambda}I$ sind die um $\tilde{\lambda}$ „verschobenen“ Eigenwerte von A. Man spricht daher auch von einer **Spektralverschiebung**.

- Da wir an den Eigenwerten von $(A-\tilde{\lambda}I)^{-1}$ interessiert sind, stellt sich die Frage, wie die Eigenwerte λ einer regulären Matrix B mit den Eigenwerten der Inversen B^{-1} zusammenhängen. Aus $Bx=\lambda x$ folgt $\frac{1}{\lambda}x = B^{-1}x$. Damit ist $\mu=\frac{1}{\lambda}$ Eigenwert von B^{-1}, falls λ Eigenwert von B ist. Zu beachten ist, daß reguläre Matrizen nur Eigenwerte ungleich Null besitzen. Andernfalls würde $Bx = 0\cdot x = 0$ für einen Vektor $x\neq 0$ gelten, was ein Widerspruch zur Regularität von B ist.

Beachte, daß die zugehörigen Eigenvektoren in beiden Fällen unverändert bleiben, d.h. ist x Eigenvektor von A bzw. B, so auch von $A-\tilde{\lambda}I$ bzw. B^{-1}. Insgesamt stehen die

Eigenwerte μ von $(A-\tilde{\lambda}I)^{-1}$ mit den Eigenwerten λ von A also in der Relation

$$\mu = \frac{1}{\lambda - \tilde{\lambda}}.$$

Die inverse Iteration von Wielandt liefert also unter den Voraussetzungen von Satz 1.4.5

- den betragsgrößten Eigenwert von $(A-\tilde{\lambda}I)^{-1}$ bzw.
- den betragskleinsten Eigenwert von $A-\tilde{\lambda}I$ bzw.
- denjenigen Eigenwert von A, für den der Abstand zu $\tilde{\lambda}$ am kleinsten ist.

Um die Voraussetzungen zur Anwendbarkeit der Potenzmethode zu erfüllen, muß A und damit auch $(A-\tilde{\lambda}I)^{-1}$ diagonalisierbar sein und es muß

$$0 < |\lambda_j - \tilde{\lambda}| < |\lambda_k - \tilde{\lambda}|$$

für alle Eigenwerte $\lambda_k \neq \lambda_j$ von A gelten. Satz 1.4.5 gilt dann analog.

Beispiel 1.4.8

Betrachte die symmetrische Matrix

$$A = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}.$$

A besitzt die Eigenwerte 2, 6 sowie den zweifachen Eigenwert 4. Im folgenden wird $x^{(i)}$, $i = 0, 1, \ldots$, durch das Verfahren von Wielandt berechnet. Für jede Iterierte wird der Vektor $\lambda^{(i)} = \left(\tilde{\lambda} + \frac{x_1^{(i)}}{y_1^{(i+1)}}, \ldots, \tilde{\lambda} + \frac{x_n^{(i)}}{y_n^{(i+1)}}\right)^\top$ berechnet.

Für den Startwert $x^{(0)} = (1,0,0,0)^\top$ und $\tilde{\lambda} = 1.5$ liefert das Verfahren von Wielandt folgenden Output:

```
iter= 1 : x= [7.229135e-01  4.252433e-01  4.252433e-01  3.401946e-01]  lambda= [2.823529e+00  1.500000e+00  1.500000e+00  1.500000e+00]
iter= 2 : x= [5.452596e-01  4.930015e-01  4.930015e-01  4.653934e-01]  lambda= [2.191682e+00  1.950000e+00  1.950000e+00  1.881356e+00]
iter= 3 : x= [5.086528e-01  4.992817e-01  4.992817e-01  4.926539e-01]  lambda= [2.036847e+00  1.994505e+00  1.994505e+00  1.973093e+00]
iter= 4 : x= [5.016749e-01  4.999225e-01  4.999225e-01  4.984749e-01]  lambda= [2.006986e+00  1.999390e+00  1.999390e+00  1.994192e+00]
iter= 5 : x= [5.003284e-01  4.999915e-01  4.999915e-01  4.996884e-01]  lambda= [2.001347e+00  1.999932e+00  1.999932e+00  1.998787e+00]
iter= 6 : x= [5.000649e-01  4.999991e-01  4.999991e-01  4.999369e-01]  lambda= [2.000263e+00  1.999992e+00  1.999992e+00  1.999751e+00]
iter= 7 : x= [5.000129e-01  4.999999e-01  4.999999e-01  4.999873e-01]  lambda= [2.000052e+00  1.999999e+00  1.999999e+00  1.999950e+00]
iter= 8 : x= [5.000026e-01  5.000000e-01  5.000000e-01  4.999975e-01]  lambda= [2.000010e+00  2.000000e+00  2.000000e+00  1.999990e+00]
iter= 9 : x= [5.000005e-01  5.000000e-01  5.000000e-01  4.999995e-01]  lambda= [2.000002e+00  2.000000e+00  2.000000e+00  1.999998e+00]
iter=10 : x= [5.000001e-01  5.000000e-01  5.000000e-01  4.999999e-01]  lambda= [2.000000e+00  2.000000e+00  2.000000e+00  2.000000e+00]
```

Die Quotienten $\tilde{\lambda} + \frac{x_j^{(i)}}{y_j^{(i+1)}}$ konvergieren für $j = 1, \ldots, 4$ offenbar gegen den betragskleinsten Eigenwert 2. Die Folge $x^{(i)}$ konvergiert gegen einen Eigenvektor zum Eigenwert 2.

Bemerkung 1.4.9 (Rayleigh-Quotienten Iteration)

Es gibt eine Variante der inversen Iteration von Wielandt, bei der der Shift-Parameter $\tilde{\lambda}$ in jeder Iteration durch

$$\tilde{\lambda}_k = \frac{(x^{(k)})^* A x^{(k)}}{(x^{(k)})^* x^{(k)}}$$

angepaßt wird, vgl. Bemerkung 1.4.6. Da sich die Matrix $A-\tilde{\lambda}_k I$ in jedem Iterationsschritt ändert, muß pro Iterationsschritt eine Zerlegung der Matrix berechnet werden, wodurch der Aufwand pro Iteration von $\mathcal{O}(n^2)$ auf $\mathcal{O}(n^3)$ ansteigt. Man kann jedoch unter geeigneten Annahmen zeigen, daß die Folge mit kubischer Konvergengeschwindigkeit konvergiert. Ein Beweis findet sich in B.N. Parlett: The symmetric eigenvalue problem. Prentice-Hall, Englewood Cliffs, 1980.
Für Beispiel 1.4.8 und Startwert $x^{(0)} = (2/3, 0, 2/3, 1)^\top$ liefert die Rayleigh-Quotienten Iteration den folgenden Output:

```
iter= 1 : x=[-3.620526e-01 -6.859943e-01 -3.048864e-01 -5.526066e-01]  lambda=[ 1.263158e+00  2.666667e+00  1.000000e+00  2.206897e+00]
iter= 2 : x=[ 5.110455e-01  4.780878e-01  5.194118e-01  4.903835e-01]  lambda=[ 2.048120e+00  1.905345e+00  2.071994e+00  1.965878e+00]
iter= 3 : x=[-4.999883e-01 -5.000228e-01 -4.999783e-01 -5.000106e-01]  lambda=[ 1.999951e+00  2.000093e+00  1.999915e+00  2.000040e+00]
iter= 4 : x=[ 5.000000e-01  5.000000e-01  5.000000e-01  5.000000e-01]  lambda=[ 2.000000e+00  2.000000e+00  2.000000e+00  2.000000e+00]
```

1.5 QR-Algorithmus

Der folgende QR-Algorithmus ist einer der wichtigsten Algorithmen zur Bestimmung von Eigenwerten. Im Gegensatz zu den bisherigen Verfahren approximiert das Verfahren nicht nur einen Eigenwert, sondern – unter geeigneten Voraussetzungen – alle Eigenwerte auf einmal. Zusätzlich erhält man noch zugehörige Eigenvektoren.
Die Grundidee des QR-Verfahrens basiert auf der Durchführung von Ähnlichkeitstranformationen

$$A = A^{(0)} \to A^{(1)} \to \cdots \to A^{(m)} \to \cdots \tag{1.6}$$

mit $A^{(i)} = (T^{(i)})^{-1} A^{(i-1)} T^{(i)}$, $i = 1, 2, \ldots$. Beachte, daß A und $A^{(i)}$ nach Satz 1.0.11 dieselben Eigenwerte besitzen. Diese Vorgehensweise ist natürlich nur dann sinnvoll, wenn die Eigenwerte (und Eigenvektoren) von $A^{(i)}$ bzw. einem Grenzwert der Folge $\{A^{(i)}\}$ einfacher zu berechnen sind als die von A. Zudem sollte darauf geachtet werden, daß die Kondition des Eigenwertproblems für $A^{(i)}$ nicht wesentlich schlechter als die des Eigenwertproblems für A ist. Ideal sind hier unitäre Matrizen $T^{(i)}$. In seiner Grundform lautet der QR-Algorithmus wie folgt.

Algorithmus 1.5.1 (QR-Algorithmus)

(0) Setze $A^{(0)} = A \in \mathbb{C}^{n\times n}$, $V^{(0)} = I$ und $k = 0$.

(1) Berechne z.B. mittels Householder-Transformationen oder Givens-Rotationen eine QR-Zerlegung der Matrix $A^{(k)}$ gemäß

$$A^{(k)} = Q^{(k)} \cdot R^{(k)}$$

mit unitärer Matrix $Q^{(k)}$ und einer rechten oberen Dreiecksmatrix $R^{(k)}$.

(2) Setze $A^{(k+1)} = R^{(k)} \cdot Q^{(k)}$, $V^{(k+1)} := V^{(k)} \cdot Q^{(k)}$, $k := k + 1$ und gehe zu (1).

Zunächst bemerkt man, daß $A^{(k)}$ und $A^{(k+1)}$ dieselben Eigenwerte besitzen, da der Übergang von $A^{(k)} = Q^{(k)} \cdot R^{(k)}$ zu

$$A^{(k+1)} = R^{(k)} \cdot Q^{(k)} = (Q^{(k)})^* \cdot A^{(k)} \cdot Q^{(k)} = (Q^{(k)})^{-1} \cdot A^{(k)} \cdot Q^{(k)}$$

eine Ähnlichkeitstransformation ist. Induktiv ergibt sich daraus

$$\begin{aligned} A^{(k)} &= (Q^{(k-1)})^{-1} \cdot A^{(k-1)} \cdot Q^{(k-1)} \\ &= (Q^{(k-1)})^{-1} \cdot (Q^{(k-2)})^{-1} \cdots (Q^{(0)})^{-1} \cdot A \cdot Q^{(0)} \cdots Q^{(k-2)} \cdot Q^{(k-1)} \\ &= (V^{(k)})^{-1} \cdot A \cdot V^{(k)}, \end{aligned} \tag{1.7}$$

wobei $V^{(k)} = Q^{(0)} \cdot Q^{(1)} \cdots Q^{(k-1)}$ unitär ist. Folglich besitzen A und sämtliche Matrizen $A^{(k)}$, $k = 1, 2, \ldots$, dieselben Eigenwerte. Eine weitere nützliche Eigenschaft liefert

Hilfsatz 1.5.2

Für die durch das QR-Verfahren 1.5.1 erzeugten Matrizen gilt für $k = 1, 2, \ldots$

$$A^k = Q^{(0)} \cdot Q^{(1)} \cdots Q^{(k-1)} \cdot R^{(k-1)} \cdots R^{(1)} \cdot R^{(0)}.$$

Beweis: Wir zeigen die Behauptung durch Induktion nach k. Für $k = 1$ gilt $A^1 = A = Q^{(0)} \cdot R^{(0)}$. Sei die Behauptung gezeigt für k. Dann folgt mit (1.7)

$$\begin{aligned} A^{k+1} &= AA^k \\ &= A^{(0)} \cdot Q^{(0)} \cdot Q^{(1)} \cdots Q^{(k-1)} \cdot R^{(k-1)} \cdot R^{(k-2)} \cdots R^{(0)} \\ &\overset{(1.7)}{=} Q^{(0)} \cdot Q^{(1)} \cdots Q^{(k-1)} \cdot A^{(k)} \cdot R^{(k-1)} \cdot R^{(k-2)} \cdots R^{(0)} \\ &= Q^{(0)} \cdot Q^{(1)} \cdots Q^{(k-1)} \cdot Q^{(k)} \cdot R^{(k)} \cdot R^{(k-1)} \cdot R^{(k-2)} \cdots R^{(0)} \end{aligned}$$

□

Es stellt sich die Frage, ob bzw. wie der Algorithmus konvergiert. Wir untersuchen dies zunächst an einem Beispiel.

Beispiel 1.5.3

Betrachte die symmetrische Matrix

$$A = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}.$$

A besitzt die Eigenwerte 2, 6 sowie den zweifachen Eigenwert 4.

Anwendung des QR-Verfahrens auf A liefert folgende Iterationen:

$$A^{(5)} = \begin{pmatrix} 5.9329 & -0.2527 & -0.2568 & -0.0000 \\ -0.2527 & 4.0311 & 0.0316 & 0.0629 \\ -0.2568 & 0.0316 & 4.0321 & 0.0640 \\ 0 & 0.0629 & 0.0640 & 2.0040 \end{pmatrix}, \quad V^{(5)} = \begin{pmatrix} -0.6230 & -0.4384 & -0.4455 & 0.4703 \\ 0.4895 & -0.5837 & 0.4150 & 0.4975 \\ 0.4895 & 0.4243 & -0.5770 & 0.4975 \\ -0.3641 & 0.5358 & 0.5445 & 0.5328 \end{pmatrix},$$

$$A^{(10)} = \begin{pmatrix} 5.9988 & -0.0347 & -0.0347 & 0.0000 \\ -0.0347 & 4.0006 & 0.0006 & -0.0020 \\ -0.0347 & 0.0006 & 4.0006 & -0.0020 \\ 0 & -0.0020 & -0.0020 & 2.0000 \end{pmatrix}, \quad V^{(10)} = \begin{pmatrix} 0.5172 & 0.4916 & 0.4917 & 0.4990 \\ -0.4998 & 0.5092 & -0.4908 & 0.5000 \\ -0.4998 & -0.4909 & 0.5091 & 0.5000 \\ 0.4825 & -0.5080 & -0.5081 & 0.5010 \end{pmatrix},$$

$$A^{(20)} = \begin{pmatrix} 6.0000 & -0.0006 & -0.0006 & 0.0000 \\ -0.0006 & 4.0000 & 0.0000 & -0.0000 \\ -0.0006 & 0.0000 & 4.0000 & -0.0000 \\ 0 & -0.0000 & -0.0000 & 2.0000 \end{pmatrix}, \quad V^{(20)} = \begin{pmatrix} 0.5003 & 0.4999 & 0.4999 & 0.5000 \\ -0.5000 & 0.5002 & -0.4998 & 0.5000 \\ -0.5000 & -0.4998 & 0.5002 & 0.5000 \\ 0.4997 & -0.5001 & -0.5001 & 0.5000 \end{pmatrix},$$

$$A^{(30)} = \begin{pmatrix} 6.0000 & -0.0000 & -0.0000 & 0.0000 \\ -0.0000 & 4.0000 & 0.0000 & -0.0000 \\ -0.0000 & 0.0000 & 4.0000 & -0.0000 \\ 0 & -0.0000 & -0.0000 & 2.0000 \end{pmatrix}, \quad V^{(30)} = \begin{pmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ -0.5000 & 0.5000 & -0.5000 & 0.5000 \\ -0.5000 & -0.5000 & 0.5000 & 0.5000 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \end{pmatrix}.$$

Offenbar konvergiert die Folge $\{A^{(k)}\}_{k\in\mathbb{N}}$ gegen eine Diagonalmatrix, die die Eigenwerte von A enthält, die zudem noch entsprechend ihrer Größe sortiert sind. Die Folge $\{Q_k\}_{k\in\mathbb{N}}$ konvergiert gegen eine Matrix, deren Spalten Eigenvektoren von A zu den jeweiligen Eigenwerten von A sind, wobei die erste Spalte Eigenvektor zum Eigenwert 6, die zweite und dritte Eigenvektoren zum Eigenwert 4 und die vierte Eigenvektor zum Eigenwert 2 sind. Ein Blick auf die Diagonalelemente der Matrizen $A^{(k)}$ zeigt ferner, daß der kleinste Eigenwert 2 sehr schnell sehr gut approximiert wird, während die größeren Eigenwerte weniger schnell approximiert werden.

Im folgenden werden wir zeigen, daß die Beobachtungen in Beispiel 1.5.3 kein Zufall waren. Eine Schwierigkeit hierbei ist der Umstand, daß QR-Zerlegungen nicht eindeutig sind. Jedoch zeigt der folgende Hilfsatz, daß QR-Zerlegungen sich nur durch sogenannte Phasen unterscheiden, welche im wesentlichen einer Skalierung der Diagonalen von R entsprechen. Ein ähnliches Resultat wurde in Numerik I bereits für die reduzierte (reelle) QR-Zerlegung nachgewiesen.

Hilfsatz 1.5.4

Sei $A \in \mathbb{C}^{n\times n}$ invertierbar und $A = Q \cdot R$ eine QR-Zerlegung mit einer unitären Matrix Q und einer rechten oberen Dreiecksmatrix R. Die QR-Zerlegung ist eindeutig, wenn die **Phasen** *$\exp(i\varphi_j)$ der Diagonalelemente $r_{j,j} = |r_{j,j}|\exp(i\varphi_j)$ mit $\varphi_j \in \mathbb{R}$, $j = 1,\dots,n$, fest vorgegeben sind. Die Matrizen Q und R hängen dann stetig von A ab.*

Beweis: Sei $A = Q \cdot R = \tilde{Q} \cdot \tilde{R}$, wobei die rechten oberen Dreiecksmatrizen R und $\tilde{R}$ dieselben Phasen besitzen, d.h.

$$r_{j,j} = |r_{j,j}|\exp(i\varphi_j), \quad \tilde{r}_{j,j} = |\tilde{r}_{j,j}|\exp(i\varphi_j), \quad j = 1,\dots,n.$$

Beachte, daß R und $\tilde{R}$ invertierbar sind. Folglich gelten $r_{j,j} \neq 0$ und $\tilde{r}_{j,j} \neq 0$, wodurch die Phasenargumente $\varphi_j \in [0, 2\pi)$ eindeutig definiert sind.
Aus $Q \cdot R = \tilde{Q} \cdot \tilde{R}$ folgt $\tilde{Q}^{-1} \cdot Q = \tilde{R} \cdot R^{-1}$. Links steht eine unitäre Matrix, rechts eine obere Dreiecksmatrix, die damit auch unitär ist. Nun kann man sich überlegen, daß jede unitäre rechte obere Dreiecksmatrix eine Diagonalmatrix sein muß, deren Diagonalelemente den Betrag eins haben. Also gilt $D = \tilde{Q}^{-1} \cdot Q = \tilde{R} \cdot R^{-1}$ mit

$$d_{j,j} = \frac{\tilde{r}_{j,j}}{r_{j,j}} = \frac{|\tilde{r}_{j,j}| \exp(i\varphi_j)}{|r_{j,j}| \exp(i\varphi_j)} = \frac{|\tilde{r}_{j,j}|}{|r_{j,j}|} > 0$$

und $|d_{j,j}| = 1$. Notwendig ist dann $D = I$ und somit $Q = \tilde{Q}$, $R = \tilde{R}$.
Die stetige Abghängigkeit der QR-Zerlegung folgt, indem man beispielsweise die Rechenschritte des Gram-Schmidt-Verfahren zur Konstruktion einer QR-Zerlegung untersucht. □

Unter Ausnutzung des Hilfresultats über die Eindeutigkeit der QR-Zerlegung zeigen wir nun das Hauptresultat dieses Abschnitts. Hierin wird die Berechnung der QR-Zerlegung nicht weiter eingeschränkt, wodurch dann jedoch Phasenmatrizen zur Normierung der QR-Zerlegung ins Spiel kommmen. Umgekehrt könnte man im folgenden Satz die Phasenmatrizen vorschreiben, was nach Hilfsatz 1.5.4 die Eindeutigkeit der QR-Zerlegung zur Folge hätte.

Satz 1.5.5 (Konvergenz des QR-Algorithmus)
Sei $A \in \mathbb{C}^{n\times n}$ diagonalisierbar mit $T^{-1}AT = \Lambda$, wobei Λ Diagonalmatrix mit den Eigenwerten $\lambda_1, \ldots, \lambda_n$ von A sei. Desweiteren gelte

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n| > 0. \tag{1.8}$$

Die Matrix T^{-1} besitze eine LR-Zerlegung mit einer normierten unteren Dreiecksmatrix L und einer oberen Dreiecksmatrix R.
Dann gibt es Phasenmatrizen

$$S^{(k)} := \begin{pmatrix} \exp(i\varphi_1^{(k)}) & & \\ & \ddots & \\ & & \exp(i\varphi_n^{(k)}) \end{pmatrix}, \qquad \varphi_j^{(k)} \in \mathbb{R},\ j = 1, \ldots, n,$$

mit

$$\lim_{k\to\infty} (S^{(k-1)})^* \cdot A^{(k)} \cdot S^{(k-1)} = \lim_{k\to\infty} (S^{(k)})^* \cdot R^{(k)} \cdot S^{(k-1)} = \begin{pmatrix} \lambda_1 & * & \cdots & * \\ & \ddots & & \vdots \\ & & \ddots & * \\ & & & \lambda_n \end{pmatrix}$$

und

$$\lim_{k\to\infty} (S^{(k-1)})^* \cdot Q^{(k)} \cdot S^{(k)} = I.$$

Insbesondere gilt

$$\lim_{k\to\infty} a_{jj}^{(k)} = \lambda_j \qquad \forall j = 1, \ldots, n.$$

Beweis: Nach Hilfsatz 1.5.2 gilt

$$A^{k+1} = Q^{(0)} \cdot Q^{(1)} \cdots Q^{(k)} \cdot R^{(k)} \cdot R^{(k-1)} \cdots R^{(0)}, \tag{1.9}$$

wodurch eine QR-Zerlegung von A^{k+1} gegeben ist (beachte: das Produkt rechter oberer Dreiecksmatrizen ist wieder eine rechte obere Dreiecksmatrix).
Da A nach Voraussetzung diagonalisierbar ist und T^{-1} eine LR-Zerlegung besitzt, ist eine weitere QR-Zerlegung von A^{k+1} gegeben durch

$$\begin{aligned} A^{k+1} &= T \cdot \Lambda^{k+1} \cdot T^{-1} \\ &= \tilde{Q} \cdot \tilde{R} \cdot \Lambda^{k+1} \cdot L \cdot R \\ &= \tilde{Q} \cdot \left(\tilde{R} \cdot \Lambda^{k+1} \cdot L \cdot \Lambda^{-(k+1)} \right) \cdot \Lambda^{k+1} \cdot R \\ &= \tilde{Q} \cdot \left(\hat{Q}^{(k+1)} \cdot \hat{R}^{(k+1)} \right) \cdot \Lambda^{k+1} \cdot R \end{aligned} \tag{1.10}$$

wobei $\tilde{Q} \cdot \tilde{R}$ eine QR-Zerlegung von T und $\hat{Q} \cdot \hat{R}$ eine QR-Zerlegung von $\tilde{R} \cdot \Lambda^{k+1} \cdot L \cdot \Lambda^{-(k+1)}$ seien. $\tilde{Q}$, $\tilde{R}$, $\hat{Q}^{(k+1)}$ und $\hat{R}^{(k+1)}$ seien dabei so gewählt, daß die Diagonalelemente von $\tilde{R}$ und $\hat{R}^{(k+1)}$ positiv sind.
Für die Elemente von $\Lambda^{k+1} \cdot L \cdot \Lambda^{-(k+1)}$ gilt

$$\left(\Lambda^{k+1} \cdot L \cdot \Lambda^{-(k+1)}\right)_{i,j} = \lambda_i^{k+1} \ell_{i,j} \lambda_j^{-(k+1)} = \begin{cases} 0, & \text{falls } i < j, \\ 1, & \text{falls } i = j, \\ \left(\frac{\lambda_i}{\lambda_j}\right)^{k+1} \ell_{i,j}, & \text{falls } i > j. \end{cases} \tag{1.11}$$

Mit (1.8) folgt daraus

$$\lim_{k\to\infty} \Lambda^{k+1} \cdot L \cdot \Lambda^{-(k+1)} = I$$

und weiter

$$\lim_{k\to\infty} \tilde{R} \cdot \Lambda^{k+1} \cdot L \cdot \Lambda^{-(k+1)} = \tilde{R} = I \cdot \tilde{R}.$$

Für die QR-zerlegung von $\tilde{R} \cdot \Lambda^{k+1} \cdot L \cdot \Lambda^{-(k+1)}$ folgt dann mit Hilfsatz 1.5.4

$$\lim_{k\to\infty} \hat{Q}^{(k+1)} = I, \qquad \lim_{k\to\infty} \hat{R}^{(k+1)} = \tilde{R} \tag{1.12}$$

und weiter

$$\lim_{k\to\infty} \tilde{Q} \cdot \hat{Q}^{(k+1)} = \tilde{Q}.$$

Da durch (1.9) und (1.10) QR-Zerlegungen von A^{k+1} gegeben sind, folgt nach Hilfsatz 1.5.4 die Existenz einer Phasenmatrix $S^{(k)}$ mit

$$\begin{aligned}(S^{(k)})^* \cdot R^{(k)} \cdots R^{(1)} \cdot R^{(0)} &= \hat{R}^{(k+1)} \cdot \Lambda^{k+1} \cdot R,\\ Q^{(0)} \cdot Q^{(1)} \cdots Q^{(k)} \cdot S^{(k)} &= \tilde{Q} \cdot \hat{Q}^{(k+1)}.\end{aligned}$$

Mit (1.12) folgt

$$\lim_{k\to\infty} Q^{(0)} \cdot Q^{(1)} \cdots Q^{(k)} \cdot S^{(k)} = \tilde{Q}.$$

Zusammen mit (1.7) erhalten wir nun die Aussagen des Satzes:

$$\begin{aligned}\lim_{k\to\infty} (S^{(k)})^* \cdot A^{(k+1)} \cdot S^{(k)} &= \lim_{k\to\infty} \left(Q^{(0)} \cdots Q^{(k)} \cdot S^{(k)}\right)^{-1} \cdot A^{(0)} \cdot Q^{(0)} \cdots Q^{(k)} \cdot S^{(k)}\\ &= \tilde{Q}^{-1} \cdot A^{(0)} \cdot \tilde{Q}\\ &= \tilde{R} \cdot T^{-1} \cdot A^{(0)} \cdot T\tilde{R}^{-1}\\ &= \tilde{R} \cdot \Lambda \cdot \tilde{R}^{-1}\\ &= \begin{pmatrix} \lambda_1 & * & \cdots & * \\ & \ddots & & \vdots \\ & & \ddots & * \\ & & & \lambda_n \end{pmatrix}\end{aligned}$$

Für die Diagonalelemente gilt

$$\left((S^{(k)})^* \cdot A^{(k+1)} \cdot S^{(k)}\right)_{j,j} = \exp(-i\varphi_j^{(k)}) a_{j,j}^{(k+1)} \exp(i\varphi_j^{(k)}) = a_{j,j}^{(k+1)},$$

woraus aus der obigen Konvergenz die Konvergenz der Diagonalelemente von $A^{(k+1)}$ gegen die Eigenwerte von A folgt.
Weiter gilt

$$\begin{aligned}\lim_{k\to\infty} (S^{(k)})^* \cdot Q^{(k+1)} \cdot S^{(k+1)} &= \lim_{k\to\infty} (S^{(k)})^{-1} \cdot (Q^{(k)})^{-1} \cdots (Q^{(0)})^{-1} \cdot Q^{(0)} \cdots Q^{(k+1)} \cdot S^{(k+1)}\\ &= \tilde{Q}^{-1} \cdot \tilde{Q}\\ &= I.\end{aligned}$$

Aus $A^{(k+1)} = Q^{(k+1)} \cdot R^{(k+1)}$ folgt

$$\begin{aligned}\lim_{k\to\infty} (S^{(k+1)})^* \cdot R^{(k+1)} \cdot S^{(k)} &= \lim_{k\to\infty} (S^{(k+1)})^* \cdot (Q^{(k+1)})^* \cdot A^{(k+1)} \cdot S^{(k)}\\ &= \lim_{k\to\infty} (S^{(k+1)})^* \cdot (Q^{(k+1)})^* \cdot S^{(k)} \cdot (S^{(k)})^* \cdot A^{(k+1)} \cdot S^{(k)}\\ &= \begin{pmatrix} \lambda_1 & * & \cdots & * \\ & \ddots & & \vdots \\ & & \ddots & * \\ & & & \lambda_n \end{pmatrix}.\end{aligned}$$

□

Folgerungen und Bemerkungen:

(i) Ist A in Satz 1.5.5 reellwertig, so erzwingt (1.8), daß A nur reelle Eigenwerte besitzen darf. Zudem sind die Diagonalelemente der Phasenmatrizen dann stets $+1$ oder -1.

(ii) Aus (1.11) läßt sich ablesen, daß die Konvergenzgeschwindigkeit des Verfahrens bestimmt wird durch die Quotienten

$$\frac{|\lambda_i|}{|\lambda_j|}, \qquad i > j.$$

Je kleiner dieser Quotient ist, desto schneller konvergiert das QR-Verfahren. Liegen die Eigenwerte hingegen nahe beieinander, so ist eine sehr langsame Konvergenz zu erwarten.

(iii) Analysiert man den Beweis genau, so stellt man fest, daß die Forderung, daß T^{-1} eine LR-Zerlegung besitze, nicht wesentlich ist. Der Beweis liesse sich analog mit einer geeigneten Zeilenpermutation von T^{-1} führen, welche stets eine LR-Zerlegung besitzt. Die Eigenwerte erscheinen dann jedoch ebenfalls permutiert auf der Diagonalen.

(iv) Hat man mit dem QR-Verfahren Approximationen von Eigenwerten berechnet, so lassen sich diese mit der inversen Iteration von Wielandt weiter verbessern.

(v) Beachte, daß die Voraussetzungen des Satzes 1.5.5 in Beispiel 1.5.3 nicht erfüllt sind. Der QR-Algorithmus liefert hier dennoch das richtige Resultat.

Bemerkung 1.5.6 (LR-Algorithmus von Rutishauser)
Ein zum QR-Algorithmus analoges Verfahren erhält man, indem in Algorithmus 1.5.1 an Stelle der QR-Zerlegung von $A^{(k)}$ eine LR-Zerlegung $A^{(k)} = L^{(k)}R^{(k)}$ mit einer normierten linken unteren Dreiecksmatrix $L^{(k)}$ und einer rechten oberen Dreiecksmatrix $R^{(k)}$ verwendet wird. Hierfür gelten ähnliche Konvergenzaussagen, jedoch sind die Ähnlichkeitstransformationen anfälliger für Rundungsfehler als beim QR-Verfahren.

1.5.1 Der QR-Algorithmus in der Praxis

In jedem Schritt des QR-Algorithmus 1.5.1 kann die benötigte QR-Zerlegung prinzipiell mit Hilfe von Householder-Transformationen oder Givens-Rotationen ermittelt werden. Dies erfordert pro Iterationsschritt jedoch einen Aufwand von $\mathcal{O}(n^3)$ Operationen.
Der Aufwand pro Iterationsschritt kann auf $\mathcal{O}(n^2)$ Operationen reduziert werden, indem die Matrix A zunächst in einem Vorschritt wie in Abschnitt 1.6 beschrieben auf

Hessenbergform bzw. Tridiagonalform transformiert wird. Dies erfordert einen einmaligen Aufwand von $\mathcal{O}(n^3)$ Operationen. Anschließend wird der QR-Algorithmus auf die Hessenberg- bzw. Tridiagonalmatrix angewendet.

Definition 1.5.7 **(Hessenbergmatrix)**

Matrizen $A = (a_{ij})$ mit $a_{ij} = 0$ für $j \leq i-2$, d.h. Matrizen der Form

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & \ddots & \ddots & \vdots \\ & & a_{n,n-1} & a_{nn} \end{pmatrix} \in \mathbb{C}^{n\times n},$$

heißen **Hessenbergmatrizen**.

Ist die Transformation auf Hessenbergmatrix erfolgt, startet das QR-Verfahren, wobei jetzt in jedem Iterationsschritt spezielle Givens-Rotationen

$$T_{ij}(c,s) = \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & c & & & -s & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & s & & & & c & \\ & & & & & & & & 1 \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \leftarrow i \\ \\ \\ \leftarrow j \\ \\ \\ \\ \end{matrix}$$

$$\begin{matrix} & & \uparrow & & \uparrow & \\ & & i & & j & \end{matrix}$$

mit geeignet gewählten Werten c und s mit $c^2 + s^2 = 1$ dazu verwendet werden, die Subdiagonalelemente der Hessenbergmatrix zu Null zu rotieren. $T_{ij}(c,s)$ beschreibt eine Drehung in der durch die Einheitsvektoren e_j und e_i aufgespannten Ebene um den Winkel ϕ, der durch $c = \cos(\phi)$ und $s = \sin(\phi)$ festgelegt ist.

Algorithmus 1.5.8 **(QR-Verfahren für Hessenbergmatrix mit Givens-Rotationen)**

(0) Sei $A \in \mathbb{C}^{n\times n}$ in Hessenbergform gegeben. Setze $A^{(0)} := A$, $V^{(0)} := I$ und $k = 0$.

(1) **Berechne QR-Zerlegung $A^{(k)} = Q^{(k)} \cdot R^{(k)}$ mittels Givens-Rotationen:**

Setze $Q^{(k)} = I$.

For $i = 1, \ldots, n-1$ **do**

$w = \sqrt{(a_{ii}^{(k)})^2 + (a_{i+1,i}^{(k)})^2}$

$(c,s) = \begin{cases} (1,0), & \text{falls } w = 0, \\ \left(\frac{a_{i,i}^{(k)}}{w}, -\frac{a_{i+1,i}^{(k)}}{w} \right), & \text{sonst} \end{cases}$

For $\ell = i, \ldots, n$ **do**

$tmp1 = c \cdot a_{i,\ell}^{(k)} - s \cdot a_{i+1,\ell}^{(k)}$

$tmp2 = s \cdot a_{i,\ell}^{(k)} + c \cdot a_{i+1,\ell}^{(k)}$

$a_{i,\ell}^{(k)} = tmp1$

$a_{i+1,\ell}^{(k)} = tmp2$

end

Setze $Q^{(k)} := Q^{(k)} \cdot (T_{i,i+1}(c,s))^*$.

end

Setze $R^{(k)} := A^{(k)}$.

(2) Setze $A^{(k+1)} = R^{(k)} \cdot Q^{(k)}$, $V^{(k+1)} := V^{(k)} \cdot Q^{(k)}$, $k := k+1$ *und gehe zu (1).*

Erläuterungen:

(i) In Schritt (1) des Algorithmus werden sukzessive die Subdiagonalelemente der Hessenbergmatrix $A^{(k)}$ durch Anwendung der Givensrotationen $T_{i,i+1}(c_i, s_i)$, $i = 1, \ldots, n-1$, zu Null rotiert, wodurch eine Zerlegung der Form

$$T_{n-1,n}(c_{n-1}, s_{n-1}) \cdots T_{1,2}(c_1, s_1) \cdot A^{(k)} = R^{(k)}$$

erzeugt wird. Mit $Q^{(k)} = T_{1,2}(c_1, s_1)^* \cdots T_{n-1,n}(c_{n-1}, s_{n-1})^*$ wird dadurch eine Zerlegung $A^{(k)} = Q^{(k)} \cdot R^{(k)}$ erzeugt. Der Aufwand beträgt $\mathcal{O}(n^2)$.

(ii) Bei den Matrixmultiplikationen in (1) und (2) wird die Strukur der Matrizen $T_{i,i+1}(c_i, s_i)$ ausgenutzt, so daß der Aufwand nur $\mathcal{O}(n^2)$ beträgt.

(iii) Mit $A^{(k)}$ ist auch die Matrix $A^{(k+1)}$ eine Hessenbergmatrix. Dies folgt aus der Darstellung

$$\begin{aligned} A^{(k+1)} &= R^{(k)} \cdot Q^{(k)} \\ &= R^{(k)} \cdot T_{1,2}(c_1, s_1)^* \cdots T_{n-1,n}(c_{n-1}, s_{n-1})^*. \end{aligned}$$

Die Multiplikation $R^{(k)} \cdot T_{1,2}(c_1, s_1)^*$ beeinflußt nur die Spalten 1 und 2 von $R^{(k)}$ und hierin nur die Zeilen 1 und 2, da $R^{(k)}$ rechte obere Dreiecksmatrix ist. $R^{(k)} \cdot T_{1,2}(c_1, s_1)^*$ ist somit mit Ausnahme des Elements $(2,1)$ immer noch rechte obere Dreiecksmatrix. Induktiv sieht man, daß die weiteren Multiplikationen lediglich Einträge auf der Subdiagonalen ergänzen, so daß $A^{(k+1)}$ Hessenbergform hat.

Analog kann man zeigen, daß Tridiagonalstruktur ebenfalls erhalten bleibt, falls A hermitesche Tridiagonalmatrix ist.

1.5.2 Konvergenzbeschleunigung durch Shift-Techniken

Im Beweis von Satz 1.5.5 haben wir gesehen, daß die Konvergenzgeschwindigkeit des QR-Verfahrens durch die Quotienten $|\lambda_i|/|\lambda_j|$ mit $i > j$ bestimmt wird. Mit Hilfe von Spektralverschiebungen (**Shift-Techniken**) kann die Konvergenz beschleunigt werden.

Algorithmus 1.5.9 (QR-Algorithmus mit Shift)

(0) Setze $A^{(0)} = A \in \mathbb{C}^{n\times n}$, $V^{(0)} = I$ *und* $k = 0$.

(1) Wähle einen Shift-Parameter μ_k *und berechne eine QR-Zerlegung*

$$A^{(k)} - \mu_k I = Q^{(k)} \cdot R^{(k)}$$

mit unitärer Matrix $Q^{(k)}$ *und einer rechten oberen Dreiecksmatrix* $R^{(k)}$.

(2) Setze $A^{(k+1)} = R^{(k)} \cdot Q^{(k)} + \mu_k I$, $V^{(k+1)} := V^{(k)} \cdot Q^{(k)}$, $k := k+1$ *und gehe zu (1).*

Die Addition des Terms $\mu_k I$ in Schritt (2) macht den Shift des Spektrums in Schritt (1) bei der Berechnung der QR-Zerlegung wieder rückgängig. Dies ist notwendig, da $A^{(k)}$ und $A^{(k+1)}$ ansonsten nicht dieselben Eigenwerte hätten.
Bei der Wahl des Schift-Parameters μ_k werden in der Regel folgende Varianten verwendet:

(i) **Rayleigh-Quotienten-Shift:** $\mu_k = a_{n,n}^{(k)}$

(ii) **Wilkinson-Shift:** μ_k ist der am nächsten an $a_{n,n}^{(k)}$ liegende Eigenwert der Matrix

$$\begin{pmatrix} a_{n-1,n-1}^{(k)} & a_{n-1,n}^{(k)} \\ a_{n,n-1}^{(k)} & a_{n,n}^{(k)} \end{pmatrix}$$

Ziel dieser Methoden ist es, den kleinsten Eigenwert von A möglichst gut zu approximieren.

In der Praxis tritt mitunter der Fall auf, daß das Element $a_{n,n-1}^{(k)}$ näherungsweise verschwindet, d.h.

$$A^{(k)} = \left(\begin{array}{c|c} A_1^{(k)} & * \\ \hline & a_{n,n}^{(k)} \end{array}\right)$$

mit einer Hessenbergmatrix $A_1^{(k)}$. Dieser Fall heißt **Deflation** und $a_{n,n}^{(k)}$ ist dann ein Eigenwert von $A^{(k)}$ bzw. A. Deswegen genügt es, das QR-Verfahren nur noch auf die kleinere Matrix $A_1^{(k)}$ anzuwenden.

Analog kann verfahren werden, wenn andere Subdiagonalelemente (näherungsweise) zu Null werden, so daß $A^{(k)}$ zerfällt in

$$A^{(k)} = \left(\begin{array}{c|c} A_1^{(k)} & * \\ \hline & A_2^{(k)} \end{array}\right)$$

mit Hessenbergmatrizen $A_1^{(k)}$ und $A_2^{(k)}$. Da das charakteristische Polynom von $A^{(k)}$ sich in diesem Fall aus dem Produkt der charakteristischen Polynome von $A_1^{(k)}$ und $A_2^{(k)}$ ergibt, genügt es, das QR-Verfahren getrennt auf $A_1^{(k)}$ und $A_2^{(k)}$ anzuwenden.

Beispiel 1.5.10

Betrachte wiederum die symmetrische Matrix

$$A = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}.$$

A besitzt die Eigenwerte 2, 6 sowie den zweifachen Eigenwert 4.
Wir wenden das QR-Verfahren mit Rayleigh-Quotienten-Shift an. Nach nur einer Iteration erhalten wir

$$A^{(1)} = \begin{pmatrix} 4.0000 & 2.0000 & -0.0000 & 0.0000 \\ 2.0000 & 4.0000 & -0.0000 & -0.0000 \\ 0 & 0.0000 & 4.0000 & -0.0000 \\ 0 & -0.0000 & -0.0000 & 4.0000 \end{pmatrix}, \quad V^{(1)} = \begin{pmatrix} 0 & -0.7071 & 0.2357 & -0.6667 \\ 0.7071 & -0.0000 & -0.6667 & -0.2357 \\ 0.7071 & -0.0000 & 0.6667 & 0.2357 \\ 0 & -0.7071 & -0.2357 & 0.6667 \end{pmatrix}.$$

Der rechte untere 2×2-Block in $A^{(1)}$ ist eine Diagonalmatrix und liefert gerade den 2-fachen Eigenwert 4. Die letzten beiden Spalten von $V^{(1)}$ sind zugehörige Eigenvektoren. Das Problem zerfällt in zwei Teilprobleme und wir können den QR-Algorithmus nun anwenden auf die Teilmatrix

$$A_1^{(1)} = \begin{pmatrix} 4.0000 & 2.0000 \\ 2.0000 & 4.0000 \end{pmatrix}.$$

Das Verfahren mit Rayleigh-Quotienten-Shift führt hier nicht weiter, sondern stagniert. Das QR-verfahren ohne Shift liefert

$$A^{(10)} = \begin{pmatrix} 6.0000 & 0.0001 \\ 0.0001 & 2.0000 \end{pmatrix}, \quad V^{(10)} = \begin{pmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{pmatrix}.$$

Übungsaufgabe: *Bestimmen Sie Eigenvektoren zu den Eigenwerten 6 und 2 mit Hilfe der Matrizen des QR-Verfahrens.*

1.6 Reduktionsverfahren

Die Idee der Reduktionsmethoden besteht darin, eine Matrix $A \in \mathbb{C}^{n\times n}$ durch eine Folge von Ähnlichkeitstransformationen (1.6) auf Tridiagonal- oder Hessenbergform zu transformieren und Eigenwerte und Eigenvektoren der transformierten Matrix anschließend mit dem QR-Verfahren zu berechnen. Gelingt es, Eigenwerte und Eigenvektoren von $\hat{A}$ zu berechnen, so sind die Eigenvektoren von A gegeben durch die Transformation

$$\hat{A}\hat{x} = \lambda\hat{x},\ \hat{x} \neq 0 \qquad \Leftrightarrow \qquad AT\hat{x} = \lambda T\hat{x},\ T\hat{x} \neq 0,$$

d.h. ist $\hat{x}$ Eigenvektor von $\hat{A}$, so ist $x = T\hat{x}$ Eigenvektor von A.

1.6.1 Das Verfahren von Lanczos

In diesem Abschnitt stellen wir das Lanczos-Verfahren zur Reduktion einer hermiteschen Matrix auf Tridiagonalgestalt vor, welches besonders für große, dünn besetzte Matrizen geeignet ist.

Das Lanczos-Verfahren basiert auf der iterativen Berechnung einer orthonormalen Basis der **Krylov-Unterräume**

$$K_m(q,A) := span\{q, Aq, A^2q, \ldots, A^{m-1}q\}, \qquad m \geq 1,\ K_0(q,A) = \{0\}$$

für einen gegebenen orthonormalen Vektor q mit $\|q\|_2 = 1$ und einer hermiteschen Matrix $A = A^* \in \mathbb{C}^{n\times n}$.

In Schritt i des Verfahrens werden eine tridiagonale Matrix

$$J^{(i)} = \begin{pmatrix} \delta_1 & \gamma_2 & & \\ \gamma_2 & \delta_2 & \ddots & \\ & \ddots & \ddots & \gamma_i \\ & & \gamma_i & \delta_i \end{pmatrix} \in \mathbb{C}^{i\times i}$$

und eine unitäre Matrix $Q^{(i)} = (q^1, \ldots, q^i) \in \mathbb{C}^{n\times i}$ konstruiert mit

$$AQ^{(i)} = Q^{(i)}J^{(i)} + \gamma_{i+1}q^{i+1}e_i^\top, \qquad e_i = (0,\ldots,0,1)^\top \in \mathbb{R}^i. \tag{1.13}$$

Diese Iteration wird solange durchgeführt, bis erstmals $\gamma_{i_0+1} = 0$ für einen Index i_0 gilt. In diesem Fall verschwindet der störende Term $\gamma_{i_0+1}q^{i_0+1}e_{i_0}^\top$ und es gilt $AQ^{(i_0)} = Q^{(i_0)}J^{(i_0)}$. Wegen

$$J^{(i_0)}x = \lambda x,\ x \neq 0 \qquad \Rightarrow \qquad AQ^{(i_0)}x = Q^{(i_0)}J^{(i_0)}x = \lambda Q^{(i_0)}x$$

ist jeder Eigenwert λ von $J^{(i_0)}$ auch Eigenwert von A. Die Umkehrung gilt allerdings nur, wenn $i_0 = n$ gilt und das Verfahren nicht vorzeitig mit einem Index $i_0 < n$ abbricht (beachte die Dimensionen von $Q^{(i)}$ und $J^{(i)}$!). Bricht das Verfahren nicht vorzeitig ab, so ist A ähnlich zur Tridiagonalmatrix $J^{(n)}$.

Es bleibt zu klären, wie die Daten δ_i, γ_i und q^i berechnet werden können. Liest man (1.13) spaltenweise, so folgt mit der Konvention $\gamma_1 q^0 := 0$ die Rekursionsformel

$$Aq^i = \gamma_i q^{i-1} + \delta_i q^i + \gamma_{i+1} q^{i+1}, \qquad i \geq 1. \tag{1.14}$$

Auflösen nach q^{i+1} und die Forderung der Normiertheit liefert für $\gamma_{i+1} \neq 0$ die Beziehungen

$$\begin{aligned} q^{i+1} &= \frac{1}{\gamma_{i+1}} r^i, && (1.15)\\ r^i &= Aq^i - \gamma_i q^{i-1} - \delta_i q^i, && (1.16)\\ \gamma_{i+1} &= \|r^i\|_2. && (1.17) \end{aligned}$$

Die gewünschte Orthonormalität der Vektoren q^i liefert weiter

$$(q^i)^* A q^i = \gamma_i (q^i)^* q^{i-1} + \delta_i (q^i)^* q^i + \gamma_{i+1} (q^i)^* q^{i+1} = \delta_i. \tag{1.18}$$

Rein formal können wir den obigen Prozess mit einem Vektor q mit $\|q\|_2 = 1$ starten und erhalten folgenden Algorithmus, wobei noch ein Hilfsvektor $u^i := Aq^i - \gamma_i q^{i-1}$ verwendet wird, so daß $r^i = u^i - \delta_i q^i$ und $\delta_i = (q^i)^* u^i$ gelten.

Algorithmus 1.6.1 **(Lanczos)**
Input: $A \in \mathbb{C}^{n\times n}$ *hermitesch,* $q \in \mathbb{C}^n$ *mit* $\|q\|_2 = 1$.
Output: i_0 *mit* $\gamma_{i_0} = 0$, δ_i, γ_i, $i = 1, \ldots, i_0 - 1$.

(0) Setze $v := q$, $r := 0$, $\gamma_1 = 1$, $i = 1$.

(1) Falls $\gamma_i = 0$, *STOP.*

(2) Falls $i \neq 1$, *setze*

$$\begin{aligned} t &:= v,\\ v &:= \frac{1}{\gamma_i} r,\\ r &:= -\gamma_i t \end{aligned}$$

(3) Setze

$$\begin{aligned} r &:= Av + r,\\ \delta_i &:= v^* r,\\ r &:= r - \delta_i v. \end{aligned}$$

(4) Setze $i_0 := i$, $i := i + 1$, $\gamma_i := \|r\|_2$. *Gehe zu (1).*

Beachte, daß nach Schritt (3) stets $v = q^i$ gilt. Pro Iteration beträgt der Aufwand $6n$ zuzüglich einer Matrix-Vektor-Multiplikation Av. Insbesondere für große, dünn besetzte Matrizen ist der Aufwand für Av abhängig von der Anzahl der Nicht-Nullelemente und daher sehr kostengünstig.
Der folgende Satz stellt den Zusammenhang zwischen der Zahl i_0 und der Dimension des Krylovraums $K_m(q, A)$ her.

Satz 1.6.2

Sei m_0 der größte Index m, so daß $\dim(K_m(q, A)) = m$ gilt (die Vektoren $q, Aq, \ldots, A^{m_0-1}q$ sind somit linear unabhängig).
Dann gilt:

(a) Der im Lanczos-Alorithmus berechnete Wert i_0 ist gleich m_0.

(b) Die durch (1.14)-(1.18) definierten Vektoren $q^1, \ldots, q^{m_0}$ definieren eine Orthonormalbasis von $K_{m_0}(q, A)$.

Beweis: Wir zeigen die Behauptung durch Induktion nach j.
Für $j = 1$ ist $q^1 = q$ wegen $\|q\|_2 = 1$ eine Orthonormalbasis von $K_1(q, A) = span\{q\}$.
Seien nun für ein $j \geq 1$ orthonormale Vektoren $q^1, \ldots, q^j$ mit (1.14)-(1.18),

$$span\{q^1, \ldots, q^i\} = K_i(q, A) \qquad \forall i \leq j$$

und $r^i \neq 0$ für $i < j$ gegeben.
Zu zeigen sind die Behauptungen für $j+1$, falls $r_j \neq 0$ gilt (andernfalls wäre $\gamma_{j+1} = 0$ und der Algorithmus terminiert). Ist $r_j \neq 0$, so ist $\gamma_{j+1} \neq 0$ und δ_j und q^{j+1} sind wohldefiniert. Nach Konstruktion gilt $\|q^{j+1}\|_2 = 1$.
Wir zeigen, daß q^{j+1} orthogonal zu q^i, $i \leq j$, ist.
Mit (1.14), (1.18) gilt für $i = j$ nach Induktionsannahme:

$$\gamma_{j+1}(q^j)^* q^{j+1} = (q^j)^* A q^j - \delta_j (q^j)^* q^j = 0.$$

Für $i = j - 1$ gilt

$$\gamma_{j+1}(q^{j-1})^* q^{j+1} = (q^{j-1})^* A q^j - \gamma_j (q^{j-1})^* q^{j-1} = (q^{j-1})^* A q^j - \gamma_j$$

und wegen

$$A q^{j-1} = \gamma_{j-1} q^{j-2} + \delta_{j-1} q^{j-1} + \gamma_j q^j$$

folgt aus der Orthonormalität von q^j und q^i für $i \leq j$ sofort $(Aq^{j-1})^* q^j = \bar{\gamma}_j = \gamma_j$ und somit $(q^{j-1})^* q^{j+1} = 0$, wobei $A = A^*$ ausgenutzt wurde.

Für $i < j-1$ folgt mit (1.14) in analoger Weise

$$\gamma_{j+1}(q^i)^* q^{j+1} = (q^i)^* A q^j = (Aq^i)^* q^j = 0.$$

Dies zeigt die Orthogonalität der Vektoren $q^1, \dots, q^{j+1}$.
Für $i \le j$ gilt

$$span\{q^1, \dots, q^i\} = K_i(q, A) \subseteq K_j(q, A).$$

Damit gilt auch $Aq^j \in K_{j+1}(q, A)$. Aus (1.15) folgt $q^{j+1} \in span\{q^{j-1}, q^j, Aq^j\} \subseteq K_{j+1}(q, A)$. Insgesamt gilt also $span\{q^1, \dots, q^{j+1}\} \subseteq K_{j+1}(q, A)$ und, da $q^1, \dots, q^{j+1}$ orthonormal sind,

$$K_{j+1}(q, A) = span\{q^1, \dots, q^{j+1}\}.$$

Insbesondere gilt deshalb auch $j+1 \le m_0 = \max_m \dim(K_m(q, A))$ und somit $i_0 \le m_0$. Andererseits gilt nach Definition von i_0

$$Aq^{i_0} \in span\{q^{i_0-1}, q^{i_0}\} \subseteq span\{q^1, \dots, q^{i_0}\} = K_{i_0}(q, A).$$

Wegen

$$Aq^i \in span\{q^1, \dots, q^{i+1}\} = K_{i+1}(q, A) \subseteq K_{i_0}(q, A)$$

für $i < i_0$ folgt $AK_{i_0}(q, A) \subseteq K_{i_0}(q, A)$. Also ist $K_{i_0}(q, A)$ ein A-invarianter Unterraum und es gilt $i_0 \ge m_0$, weil $K_{m_0}(q, A)$ der erste A-invariante Teilraum unter den $K_m(q, A)$ ist. Damit ist $i_0 = m_0$ gezeigt. □

Theoretisch kann das Lanczos-Verfahren mit einem Index $i_0 < n$ abbrechen. In der Praxis wird der Fall $\gamma_{i_0+1} = 0$ jedoch auf Grund von Rundungsfehlern in der Regel nicht auftreten.
Die vom Verfahren berechneten Vektoren q^i sind theoretisch orthonormal, was in der Praxis allerdings wiederum durch Rundungsfehler schnell verloren gehen kann. Es gibt Varianten des Verfahrens, die die berechneten Vektoren in jedem Schritt mit einem Gram-Schmidt'schen Verfahren re-orthonormalisieren.
Die Eigenwerte der Matrix $J^{(i_0)}$ können effizient mit dem QR-Verfahren aus Abschnitt 1.5 berechnet werden. Eine alternative Methode basiert auf der Berechnung der Nullstellen des charakteristischen Polynoms von $J^{(i_0)}$, welches für hermitesche Tridiagonalmatrizen rekursiv und effizient ausgewertet werden kann. Die Berechnung der Nullstellen kann mit dem Bisektionsverfahren oder dem Newtonverfahren erfolgen. (→ **Übungsaufgabe**!)

Beispiel 1.6.3

Betrachte die symmetrische Matrix

$$A = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}.$$

A besitzt die Eigenwerte 2, 6 *sowie den zweifachen Eigenwert* 4.
Das Lanczos-Verfahren mit $q = (1,0,0,0)^\top$ *bricht mit* $i_0 = 3$ *(also* $\gamma_4 = 0$*) ab und berechnet eine Zerlegung* $AQ^{(3)} = Q^{(3)}J^{(3)}$ *mit*

$$J^{(3)} = \begin{pmatrix} 4 & \sqrt{2} & 0 \\ \sqrt{2} & 4 & \sqrt{2} \\ 0 & \sqrt{2} & 4 \end{pmatrix}, \qquad Q^{(3)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1/\sqrt{2} & 0 \\ 0 & -1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Die Matrix $J^{(3)}$ *besitzt die Eigenwerte* $2, 4, 6$.

Bemerkung 1.6.4

Es gibt eine Variante des Verfahrens für nicht hermitesche Matrizen, welche auf das BiCGSTAB-Verfahren führt.

1.6.2 Reduktion auf Hessenbergform

Ziel ist es, eine allgemeine Matrix $A \in \mathbb{C}^{n\times n}$ auf sogenannte Hessenberg-Gestalt zu transformieren, vgl. Definition 1.5.7.
Im folgenden verwenden wir unitäre Householder-Matrizen $T^{(i)}$ in (1.6). Wir nehmen an, daß das Verfahren startend mit $A^{(0)} = A$ für ein $0 \le i < n-1$ fortgeschritten ist bis

$$A^{(i)} = \left(\begin{array}{ccccc|ccc} a_{1,1}^{(i)} & a_{1,2}^{(i)} & \cdots & a_{1,i}^{(i)} & a_{1,i+1}^{(i)} & a_{1,i+2}^{(i)} & \cdots & a_{1,n}^{(i)} \\ a_{2,1}^{(i)} & a_{2,2}^{(i)} & \cdots & a_{2,i}^{(i)} & a_{2,i+1}^{(i)} & a_{2,i+2}^{(i)} & \cdots & a_{2,n}^{(i)} \\ & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & a_{i,i-1}^{(i)} & a_{i,i}^{(i)} & a_{i,i+1}^{(i)} & a_{i,i+2}^{(i)} & \cdots & a_{i,n}^{(i)} \\ & & & a_{i+1,i}^{(i)} & a_{i+1,i+1}^{(i)} & a_{i+1,i+2}^{(i)} & \cdots & a_{i+1,n}^{(i)} \\ \hline & & & & a_{i+2,i+1}^{(i)} & a_{i+2,i+2}^{(i)} & \cdots & a_{i+2,n}^{(i)} \\ & & & & \vdots & \vdots & \ddots & \vdots \\ & & & & a_{n,i+1}^{(i)} & a_{n,i+2}^{(i)} & \cdots & a_{n,n}^{(i)} \end{array}\right) =: \left(\begin{array}{c|c|c} A_1^{(i)} & & A_2^{(i)} \\ \hline 0 & a^{(i)} & A_3^{(i)} \end{array}\right), \tag{1.19}$$

wobei $A_1^{(i)} \in \mathbb{C}^{(i+1)\times(i+1)}$ Hessenbergmatrix sei und $A_2^{(i)} \in \mathbb{C}^{(i+1)\times(n-i-1)}$, $A_3^{(i)} \in \mathbb{C}^{(n-i-1)\times(n-i-1)}$, $0 \neq a^{(i)} = (a_{i+2,i+1}^{(i)}, \ldots, a_{n,i+1}^{(i)})^\top \in \mathbb{C}^{n-i-1}$.
Die ersten i Spalten von $A^{(i)}$ liegen also bereits in Hessenbergform vor.
Da $a^{(i)} \neq 0$ vorausgesetzt ist, können wir eine unitäre Householdermatrix

$$H_i = I - \frac{2}{v_i^* v_i} v_i v_i^* \in \mathbb{C}^{(n-i-1)\times(n-i-1)}$$

konstruieren mit $H_i a^{(i)} = k_i e_i$, wobei $e_i = (1,0,\ldots,0)^\top \in \mathbb{R}^{n-i-1}$ und $k_i = -\frac{a_{i+2,i+1}^{(i)}}{|a_{i+2,i+1}^{(i)}|}\|a^{(i)}\|_2$ gelten, wobei die Konvention $0/0 = 1$ im Fall $a_{i+2,i+1}^{(i)} = 0$ beachtet wird. Die Matrix H_i

ist definiert durch den Vektor

$$v_i = \frac{1}{\|a^{(i)}\|_2} a^{(i)} + \frac{a^{(i)}_{i+2,i+1}}{|a^{(i)}_{i+2,i+1}|} e_i.$$

Definiere die unitäre Matrix $T^{(i+1)}$ mit $(T^{(i+1)})^* = (T^{(i+1)})^{-1} = T^{(i+1)}$ durch

$$T^{(i+1)} = \left(\begin{array}{c|c} I & 0 \\ \hline 0 & H_i \end{array}\right)$$

und

$$\begin{aligned}
A^{(i+1)} &= (T^{(i+1)})^{-1} \cdot A^{(i)} \cdot T^{(i+1)} = T^{(i+1)} \cdot A^{(i)} \cdot T^{(i+1)} \\
&= \left(\begin{array}{c|c} I & 0 \\ \hline 0 & H_i \end{array}\right) \left(\begin{array}{c|c} A_1^{(i)} & A_2^{(i)} \\ \hline 0 \;|\; a^{(i)} & A_3^{(i)} \end{array}\right) \left(\begin{array}{c|c} I & 0 \\ \hline 0 & H_i \end{array}\right) \\
&= \left(\begin{array}{c|c} A_1^{(i)} & A_2^{(i)} \\ \hline 0 \;|\; H_i a^{(i)} & H_i A_3^{(i)} \end{array}\right) \left(\begin{array}{c|c} I & 0 \\ \hline 0 & H_i \end{array}\right) \\
&= \left(\begin{array}{c|c} A_1^{(i)} & A_2^{(i)} H_i \\ \hline 0 \;|\; H_i a^{(i)} & H_i A_3^{(i)} H_i \end{array}\right) \\
&= \left(\begin{array}{c|c} A_1^{(i)} & A_2^{(i)} H_i \\ \hline 0 \;|\; k_i e_i & H_i A_3^{(i)} H_i \end{array}\right).
\end{aligned}$$

Ist $a^{(i)} = 0$, so kann $H_i = I$ gewählt werden. Die Matrix $A^{(i+1)}$ hat formal dieselbe Struktur wie $A^{(i)}$, wobei nun die ersten $i+1$ Spalten in Hessenbergform vorliegen. Das Verfahren kann nun mit der Matrix $H_i A_3^{(i)} H_i$ fortgesetzt werden. Insgesamt wird die Matrix $A = A^{(0)}$ so in $n-2$ Schritten auf eine ähnliche Hessenbergmatrix

$$\hat{A} = T^{-1} \cdot A \cdot T, \qquad T = T^{(1)} \cdot T^{(2)} \cdots T^{(n-2)},$$

transformiert, welche dieselben Eigenwerte wie A besitzt. Zusammenfassend erhalten wir

Algorithmus 1.6.5 **(Reduktion auf Hessenbergform)**
Gegeben: Matrix $A \in \mathbb{C}^{n\times n}$, Setze $A^{(0)} := A$.
Output: Matrix $A^{(n-2)} \in \mathbb{C}^{n\times n}$ in Hessenbergform, welche ähnlich zu A ist.

Für $i = 0, 2, \ldots, n-2$:

Seien $A_1^{(i)}$, $A_2^{(i)}$, $A_3^{(i)}$ und $a^{(i)}$ definiert gemäß (1.19).

Falls $\|a^{(i)}\|_2 = 0$, setze $H_i = I$, $A_2^{(i+1)} = A_2^{(i)}$ und $A_3^{(i+1)} = A_3^{(i)}$.

Andernfalls setze

$$\begin{aligned} v &= \frac{1}{\|a^{(i)}\|_2} a^{(i)} + \frac{a^{(i)}_{i+2,i+1}}{|a^{(i)}_{i+2,i+1}|} e_i \\ \beta &= \frac{2}{v^* v}, \\ w &= \beta (A_3^{(i)})^* v, \\ y &= \beta A_2^{(i)} v, \\ z &= \beta A_3^{(i)} v, \end{aligned}$$

und setze $A_2^{(i+1)} = A_2^{(i)} - y \cdot v^*$ *and* $A_3^{(i+1)} = A_3^{(i)} - v \cdot w^* - z \cdot v^* + \beta w^* v \cdot v \cdot v^*$.

Der Aufwand (Multiplikationen und Divisionen) im i-ten Schleifendurchlauf beträgt

$$\begin{aligned} v &: 2(n-i-1) \\ \beta &: n-i-1 \\ w, z &: 2(n-i-1)(n-i-1) + 2(n-i-1) \\ y &: (i+1)(n-i-1) + (i+1) \\ A_2^{(i)} &: (i+1)(n-i-1) \\ A_3^{(i)} &: 4(n-i-1)(n-i-1) + (n-i-1) \end{aligned}$$

Insgesamt ergibt sich ein Aufwand von

$$\sum_{i=0}^{n-2} (6(n-i-1)(n-i-1) + (n-i-1)(6+2(i+1)) + i + 1) = \frac{7}{3} n^3 + \mathcal{O}(n^2).$$

Spezialfall:
Ist A hermitesch, so liefert die obige Prozedur eine hermitesche Tridiagonalmatrix $\hat{A}$.
Denn: Im hermiteschen Fall gilt beginnend mit $i = 0$ induktiv $A_3^{(i)} = (A_3^{(i)})^*$ und

$$A_2^{(i)} = \left(\begin{array}{c|c} 0 & a^{(i)} \end{array} \right)^* = \begin{pmatrix} 0 \\ (a^{(i)})^* \end{pmatrix}$$

und weiter

$$A_2^{(i)} H_i = \begin{pmatrix} 0 \\ (a^{(i)})^* \end{pmatrix} H_i = \begin{pmatrix} 0 \\ (a^{(i)})^* H_i \end{pmatrix} = \begin{pmatrix} 0 \\ (H_i a^{(i)})^* \end{pmatrix} = \begin{pmatrix} 0 \\ \bar{k}_i e_i^* \end{pmatrix}.$$

Damit ist auch $A^{(i+1)}$ wieder hermitesch und die ersten $i+1$ Spalten besitzen Tridiagonalstruktur.

Beispiel 1.6.6

Betrachte die symmetrische Matrix

$$A = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}.$$

A besitzt die Eigenwerte 2, 6 sowie den zweifachen Eigenwert 4.
Das Reduktionsverfahren liefert

$$\hat{A} = A^{(3)} = \begin{pmatrix} 4 & \sqrt{2} & 0 & 0 \\ \sqrt{2} & 4 & -\sqrt{2} & 0 \\ 0 & -\sqrt{2} & 4 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}, \qquad T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & -1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 0 & 0 & -1 & 0 \end{pmatrix}.$$

Die Matrix $\hat{A}$ ist tridiagonal und symmetrisch und besitzt ebenfalls die Eigenwerte $2, 4, 4, 6$.

Bemerkung 1.6.7

Analog zur Verwendung von Householder-Transformationen können auch Gauss- Eliminationsschritte mit Pivoting verwendet werden, um eine Matrix auf Hessenbergform zu transformieren. Diese sind kostengünstiger, aber weniger stabil als Householder-Transformationen.

Kapitel 2

Einschrittverfahren zur Lösung von Anfangswertproblemen

Differentialgleichungen treten in vielen Bereichen auf. Wir betrachten einige einführende Beispiele.

Beispiel 2.0.1 **(Aufstieg einer Rakete)**

Eine Rakete mit der Anfangsmasse $m(0) = m_0$ startet zum Zeitpunkt $t = 0$ aus der Ruhelage auf der Erdoberfläche mit Anfangshöhe $h(0) = 0$ und wird senkrecht nach oben geschossen, vgl. Abbildung 2.1.

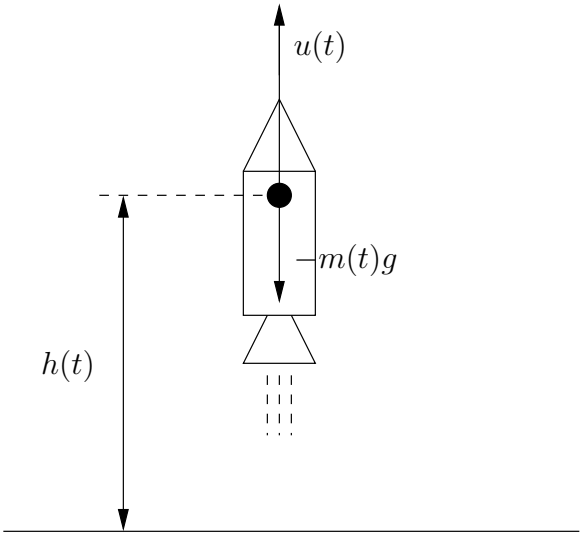


Abbildung 2.1: Das Raketen-Problem: Vertikaler Aufstieg einer Rakete.

Die Funktion $u(t) \geq 0$ sei eine gegebene Funktion, die die Schubkraft zum Zeitpunkt t angibt. Die zeitliche Höhenbewegung der Rakete ist durch das Newton'sche Gesetz

$$\textit{Kraft} = \textit{Masse} \times \textit{Beschleunigung}$$

gegeben:

$$m(t)h''(t) = \underbrace{-m(t)g}_{\textit{Erdanziehung}} + \underbrace{u(t)}_{\textit{Schub}} .$$

Wir nehmen an, daß der Treibstoffverbrauch proportional zur Schubkraft ist, so daß die Änderung der Masse durch die Differentialgleichung

$$m'(t) = -c \cdot u(t)$$

gegeben ist. Luftwiderstände werden vernachlässigt.
Insgesamt erhalten wir folgendes Anfangswertproblem erster Ordnung, welches den Aufstieg der Rakete beschreibt:

$$\begin{aligned} h'(t) &= v(t), & h(0) &= 0, \\ v'(t) &= -g + \tfrac{u(t)}{m(t)}, & v(0) &= 0, \\ m'(t) &= -c \cdot u(t), & m(0) &= m_0. \end{aligned}$$

Hierin sind $g = 9.81$ [m/s] und $c > 0$ gegebene Konstanten und $u(t) \geq 0$ eine gegebene Funktion.

Das folgende Beispiel liefert ein Modell für die Radaufhängung eines Autos, welches häufig für Komfortuntersuchungen verwendet wird.

Beispiel 2.0.2 (Gedämpfte Schwingung)
Eine von außen durch eine Kraft $u(t)$ angeregte Schwingung genügt unter der Voraussetzung, daß die rückstellende Federkraft proportional zur Auslenkung $x(t)$ ist, der Bewegungsgleichung

$$mx''(t) + dx'(t) + cx(t) = u(t),$$

wobei $c > 0$ die Federkonstante, $d > 0$ die Dämpferkonstante und $m > 0$ die Masse des Schwingers bezeichnen, vgl. Abbildung 2.2. Dies ist ein sehr einfaches Modell für die Radaufhängung eines Autos. Zum Zeitpunkt $t = 0$ sei das System (etwa durch eine Bodenunebenheit) ausgelenkt: $x(0) = x_0$ und $x'(0) = x_0'$.

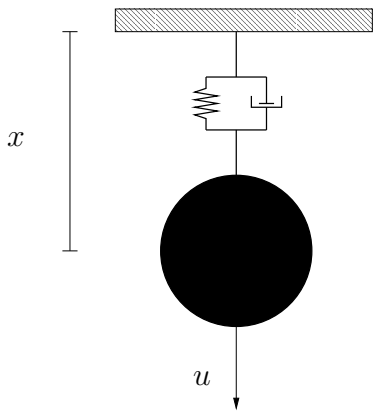


Abbildung 2.2: Modell einer Radaufhängung.

Die Bewegung eines Autos kann durch das folgende Einspurmodell beschrieben werden.

Beispiel 2.0.3 (Modell eines Autos)

Ein beliebtes (einfaches) Automodell ist das sogenannte Einspurmodell. Es unterliegt gewissen vereinfachenden Annahmen (Wank- und Nickbewegungen werden vernachlässigt), so daß die Räder einer Achse zu einem virtuellen Rad in der Mitte zusammengefaßt werden können. Zudem kann angenommen werden, daß der Schwerpunkt auf Fahrbahnhöhe liegt, so daß es ausreicht, die Bewegung in der Ebene zu untersuchen.

Das folgende Fahrzeugmodell besitzt vier Eingangsgrößen (Steuerungen), die vom Fahrer gewählt werden können: Die Lenkwinkelgeschwindigkeit $w_\delta(t)$, die Bremskraft $F_B(t)$, den Gang $\mu(t) \in \{1,2,3,4,5\}$ und die Gaspedalstellung $\phi(t) \in [0,1]$. Die Konfiguration des Fahrzeugs ist in Abbildung 2.3 dargestellt.

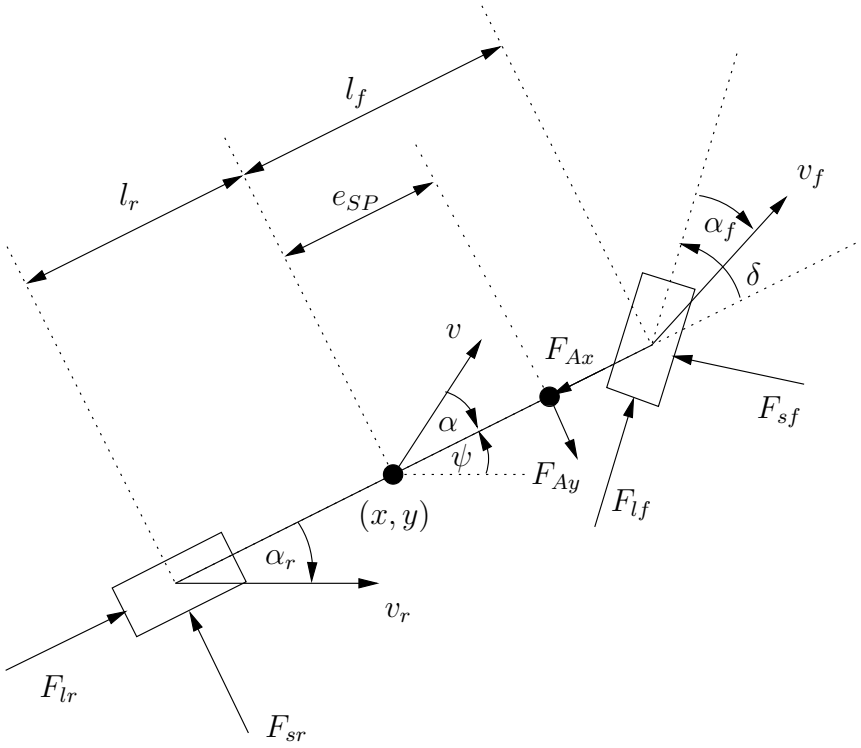


Abbildung 2.3: Geometrische Beschreibung des Einspurmodells.

Es bezeichnen:

- (x,y)*: Schwerpunkt des Autos*
- v, v_f, v_r*: Geschwindigkeit des Autos bzw. des Vorder- bzw. Hinterrads*
- δ, β, ψ*: Lenkwinkel, Schräglaufwinkel, Gierwinkel*

- α_f, α_r: *Schlupfwinkel am Vorder- bzw. Hinterrad*
- F_{sf}, F_{sr}: *Reifenseitenkräfte vorne und hinten*
- F_{lf}, F_{lr}: *Reifenlängskräfte vorne und hinten*
- l_f, l_r, e_{SP}: *Abstände*
- F_{Ax}, F_{Ay}: *Luftwiderstand in x-Richtung bzw. in y-Richtung*
- m: *Masse*

Die Bewegung des Einspurmodells genügt den folgenden Differentialgleichungen, wobei die Abhängigkeit von der Zeit t weggelassen wurde:

$$\begin{aligned}
x' &= v\cos(\psi-\beta),\\
y' &= v\sin(\psi-\beta),\\
v' &= \frac{1}{m}\Big[(F_{lr}-F_{Ax})\cos\beta + F_{lf}\cos(\delta+\beta) - (F_{sr}-F_{Ay})\sin\beta\\
&\qquad\qquad -F_{sf}\sin(\delta+\beta)\Big],\\
\beta' &= w_z - \frac{1}{m\cdot v}\Big[(F_{lr}-F_{Ax})\sin\beta + F_{lf}\sin(\delta+\beta)\\
&\qquad\qquad +(F_{sr}-F_{Ay})\cos\beta + F_{sf}\cos(\delta+\beta)\Big],\\
\psi' &= w_z,\\
w_z' &= \frac{1}{I_{zz}}\Big[F_{sf}\cdot l_f\cdot\cos\delta - F_{sr}\cdot l_r - F_{Ay}\cdot e_{SP} + F_{lf}\cdot l_f\cdot\sin\delta\Big],\\
\delta' &= w_\delta.
\end{aligned}$$

Die Reifenseitenkräfte hängen von den Schlupfwinkeln ab. Ein berühmtes Modell ist die 'magic formula' von Pacejka [PB93]:

$$\begin{aligned}
F_{sf}(\alpha_f) &= D_f\sin\left(C_f\arctan\left(B_f\alpha_f - E_f\left(B_f\alpha_f - \arctan(B_f\alpha_f)\right)\right)\right),\\
F_{sr}(\alpha_r) &= D_r\sin\left(C_r\arctan\left(B_r\alpha_r - E_r\left(B_r\alpha_r - \arctan(B_r\alpha_r)\right)\right)\right),
\end{aligned}$$

vgl. Abbildung 2.4. $B_f, B_r, C_f, C_r, D_f, D_r, E_f, E_r$ *sind reifenabhängige Konstanten.*

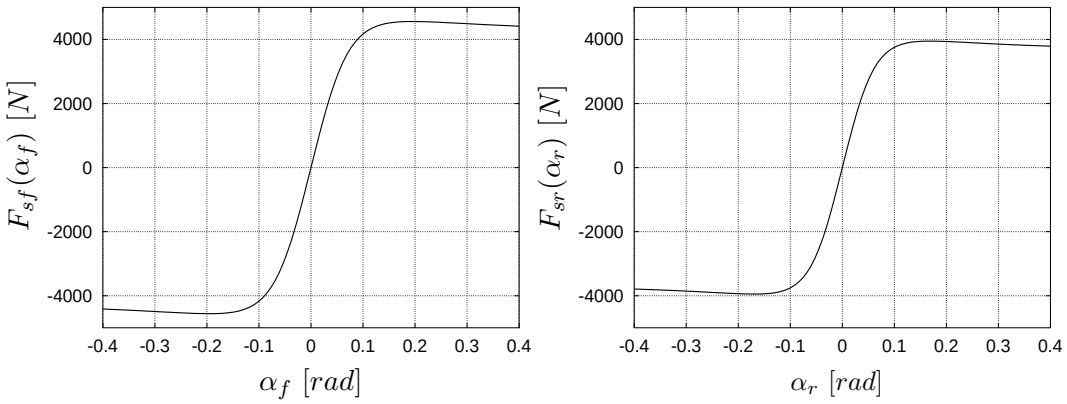


Abbildung 2.4: Reifenseitenkräfte am Vorder- (links) und Hinterrad (rechts) in Abhängigkeit vom Schlupfwinkel.

Die Schlupfwinkel berechnen sich zu

$$\alpha_f = \delta - \arctan\left(\frac{l_f\dot{\psi} - v\sin\beta}{v\cos\beta}\right), \qquad \alpha_r = \arctan\left(\frac{l_r\dot{\psi} + v\sin\beta}{v\cos\beta}\right).$$

Der Luftwiderstand beträgt

$$F_{Ax} = \frac{1}{2}\cdot c_w \cdot \rho \cdot A \cdot v^2,$$

wobei c_w den cw-Wert, ρ die Luftdichte und A die effektive Anströmfläche bezeichnen. Zur Vereinfachung trete kein Seitenwind auf: $F_{Ay} = 0$.
Unter der Annahme, daß das Auto Heckantrieb besitzt, ist die Reifenlängskraft am Vorderrad gegeben durch

$$F_{lf} = -F_{Bf} - F_{Rf},$$

wobei F_{Bf} die Bremskraft und F_{Rf} den Rollwiderstand am Vorderrad bezeichnet. Die Reifenlängskraft am Hinterrad lautet entsprechend

$$F_{lr} = \frac{M_{wheel}(\phi,\mu)}{R} - F_{Br} - F_{Rr}$$

wobei $M_{wheel}(\phi,\mu)$ das Antriebsmoment des Motors am Hinterrad bezeichnet.
Wir nehmen an, daß die Bremskraft F_B, die vom Fahrer gesteuert wird, wie folgt verteilt ist:

$$F_{Bf} = \frac{2}{3}F_B, \qquad F_{Br} = \frac{1}{3}F_B.$$

Weitere Details zur Modellierung von $M_{wheel}(\phi,\mu)$, F_{Rf} und F_{Rr}, sowie Parameterwerte können in Gerdts [Ger05] nachgelesen werden.

Allgemein erlauben mechanische Systeme folgende Beschreibung.

Beispiel 2.0.4 (Mechanische Mehrkörpersysteme)
Die Lagrangeschen Bewegungsgleichungen für mechanische Mehrkörpersysteme lauten

$$M(q(t))q''(t) = f(t, q(t), q'(t))$$

mit dem Vektor q der verallgemeinerten Lagekoordinaten, der symmetrischen und positiv definiten Massenmatrix M und dem Vektor f der verallgemeinerten Kräfte und Momente (resultieren z.B. aus dem Impuls- und Drallsatz). Dieses System kann durch Einführung der verallgemeinerten Geschwindigkeiten $v(t) := q'(t)$ und Invertierung von M auf ein System erster Ordnung transformiert werden:

$$\begin{aligned} q'(t) &= v(t), \\ v'(t) &= M(q(t))^{-1} \cdot f(t, q(t), v(t)). \end{aligned}$$

Es gibt Softwarepakete, die diese Bewegungsgleichungen für sehr komplizierte Systeme automatisch generieren, z.B. SIMPACK oder ADAMS.

Differentialgleichungen treten auch in der Chemie auf, um beispielsweise chemische Reaktionen zu beschreiben.

Beispiel 2.0.5 (Chemische Reaktionen)
Das Substrat S reagiert mit einem Enzym E zu einem Komplex SE. Der Komplex SE seinerseits reagiert zu einem Produkt P und dem Enzym E. Dies liefert die chemischen Reaktionsgleichungen

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} SE \quad , \qquad SE \xrightarrow{k_2} P + E$$

Darin sind k_1, k_1 und k_2 Reaktionsgeschwindigkeiten. Das Massenwirkunsgesetz (Die Konzentrationsänderung ist proportional zum Produkt der Konzentrationen der Reaktionspartner) liefert das Differentialgleichungssystem

$$\begin{aligned} [S]' &= -k_1 \cdot [E] \cdot [S] + k_{-1} \cdot [SE], \\ [E]' &= -k_1 \cdot [E] \cdot [S] + k_{-1} \cdot [SE] + k_2 \cdot [SE], \\ [SE]' &= k_1 \cdot [E] \cdot [S] - k_{-1} \cdot [SE] - k_2 \cdot [SE], \\ [P]' &= k_2 \cdot [SE], \end{aligned}$$

vgl. Murray [Mur93], Michaelis und Menten.

Auch in der Biologie werden Differentialgleichungen verwendet, um beispielsweise Räuber-Beute-Modelle bzw. das Zusammenspiel von Spezies zu beschreiben.

Beispiel 2.0.6 (Räuber-Beute-Modell von Lotka-Volterra in der Biologie)
Sei $N(t)$ die Beute-Population zum Zeitpunkt t und $P(t)$ die Räuber-Population zum

Zeitpunkt t. Ein einfaches Modell zur Beschreibung der Entwicklung der Populationen ist gegeben durch das Differentialgleichungssystem

$$\begin{aligned} N'(t) &= a \cdot N(t) - b \cdot N(t) \cdot P(t), \\ P'(t) &= c \cdot N(t) \cdot P(t) - d \cdot P(t), \end{aligned}$$

vgl. Murray [Mur93]. Darin bedeuten die Konstanten a, b, c, d folgendes

$a > 0$	*Geburtenrate der Beute*
$b > 0$	*Verlustrate der Beute durch Räuber*
	(falls $P = 0$ und $a > 0$ erfolgt ungehemmtes Wachstum)
$c > 0$	*Wachstumsrate der Räuberpopulation*
	(falls $N = 0$ und $d > 0$ sterben die Räuber aus)
$d > 0$	*Sterberate der Räuber.*

Wie die obigen Beispiele zeigen, treten (gewöhnliche) Differentialgleichungen in nahezu allen Disziplinen auf. Viele Phänomene (Ausbreitung von Wellen, Wärmeleitung, elektrische Potentiale, Strömungen) werden darüber hinaus durch partielle Differentialgleichungen beschrieben, auf die wir in dieser Vorlesung nicht eingehen können. Anders als bei gewöhnlichen Differentialgleichungen, bei denen die gesuchten Funktionen nur von einer unabhängigen Variable abhängen (in den Beispielen war dies die Zeit t), hängen die gesuchten Funktionen bei partiellen Differentialgleichungen von mindestens zwei unabhängigen Variablen ab (in der Regel sind dies Zeit und Ort).

2.1 Anfangswertprobleme

Im Rahmen dieser Vorlesung beschäftigen wir uns mit gewöhnlichen Differentialgleichungen der Form

$$x'(t) = f(t, x(t)), \quad x(t) \in \mathbb{R}^n,\ t \in [a, b],$$

mit einer gegebenen Funktion $f : [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$. Hierin interpretieren wir t als Zeit, $x(t)$ als Zustand des Systems mit Ableitung $x'(t)$ zum Zeitpunkt t. Gibt man noch einen Anfangswert $x(a) = x_a \in \mathbb{R}^n$ vor, entsteht das folgende Problem.

Problem 2.1.1 **(Anfangswertproblem (AWP))**
Seien $x_0 \in \mathbb{R}^n$, $f : [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$, $a, b \in \mathbb{R}$ gegeben. Bestimme in $[a, b]$ eine stetig differenzierbare Funktion $x : [a, b] \to \mathbb{R}^n$ mit

$$\begin{aligned} x'(t) &= f(t, x(t)), && (2.1) \\ x(a) &= x_a. && (2.2) \end{aligned}$$

Das Problem bei dieser Aufgabenstellung ist, daß durch (2.1) zwar eine Beziehung zwischen der Ableitung x' und der Funktion x bekannt ist, die Funktion x ist aber – bis auf den Anfangswert in (2.2) – unbekannt. Mit Ausnahme einiger weniger Spezialfälle, z.B. lineare homogene Differentialgleichungssysteme mit konstanten Koeffizienten, kann die Lösung des Anfangswertproblems **nicht** explizit angegeben werden. Es werden daher numerische Methoden zur Approximation einer Lösung von AWP benötigt.
Unter einer Lösung des AWP verstehen wir eine auf $[a,b]$ stetig differenziebare Funktion x, die (2.1) und (2.2) für alle $t \in [a,b]$ erfüllt. Man kann den Begriff einer Lösung jedoch auch in abgeschwächter Form betrachten, indem x als Lösung angesehen wird, falls

$$x(t) = x_a + \int_a^t f(s, x(s))ds$$

für alle $t \in [a,b]$ gilt. Dies würde sogenannte absolutstetige Funktionen als Lösung zulassen. Dies sind stetige Funktionen, deren Ableitung im Lebesgue-Sinne integrierbar ist. Die Integraldarstellung zeigt auch die enge Verbindung von Anfangswertproblemen und Integration, was auch numerisch durch numerische Integrationsverfahren ausgenutzt werden kann.

Bemerkung 2.1.2

Ist $f : [a,b] \times \mathbb{R}^n \to \mathbb{R}^n$ eine k-mal stetig differenzierbare Funktion (bzgl. beider Argumente), so ist die Lösung x des AWP mindestens $(k+1)$-mal stetig differenzierbar.

2.2 Existenz- und Eindeutigkeit

Ohne Beweis zitieren wir einige Standardresultate zur Existenz- und Eindeutigkeit von Lösungen des AWP. Hierbei ist es wichtig, zwischen lokaler und globaler Existenz- und Eindeutigkeit einer Lösung zu unterscheiden. Während sich der Begriff „lokal“ nur auf eine (möglicherweise kleine) Umgebung der Stelle a bezieht, bezieht sich „global“ auf das gesamte Intervall $[a,b]$, in dem eine Lösung von AWP gesucht wird.
Die lokale Existenz einer Lösung ist garantiert, falls f stetig ist:

Satz 2.2.1 (Existenzsatz von Peano)

(a) **Lokale Existenz:**
Sei $D \subseteq \mathbb{R} \times \mathbb{R}^n$ ein Gebiet und $(a, x_a) \in D$. Ferner sei f stetig in D. Dann besitzt das Anfangswertproblem 2.1.1 mindestens eine lokal um a definierte Lösung, die bis zum Rand von D fortgesetzt werden kann.

(b) **Globale Existenz:**
Sei f stetig und beschränkt im Streifen $[a,b]\times\mathbb{R}^n$. Dann existiert mindestens eine in $[a,b]$ differenzierbare Lösung des Anfangswertproblems 2.1.1.

Beweis: siehe Walter [Wal90], Paragraphen 7,10. □

(Lokale) Eindeutigkeit erhält man, wenn f zusätzlich Lipschitz-stetig ist:

Satz 2.2.2 (Existenz- und Eindeutigkeitssatz)

(a) **Lokale Existenz und Eindeutigkeit:**
Sei $D\subseteq\mathbb{R}\times\mathbb{R}^n$ ein Gebiet und $(a,x_a)\in D$. f sei stetig in D und genüge einer lokalen Lipschitz-Bedingung bzgl. x, d.h. zu jedem $(\hat{t},\hat{x})\in D$ gibt es eine Umgebung $U_\varepsilon(\hat{t},\hat{x})$, so daß eine Konstante $L=L(\hat{t},\hat{x})$ existiert mit

$$\|f(t,y)-f(t,z)\|\le L\|y-z\| \qquad \forall (t,y),(t,z)\in D\cap U_\varepsilon(\hat{t},\hat{x}).$$

Dann besitzt das Anfangswertproblem 2.1.1 genau eine lokal um a definierte Lösung, die sich bis zum Rand von D fortsetzen läßt.

(b) **Globale Existenz und Eindeutigkeit:**
Sei f stetig im Streifen $[a,b]\times\mathbb{R}^n$ und (global!) lipschitzstetig bzgl. x, d.h. es gebe eine Konstante L mit

$$\|f(t,y)-f(t,z)\|\le L\|y-z\| \qquad \forall (t,y),(t,z)\in [a,b]\times\mathbb{R}^{n_x}.$$

Dann gibt es genau eine Lösung des Anfangswertproblems 2.1.1 in $[a,b]$.

Beweis: siehe Walter [Wal90], Paragraph 10. □

Wir illustrieren die Begriffe an drei Beispielen.

Beispiel 2.2.3

- **Mehrere Lösungen:**
  Betrachte

$$x'(t)=-2\sqrt{1-x(t)}=:f(x(t)), \qquad x(0)=1.$$

  Man überprüft leicht, daß

$$\begin{aligned} x(t) &= 1,\\ x(t) &= 1-t^2 \end{aligned}$$

  Lösungen sind. f erfüllt keine Lipschitz-Bedingung in $x=1$.

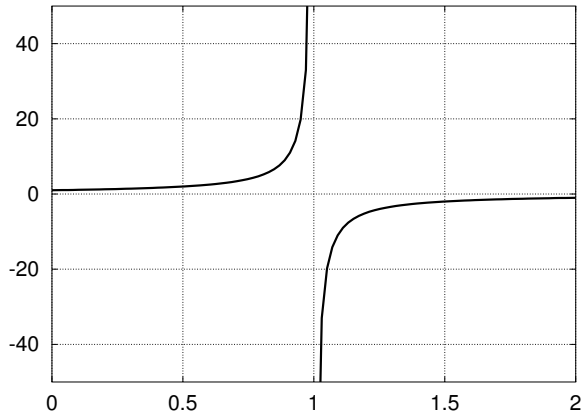


Abbildung 2.5: Lokal eindeutige Lösung des Anfangswertproblems $x'(t) = x(t)^2$, $x(0) = 1$.

- **Eindeutige, lokal definierte Lösung:**
  Betrachte

  $$x'(t) = x(t)^2 =: f(x(t)), \quad x(0) = x_a \in \mathbb{R}.$$

  $f(x) = x^2$ *ist* **nicht** *lipschitzstetig auf* $\mathbb{R}$*, da* $f'(x) = 2x$ *für* $x \to \pm\infty$ *unbeschränkt ist. Die Funktion ist nur* **lokal** *lipschitzstetig. Die lokal eindeutige Lösung lautet*

  $$x(t) = -\frac{x_a}{x_a t - 1}.$$

  Für $t = 1/x_a$*,* $x_a \neq 0$ *ist sie unbeschränkt. Für* $x_a > 0$ *ist sie definiert in* $(-\infty, 1/x_a)$*. Für* $x_a < 0$ *ist sie definiert in* $(1/x_a, \infty)$*. Für* $x_a = 0$ *ist* $x(t) \equiv 0$ *auf ganz* $\mathbb{R}$*. Abbildung 2.5 zeigt die Lösung für* $x(0) = x_a = 1$*.*

- **Global eindeutige Lösung:**
  Betrachte

  $$x'(t) = \lambda x(t) =: f(x(t)), \qquad x(0) = x_a \in \mathbb{R}$$

  mit $\lambda \in \mathbb{R}$*.* f *ist global lipschitzstetig. Die eindeutige Lösung lautet*

  $$x(t) = x_a \cdot \exp(\lambda t).$$

Wir werden später sehen, daß die Lipschitz-Stetigkeit von f auch bei numerischen Approximationsverfahren eine wichtige Rolle spielt ($\to$ Stabilität von Diskretisierungsverfahren).

2.3 Diskretisierung mittels Einschrittverfahren

Allen folgenden Diskretisierungsverfahren zur approximativen Lösung von Anfangswertproblemen liegt eine Diskretisierung des Zeitintervalls $[a,b]$ in Form eines **Gitters**

$$\mathbb{G}_N := \{a = t_0 < t_1 < t_2 < \ldots < t_{N-1} < t_N = b\} \tag{2.3}$$

zu Grunde. Die **Gitterpunkte** t_i, $i = 0,1,\ldots,N$, $N \in \mathbb{N}$, und die **Schrittweiten** $h_i = t_{i+1} - t_i$, $i = 0,1,\ldots,N-1$, werden zur Vereinfachung häufig äquidistant mit $h_i = h = (b-a)/N$ für $i = 0,\ldots,N-1$ gewählt. Insbesondere in praktischen Anwendungen ist es in der Regel jedoch effizienter, die Schrittweiten (bzw. die Gitterpunkte) adaptiv an die Lösung des AWP anzupassen (was nicht so einfach ist, da die Lösung ja nicht bekannt ist). Wie dies konkret realisiert werden kann, werden wir im Abschnitt über Schrittweitensteuerung sehen.
Ziel der Diskretisierungsverfahren ist es nun, eine sogenannte **Gitterfunktion** $x_N : \mathbb{G}_N \to \mathbb{R}^n$ mit $t \mapsto x_N(t)$ für $t \in \mathbb{G}_N$ zu konstruieren, die die gesuchte Lösung $\hat{x}$ des AWP zumindest an den Gitterpunkten im Gitter $\mathbb{G}_N$ in einem noch zu definierenden Sinne approximiert, d.h. für alle Gitterpunkte $\mathbb{G}_N$ soll gelten

$$x_N(t_i) \approx \hat{x}(t_i), \qquad i = 0,\ldots,N.$$

Da Gitterfunktionen nur auf dem Gitter $\mathbb{G}_N$ definiert sind, und somit durch $N+1$ Werte beschrieben werden, schreiben wir zur Abkürzung auch einfach

$$x_i := x_N(t_i), \qquad i = 0,\ldots,N.$$

Die Interpretation der Werte x_i, $i = 0,\ldots,N$, als Gitterfunktion ist jedoch nützlich, da letztendlich zur Bestimmung des Approximationsfehlers die auf ganz $[a,b]$ definierte Funktion x mit der nur an den Gitterpunkten definierten Funktion x_N verglichen werden muß. Dies kann im Prinzip auf zwei Arten erfolgen: Entweder wird die Gitterfunktion x_N auf das ganze Intervall $[a,b]$ fortgesetzt, etwa durch Splineinterpolation, oder die Lösung x des AWP wird auf das Gitter $\mathbb{G}_N$ eingeschränkt. Wir folgen dem zweiten Ansatz. Aus mathematischer Sicht ist nun insbesondere interessant, ob die Folge von Gitterfunktionen $\{x_N\}_{N\in\mathbb{N}}$ beim Grenzübergang $N \to \infty$ gegen die Lösung des AWP konvergiert. Hierzu wird es i.a. notwendig sein, daß die sogenannte **Maschenweite (Feinheit des Gitters)**

$$h = \max_{i=0,\ldots,N-1} h_i$$

gegen Null konvergiert, so daß jeder Bereich von $[a,b]$ auch tatsächlich durch Gitterpunkte abgedeckt ist. Bevor wir uns mit der Konvergenz von Diskretisierungsverfahren beschäftigen, betrachten wir zunächst gängige Einschrittverfahren.

2.3.1 Das Eulerverfahren

Das wohl einfachste Verfahren ist das explizite Eulerverfahren. Es arbeitet die Gitterpunkte im Gitter $\mathbb{G}_N$ schrittweise ab beginnend mit dem gegebenen Anfangswert $\hat{x}(a) = x_a$, welcher den Wert der Gitterfunktion $x_0 = x_N(t_0) = x_a$ festlegt. In $t_0 = a$ ist die Ableitung der Lösung bekannt, sie ist nämlich gerade $\hat{x}'(t_0) = f(t_0, x_0)$. An der Stelle $t_1 = a + h_0$ kann die Lösung (bei hinreichender Glätte) durch Taylorentwicklung approximiert werden durch

$$\hat{x}(t_1) \approx \hat{x}(t_0) + \hat{x}'(t_0)(t_1 - t_0) = x_0 + h_0 f(t_0, x_0).$$

Dieser Wert wird als Approximation am Gitterpunkt t_1 verwendet:

$$x_1 = x_N(t_1) := x_0 + h_0 f(t_0, x_0).$$

An der Näherung x_1 kann nun wieder die Funktion f ausgewertet und analog eine Approximation x_2 am Gitterpunkt t_2 bestimmt werden, wobei man sich jetzt allerdings nicht mehr in der exakten Lösung befindet, sondern nur in der Approximation. Dies wird solange wiederholt, bis der Zeitpunkt $t_N = b$ erreicht wird, vgl. Abbildung 2.6.

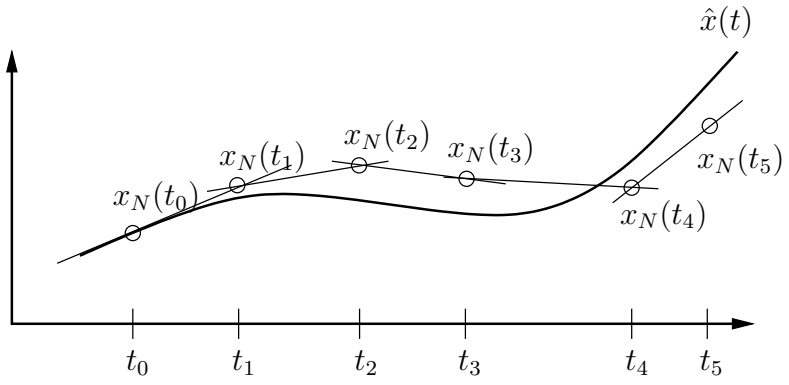


Abbildung 2.6: Idee des expliziten Eulerverfahrens: Approximation durch lokale Linearisierung.

Insgesamt entsteht das explizite Eulerverfahren.

Algorithmus 2.3.1 (Explizites Eulerverfahren)

(0) Gegeben sei das AWP 2.1.1. Wähle ein Gitter $\mathbb{G}_N$ gemäß (2.3).

(1) Setze $x_0 = x_N(t_0) = x_a$.

(2) Berechne für $i = 0, 1, \ldots, N-1$:

$$x_N(t_{i+1}) \quad = \quad x_N(t_i) + h_i f(t_i, x_N(t_i)).$$

Bei der Berechnung des Wertes $x_N(t_{i+1})$ im expliziten Eulerverfahren wird nur auf den Wert $x_N(t_i)$ am vorangehenden Gitterpunkt zurück gegriffen. Daher handelt es sich um ein **Einschrittverfahren**. Anders als Einschrittverfahren greifen **Mehrschrittverfahren** nicht nur auf $x_N(t_i)$ zurück, sondern sie verwenden auch $x_N(t_{i-1}), x_N(t_{i-2}), \ldots$.

Bemerkung 2.3.2 (Alternative Motivationen des Eulerverfahrens)

- *Das Eulerverfahren kann auch dadurch motiviert werden, daß die Ableitung $x'(t)$ in AWP durch die finite Differenzenapproximation*

$$x'(t_i) \approx \frac{x(t_{i+1}) - x(t_i)}{h_i}$$

  ersetzt wird und die rechte Seite in t_i ausgewertet wird.

- *Eine weitere Motivation basiert auf der Darstellung*

$$x(t_{i+1}) - x(t_i) = \int_{t_i}^{t_{i+1}} f(t, x(t))dt$$

  der Differentialgleichung. Das explizite Eulerverfahren entsteht, wenn das Integral durch $(t_{i+1} - t_i)f(t_i, x(t_i))$ approximiert wird.

Anstatt die Lösung im Zeitpunkt t_{i+1} um t_i nach Taylor zu entwickeln, kann umgekehrt auch folgende Taylorentwicklung verwendet werden:

$$x_0 = \hat{x}(t_0) \approx \hat{x}(t_1) + \hat{x}'(t_1)(t_0 - t_1) = x_1 - h_0 f(t_1, x_1)$$

bzw.

$$x_1 = x_0 + h_0 f(t_1, x_1).$$

Diese Vorschrift führt auf das implizite Eulerverfahren.

Algorithmus 2.3.3 (Implizites Eulerverfahren)

(0) Gegeben sei das AWP 2.1.1. Wähle ein Gitter $\mathbb{G}_N$ gemäß (2.3).

(1) Setze $x_0 = x_N(t_0) = x_a$.

(2) Berechne für $i = 0, 1, \ldots, N-1$:

$$x_N(t_{i+1}) = x_N(t_i) + h_i f(t_{i+1}, x_N(t_{i+1})). \tag{2.4}$$

Die Schwierigkeit beim impliziten Eulerverfahren besteht darin, daß (2.4) eine **nichtlineare Gleichung** für den unbekannten Wert $x_N(t_{i+1})$ darstellt. Diese nichtlineare Gleichung kann durch Fixpunktiteration oder mit dem Newtonverfahren gelöst werden. Da dies in jedem Schritt erfolgen muß, steigt der Aufwand im Vergleich zum expliziten Eulerverfahren stark an. Dennoch hat das implizite Eulerverfahren seine Existenzberechtigung, da es bessere Stabilitätseigenschaften als das explizite Eulerverfahren besitzt (→ A-Stabilität und steife Differentialgleichungen).
Formal ist auch das implizite Eulerverfahren ein Einschrittverfahren, da zur Berechnung von $x_N(t_{i+1})$ nur der Wert $x_N(t_i)$ verwendet wird.

2.3.2 Runge-Kutta-Verfahren

Bereits die erste Approximation x_1 beim Eulerverfahren ist lediglich eine Approximation an die exakte Lösung $\hat{x}(t_1)$. Dieser Approximationsfehler pflanzt sich mit jedem Schritt fort, da f in der Approximation ausgewertet wird. Runge-Kutta-Verfahren versuchen nun, den Approximationsfehler zu reduzieren, indem potenziell bessere Approximationen berechnet werden. Wir illustrieren dies für das Verfahren von Heun. Hierzu greifen wir wiederum auf die Integraldarstellung

$$x(t_{i+1}) - x(t_i) = \int_{t_i}^{t_{i+1}} f(t, x(t))dt$$

zurück. Approximation des Integrals durch die Trapezsumme liefert

$$x(t_{i+1}) - x(t_i) = \int_{t_i}^{t_{i+1}} f(t, x(t))dt \approx \frac{h_i}{2} \left(f(t_i, x(t_i)) + f(t_{i+1}, x(t_{i+1})) \right).$$

Da $x(t_{i+1})$ wie beim impliziten Eulerverfahren implizit durch diese Gleichung gegeben ist, approximieren wir $x(t_{i+1})$ im letzten Term durch einen expliziten Eulerschritt und erhalten

$$x(t_{i+1}) - x(t_i) \approx \frac{h_i}{2} \left(f(t_i, x(t_i)) + f(t_{i+1}, x(t_i) + h_i f(t_i, x(t_i))) \right).$$

Dies ist nun ein explizites Verfahren, das sogenannte Verfahren von Heun.

Algorithmus 2.3.4 (Verfahren von Heun)

(0) Gegeben sei das AWP 2.1.1. Wähle ein Gitter $\mathbb{G}_N$ gemäß (2.3).

(1) Setze $x_0 = x_N(t_0) = x_a$.

(2) Berechne für $i = 0, 1, \dots, N-1$:

$$\begin{aligned} k_1 &= f(t_i, x_N(t_i)), \\ k_2 &= f(t_i + h_i, x_N(t_i) + h_i k_1), \\ x_N(t_{i+1}) &= x_N(t_i) + \frac{h_i}{2}(k_1 + k_2). \end{aligned}$$

Da die Trapezregel eine bessere Approximation als die beim Eulerverfahren verwendete Riemann-Summe an das Integral ist, kann man erwarten, daß das Verfahren von Heun bessere Approximationen als das explizite Eulerverfahren liefert. Dies werden wir später formal untersuchen. Das Verfahren von Heun und auch die beiden Eulerverfahren sind Beispiele für Runge-Kutta-Verfahren, die wie folgt definiert sind.

Definition 2.3.5 (Runge-Kutta-Verfahren)
Sei $\mathbb{G}_N$ ein Gitter gemäß (2.3). Für $s \in \mathbb{N}$ und Konstanten b_j, c_j, a_{ij}, $i, j = 1, \dots, s$ ist das **s-stufige Runge-Kutta-Verfahren** *definiert durch*

$$\begin{aligned} x_N(t_{i+1}) &= x_N(t_i) + h_i \sum_{j=1}^{s} b_j k_j(t_i, x_N(t_i); h_i) \\ k_j(t_i, x_N(t_i); h_i) &= f\left(t_i + c_j h_i, x_N(t_i) + h_i \sum_{\ell=1}^{s} a_{j\ell} k_\ell(t_i, x_N(t_i); h_i) \right), \quad j = 1, \dots, s. \end{aligned}$$

Die Funktionen k_j, $j = 1, \dots, s$, heißen auch **Stufenableitungen**.
Das s-stufige Runge-Kutta-Verfahren heißt **explizit**, *falls $a_{ij} = 0$ für $j \geq i$ gilt. Andernfalls heißt das Verfahren* **implizit**.

Bemerkung 2.3.6 (Äquivalente Darstellung)
Eine äquivalente Darstellung des s-stufigen Runge-Kutta-Verfahrens ist gegeben durch

$$\begin{aligned} x_N(t_{i+1}) &= x_N(t_i) + h_i \sum_{j=1}^{s} b_j f(t_i + c_j h_i, \eta_{i+1}^{(j)}), \\ \eta_{i+1}^{(j)} &= x_N(t_i) + h_i \sum_{\ell=1}^{s} a_{j\ell} f(t_i + c_\ell h, \eta_{i+1}^{(l)}), \quad j = 1, \dots, s. \end{aligned}$$

Die Werte $\eta_{i+1}^{(j)}$ heißen **Stufenapproximationen**.

Bei expliziten Runge-Kutta-Verfahren hängt die Stufenableitung k_j nur von den Werten $k_1, \dots, k_{j-1}$ ab. Die Stufenableitungen können daher beginnend mit k_1 schrittweise berechnet werden.

Implizite Runge-Kutta-Verfahren, wie z.B. das implizite Eulerverfahren, erfordern in jedem Schritt die Lösung eines $n \cdot s$-dimensionalen nichtlinearen Gleichungssystems.
Runge-Kutta-Verfahren werden durch das sogenannte **Butcher-Tableau** beschrieben:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \quad \Leftrightarrow \quad \begin{array}{c|c} c & A \\ \hline & b^\top \end{array}$$

Für explizite Verfahren ist nur die linke untere Dreiecksmatrix exklusive der Diagonalen ungleich Null, so daß **explizite Verfahren** folgende Struktur des Butcher-Tableaus besitzen:

$$\begin{array}{c|ccccc} c_1 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

Beispiel 2.3.7 **(Klassisches Runge-Kutta-Verfahren)**
Das **klassische Runge-Kutta-Verfahren** *ist ein 4-stufiges, explizites Runge-Kutta-Verfahren mit dem Butcher-Tableau*

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

und entspricht der Vorschrift

$$\begin{aligned} x_N(t_{i+1}) &= x_N(t_i) + h_i \left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 \right) \\ k_1 &= f(t_i, x_N(t_i)), \\ k_2 &= f\left(t_i + \frac{h_i}{2}, x_N(t_i) + \frac{h_i}{2}k_1 \right), \\ k_3 &= f\left(t_i + \frac{h_i}{2}, x_N(t_i) + \frac{h_i}{2}k_2 \right), \\ k_4 &= f\left(t_i + h_i, x_N(t_i) + h_i k_3 \right). \end{aligned}$$

Beispiel 2.3.8 **(Radau-IIA-Verfahren)**

Ein gebräuchliches implizites, 2-stufiges Verfahren ist das **Radau-IIA-Verfahren***:*

$$\begin{array}{c|cc} 1/3 & 5/12 & -1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array}$$

mit

$$\begin{aligned} x_N(t_{i+1}) &= x_N(t_i) + h_i \left(\frac{3}{4} k_1 + \frac{1}{4} k_2 \right) \\ k_1 &= f\left(t_i + \frac{h_i}{3}, x_N(t_i) + h_i \left(\frac{5}{12} k_1 - \frac{1}{12} k_2 \right) \right), \\ k_2 &= f\left(t_i + h_i, x_N(t_i) + h_i \left(\frac{3}{4} k_1 + \frac{1}{4} k_2 \right) \right). \end{aligned}$$

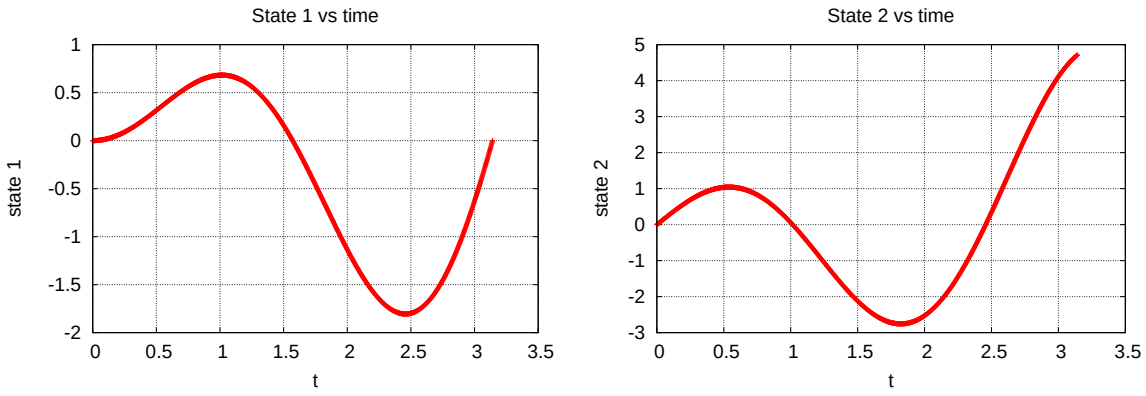
Beispiel 2.3.9

Das Anfangswertproblem

$$\begin{aligned} x_1'(t) &= x_2(t), \qquad x_1(0) = 0, \\ x_2'(t) &= -4x_1(t) + 3\cos(2t), \qquad x_2(0) = 0, \end{aligned}$$

besitzt – wie man leicht nachrechnet – die Lösung

$$\begin{aligned} \hat{x}_1(t) &= \frac{3}{4} t \sin(2t), \\ \hat{x}_2(t) &= \frac{3}{4} \sin(2t) + \frac{3}{2} t \cos(2t). \end{aligned}$$



Wir berechnen für $[a,b] = [0,\pi]$ *und Gitter* $\mathbb{G}_N$ *mit* $N = 5, 10, 20, 40, 80, 160, 320, 640, 1280$ *Approximationen* x_N *mit dem expliziten Eulerverfahren, dem Verfahren von Heun und*

dem klassischen Runge-Kutta-Verfahren und vergleichen die Fehler

$$\|x_N - \hat{x}\|_\infty = \max_{t_i \in \mathbb{G}_N} \|x_N(t_i) - \hat{x}(t_i)\|_2.$$

Zusätzlich schätzen wir die Fehlerordnung p aus $\|x_N - \hat{x}\|_\infty = C(1/N)^p$ gemäß

$$\frac{\|x_N - \hat{x}\|_\infty}{\|x_{2N} - \hat{x}\|_\infty} = \frac{1/N^p}{1/(2N)^p} = 2^p \quad \Rightarrow \quad p = \log_2\left(\frac{\|x_N - \hat{x}\|_\infty}{\|x_{2N} - \hat{x}\|_\infty}\right) = \frac{\log\left(\frac{\|x_N - \hat{x}\|_\infty}{\|x_{2N} - \hat{x}\|_\infty}\right)}{\log(2)}.$$

Es ergeben sich folgende Werte:

N	EULER		HEUN		RK	
	ERR	P	ERR	P	ERR	P
5	0.1892E+02	0.0000E+00	0.6117E+01	0.0000E+00	0.3301E+00	0.0000E+00
10	0.6456E+01	0.1551E+01	0.1024E+01	0.2578E+01	0.2184E-01	0.3918E+01
20	0.2808E+01	0.1201E+01	0.2453E+00	0.2062E+01	0.1327E-02	0.4040E+01
40	0.1374E+01	0.1031E+01	0.6058E-01	0.2018E+01	0.8146E-04	0.4026E+01
80	0.6604E+00	0.1057E+01	0.1506E-01	0.2008E+01	0.5041E-05	0.4014E+01
160	0.3219E+00	0.1037E+01	0.3753E-02	0.2005E+01	0.3136E-06	0.4007E+01
320	0.1587E+00	0.1020E+01	0.9364E-03	0.2003E+01	0.1955E-07	0.4004E+01
640	0.7879E-01	0.1010E+01	0.2339E-03	0.2001E+01	0.1221E-08	0.4002E+01
1280	0.3925E-01	0.1005E+01	0.5845E-04	0.2001E+01	0.7624E-10	0.4001E+01

Man sieht sehr schön, daß das Eulerverfahren die Ordnung 1, das Heun-Verfahren die Ordnung 2 und das klassische Runge-Kutta-Verfahren die Ordnung 4 besitzt. Für $N = 1280$ erreicht das explizite Eulerverfahren einer Genauigkeit von etwa $4 \cdot 10^{-2}$. Hierzu benötigt es 1280 Funktionsauswertungen von f. Das Heun-Verfahren übertrifft diese Genauigkeit bereits mit $N = 80$, wofür es nur 160 Funktionsauswertungen benötigt. Das klassische Runge-Kutta-Verfahren erreicht die Genauigkeit des Eulerverfahrens bereits für $N = 10$, wofür es lediglich 40 Funktionsauswertungen benötigt.

2.3.3 Allgemeine Einschrittverfahren

Die Eulerverfahren und Runge-Kutta-Verfahren lassen sich in die allgemeine Klasse der Einschrittverfahren einordnen.

Definition 2.3.10 (Einschrittverfahren)

Sei $\Phi : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ eine gegebene stetige Funktion und $\mathbb{G}_N$ ein Gitter gemäß 2.3. Das Verfahren

$$x_N(t_0) = x_a, \tag{2.5}$$

$$x_N(t_{i+1}) = x_N(t_i) + h_i \Phi(t_i, x_N(t_i), h_i), \quad i = 0, \dots, N-1, \tag{2.6}$$

zur Approximation einer Lösung des AWP 2.1.1 heißt **Einschrittverfahren**. *Die Funktion Φ heißt* **Inkrementfunktion**.

Der Name Einschrittverfahren ist darin begründet, daß Φ nur vom vorhergehenden Wert $x_N(t_i)$ abhängt. Bei Mehrschrittverfahren kann Φ zusätzlich von $x_N(t_{i-1}), x_N(t_{i-2}), \dots$ abhängen.

Beispiel 2.3.11 (Inkrementfunktion einiger Einschrittverfahren)
Die Inkrementfunktion des expliziten Eulerverfahrens ist gegeben durch

$$\Phi(t,x,h) = f(t,x),$$

die des Heun-Verfahrens lautet

$$\Phi(t,x,h) = \frac{1}{2}\left(f(t,x) + f(t+h, x+hf(t,x))\right).$$

Die Inkrementfunktion des impliziten Eulerverfahrens ist formal implizit gegeben durch

$$\Phi(t,x,h) = f(t+h, x+h\Phi(t,x,h)).$$

Die Inkrementfunktion eines Runge-Kutta-Verfahrens lautet je nach Darstellung des Runge-Kutta-Verfahrens

$$\Phi(t,x,h) = \sum_{j=1}^{s} b_j k_j(t,x;h)$$

bzw.

$$\Phi(t,x,h) = \sum_{j=1}^{s} b_j f(t+c_j h, \eta_{i+1}^{(j)}),$$

wobei die k_j bzw. $\eta^{(j)}$ bei impliziten Runge-Kutta-Verfahren wiederum implizit gegeben sind.

2.4 Konsistenz, Stabilität und Konvergenz von Einschrittverfahren

Wir beschränken uns im folgenden auf äquidistante Gitter $\mathbb{G}_N$ mit Schrittweite $h = \frac{b-a}{N}$ und betrachten das Einschrittverfahren (2.5)-(2.6).
Es bezeichnen $\hat{x}$ die exakte Lösung des Anfangswertproblems 2.1.1 und

$$\Delta_N : \{x : [a,b] \to \mathbb{R}^n\} \to \{x_N : \mathbb{G}_N \to \mathbb{R}^n\}, \qquad \Delta_N(x)(t) = x(t) \text{ für } t \in \mathbb{G}_N,$$

den Restriktionsoperator auf das Gitter $\mathbb{G}_N$. Auf dem Raum aller Gitterfunktionen ist durch

$$\|x_N\|_\infty = \max_{t_i \in \mathbb{G}_N} \|x_N(t_i)\|_2$$

eine Norm definiert. Damit kann der globale Fehler und der Begriff Konvergenz definiert werden.

Definition 2.4.1 (Globaler Fehler, Konvergenz)
Der **globale Fehler** $e_N : \mathbb{G}_N \to \mathbb{R}^n$ *ist definiert durch*

$$e_N := x_N - \Delta_N(\hat{x}).$$

Das Einschrittverfahren (2.5)-(2.6) heißt **konvergent**, *wenn*

$$\lim_{N\to\infty} \|e_N\|_\infty = 0.$$

Das Einschrittverfahren (2.5)-(2.6) besitzt die **Konvergenzordnung** p, *wenn*

$$\|e_N\|_\infty = \mathcal{O}\left(\frac{1}{N^p}\right) \qquad \text{für } N \to \infty.$$

Neben dem globalen Fehler ist der lokale Diskretisierungsfehler von großer Bedeutung.

Definition 2.4.2 **(Lokaler Diskretisierungsfehler, Konsistenz)**
Seien $\hat{x} \in \mathbb{R}^n$, $\hat{t} \in [a,b]$ und das Einschrittverfahren (2.5)-(2.6) gegeben. Es bezeichne y die Lösung des Anfangswertproblems

$$y'(t) = f(t, y(t)), \qquad y(\hat{t}) = \hat{x}.$$

Der **lokale Diskretisierungsfehler in** $(\hat{t}, \hat{x})$ *ist definiert durch*

$$\ell_h(\hat{t}, \hat{x}) := \frac{y(\hat{t}+h) - \hat{x}}{h} - \Phi(\hat{t}, \hat{x}, h).$$

Das Einschrittverfahren heißt **konsistent in einer Lösung x des AWP (2.1)-(2.2)** , *wenn*

$$\lim_{h\to 0} \left(\max_{t\in[a,b-h]} \|\ell_h(t, x(t))\| \right) = 0.$$

Das Einschrittverfahren besitzt die **Konsistenzordnung p in einer Lösung x des AWP (2.1)-(2.2)**, *wenn es eine von h unabhängige Konstante $C > 0$ und eine Konstante $h_0 > 0$ gibt mit*

$$\max_{t\in[a,b-h]} \|\ell_h(t, x(t))\| \le Ch^p \qquad \forall 0 < h \le h_0.$$

Anschaulich gibt der lokale Diskretisierungsfehler an, wie gut die exakte Lösung y des AWP mit Startpunkt in $(\hat{t}, \hat{x})$ das Einschrittverfahren erfüllt. Beachte, daß es für die Konsistenz **nicht** ausreicht, wenn der lokale Fehler $\|y(\hat{t}+h) - x_N(\hat{t}+h)\|$ für $h \to 0$ gegen Null strebt.

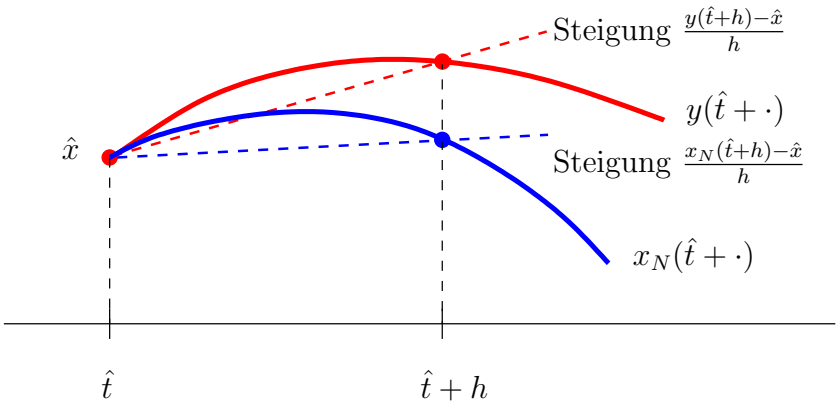


Abbildung 2.7: Lokaler Diskretisierungsfehler und Konsistenz: $x_N(\hat{t}+\cdot)$ bezeichnet die Lösung des Einschrittverfahrens als Funktion der Schrittweite h.

Bemerkung 2.4.3

- *Wegen*

$$\ell_h(\hat{t},\hat{x}) = \frac{y(\hat{t}+h) - \big(\hat{x} + h\Phi(\hat{t},\hat{x},h)\big)}{h} = \frac{y(\hat{t}+h) - x_N(\hat{t}+h)}{h}$$

  kann der lokale Diskretisierungsfehler auch als lokaler Fehler pro Schrittweite h (engl. local error per unit step) interpretiert werden. Beachte, daß der lokale Fehler im Zähler die Ordnung $p+1$ besitzt, falls das Verfahren die Konsistenzordnung p hat.

- *Konsistenz kann auch unabhängig von einer Lösung x des AWP definiert werden. Dazu muß etwa*

$$\lim_{h\to 0} \left(\sup_{(\hat{t},\hat{x})\in[a,b]\times D} \|\ell_h(\hat{t},\hat{x})\| \right) = 0$$

  gelten, wobei $D \subseteq \mathbb{R}^n$ eine geeignet gewählte Menge ist.

- *Wegen*

$$\lim_{h\to 0} \frac{x(t+h)-x(t)}{h} = x'(t) = f(t,x(t))$$

  ist Konsistenz gleichbedeutend mit der Bedingung

$$\lim_{h\to 0} \Phi(t,x(t),h) = f(t,x(t)),$$

  die gleichmäßig in t gelten muss.

Beispiel 2.4.4

Sei x Lösung des AWP (2.1)-(2.2). Diese erfüllt dann

$$\lim_{h\to 0}\frac{x(t+h)-x(t)}{h} = x'(t) = f(t,x(t)).$$

Betrachte das explizite Eulerverfahren

$$x_N(t_{i+1}) = x_N(t_i) + hf(t_i, x_N(t_i))$$

mit einer stetigen Funktion f. Dann gilt für den lokalen Diskretisierungsfehler

$$\lim_{h\to 0}\|\ell_h(t,x(t))\| = \lim_{h\to 0}\left\|\frac{x(t+h)-x(t)}{h} - f(t,x(t))\right\| = 0$$

gleichmäßig in t, da x' stetig im Kompaktum $[a,b]$ und nach dem Satz von Heine somit gleichmäßig stetig in $[a,b]$ ist, und das explizite Eulerverfahren ist konsistent.
Ist f sogar stetig differenzierbar (und x somit zweimal stetig differenzierbar), so liefert Taylorentwicklung von x um $t \in [a, b-h]$ mit $h > 0$ die Beziehung

$$\begin{aligned}
\|\ell_h(t,x(t))\| &= \left\|\frac{x(t+h)-x(t)}{h} - f(t,x(t))\right\| \\
&= \left\|\frac{x'(t)h + \frac{1}{2}\int_0^1 x''(t+sh)h^2 ds}{h} - f(t,x(t))\right\| \\
&= \left\|\frac{f(t,x(t))h + \frac{1}{2}\int_0^1 x''(t+sh)h^2 ds}{h} - f(t,x(t))\right\| \\
&\le Ch
\end{aligned}$$

mit $C = \max_{t\in[a,b]}\|x''(t)\|$. Damit gilt also

$$\max_{t\in[a,b-h]}\|\ell_h(t,x(t))\| \le Ch$$

und das explizite Eulerverfahren besitzt die Konsistenzordnung 1. Eine höhere Konsistenzordnung ist mit dem expliziten Eulerverfahren i.a. nicht zu erreichen.

Zum Nachweis der Konsistenzordnung für Runge-Kutta-Verfahren müssen die Lösung y und die rechte Seite f nach Taylor entwickelt werden. Da es sich bei f um eine vektorwertige Funktionen mit vektorwertigem Argument handelt, sind die Ableitungen i.a. Tensoren, d.h. die j-te Ableitung von f an der Stelle $z = (t,x) \in \mathbb{R}^{n+1}$ ist eine j-lineare Abbildung

$$f^{(j)}(z)\cdot(h_1,\dots,h_j) = \sum_{i_1,\dots,i_j=1}^{n+1}\frac{\partial^j f(z)}{\partial z_{i_1}\cdots\partial z_{i_j}} h_{1,i_1}\cdots h_{j,i_j}$$

mit $h_k \in \mathbb{R}^{n+1}$, $k = 1, \dots, j$. Die multivariate Taylorformel für eine $(p+1)$-mal stetig differenzierbare Funktion f lautet

$$f(z+h) = \sum_{j=0}^{p} \frac{1}{j!} f^{(j)}(z) \cdot \underbrace{(h, \dots, h)}_{j-mal} + R_{p+1}(z;h)$$

mit der Restglieddarstellung

$$R_{p+1}(z;h) = \frac{1}{(p+1)!} \int_0^1 f^{(p+1)}(z+th) \cdot \underbrace{(h, \dots, h)}_{(p+1)-mal} dt$$

und man kann hiermit wie im eindimensionalen Fall rechnen, vgl. [DB02, S. 138] und [Wlo71, S.174]. Ist $f^{(p+1)}$ beschränkt, so gilt

$$\|R_{p+1}(z;h)\| \le C\|h\|^{p+1} = \mathcal{O}(\|h\|^{p+1}), \qquad C = \frac{1}{(p+1)!} \max_{y \in [z, z+h]} \|f^{(p+1)}(y)\|.$$

Beispiel 2.4.5

Sei f zweimal stetig differenzierbar (und x somit dreimal stetig differenzierbar). Taylorentwicklung der Lösung x um $t \in [a, b-h]$ mit $h > 0$ liefert

$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2} x''(t) + \mathcal{O}(h^3) = x(t) + hf + \frac{h^2}{2} (f_t + f_x \cdot f) + \mathcal{O}(h^3),$$

wobei f und deren Ableitungen jeweils in $(t, x(t))$ ausgewertet werden. Die Inkrementfunktion des Heun-Verfahrens lautet

$$\Phi(t, x, h) = \frac{1}{2} \left(f(t,x) + f(t+h, x + hf(t,x)) \right).$$

Taylorentwicklung der rechten Seite liefert

$$f(t+h, x(t) + hf(t, x(t))) \;=\; f + h\,(f_t + f_x \cdot f) + \mathcal{O}(h^2),$$

wobei f und deren Ableitungen jeweils in $(t, x(t))$ ausgewertet werden. Lokaler Diskretisierungsfehler:

$$\begin{aligned} \|\ell_h(t, x(t))\| &= \left\| \frac{x(t+h) - x(t)}{h} - \Phi(t, x(t), h) \right\| \\ &= \left\| f + \frac{h}{2} (f_t + f_x \cdot f) + \mathcal{O}(h^2) - \frac{1}{2} \left(f + f + hf_t + hf_x \cdot f + \mathcal{O}(h^2) \right) \right\| \\ &= \mathcal{O}(h^2). \end{aligned}$$

Das Verfahren von Heun hat also die Konsistenzordnung 2, falls f zweimal stetig differenzierbar ist.

Beispiel 2.4.6

Bei hinreichender Glätte von f können mittels Taylorentwicklung die folgenden Ord-

nungsbedingungen für Runge-Kutta-Verfahren gezeigt werden, vgl. z.B. Strehmel und Weiner [SW95], S. 50:

$$
\begin{aligned}
p=1 \quad &: \quad \sum_{i=1}^{s} b_i = 1, \\
p=2 \quad &: \quad \sum_{i=1}^{s} b_i c_i = \frac{1}{2}, \\
p=3 \quad &: \quad \sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}, \quad \sum_{i,j=1}^{s} b_i a_{ij} c_j = \frac{1}{6}, \\
p=4 \quad &: \quad \sum_{i=1}^{s} b_i c_i^3 = \frac{1}{4}, \quad \sum_{i,j=1}^{s} b_i c_i a_{ij} c_j = \frac{1}{8}, \quad \sum_{i,j=1}^{s} b_i a_{ij} c_j^2 = \frac{1}{12}, \quad \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k = \frac{1}{24}.
\end{aligned}
$$

Hierbei ist die Gültigkeit der Knotenbedingungen

$$c_i = \sum_{j=1}^{s} a_{ij}, \qquad i = 1, \dots, s$$

vorausgesetzt worden.

Die Konsistenz des Einschrittverfahrens alleine ist nicht ausreichend, um auch dessen Konvergenz zeigen zu können. Zusätzlich zur Konsistenz wird noch die Stabilität benötigt.

Definition 2.4.7 (Stabilität)

Es seien $\{x_N\}_{N\in\mathbb{N}}$ Gitterfunktionen mit (2.5)-(2.6) und $h = (b-a)/N$, $N \in \mathbb{N}$. Desweiteren seien $\{y_N\}_{N\in\mathbb{N}}$ Gitterfunktionen $y_N : \mathbb{G}_N \to \mathbb{R}^n$ mit

$$
\begin{aligned}
\delta_N(t_0) &:= y_N(t_0) - x_a, \\
\delta_N(t_i) &:= \frac{y_N(t_i) - y_N(t_{i-1})}{h} - \Phi(t_{i-1}, y_N(t_{i-1}), h), \qquad i = 1, \dots, N.
\end{aligned}
$$

Die Funktion $\delta_N : \mathbb{G}_N \to \mathbb{R}^n$ wird als **Defekt** *von y_N bezeichnet.*
Das Einschrittverfahren heißt **stabil in** *$\{x_N\}_{N\in\mathbb{N}}$, falls es von N unabhängige Konstanten $S, R \geq 0$ gibt, so daß für fast alle $h = (b-a)/N$, $N \in \mathbb{N}$, folgendes gilt:*
Aus

$$\|\delta_N\|_\infty < R$$

folgt

$$\|y_N - x_N\|_\infty \leq S \|\delta_N\|_\infty.$$

Die Konstante R heißt **Stabilitätsschwelle** *und S heißt* **Stabilitätsschranke**.

Konsistenz und Stabilität sichern die Konvergenz des Verfahrens.

Satz 2.4.8 (Konvergenzsatz)

Das Einschrittverfahren sei konsistent in einer Lösung $\hat{x}$ des AWP (2.1)-(2.2) und stabil in der durch das Einschrittverfahren erzeugten Folge $\{x_N\}_{N\in\mathbb{N}}$.
Dann ist das Einschrittverfahren konvergent.
Besitzt das Einschrittverfahren darüber hinaus die Konsistenzordnung p in $\hat{x}$, so besitzt es die Konvergenzordnung p.

Beweis: Eine Lösung $\hat{x}$ des AWP erfüllt die Vorschrift des Einschrittverfahrens mit Schrittweite h i.a. nicht exakt und liefert einen Defekt

$$\begin{aligned}\delta_N(t_0) &= \hat{x}(t_0) - x_a = 0,\\ \delta_N(t_i) &= \frac{\hat{x}(t_i)-\hat{x}(t_{i-1})}{h} - \Phi(t_{i-1},\hat{x}(t_{i-1}),h), \qquad i=1,\dots,N.\end{aligned}$$

Da das Verfahren konsistent ist, ist es für hinreichend kleine Schrittweiten $h=(b-a)/N$ (bzw. hinreichend großes N) stets möglich, $\|\delta_N\|_\infty < R$ zu erreichen, da wegen $\delta_N(t_{i+1}) = \ell_h(t_i,\hat{x}(t_i))$ auch

$$0 = \lim_{h\to 0}\left(\max_{i=0,\dots,N-1}\|\ell_h(t_i,\hat{x}(t_i))\|\right) = \lim_{N\to\infty}\left(\max_{i=0,\dots,N-1}\|\delta_N(t_{i+1})\|\right)$$

gilt. Da das Verfahren stabil ist, folgt (mit $y_N = \Delta_N(x)$ in Definition 2.4.7)

$$\|e_N\|_\infty = \|x_N - \Delta_N(\hat{x})\|_\infty \le S\|\delta_N\|_\infty.$$

Wegen $\delta_N(t_0)=0$ und $\delta_N(t_i) = \ell_N(t_{i-1},\hat{x}(t_{i-1}))$, $i=1,\dots,N$, folgt aus der Konsistenz

$$\lim_{N\to\infty}\|\delta_N\|_\infty = 0$$

bzw. aus der Konsistenzordnung p

$$\|\delta_N\|_\infty = \mathcal{O}\left(\frac{1}{N^p}\right) \qquad \text{für } N\to\infty.$$

Dies zeigt die Konvergenz bzw. die Konvergenz mit Ordnung p. □

Natürlich ist die Stabilitätsdefinition so noch nicht nachprüfbar. Ein hinreichendes Kriterium ist eng mit der Existenz und Eindeutigkeit einer Lösung verknüpft. Zunächst zeigen wir das folgende Resultat.

Lemma 2.4.9 (diskretes Gronwall-Lemma)

Für Zahlen $h>0$, $L>0$, $e_k\ge 0$, $d_k\ge 0$, $k=1,\dots,N$, mit

$$a_k \le (1+hL)a_{k-1} + hd_k, \quad k=1,\dots,N,$$

gilt

$$a_k \le \exp(khL)\left(a_0 + kh\max_{j=1,\dots,k} d_j\right), \quad k=0,\dots,N.$$

Beweis: Wir zeigen die Aussage per Induktion nach k. Für $k = 0$ ist die Aussage richtig. Sei die Aussage nun gezeigt für $k \in \mathbb{N}_0$. Dann gilt

$$\begin{aligned}
a_{k+1} &\leq (1+hL)a_k + hd_{k+1} \\
&\overset{1+hL>0}{\leq} (1+hL)\exp(khL)\left(a_0 + kh \max_{j=1,\dots,k} d_j\right) + hd_{k+1} \\
&\overset{1+hL\leq\exp(hL)}{\leq} \exp((k+1)hL)\left(a_0 + kh \max_{j=1,\dots,k} d_j\right) + hd_{k+1} \\
&\overset{\exp((k+1)hL)\geq 1}{\leq} \exp((k+1)hL)\left(a_0 + kh \max_{j=1,\dots,k} d_j + hd_{k+1}\right) \\
&\leq \exp((k+1)hL)\left(a_0 + (k+1)h \max_{j=1,\dots,k+1} d_j\right).
\end{aligned}$$

□

Nun folgt ein hinreichendes Kriterium für Stabilität.

Hilfsatz 2.4.10

Es gebe Konstanten $h_0 > 0$ und $L > 0$, so daß die Inkrementfunktion Φ des Einschrittverfahrens für alle $0 < h \leq h_0$ und alle $t \in [a,b]$ die Lipschitzbedingung

$$\|\Phi(t,y,h) - \Phi(t,z,h)\| \leq L\|y-z\| \qquad \forall y,z \in \mathbb{R}^n$$

erfüllt. Dann ist das Einschrittverfahren stabil.

Beweis: Sei $\mathbb{G}_N$ Gitter mit $0 < h = \frac{b-a}{N} \leq h_0$. Desweiteren sei y_N Gitterfunktion mit Defekt δ_N und

$$\|\delta_N\|_\infty < R.$$

Dann gilt für $j = 1, \dots, N$:

$$\begin{aligned}
\|y_N(t_0) - x_N(t_0)\| &= \|x_0 + \delta_N(t_0) - x_0\| = \|\delta_N(t_0)\|, \\
\|y_N(t_j) - x_N(t_j)\| &= \| \quad y_N(t_{j-1}) + h\Phi(t_{j-1}, y_N(t_{j-1}), h) + h\delta_N(t_j) \\
&\qquad - x_N(t_{j-1}) - h\Phi(t_{j-1}, x_N(t_{j-1}), h)\| \\
&\leq \|y_N(t_{j-1}) - x_N(t_{j-1})\| + h\|\Phi(t_{j-1}, y_N(t_{j-1}), h) - \Phi(t_{j-1}, x_N(t_{j-1}), h)\| \\
&\quad + h\|\delta_N(t_j)\| \\
&\leq (1+hL)\|y_N(t_{j-1}) - x_N(t_{j-1})\| + h\|\delta_N(t_j)\|.
\end{aligned}$$

Anwendung des diskreten Gronwall-Lemmas 2.4.9 mit $a_k = \|y_N(t_k) - x_N(t_k)\|$, $d_k =$

$\|\delta_N(t_k)\|$ liefert

$$\begin{aligned}\|y_N(t_j) - x_N(t_j)\| &\leq \exp(jhL)\left(\|y_N(t_0) - x_N(t_0)\| + jh \max_{k=1,\ldots,j} \|\delta_N(t_k)\|\right)\\ &= \exp(jhL)\left(\|\delta_N(t_0)\| + jh \max_{k=1,\ldots,j} \|\delta_N(t_k)\|\right)\\ &\leq C_j \exp(jhL) \max_{k=0,\ldots,j} \|\delta_N(t_k)\|\end{aligned}$$

mit $C_j = 1 + jh$ für $j = 0, \ldots, N$. Mit $t_j - t_0 = jh \leq Nh = b - a$ folgt daraus schließlich

$$\|y_N - x_N\|_\infty \leq S\|\delta_N\|_\infty$$

mit $S := C_N \exp((b-a)L)$. □

Für Runge-Kutta-Verfahren folgt die Lipschitzstetigkeit der Inkrementfunktion Φ aus der Lipschitzstetigkeit der Funktion f, d.h. ein Runge-Kutta-Verfahren ist stabil, falls f lipschitzstetig bzgl. x gleichmäßig in $t \in [a, b]$ ist. Nach dem Existenz- und Eindeutigkeit existiert in diesem Fall eine (lokal) eindeutige Lösung des AWP. Mit anderen Worten: Stabilität des Verfahrens und Eindeutigkeit der Lösung hängen für Einschrittverfahren eng zusammen.

Bemerkung 2.4.11

- *Hilfssatz 2.4.10 läßt sich weiter abschwächen. Es genügt, daß Φ lokal lipschitzstetig in der exakten Lösung $\hat{x}$ des Anfangswertproblems 2.1.1 ist.*
- *Bei der Definition der Stabilität sind wir insgeheim davon ausgegangen, daß x_N die Gleichungen (2.5)-(2.6) exakt erfüllt. Dies ist in der Praxis nicht der Fall, da i.a. Rundungsfehler auftreten. Die obige Konvergenzanalyse kann jedoch entsprechend erweitert werden, so daß auch der Rundungsfehlereinfluß berücksichtigt wird, vgl. z.B. Stetter [Ste73] und Demailly [Dem91]. Stetter [Ste73] entwickelt eine sehr allgemeine Konvergenztheorie, die auch auf viele andere Problemklassen anwendbar ist.*
- *Neben dem hier verwendeten Stabilitätsbegriff spielen andere Stabilitätsbegriffe für die numerische Lösung eine große Rolle. Im Zusammenhang mit steifen Differentialgleichungen werden A-stabile bzw. A(α)-stabile Verfahren benötigt, was im Zusammenhang mit Runge-Kutta-Verfahren zwangsläufig auf implizite Verfahren führt.*

2.5 Schrittweitensteuerung

Aus Effizienzgründen ist es in der Regel nicht sinnvoll, mit einer konstanten Schrittweite h zu arbeiten, sondern einen Algorithmus zur automatischen Schrittweitenanpassung

zu verwenden. Ziel dieses Algorithmus ist es, das Gitter $\mathbb{G}_N = \{t_0 < t_1 < \ldots < t_N\}$ mit Schrittweiten $h_i = t_{i+1} - t_i$, $i = 0, \ldots, N-1$, automatisch zu bestimmen. Wir demonstrieren die Notwendigkeit einer solchen Prozedur an folgendem Beispiel.

Beispiel 2.5.1

Das folgende Anfangswertproblem beschreibt die Bewegung eines Satelliten um das System Erde–Mond und liefert eine periodische Bahn, den sogenannten Ahrenstorf-Orbit:

$$\begin{aligned} x''(t) &= x(t) + 2y'(t) - \bar{\mu}\tfrac{x(t)+\mu}{D_1} - \mu\tfrac{x(t)-\bar{\mu}}{D_2}, \\ y''(t) &= y(t) - 2x'(t) - \bar{\mu}\tfrac{y(t)}{D_1} - \mu\tfrac{y(t)}{D_2}, \end{aligned}$$

mit $\mu = 0.012277471$, $\bar{\mu} = 1 - \mu$ *und*

$$D_1 = \sqrt{((x(t)+\mu)^2 + y(t)^2)^3}, \qquad D_2 = \sqrt{((x(t)-\bar{\mu})^2 + y(t)^2)^3}$$

und

$$\begin{aligned} x(0) &= 0.994, \quad & y(0) &= 0, \\ x'(0) &= 0, & y'(0) &= -2.001585106379. \end{aligned}$$

Das Differentialgleichungssystem zweiter Ordnung kann durch $x_1 := x$, $x_2 := y$, $x_3 := x' = x_1'$ *und* $x_4 := y' = x_2'$ *auf ein System erster Ordnung transformiert werden:*

$$\begin{aligned} x_1'(t) &= x_3(t), \\ x_2'(t) &= x_4(t), \\ x_3'(t) &= x(t) + 2x_4(t) - \bar{\mu}\tfrac{x_1(t)+\mu}{D_1} - \mu\tfrac{x_1(t)-\bar{\mu}}{D_2}, \\ x_4'(t) &= x_2(t) - 2x_3(t) - \bar{\mu}\tfrac{x_2(t)}{D_1} - \mu\tfrac{x_2(t)}{D_2}, \end{aligned}$$

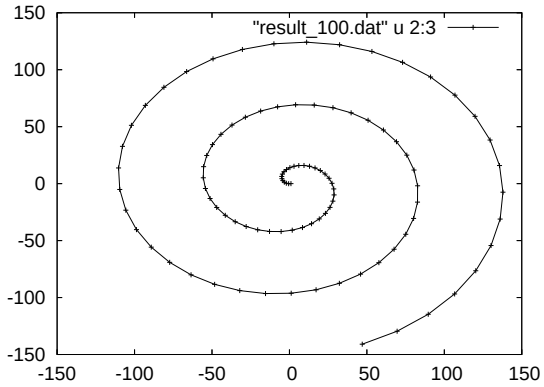
mit

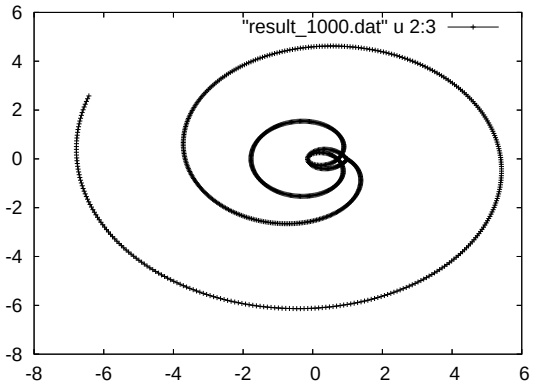
$$D_1 = \sqrt{((x_1(t)+\mu)^2 + x_2(t)^2)^3}, \qquad D_2 = \sqrt{((x_1(t)-\bar{\mu})^2 + x_2(t)^2)^3}$$

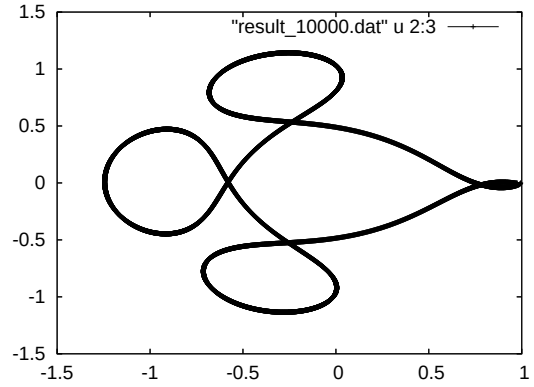
und

$$\begin{aligned} x_1(0) &= 0.994, \quad & x_2(0) &= 0, \\ x_3(0) &= 0, & x_4(0) &= -2.001585106379. \end{aligned}$$

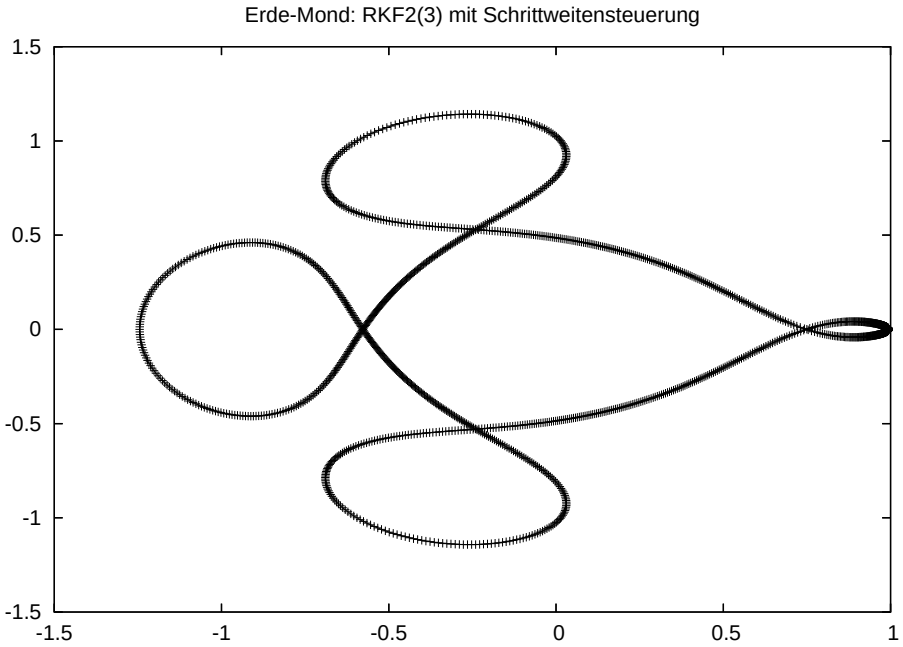
Die folgenden Abbildungen zeigen die Komponenten $(x(t), y(t))$ *der numerischen Lösung, die mit dem klassischen Runge-Kutta-Verfahren für das Intervall* $[a, b]$ *mit* $a = 0$ *und* $b = 17.065216560158$ *und fester Schrittweite* $h = (b-a)/N$ *berechnet wurden. Erst für* $N = 10000$ *ergibt sich eine Lösung, die mit der Referenzlösung im untersten Bild übereinstimmt. Hierfür sind 40000 Funktionsauswertungen der rechten Seite nötig:*







Die Referenzlösung im folgenden Bild wurde mit dem Runge-Kutta-Verfahren RKF2(3) der Ordnung 2 berechnet, welches eine automatische Schrittweitensteuerung verwendet und nur 6368 Funktionsauswertungen benötigt.



Das Beispiel zeigt, daß es bei äquidistanter Schrittweite i.a. notwendig ist, eine sehr kleine Schrittweite zu wählen, um eine hinreichend gute Näherung zu erhalten. Bei der Lösung mit adaptiv angepassten Schrittweiten erkennt man am Abstand der Punkte, wo kleine und wo große Schritte geählt wurden.

Gängige Schrittweitenstrategien basieren auf der numerischen Schätzung des lokalen (oder globalen) Fehlers und versuchen, diesen unter einer gegebenen Genauigkeitsschranke zu halten. Dort wo sich die Lösung kaum ändert, kann eine große Schrittweite gewählt werden, während an Stellen mit starker Änderung der Lösung aus Genauigkeitsgründen eine kleine Schrittweite gewählt werden muß. Die passende Schrittweite wird auf Basis einer Schätzung des lokalen Diskretisierungsfehlers bestimmt.
Zunächst überlegen wir uns, was eine automatische Schrittweitensteuerung leisten soll. Dazu nehmen wir an, das Einschrittverfahren sei zum Gitterpunkt t_i fortgeschritten. Zusätzlich sei ein Schrittweitenvorschlag h für den nächsten Integrationsschritt gegeben. Eine automatische Schrittweitensteuerung sollte folgendes leisten:

(1) Für den aktuellen Schritt von t_i nach $t_{i+1} = t_i + h$ mit Schrittweite h soll entschieden werden, ob h akzeptabel ist. Ist dies der Fall, so wird ein Schritt mit einem zu Grunde liegenden Einschrittverfahren mit Schrittweite h durchgeführt.

(2) Ist die Schrittweite h akzeptabel, so soll zusätzlich eine Schrittweite h_{neu} für den nächsten Schritt von t_{i+1} nach $t_{i+1} + h_{neu}$ vorgeschlagen werden.

(3) Ist die Schrittweite h nicht akzeptabel, so soll eine Schrittweite h_{neu} vorgeschlagen werden, mit der der Schritt von t_i nach $t_i + h_{neu}$ wiederholt wird.

Zur Berechnung einer Schätzung des lokalen Diskretisierungsfehlers gibt es im wesentlichen zwei Varianten: **Zwei Verfahren-Eine Schrittweite** bzw. **eingebettete Runge-Kutta-Verfahren**.

2.5.1 Ein Verfahren-Zwei Schrittweiten

Wir betrachten ein Einschrittverfahren der Ordnung p mit stetig differenzierbarer Inkrementfunktion Φ, welches bis zum Gitterpunkt t_i mit Näherung x_i fortgeschritten sei. Wir möchten den lokalen Fehler schätzen. Es bezeichne $y(\cdot)$ die exakte Lösung des AWP

$$y'(t) = f(t, y(t)), \quad y(t_i) = x_i. \tag{2.7}$$

Ausgehend von (t_i, x_i) wird eine Näherung η_h mit Schrittweite h berechnet:

$$\eta_h = x_i + h\Phi\left(t_i, x_i, h\right).$$

Ausserdem bezeichne $\eta_{2\times\frac{h}{2}}$ diejenige Näherung an der Stelle $t_i + h$, die entsteht, wenn ausgehend von (t_i, x_i) **zwei Schritte** des Einschrittverfahrens mit Schrittweite $h/2$ durchgeführt werden, d.h.

$$\eta_{2\times\frac{h}{2}} = \eta_{\frac{h}{2}} + \frac{h}{2}\Phi\left(t_i + h/2, \eta_{\frac{h}{2}}, h/2\right)$$

mit

$$\eta_{\frac{h}{2}} = x_i + \frac{h}{2}\Phi\left(t_i, x_i, h/2\right).$$

Weiterhin sei $\hat{\eta}_{\frac{h}{2}}$ die Näherung, die man erhält, wenn von $(t_i + h/2, y(t_i + h/2))$ ein Schritt des Verfahrens mit Schrittweite $h/2$ durchgeführt wird, d.h.

$$\hat{\eta}_{\frac{h}{2}} = y(t_i + h/2) + \frac{h}{2}\Phi\left(t_i + h/2, y(t_i + h/2), h/2\right).$$

Da das Einschrittverfahren die (Konsistenz-)Ordnung p besitzt, gilt für die lokalen Fehler von η_h, $\eta_{\frac{h}{2}}$ und $\hat{\eta}_{\frac{h}{2}}$

$$\begin{aligned} y(t_i + h) - \eta_h &= C(t_i)h^{p+1} + \mathcal{O}(h^{p+2}), \\ y(t_i + h/2) - \eta_{\frac{h}{2}} &= C(t_i)\left(\frac{h}{2}\right)^{p+1} + \mathcal{O}(h^{p+2}), \\ y(t_i + h) - \hat{\eta}_{\frac{h}{2}} &= C(t_i + h/2)\left(\frac{h}{2}\right)^{p+1} + \mathcal{O}(h^{p+2}). \end{aligned} \tag{2.8}$$

Damit folgt

$$\begin{aligned}
y(t_i+h)-\eta_{2\times\frac{h}{2}} &= y(t_i+h)-\hat\eta_{\frac{h}{2}}+\hat\eta_{\frac{h}{2}}-\eta_{2\times\frac{h}{2}}\\
&= C(t_i+h/2)\left(\frac{h}{2}\right)^{p+1}+\hat\eta_{\frac{h}{2}}-\eta_{2\times\frac{h}{2}}+\mathcal{O}(h^{p+2})\\
&= C(t_i+h/2)\left(\frac{h}{2}\right)^{p+1}+y(t_i+h/2)+\frac{h}{2}\Phi\left(t_i+h/2,y(t_i+h/2),h/2\right)\\
&\quad -\eta_{\frac{h}{2}}-\frac{h}{2}\Phi\left(t_i+h/2,\eta_{\frac{h}{2}},h/2\right)+\mathcal{O}(h^{p+2})\\
&= C(t_i+h/2)\left(\frac{h}{2}\right)^{p+1}+C(t_i)\left(\frac{h}{2}\right)^{p+1}\\
&\quad +\frac{h}{2}\left(\Phi\left(t_i+h/2,y(t_i+h/2),h/2\right)-\Phi\left(t_i+h/2,\eta_{\frac{h}{2}},h/2\right)\right)+\mathcal{O}(h^{p+2}).
\end{aligned}$$

Der Mittelwertsatz in Integralform liefert

$$\begin{aligned}
&\Phi\left(t_i+h/2,y(t_i+h/2),h/2\right)-\Phi\left(t_i+h/2,\eta_{\frac{h}{2}},h/2\right)\\
=&\int_0^1\Phi'\left(t_i+h/2,\eta_{\frac{h}{2}}+s\left(y(t_i+h/2)-\eta_{\frac{h}{2}}\right),h/2\right)\underbrace{\left(y(t_i+h/2)-\eta_{\frac{h}{2}}\right)}_{=\mathcal{O}(h^{p+1})}ds.
\end{aligned}$$

Unter der Annahme, daß Φ' beschränkt ist, folgt damit

$$y(t_i+h)-\eta_{2\times\frac{h}{2}} = \left(C(t_i+h/2)+C(t_i)\right)\left(\frac{h}{2}\right)^{p+1}+\mathcal{O}(h^{p+2}).$$

Vernachlässigt man die Terme mit $\mathcal{O}(h^{p+2})$ und nutzt aus, daß für kleine $h>0$ $C(t_i+h/2)=C(t_i)+\mathcal{O}(h)$ gilt[1], so erhalten wir mit (2.8) die folgende Schätzung für die Hauptfehlerkonstante $C(t_i)$:

$$C(t_i)=\frac{2^p}{2^p-1}\left(\eta_{2\times\frac{h}{2}}-\eta_h\right)\cdot h^{-(p+1)}+\mathcal{O}(h).$$

Damit folgt dann

$$y(t_i+h)-\eta_{2\times\frac{h}{2}}=\frac{\eta_{2\times\frac{h}{2}}-\eta_h}{2^p-1}+\mathcal{O}(h^{p+2}).$$

Insbesondere liest man hieraus sofort ab, daß

$$\eta_{2\times\frac{h}{2}}+\frac{\eta_{2\times\frac{h}{2}}-\eta_h}{2^p-1} \tag{2.9}$$

eine Näherung der Konsistenzordnung $p+1$ ist. Der Term

$$err:=\frac{\|\eta_{2\times\frac{h}{2}}-\eta_h\|}{2^p-1}$$

[1]Dies ist gerechtfertigt, falls die rechte Seite f und damit auch Φ hinreichend glatt sind. Die Konstante C enthält Funktionswerte von f und deren Ableitungen, die in (t_i,x_i) ausgewertet werden.

dient als Schätzung des lokalen Fehlers.

Ist h akzeptabel?
Um zu entscheiden, ob die Schrittweite h akzeptabel ist, wird geprüft, ob die Schätzung des lokalen Fehlers kleiner als eine gewisse vom Benutzer vorgegebene Toleranz *tol* ist, d.h. es muß

$$err = \frac{\|\eta_{2\times\frac{h}{2}} - \eta_h\|}{2^p - 1} \leq tol$$

gelten. Ist dies der Fall, so wird die neue Näherung $\eta_{2\times\frac{h}{2}}$ oder besser (2.9) akzeptiert.
Ist dies nicht der Fall, so muß die Schrittweite h angepasst werden. Mit den obigen Betrachtungen erhält man für den Schritt von t_i nach $t_i + h_{neu}$ mit einer neuen Schrittweite h_{neu} die Fehlerabschätzung

$$\begin{aligned} y(t_i + h_{neu}) - \eta_{2\times\frac{h_{neu}}{2}} &= 2C(t_i)\left(\frac{h_{neu}}{2}\right)^{p+1} + \mathcal{O}(h_{neu}^{p+2}) \\ &= \frac{\eta_{2\times\frac{h}{2}} - \eta_h}{2^p - 1}\left(\frac{h_{neu}}{h}\right)^{p+1} + \mathcal{O}(hh_{neu}^{p+1}) + \mathcal{O}(h_{neu}^{p+2}). \end{aligned}$$

Vernachlässigung der Terme höherer Ordnung und die Forderung, der lokale Fehler möge kleiner als *tol* sein, führt auf die Forderung

$$\frac{\|\eta_{2\times\frac{h}{2}} - \eta_h\|}{2^p - 1}\left(\frac{h_{neu}}{h}\right)^{p+1} \leq tol$$

bzw.

$$h_{neu} \leq \left(\frac{tol}{err}\right)^{\frac{1}{p+1}} \cdot h.$$

Häufig wird noch ein Sicherheitsfaktor $0 < \alpha < 1$ eingeführt (oft $\alpha = 0.8$), um häufige Änderungen der Schrittweite zu vermeiden, so daß der Schrittweitenvorschlag

$$h_{neu} = \alpha\left(\frac{tol}{err}\right)^{\frac{1}{p+1}} \cdot h \tag{2.10}$$

resultiert. Zusätzlich kann der Fehler *err* noch geeignet komponentenweise gewichtet werden.

Wie soll h_{neu} für den nächsten Integrationsschritt gewählt werden?
Für den Schritt von t_{i+1} mit Näherung x_{i+1} zu $t_{i+1} + h_{neu}$ gilt analog zu oben

$$y(t_{i+1} + h_{neu}) - \eta_{2\times\frac{h_{neu}}{2}} = 2C(t_{i+1})\left(\frac{h_{neu}}{2}\right)^{p+1} + \mathcal{O}(h_{neu}^{p+2}).$$

Ist C differenzierbar (falls f bzw. Φ hinreichend glatt!), so gilt $C(t_{i+1}) = C(t_i + h) = C(t_i) + \mathcal{O}(h)$ und somit

$$y(t_{i+1} + h_{neu}) - \eta_{2\times\frac{h_{neu}}{2}} \;=\; 2C(t_i)\left(\frac{h_{neu}}{2}\right)^{p+1} + \mathcal{O}(hh_{neu}^{p+1}) + \mathcal{O}(h_{neu}^{p+2}).$$

Analog zu oben erhält man wiederum den Schrittweitenvorschlag h_{neu} gemäß (2.10).

Aufwand

Der Nachteil dieser Methode zur Schrittweitensteuerung ist, daß der Aufwand pro Integrationsschritt im Vergleich zum Einschrittverfahren ohne Schrittweitensteuerung dreimal so hoch ist, da err (bzw. $\eta_{2\times\frac{h}{2}}$ und η_h) berechnet werden muß. Ein Vorteil ist, daß mit (2.9) eine Näherung höherer Ordnung nebenbei mitberechnet wird.

2.5.2 Eingebettete Runge-Kutta-Verfahren

An Stelle nur eines Verfahrens können auch zwei Einschrittverfahren **benachbarter Ordnung** verwendet werden. Insbesondere bietet sich diese Vorgehensweise für Runge-Kutta-Verfahren an, die den gleichen Knotenvektor c und die gleiche Verfahrensmatrix A verwenden. Nur der Gewichtsvektor b unterscheidet sich. Es seien also beispielsweise zwei explizite Verfahren durch das Butcher-Schema gegeben:

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
c_s	a_{s1}	a_{s2}	$\cdots$	a_{ss-1}	
$RK1$	b_1	b_2	$\cdots$	b_{s-1}	b_s
$RK2$	$\hat{b}_1$	$\hat{b}_2$	$\cdots$	$\hat{b}_{s-1}$	$\hat{b}_s$

Das Verfahren $RK1$ besitze Konsistenzordnung p, das Verfahren $RK2$ besitze Konsistenzordnung q mit $q = p + 1$ oder $q = p - 1$. Man spricht dann auch von **eingebetteten Runge-Kutta-Verfahren**. Ein Beispiel ist das eingebettete Verfahren der Ordnung $p(q) = 2(3)$:

0				
$1/4$	$1/4$			
$27/40$	$-189/800$	$729/800$		
1	$214/891$	$1/33$	$650/891$	
$RK1$	$214/891$	$1/33$	$650/891$	0
$RK2$	$533/2106$	0	$800/1053$	$-1/78$

Die Berechnung der Näherungen η_{i+1} mit $RK1$ und $\hat{\eta}_{i+1}$ mit $RK2$ zur gleichen Schrittweite h ist sehr kostengünstig, da der Knotenvektor c und die Matrix A bei beiden Verfahren übereinstimmen, d.h. die Zwischenwerte $\eta_{i+1}^{(j)}$ müssen nur einmalig berechnet werden.
Gegeben seien nun also zwei Runge-Kutta-Verfahren mit benachbarten Konvergenzordnungen p und $p+1$ und Inkrementfunktionen Φ und $\bar{\Phi}$. Das Verfahren sei bis zum Zeitpunkt t_i mit Näherung x_i fortgeschritten.
Mit beiden Verfahren wird in (t_i, x_i) jeweils ein Schritt mit Schrittweite h durchgeführt:

$$\begin{aligned} \eta_h &= x_i + h\Phi(t_i, x_i, h), \\ \bar{\eta}_h &= x_i + h\bar{\Phi}(t_i, x_i, h) \end{aligned}$$

Die lokalen Fehler der beiden Verfahrens erfüllen

$$\begin{aligned} y(t_i + h) - \eta_h &= C(t_i)h^{p+1} + \mathcal{O}(h^{p+2}), \\ y(t_i + h) - \bar{\eta}_h &= \bar{C}(t_i)h^{p+2} + \mathcal{O}(h^{p+3}), \end{aligned}$$

wobei y wieder die Lösung des AWP (2.7) bezeichnet. Subtraktion der zweiten von der ersten Gleichung liefert

$$C(t_i) = \frac{1}{h^{p+1}} (\bar{\eta}_h - \eta_h) + \mathcal{O}(h).$$

Damit erhalten wir auch eine Schätzung des lokalen Fehlers für das erste Verfahren (mit niedrigerer Ordnung p):

$$y(t_i + h) - \eta_h = (\bar{\eta}_h - \eta_h) + \mathcal{O}(h^{p+2})$$

Analog zum vorigen Abschnitt ist h akzeptabel, falls

$$err := \|\bar{\eta}_h - \eta_h\| \leq tol$$

gilt. Eine neue Schrittweite, falls h nicht akzeptabel ist bzw. für den nächsten Schritt, muß wieder

$$h_{neu} \leq \left(\frac{tol}{err}\right)^{\frac{1}{p+1}} \cdot h$$

erfüllen. Diese Betrachtungen führen unter Ergänzung einiger technischer Details auf den folgenden Schrittweitenalgorithmus, vgl. Strehmel und Weiner [SW95], S. 62.

Algorithmus 2.5.2 (Schrittweitensteuerung)

(0) Initialisierung: $t = a$, $x = x_a$. Wähle Anfangsschrittweite h.

(1) Falls $t + h > b$, setze $h = b - t$.

(2) Berechne mit RK_1 bzw. RK_2 ausgehend von x Näherungen η und $\bar{\eta}$ an der Stelle $t+h$.

(3) Berechne err und h_{neu} gemäß

$$err = \max_{i=1,\dots,n_x} \left(\frac{|\eta_i - \bar{\eta}_i|}{sk_i} \right)$$

mit Skalierungsfaktoren $sk_i = atol + \max(|\eta_i|, |x_i|) \cdot rtol$, absoluter Fehlertoleranz $atol = 10^{-7}$ und relativer Fehlertoleranz $rtol = 10^{-7}$ (η_i, $\bar{\eta}_i$ und x_i bezeichnen jeweils die Komponenten von η, $\bar{\eta}$ und x), sowie

$$h_{neu} = \min(\alpha_{max}, \max(\alpha_{min}, \alpha \cdot (1/err)^{1/(1+p)})) \cdot h$$

mit $\alpha_{max} = 1.5$, $\alpha_{min} = 0.2$ und $\alpha = 0.8$.

(4) Falls $h_{neu} < h_{min} := 10^{-8}$, Abbruch mit Fehlermeldung.

(5) Falls $err \leq 1$ (Schritt wird akzeptiert):

(i) Setze $x = \eta$, $t = t + h$.
(ii) Falls $|t - b| < 10^{-8}$, Abbruch mit Erfolg.
(iii) Setze $h = h_{neu}$ und gehe zu (1).

Falls $err > 1$ (Schritt wird wiederholt): Setze $h = h_{neu}$ und gehe zu (1).

Kapitel 3

Mehrschrittverfahren

In diesem Kapitel werden Diskretisierungsverfahren für das Anfangswertproblem 2.1.1 behandelt, die – im Gegensatz zu Einschrittverfahren – nicht nur auf die Approximation am zurückliegenden benachbarten Gitterpunkt zugreifen, sondern auch weiter zurück liegende Werte verwenden. Zunächst diskutieren wir einige konkrete Mehrschrittverfahren. Anschließend werden Konsistenz, Stabilität und Konvergenz der Verfahren untersucht. Hierbei beschränken wir uns auf lineare Mehrschrittverfahren, die überwiegend in der Praxis eingesetzt werden. Grundlage der Verfahren ist wiederum die Konstruktion von Approximationen auf Gittern $\mathbb{G}_N$, $N \in \mathbb{N}$, welche wie bei Einschrittverfahren äquidistant oder adaptiv gewählt werden können. Im folgenden beschränken wir uns jedoch auf äquidistante Gitter mit Schrittweite $h = (b-a)/N$, $N \in \mathbb{N}$.

3.1 Beispiele für Mehrschrittverfahren

Im folgenden seien $k \in \mathbb{N}$ fest gegeben und t_{i+j}, $i = 0, \ldots, N-k$, $j = 0, \ldots, k$, Gitterpunkte in $\mathbb{G}_N$. Wir betrachten den Integrationsschritt von t_{i+k-1} nach t_{i+k} und nehmen an, daß bereits Näherungen $x_i := x_N(t_i), x_{i+1} := x_N(t_{i+1}), \ldots, x_{i+k-1} := x_N(t_{i+k-1})$ berechnet wurden. Ziel ist es, eine Approximation $x_{i+k} := x_N(t_{i+k})$ zu berechnen.

3.1.1 Adams-Verfahren

Die Adams-Verfahren basieren auf Quadraturverfahren, die auf die Integraldarstellung

$$x(t_{i+k}) - x(t_{i+k-1}) = \int_{t_{i+k-1}}^{t_{i+k}} f(t, x(t))dt \tag{3.1}$$

angewendet werden. Ähnlich wie bei den Newton-Cotes-Formeln zur numerischen Integration wird der Integrand f durch ein interpolierendes Polynom P ersetzt (vgl. Abbildung 3.1), woraus die folgende Vorschrift zur Bestimmung von $x_{i+k} = x_N(t_{i+k})$ resultiert:

$$x_{i+k} - x_{i+k-1} = \int_{t_{i+k-1}}^{t_{i+k}} P(t)dt. \tag{3.2}$$

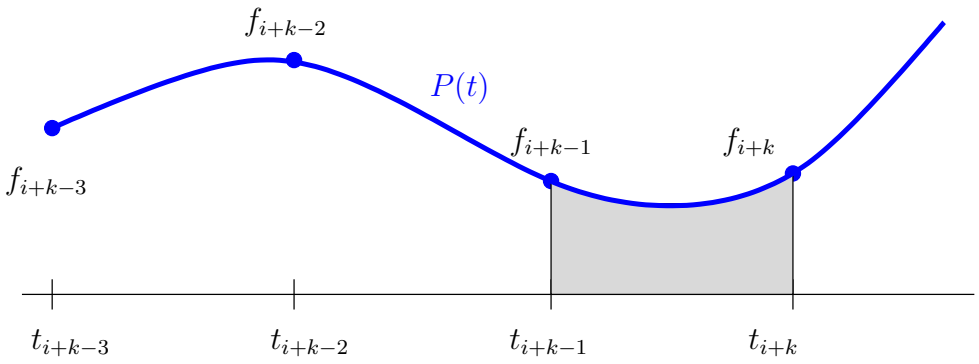


Abbildung 3.1: Idee der Adams-Verfahren: Polynominterpolation der Funktion f an Gitterpunkten

Je nachdem, welche der Stützwerte

$$(t_{i+j}, f_{i+j}), \qquad j = 0, \dots, k,$$

mit

$$f_{i+j} := f(t_{i+j}, x_N(t_{i+j}))), \qquad j = 0, \dots, k,$$

interpoliert werden, erhält man die **Adams-Bashforth-Verfahren**, die zur Klasse der expliziten linearen Mehrschrittverfahren gehören, oder die **Adams-Moulton-Verfahren**, die zur Klasse der impliziten linearen Mehrschrittverfahren gehören.

Adams-Bashforth-Verfahren

Bei den Adams-Bashforth-Verfahren wird das interpolierende Polynom P vom Höchstgrad $k-1$ zu den Stützpunkten

$$(t_{i+j}, f_{i+j}), \qquad j = 0, \dots, k-1,$$

berechnet. Da die gesuchte Näherung x_{i+k} nicht in die Konstruktion von P eingeht, handelt es sich beim resultierenden Verfahren (3.2) um ein explizites Verfahren. P besitzt die Darstellung

$$P(t) = \sum_{j=0}^{k-1} f_{i+j} L_j(t), \qquad L_j(t) = \prod_{\ell=0, \ell\neq j}^{k-1} \frac{t - t_{i+\ell}}{t_{i+j} - t_{i+\ell}}.$$

Einsetzen in (3.2) liefert

$$x_{i+k} - x_{i+k-1} = \sum_{j=0}^{k-1} f_{i+j} \int_{t_{i+k-1}}^{t_{i+k}} L_j(t)dt =: h \sum_{j=0}^{k-1} \beta_{ij} f_{i+j}$$

mit

$$\beta_{ij} = \frac{1}{h} \int_{t_{i+k-1}}^{t_{i+k}} L_j(t)dt = \frac{1}{h} \int_{t_{i+k-1}}^{t_{i+k}} \prod_{\ell=0, \ell\neq j}^{k-1} \frac{t - t_{i+\ell}}{t_{i+j} - t_{i+\ell}} dt.$$

Für äquidistante Gitter mit Schrittweite h folgt weiter

$$\begin{aligned}\beta_{ij} &= \frac{1}{h}\int_{t_{i+k-1}}^{t_{i+k}} L_j(t)dt \\ &= \int_0^1 L_j(t_{i+k-1}+sh)ds \\ &= \int_0^1 \prod_{\ell=0,\ell\neq j}^{k-1} \frac{t_{i+k-1}+sh-t_{i+\ell}}{t_{i+j}-t_{i+\ell}}ds \\ &= \int_0^1 \prod_{\ell=0,\ell\neq j}^{k-1} \frac{(k-\ell-1)+s}{j-\ell}ds, \qquad j=0,\dots,k-1.\end{aligned}$$

Insbesondere hängen die Koeffizienten $\beta_j := \beta_{ij}$ bei äquidistanten Gittern nicht vom Index i ab. Für nicht äquidistante Gitter muss das Interpolationspolynom in jedem Schritt neu aufgestellt werden, wobei es hierfür effiziente Updateregeln gibt.

Beispiel 3.1.1

Für $k=1$ ergibt sich $\beta_0 = 1$ und es entsteht das explizite Eulerverfahren

$$x_{i+1} - x_i = hf(t_i, x_i).$$

Für $k=2$ ergibt sich $\beta_0 = -1/2$ und $\beta_1 = 3/2$ und das resultierende Verfahren lautet

$$x_{i+2} - x_{i+1} = \frac{h}{2}\left(-f_i + 3f_{i+1}\right).$$

In analoger Weise erhält man

$$\begin{aligned}k=3 \;&:\; x_{i+3} = x_{i+2} + \frac{h}{12}\left(23f_{i+2} - 16f_{i+1} + 5f_i\right) \\ k=4 \;&:\; x_{i+4} = x_{i+3} + \frac{h}{24}\left(55f_{i+3} - 59f_{i+2} + 37f_{i+1} - 9f_i\right)\end{aligned}$$

Adams-Moulton-Verfahren

Anders als bei den Adams-Bashforth-Verfahren wird bei den Adams-Moulton-Verfahren zusätzlich der Punkt (t_{i+k}, f_{i+k}) durch das Polynom P interpoliert, d.h. P ist das interpolierende Polynom vom Höchstgrad k zu den Stützpunkten

$$(t_{i+j}, f_{i+j}), \qquad j=0,\dots,k.$$

Da $f_{i+k} = f(t_{i+k}, x_{i+k})$ von der gesuchten Näherung x_{i+k} abhängt, ist das resultierende Verfahren implizit, d.h. x_{i+k} ist implizit durch die nichtlineare Gleichung (3.2) gegeben. P besitzt die Darstellung

$$P(t) = \sum_{j=0}^{k} f_{i+j}L_j(t), \qquad L_j(t) = \prod_{\ell=0,\ell\neq j}^{k} \frac{t-t_{i+\ell}}{t_{i+j}-t_{i+\ell}}.$$

Einsetzen in (3.2) liefert

$$x_{i+k} - x_{i+k-1} = \sum_{j=0}^{k} f_{i+j} \int_{t_{i+k-1}}^{t_{i+k}} L_j(t)dt =: h \sum_{j=0}^{k} \beta_{ij} f_{i+j}$$

mit

$$\beta_{ij} = \frac{1}{h} \int_{t_{i+k-1}}^{t_{i+k}} L_j(t)dt = \frac{1}{h} \int_{t_{i+k-1}}^{t_{i+k}} \prod_{\ell=0,\ell\neq j}^{k} \frac{t - t_{i+\ell}}{t_{i+j} - t_{i+\ell}} dt.$$

Für äquidistante Gitter mit Schrittweite h folgt weiter

$$\begin{aligned}
\beta_j := \beta_{ij} &= \frac{1}{h} \int_{t_{i+k-1}}^{t_{i+k}} L_j(t)dt \\
&= \int_0^1 L_j(t_{i+k-1} + sh)ds \\
&= \int_0^1 \prod_{\ell=0,\ell\neq j}^{k} \frac{t_{i+k-1} + sh - t_{i+\ell}}{t_{i+j} - t_{i+\ell}} ds \\
&= \int_0^1 \prod_{\ell=0,\ell\neq j}^{k} \frac{(k-\ell-1) + s}{j - \ell} ds, \qquad j = 0, \ldots, k.
\end{aligned}$$

Beispiel 3.1.2

Für $k = 0$ folgt $\beta_0 = 1$ und es resultiert das implizite Eulerverfahren

$$x_i - x_{i-1} = hf(t_i, x_i).$$

Für $k = 1$ folgt $\beta_0 = \beta_1 = 1/2$ und es resultiert die **Trapezregel**

$$x_{i+1} - x_i = \frac{h}{2} \left(f_i + f_{i+1} \right).$$

In analoger Weise erhält man

$$\begin{aligned}
k = 2 \quad &: \quad x_{i+2} = x_{i+1} + \frac{h}{12} \left(5f_{i+2} + 8f_{i+1} - f_i \right) \\
k = 3 \quad &: \quad x_{i+3} = x_{i+2} + \frac{h}{24} \left(9f_{i+3} + 19f_{i+2} - 5f_{i+1} + f_i \right)
\end{aligned}$$

Bemerkung 3.1.3 (Nyström-Verfahren und Milne-Simpson-Verfahren)

Anstatt das Integral in (3.1) von t_{i+k-1} bis t_{i+k} zu betrachten, sind auch andere Intervallgrenzen möglich. Beispielsweise führt der Ansatz

$$x(t_{i+k}) - x(t_{i+k-2}) = \int_{t_{i+k-2}}^{t_{i+k}} f(t, x(t))dt$$

analog zu den Adams-Verfahren auf Nyström-Verfahren (explizit) oder Milne-Simpson-Verfahren (implizit). Speziell ergibt sich für das Nyström-Verfahren mit $k = 2$ die Mittelpunktregel

$$x_{i+2} - x_i = 2hf(t_{i+1}, x_{i+1}).$$

3.1.2 BDF-Verfahren

Die Backward Differentiation Formulae (BDF) wurden durch Curtiss und Hirschfelder [CH52] und Gear [Gea71] eingeführt und gehören zur Klasse der impliziten linearen Mehrschrittverfahren. Anders als bei den Adams-Verfahren wird nicht der Integrand in der Integraldarstellung interpoliert, sondern die Näherungen x_{i+j}, $j = 0, \dots, k$, selbst werden interpoliert, vgl. Abbildung 3.2.

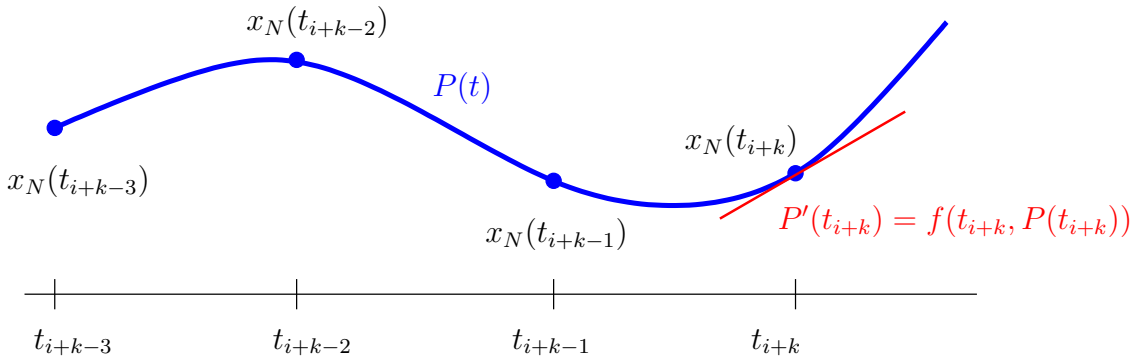


Abbildung 3.2: Idee der BDF-Verfahren: Polynominterpolation der Approximationen

Ziel ist es wiederum, x_{i+k} zu berechnen. Hierzu wird das interpolierende Polynom $P(t)$ vom Höchstgrad k mit

$$P(t_{i+j}) = x_{i+j}, \qquad j = 0, \dots, k,$$

berechnet. Beachte, daß P den noch unbekannten Vektor x_{i+k} bereits interpoliert. Zur Bestimmung von x_{i+k} wird gefordert, daß P die Differentialgleichung (2.1) am Gitterpunkt t_{i+k} erfüllen soll, d.h. es soll

$$P'(t_{i+k}) = f(t_{i+k}, P(t_{i+k})) = f(t_{i+k}, x_{i+k})$$

gelten. Mit Hilfe der Lagrange-Darstellung

$$P(t) = \sum_{j=0}^{k} x_{i+j} L_j(t), \qquad L_j(t) = \prod_{\ell=0, \ell \neq j}^{k} \frac{t - t_{i+\ell}}{t_{i+j} - t_{i+\ell}},$$

des Interpolationspolynoms läßt sich die Ableitung $P'(t_{i+k})$ darstellen als

$$P'(t_{i+k}) = \sum_{j=0}^{k} x_{i+j} L_j'(t_{i+k}) =: \frac{1}{h} \sum_{j=0}^{k} \alpha_j x_{i+j},$$

mit

$$\begin{aligned}\alpha_j &= hL_j'(t_{i+k}) \\ &= h\sum_{\kappa=0,\kappa\neq j}^{k}\frac{1}{t_{i+j}-t_{i+\kappa}}\prod_{\ell=0,\ell\neq j,\ell\neq\kappa}^{k}\frac{t_{i+k}-t_{i+\ell}}{t_{i+j}-t_{i+\ell}} \\ &= \sum_{\kappa=0,\kappa\neq j}^{k}\frac{1}{j-\kappa}\prod_{\ell=0,\ell\neq j,\ell\neq\kappa}^{k}\frac{k-\ell}{j-\ell}\end{aligned}$$

im Falle äquidistanter Stützstellen. Einsetzen führt auf die nichtlineare Gleichung

$$\sum_{j=0}^{k}\alpha_j x_{i+j} = hf(t_{i+k}, x_{i+k})$$

für x_{i+k}, welche mit dem Newtonverfahren gelöst werden kann.
Unter den impliziten Verfahren sind BDF-Verfahren beliebt, weil sie lediglich die Lösung eines n-dimensionalen nichtlinearen Gleichungssystems erfordern, während ein implizites, s-stufiges Runge-Kutta-Verfahren die Lösung eines $n \cdot s$-dimensionalen nichtlinearen Gleichungssystems erfordern.
Numerisch relevant sind nur die BDF-Verfahren mit $k \leq 6$, da sie für $k > 6$ nicht stabil sind. Die BDF-Verfahren bis $k = 6$ lauten wie folgt, vgl. [SW95, S. 335]:

$$\begin{aligned}
k=1 &: hf_{i+1} = x_{i+1} - x_i \quad \text{(implizites Eulerverfahren)} \\
k=2 &: hf_{i+2} = \frac{1}{2}(3x_{i+2} - 4x_{i+1} + x_i) \\
k=3 &: hf_{i+3} = \frac{1}{6}(11x_{i+3} - 18x_{i+2} + 9x_{i+1} - 2x_i) \\
k=4 &: hf_{i+4} = \frac{1}{12}(25x_{i+4} - 48x_{i+3} + 36x_{i+2} - 16x_{i+1} + 3x_i) \\
k=5 &: hf_{i+5} = \frac{1}{60}(137x_{i+5} - 300x_{i+4} + 300x_{i+3} - 200x_{i+2} + 75x_{i+1} - 12x_i) \\
k=6 &: hf_{i+6} = \frac{1}{60}(147x_{i+6} - 360x_{i+5} + 450x_{i+4} - 400x_{i+3} + 225x_{i+2} - 72x_{i+1} + 10x_i)
\end{aligned}$$

Abkürzungen: $f_{i+j} = f(t_{i+j}, x_N(t_{i+j}))$, $x_{i+j} = x_N(t_{i+j})$, $j = 0, \ldots, 6$.

3.1.3 Lineare Mehrschrittverfahren

Die bisher diskutierten Mehrschrittverfahren gehören zur Klasse der linearen k-stufigen Mehrschrittverfahren.

Definition 3.1.4 (lineares Mehrschrittverfahren)

Gegeben seien $k \in \mathbb{N}$, Koeffizienten $\alpha_j, \beta_j \in \mathbb{R}$, $j = 0, \ldots, k$, ein äquidistantes Gitter $\mathbb{G}_N$ gemäß (2.3) mit Schrittweite $h = (b-a)/N$, sowie Startwerte $x_j = x_N(t_j)$ für $j = 0, \ldots, k-1$.

Die Vorschrift

$$\sum_{j=0}^{k} \alpha_j x_{i+j} = h \sum_{j=0}^{k} \beta_j f(t_{i+j}, x_{i+j}), \qquad i = 0, 1, \ldots, N-k, \tag{3.3}$$

zur Berechnung von x_{i+k} aus den Werten x_{i+j}, $j = 0, \ldots, k-1$, mit $\alpha_k \neq 0$ und $|\alpha_0| + |\beta_0| \neq 0$ heißt **k-stufiges lineares Mehrschrittverfahren**. *Die Funktion*

$$\Phi(t, x^0, \ldots, x^k, h) = \sum_{j=0}^{k} \beta_j f(t_{i+j}, x^j)$$

heißt **Verfahrensfunktion**.
Das k-stufige lineare Mehrschrittverfahren heißt **explizit**, *falls $\beta_k = 0$ gilt, sonst* **implizit**.

Das Mehrschrittverfahren (3.3) heißt linear, weil die Vorschrift linear von den Werten f_{i+j}, $j = 0, \ldots, k$, abhängt.

Bemerkung 3.1.5 **(Initialisierung von Mehrschrittverfahren)**
Zur Durchführung eines k-stufigen linearen Mehrschrittverfahrens werden Startwerte $x_0, \ldots, x_{k-1}$ benötigt. Diese können wie folgt berechnet werden:

- *Mit Hilfe eines geeigneten Einschrittverfahrens, etwa Runge-Kutta-Verfahren, werden ausgehend von $x_0 = x_a$ $k-1$ Schritte des Verfahrens zur Berechnung von $x_1, \ldots, x_{k-1}$ durchgeführt.*
- *Ausgehend von $x_0 = x_a$ wird die Näherung x_j für $j = 1, \ldots, k-1$ mit einem j-stufigen linearen Mehrschrittverfahren berechnet. Die Stufenanzahl des Mehrschrittverfahrens baut sich in der Initialisierungsphase somit schrittweise auf.*

Bei der Initialisierung ist darauf zu achten, daß die Näherungen $x_1, \ldots, x_{k-1}$ mit der richtigen Ordnung approximiert werden, da ansonsten ein Ordnungsverlust im k-stufigen linearen Mehrschrittverfahren erfolgen kann.

Wir interessieren uns für die Konvergenz k-stufiger linearer Mehrschrittverfahren. Der globale Fehler ist wie bei Einschrittverfahren definiert. Beachte hierbei, daß die folgende Definition die Konvergenz der Startwerte $x_0, \ldots, x_{k-1}$ mit entsprechender Ordnung enthält.

Definition 3.1.6 **(Globaler Fehler, Konvergenz)**
Der **globale Fehler** $e_N : \mathbb{G}_N \to \mathbb{R}^n$ *ist definiert durch*

$$e_N := x_N - \Delta_N(\hat{x}).$$

Das Mehrschrittverfahren (3.3) heißt **konvergent**, *wenn*

$$\lim_{N\to\infty} \|e_N\|_\infty = 0.$$

Das Mehrschrittverfahren (3.3) besitzt die **Konvergenzordnung** p, *wenn*

$$\|e_N\|_\infty = \mathcal{O}\left(\frac{1}{N^p}\right) \qquad \text{für } N \to \infty.$$

Den lokalen Diskretisierungsfehler erhält man wie bei Einschrittverfahren, indem die exakte Lösung in die Verfahrensvorschrift (3.3) eingesetzt wird.

Definition 3.1.7 (Lokaler Diskretisierungsfehler, Konsistenz)
Seien $\hat{x} \in \mathbb{R}^n$, $\hat{t} \in [a,b]$ und das k-stufige lineare Mehrschrittverfahren (3.3) gegeben. Es bezeichne y die Lösung des Anfangswertproblems

$$y'(t) = f(t, y(t)), \qquad y(\hat{t}) = \hat{x}.$$

Der **lokale Diskretisierungsfehler in** $(\hat{t}, \hat{x})$ *ist definiert durch*

$$\ell_h(\hat{t}, \hat{x}) := \frac{1}{h}\sum_{j=0}^{k} \alpha_j y(\hat{t} + jh) - \sum_{j=0}^{k} \beta_j f(\hat{t} + jh, y(\hat{t} + jh))$$

Das Mehrschrittverfahren heißt **konsistent in einer Lösung x des AWP (2.1)-(2.2)**, *wenn*

$$\lim_{h\to 0} \left(\max_{t\in[a,b-kh]} \|\ell_h(t, x(t))\| \right) = 0.$$

Das Mehrschrittverfahren besitzt die **Konsistenzordnung p in einer Lösung x des AWP (2.1)-(2.2)**, *wenn es eine von h unabhängige Konstante $C > 0$ und eine Konstante $h_0 > 0$ gibt mit*

$$\max_{t\in[a,b-kh]} \|\ell_h(t, x(t))\| \le Ch^p \qquad \forall 0 < h \le h_0.$$

Wir untersuchen zunächst die Konsistenz eines k-stufigen linearen Mehrschrittverfahrens. Seien dazu $t \in [a, b-kh]$ und x Lösung des AWP (2.1)-(2.2).
Ist f stetig, so gilt mit dem Mittelwertsatz für vektorwertige Funktionen die Beziehung

$$\begin{aligned}
\ell_h(t, x(t)) &= \frac{1}{h}\sum_{j=0}^{k} \alpha_j x(t + jh) - \sum_{j=0}^{k} \beta_j f(t + jh, x(t + jh)) \\
&= \frac{1}{h}\sum_{j=0}^{k} \alpha_j \left(x(t) + \int_0^1 x'(t + sjh) jh ds \right) - \sum_{j=0}^{k} \beta_j x'(t + jh) \\
&= x(t)\sum_{j=0}^{k} \frac{\alpha_j}{h} + \sum_{j=0}^{k} \int_0^1 j\alpha_j x'(t + sjh) - \beta_j x'(t + jh) ds.
\end{aligned}$$

Da x' stetig auf dem Kompaktum $[a,b]$ ist, konvergiert ℓ_h für $h \to 0$ gleichmäßig in t gegen Null, falls noch

$$\sum_{j=0}^{k} \alpha_j = 0 \qquad \text{und} \qquad \sum_{j=0}^{k} (j\alpha_j - \beta_j) = 0$$

gelten. In diesem Fall, ist das Verfahren also konsistent. Wie bei Einschrittverfahren wird die Konsistenzordnung p eines linearen Mehrschrittverfahrens durch Taylorentwicklung des lokalen Diskretisierungsfehlers nachgewiesen:

$$\begin{aligned}
\ell_h(t,x(t)) &= \frac{1}{h}\sum_{j=0}^{k} \alpha_j x(t+jh) - \sum_{j=0}^{k} \beta_j f(t+jh, x(t+jh)) \\
&= \frac{1}{h}\sum_{j=0}^{k} \alpha_j x(t+jh) - \sum_{j=0}^{k} \beta_j x'(t+jh) \\
&= \sum_{j=0}^{k} \left(\frac{\alpha_j}{h} \sum_{\ell=0}^{p} \frac{1}{\ell!} x^{(\ell)}(t)(jh)^\ell - \beta_j \sum_{\ell=0}^{p-1} \frac{1}{\ell!} x^{(\ell+1)}(t)(jh)^\ell \right) + \mathcal{O}(h^p) \\
&= \sum_{j=0}^{k} \left(\frac{\alpha_j}{h} \sum_{\ell=0}^{p} \frac{1}{\ell!} x^{(\ell)}(t)(jh)^\ell - \beta_j \sum_{\ell=1}^{p} \frac{1}{(\ell-1)!} x^{(\ell)}(t)(jh)^{\ell-1} \right) + \mathcal{O}(h^p) \\
&= \sum_{j=0}^{k} \left(\frac{\alpha_j}{h} x(t) + \sum_{\ell=1}^{p} \left(\frac{j^\ell}{\ell!}\alpha_j - \frac{j^{\ell-1}}{(\ell-1)!}\beta_j \right) x^{(\ell)}(t) h^{\ell-1} \right) + \mathcal{O}(h^p)
\end{aligned}$$

Damit ist folgender Satz gezeigt.

Satz 3.1.8 (Konsistenzbedingungen linearer Mehrschrittverfahren)
Sei $f : [a,b]\times\mathbb{R}^n \to \mathbb{R}^n$ p-mal stetig differenzierbar (bzw. x p+1-mal stetig differenzierbar). Dann besitzt das k-stufige lineare Mehrschrittverfahren (3.3) die Konsistenzordnung p, falls die Bedingungen

$$\sum_{j=0}^{k} \alpha_j = 0, \qquad \sum_{j=0}^{k} \left(\frac{j^\ell}{\ell!}\alpha_j - \frac{j^{\ell-1}}{(\ell-1)!}\beta_j \right) = 0 \qquad \forall \ell = 1,\dots,p, \tag{3.4}$$

gelten. Ist f nur stetig, so ist das k-stufige lineare Mehrschrittverfahren konsistent, falls

$$\sum_{j=0}^{k} \alpha_j = 0 \qquad \textit{und} \qquad \sum_{j=0}^{k} (j\alpha_j - \beta_j) = 0$$

gelten.

Da die bisher betrachteten Mehrschrittverfahren (Adams-Verfahren, BDF-Verfahren) durch Interpolation hervorgegangen sind, kann die Konsistenzordnung auch durch Ausnutzen der

Fehlerdarstellung für interpolierende Polynome gezeigt werden. Wir illustrieren dies für Adams-Bashforth-Verfahren.

Satz 3.1.9

Sei $f : [a,b]\times\mathbb{R}^n \to \mathbb{R}^n$ k-mal stetig differenzierbar (bzw. x k+1-mal stetig differenzierbar). Dann besitzt das k-stufige Adams-Bashforth-Verfahren

$$x_{i+k} - x_{i+k-1} = h\sum_{j=0}^{k-1} \beta_{ij} f_{i+j}$$

mit

$$\beta_{ij} = \frac{1}{h}\int_{t_{i+k-1}}^{t_{i+k}} L_j(t)dt, \qquad j = 0,\dots,k-1,$$

die Konsistenzordnung k.

Beweis: Für den lokalen Diskretisierungsfehler (mit äquidistanten Schritten) gilt

$$\begin{aligned}
\ell_h(t,x(t)) &= \frac{1}{h}\left(x(t+kh) - x(t+(k-1)h)\right) - \sum_{j=0}^{k-1}\beta_{ji} f(t+jh, x(t+jh)) \\
&= \frac{1}{h}\int_{t+(k-1)h}^{t+kh} x'(s)ds - \frac{1}{h}\int_{t+(k-1)h}^{t+kh}\sum_{j=0}^{k-1} L_j(s) f(t+jh,x(t+jh))ds \\
&= \frac{1}{h}\int_{t+(k-1)h}^{t+kh}\left(x'(s) - \sum_{j=0}^{k-1} L_j(s) f(t+jh,x(t+jh))\right)ds \\
&= \frac{1}{h}\int_{t+(k-1)h}^{t+kh}\left(f(s,x(s)) - P(s)\right)ds,
\end{aligned}$$

wobei P gerade das interpolierende Polynom zu den Stützpunkten $(t+jh, f(t+jh, x(t+jh)))$, $j = 0,\dots,k-1$, ist. Der Interpolationsfehler kann unter der Voraussetzung, daß f k-mal stetig differenzierbar ist, abgeschätzt werden durch

$$\|f(s,x(s)) - P(s)\| \le \frac{\|f^{(k)}\|_\infty}{k!}\prod_{j=0}^{k-1}|s-(t+jh)| \le \frac{\|f^{(k)}\|_\infty}{k!}\prod_{j=0}^{k-1} kh = Ch^k$$

mit $C = \frac{\|f^{(k)}\|_\infty}{k!}k^k$, wobei $s \in [t+(k-1)h, t+kh]$ ausgenutzt wurde.
Einsetzen liefert die Abschätzung

$$\|\ell_h(t,x(t))\| \le \frac{1}{h}\int_{t+(k-1)h}^{t+kh} Ch^k ds = \frac{1}{h}Ch^{k+1} = Ch^k,$$

so daß die Konsistenzordnung k gezeigt ist. □

Analog zeigt man, daß das k-stufige Adams-Moulton-Verfahren sogar die Ordnung $k+1$ besitzt. Zu beachten ist dabei nur, daß auch der Punkt $(t+kh, f(t+kh, x(t+kh))$ interpoliert wird und das Interpolationspolynom somit den Grad k besitzt.

Im Abschnitt über Einschrittverfahren hatten wir gesehen, daß Konsistenz und Lipschitzstetigkeit der Inkrementfunktion hinreichend für die Konvergenz des Einschrittverfahrens waren. Das folgende Beispiel zeigt, daß Konsistenz und Lipschitzstetigkeit der Verfahrensfunktion für lineare Mehrschrittverfahren i.a. nicht ausreicht, um Konvergenz zu zeigen.

Beispiel 3.1.10 (Stabilität bei Mehrschrittverfahren)

Betrachte das Anfangswertproblem

$$x'(t) = -x(t), \qquad x(0) = 1,$$

im Intervall $[0,1]$ *welches die Lösung*

$$\hat{x}(t) = \exp(-t)$$

besitzt. Zunächst wenden wir das 3-stufige Adams-Bashforth-Verfahren

$$x_{i+3} = x_{i+2} + \frac{h}{12}\left(23 f_{i+2} - 16 f_{i+1} + 5 f_i\right)$$

mit exakten Startwerten $x_0 = 1$, $x_1 = \exp(-h)$, $x_2 = \exp(-2h)$ *und* $h = 1/N$, $N \in \mathbb{N}$, *auf das Anfangswertproblem an und erhalten in Abhängigkeit von* N *folgende globale Fehler:*

N	MSV ERROR	ORDER P
5	0.8250E-03	0.0000E+00
10	0.1230E-03	0.2746E+01
20	0.1638E-04	0.2908E+01
40	0.2103E-05	0.2961E+01
80	0.2663E-06	0.2982E+01
160	0.3348E-07	0.2991E+01
320	0.4198E-08	0.2996E+01
640	0.5255E-09	0.2998E+01

Das 3-stufige Adams-Bashforth-Verfahren besitzt die Konsistenzordnung 3 *und die numerischen Ergebnisse zeigen auch die Konvergenzordnung* 3.
Nun wenden das explizite 2-stufige lineare Mehrschrittverfahren

$$x_{i+2} + 4x_{i+1} - 5x_i = h\left(2f(t_i, x_i) + 4f(t_{i+1}, x_{i+1})\right)$$

mit exakten Startwerten $x_0 = 1$, $x_1 = \exp(-h)$ *und* $h = 1/N$, $N \in \mathbb{N}$, *auf das Anfangswertproblem an und erhalten in Abhängigkeit von* N *folgende globale Fehler:*

N	MSV ERROR
5	0.3069E-01
10	0.7045E+01
20	0.4652E+07
40	0.2883E+20
80	0.1671E+47
160	0.8723+101
320	0.3748+212
640	+Infinity

Die Lösung für $N = 20$ zeigt am Ende des Zeitintervalls starke Oszillationen, die sich mit wachsendem N noch deutlich verstärken.

T	X(T)
0.0000000000000000E+00	0.1000000000000000E+01
0.5000000000000000E-01	0.9512294245007140E+00
0.1000000000000000E+00	0.9048364170970011E+00
0.1500000000000000E+00	0.8607112282460939E+00
0.2000000000000000E+00	0.8187112851417111E+00
0.2500000000000000E+00	0.7788976208106736E+00
0.3000000000000000E+00	0.7403152897895553E+00
0.3500000000000000E+00	0.7072741248561684E+00
0.4000000000000000E+00	0.6569935955729138E+00
0.4500000000000000E+00	0.7062701103889869E+00
0.5000000000000000E+00	0.2529341546735327E+00
0.5500000000000000E+00	0.2398400091277198E+01
0.6000000000000001E+00	-0.8833903025463922E+01
0.6500000000000001E+00	0.4885455315420674E+02
0.7000000000000002E+00	-0.2484752480724415E+03
0.7500000000000001E+00	0.1282983352359867E+04
0.8000000000000002E+00	-0.6606058795466407E+04
0.8500000000000001E+00	0.3403206536752226E+05
0.9000000000000001E+00	-0.1753043626413789E+06
0.9500000000000002E+00	0.9030354433946502E+06
0.1000000000000000E+01	-0.4651740239200287E+07

Obwohl das Mehrschrittverfahren nach (3.4) sogar die Konsistenzordnung 3 besitzt und $f(t,x) = -x$ (und somit auch die Verfahrensfunktion des Mehrschrittverfahrens) lipschitzstetig ist, konvergiert das Verfahren offenbar nicht.

Der Grund für das im Beispiel beobachtete Verhalten des zweiten Verfahrens liegt in der fehlenden Stabilität des Verfahrens. Zwar war die Lipschitzstetigkeit der Verfahrensfunktion bei Einschrittverfahren hinreichend für Stabilität, bei Mehrschrittverfahren ist die Untersuchung der Stabilität jedoch komplizierter.

Definition 3.1.11 (Stabilität)

Es seien $\{x_N\}_{N\in\mathbb{N}}$ Gitterfunktionen mit (3.3) und $h = (b-a)/N$, $N \in \mathbb{N}$. Desweiteren seien $\{y_N\}_{N\in\mathbb{N}}$ Gitterfunktionen $y_N : \mathbb{G}_N \to \mathbb{R}^n$ mit

$$\begin{aligned} \delta_N(t_i) &:= y_N(t_i) - x_N(t_i), \quad i = 0,\dots,k-1, \\ \delta_N(t_{i+k}) &:= \frac{1}{h}\sum_{j=0}^{k} \alpha_j y_N(t_i+jh) - \sum_{j=0}^{k} \beta_j f(t_i+jh, y_N(t_i+jh)), \quad i = 0,\dots,N-k. \end{aligned}$$

Die Funktion $\delta_N : \mathbb{G}_N \to \mathbb{R}^n$ wird als **Defekt** *von y_N bezeichnet.*

Das k-stufige lineare Mehrschrittverfahren heißt **stabil in** *$\{x_N\}_{N\in\mathbb{N}}$, falls es von N unabhängige Konstanten $S, R \geq 0$ gibt, so daß für fast alle $h = (b-a)/N$, $N \in \mathbb{N}$, folgendes gilt:*

Aus

$$\|\delta_N\|_\infty < R$$

folgt

$$\|y_N - x_N\|_\infty \le S\|\delta_N\|_\infty.$$

Die Konstante R heißt **Stabilitätsschwelle** *und S heißt* **Stabilitätsschranke**.

Konsistenz und Stabilität sichern die Konvergenz des Verfahrens, falls die Startwerte hinreichend gut gewählt werden.

Satz 3.1.12 (Konvergenzsatz)
Das k-stufige lineare Mehrschrittverfahren sei konsistent in einer Lösung $\hat{x}$ des AWP (2.1)-(2.2) und stabil in der durch das Mehrschrittverfahren erzeugten Folge $\{x_N\}_{N\in\mathbb{N}}$. Darüber hinaus gelte

$$\lim_{h\to 0} \|x_N(t_i) - \hat{x}(t_i)\| = 0 \qquad \forall i = 0,\dots,k-1. \tag{3.5}$$

Dann ist das Mehrschrittverfahren konvergent.
Besitzt das Mehrschrittverfahren darüber hinaus die Konsistenzordnung p in $\hat{x}$ und gilt

$$\|x_N(t_i) - \hat{x}(t_i)\| = \mathcal{O}(h^p) \qquad \forall i = 0,\dots,k-1, \tag{3.6}$$

so besitzt es die Konvergenzordnung p.

Beweis: Eine Lösung $\hat{x}$ des AWP erfüllt die Vorschrift des Mehrschrittverfahrens mit Schrittweite h i.a. nicht exakt und liefert einen Defekt

$$\begin{aligned}
\delta_N(t_i) &= \hat{x}(t_i) - x_N(t_i), \qquad i = 0,\dots,k-1,\\
\delta_N(t_{i+k}) &= \frac{1}{h}\sum_{j=0}^{k} \alpha_j \hat{x}(t_i + jh) - \sum_{j=0}^{k} \beta_j f(t_i + jh, \hat{x}(t_i + jh)), \qquad i = 0,\dots,N-k.
\end{aligned}$$

Da das Verfahren konsistent ist und (3.5) vorausgesetzt ist, ist es für hinreichend kleine Schrittweiten $h = (b-a)/N$ (bzw. hinreichend großes N) stets möglich, $\|\delta_N\|_\infty < R$ zu erreichen, da wegen $\delta_N(t_{i+k}) = \ell_h(t_i, \hat{x}(t_i))$ auch

$$0 = \lim_{h\to 0} \left(\max_{i=0,\dots,N-k} \|\ell_h(t_i, \hat{x}(t_i))\| \right) = \lim_{N\to\infty} \left(\max_{i=0,\dots,N-k} \|\delta_N(t_{i+k})\| \right)$$

gilt. Da das Verfahren stabil ist, folgt (mit $y_N = \Delta_N(x)$ in Definition 3.1.11)

$$\|e_N\|_\infty = \|x_N - \Delta_N(\hat{x})\|_\infty \le S\|\delta_N\|_\infty.$$

Wegen (3.5) und $\delta_N(t_{i+k}) = \ell_h(t_i, \hat{x}(t_i))$, $i = 0,\dots,N-k$, folgt aus der Konsistenz

$$\lim_{N\to\infty} \|\delta_N\|_\infty = 0$$

bzw. mit (3.6) und der Konsistenzordnung p

$$\|\delta_N\|_\infty = \mathcal{O}\left(\frac{1}{N^p}\right) \qquad \text{für } N \to \infty.$$

Dies zeigt die Konvergenz bzw. die Konvergenz mit Ordnung p. □

Wir versuchen nun, eine hinreichende Bedingung für die Stabilität eines k-stufigen linearen Mehrschrittverfahrens zu finden. Dazu seien Gitterfunktionen x_N und y_N mit Startwerten $x_N(t_i)$ und $y_N(t_i)$, $i = 0, \dots, k-1$, und

$$\begin{aligned}
\sum_{j=0}^{k} \alpha_j x_N(t_i + jh) &= h \sum_{j=0}^{k} \beta_j f(t_i + jh, x_N(t_i + jh)), \\
\sum_{j=0}^{k} \alpha_j y_N(t_i + jh) &= h \sum_{j=0}^{k} \beta_j f(t_i + jh, y_N(t_i + jh)) + h\delta_N(t_{i+k})
\end{aligned}$$

für $i = 0, \dots, N-k$ gegeben, wobei noch $\|\delta_N\|_\infty < R$ gelte. Subtraktion der ersten von der zweiten Gleichung und Division durch $\alpha_k \neq 0$ liefert dann für $i = 0, \dots, N-k$ die Beziehungen

$$\begin{aligned}
\sum_{j=0}^{k} \frac{\alpha_j}{\alpha_k} \left(y_N(t_{i+j}) - x_N(t_{i+j})\right) &= h \sum_{j=0}^{k} \frac{\beta_j}{\alpha_k} \left(f(t_{i+j}, y_N(t_{i+j})) - f(t_{i+j}, x_N(t_{i+j}))\right) \\
&\quad + \frac{h}{\alpha_k} \delta_N(t_{i+k}) \\
&=: b_i,
\end{aligned}$$

bzw.

$$y_N(t_{i+k}) - x_N(t_{i+k}) = b_i - \sum_{j=0}^{k-1} \frac{\alpha_j}{\alpha_k} \left(y_N(t_{i+j}) - x_N(t_{i+j})\right). \tag{3.7}$$

Mit $z_i := y_N(t_i) - x_N(t_i)$ für $i = 0, \dots, N$ und

$$Z_i := \begin{pmatrix} z_i \\ z_{i+1} \\ \vdots \\ z_{i+k-1} \end{pmatrix}, B_i := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ b_i \end{pmatrix}, A := \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -\frac{\alpha_0}{\alpha_k} & -\frac{\alpha_1}{\alpha_k} & \cdots & -\frac{\alpha_{k-2}}{\alpha_k} & -\frac{\alpha_{k-1}}{\alpha_k} \end{pmatrix}$$

für $i = 0, \dots, N-k$ läßt sich (3.7) schreiben als

$$Z_{i+1} = AZ_i + B_i, \qquad i = 0, \dots, N-1, \tag{3.8}$$

mit Startwert $Z_0 = (z_0, \dots, z_{k-1})^\top$. Dies ist formal ein Einschrittverfahren für Z_i und wir können die Stabilität von (3.8) untersuchen. Ziel dabei ist es, eine Bedingung zu finden,

die garantiert, daß die Werte Z_i für $i = 0, 1, 2, \ldots$ beschränkt bleiben. Beachte, daß die Nichtlinearitäten aus (3.7) in der Inhomogenität B_i versteckt sind, wobei B_i noch von y_N und x_N abhängt. Zunächst aber analysieren wir (3.8) ohne die Störungen B_i, $i = 0, 1, \ldots$. Dann folgt $Z_i = A^i Z_0$ und es stellt sich die Frage, wann Z_i bzw. A^i für jeden Startwert Z_0 beschränkt bleiben.

Hilfsatz 3.1.13

Für $A \in \mathbb{R}^{n\times n}$ sind folgende Aussagen äquivalent:

(a) Die Folge $\{A^i\}_{i\in\mathbb{N}}$ ist beschränkt, d.h. es gibt eine Konstante $C \geq 0$ mit

$$\|A^i\| \leq C \qquad \forall i = 0, 1, 2, \ldots.$$

(b) Es gilt $|\lambda| \leq 1$ für jeden Eigenwert λ von A und für jeden Eigenwert λ mit $|\lambda| = 1$ stimmen algebraische und geometrische Vielfachheit überein.

Beweis: Der Beweis kann mit Hilfe der Jordanzerlegung von A geführt werden. Einen ähnlichen Beweis hatten wir schon in Numerik I, Satz 2.6.8 im Zusammenhang mit iterativen Lösungsverfahren zur Lösung von Gleichungssystemen geführt. Wir überlassen die technischen Details dem Leser. □

Beachte, daß A gerade die Transponierte der Frobenius-Begleitmatrix (vgl. Beispiel 1.2.3) zum Polynom

$$\mu^k + \sum_{j=0}^{k-1} \frac{\alpha_j}{\alpha_k} \mu^j$$

ist. Die Transponierte hat dieselben Eigenwerte wie die Frobenius-Begleitmatrix. Damit sind die Eigenwerte von A gerade durch die Nullstellen des sogenannten **erzeugenden Polynoms des k-stufigen linearen Mehrschrittverfahrens**

$$q(\mu) = \alpha_k \mu^k + \alpha_{k-1} \mu^{k-1} + \ldots + \alpha_0$$

gegeben. Die Stabilitätsbedingung aus Hilfsatz 3.1.13 wird zu Ehren von dessen Entdecker Dahlquist als Wurzelbedingung oder Nullstabilität bezeichnet.

Definition 3.1.14 (Wurzelbedingung von Dahlquist, Nullstabilität)

Das k-stufige lineare Mehrschrittverfahren (3.3) zur Lösung des AWP (2.1)-(2.2) heißt **nullstabil** *(bzw. erfüllt die* **Wurzelbedingung von Dahlquist***), falls für das erzeugende Polynom*

$$q(\mu) = \alpha_k \mu^k + \alpha_{k-1} \mu^{k-1} + \ldots + \alpha_0$$

die folgenden Bedingungen erfüllt sind:

(i) *Für jede Nullstelle λ von q gilt $|\lambda| \leq 1$.*

(ii) *Jede Nullstelle λ von q mit $|\lambda| = 1$ ist λ einfache Nullstelle von q.*

Bemerkung 3.1.15

Aus dem Beweis zu Satz 3.1.8 folgt, daß ein konsistentes Mehrschrittverfahren notwendig die Bedingung $0 = \sum_{j=0}^{k} \alpha_k = q(1)$ erfüllen muß. Damit besitzt ein konsistentes Mehrschrittverfahren stets 1 als Nullstelle des erzeugenden Polynoms.

Beispiel 3.1.16

(a) *Das erzeugende Polynom q für das Mehrschrittverfahren in Beispiel 3.1.10 lautet*

$$q(\mu) = \mu^2 + 4\mu - 5$$

und besitzt die Nullstellen $\lambda_1 = -5$ und $\lambda_2 = 1$, wobei λ_2 eine zweifache Nullstelle ist. Somit ist dieses Verfahren nicht nullstabil.

(b) *Das k-stufige Adams-Bashforth-Verfahren*

$$x_{i+k} - x_{i+k-1} = h \sum_{j=0}^{k-1} \beta_{ij} f_{i+j}$$

und das k-stufige Adams-Moulton-Verfahren

$$x_{i+k} - x_{i+k-1} = h \sum_{j=0}^{k} \beta_{ij} f_{i+j}$$

besitzen das erzeugende Polynom $q(\mu) = \mu^k - \mu^{k-1} = \mu^{k-1}(\mu - 1)$ und somit die einfache Nullstelle $\lambda = 1$ und die $(k-1)$-fache Nullstelle $\lambda = 0$. Damit sind diese Verfahren nullstabil.

Bemerkung 3.1.17 (1. Dahlquist-Schranke)

Strehmel und Weiner [SW95, Satz 4.3.1] zeigen, daß ein nullstabiles k-stufiges lineares Mehrschrittverfahren höchstens folgende Konsistenzordnung p besitzen kann:

(i) *Für k gerade, ist die maximale Konsistenzordnung $p \leq k + 2$.*

(ii) *Für k ungerade, ist die maximale Konsistenzordnung $p \leq k + 1$.*

(iii) *Für $\beta_k / \alpha_k \leq 0$, ist die maximale Konsistenzordnung $p \leq k$.*

Diese Maximalordnungen werden auch angenommen.

Nun kommen wir zum entscheidenden Satz.

Satz 3.1.18 (Stabilität von Mehrschrittverfahren)
Das k-stufige lineare Mehrschrittverfahren (3.3) sei nullstabil und $f : [a,b] \times \mathbb{R}^n \to \mathbb{R}^n$ erfülle die Lipschitzbedingung

$$\|f(t,x) - f(t,y)\| \leq L\|x-y\| \qquad \forall x, y \in \mathbb{R}^n$$

mit einer Konstanten $L \geq 0$. Dann ist das Verfahren stabil im Sinne von Definition 3.1.11.

Beweis: Aus (3.8) folgt induktiv, daß

$$Z_i = A^i Z_0 + \sum_{j=0}^{i-1} A^{i-1-j} B_j$$

für $i = 0, \ldots, N-k+1$ gilt. Daraus folgt sofort die Abschätzung

$$\|Z_i\|_\infty \leq \|A^i\|_\infty \cdot \|Z_0\|_\infty + \sum_{j=0}^{i-1} \|A^{i-1-j}\|_\infty \cdot \|B_j\|_\infty.$$

Nach Hilfsatz 3.1.13 gibt es eine Konstante C mit $\|A^i\|_\infty \leq C$ für alle $i = 0, 1, 2, \ldots$, und somit gilt

$$\|Z_i\|_\infty \leq C \left(\|Z_0\|_\infty + \sum_{j=0}^{i-1} \|B_j\|_\infty \right). \tag{3.9}$$

Nach Definition von B_i folgt mit $\beta := \max\limits_{j=0,\ldots,k} |\beta_j|$ weiter

$$\begin{aligned}
\|B_i\|_\infty &= \|b_i\|_\infty \\
&= \frac{h}{\alpha_k} \left\| \sum_{j=0}^{k} \beta_j \left(f(t_{i+j}, y_N(t_{i+j})) - f(t_{i+j}, x_N(t_{i+j})) \right) + \delta_N(t_{i+k}) \right\|_\infty \\
&\leq \frac{h}{\alpha_k} \beta L \sum_{j=0}^{k} \|z_{i+j}\|_\infty + \frac{h}{\alpha_k} \|\delta_N(t_{i+k})\|_\infty \\
&\leq \frac{h}{\alpha_k} \beta L \left(k\|Z_i\|_\infty + \|Z_{i+1}\|_\infty \right) + \frac{h}{\alpha_k} \max_{i=0,\ldots,N-k} \|\delta_N(t_{i+k})\|_\infty.
\end{aligned}$$

Summation mit $i \le N$ und $h = (b-a)/N$ liefert

$$\begin{aligned}\sum_{j=0}^{i-1} \|B_j\|_\infty &\le \frac{h}{\alpha_k}\beta L \sum_{j=0}^{i-1} (k\|Z_j\|_\infty + \|Z_{j+1}\|_\infty) + N\frac{h}{\alpha_k} \max_{j=0,\dots,N-k} \|\delta_N(t_{j+k})\|_\infty \\ &= \frac{h}{\alpha_k}\beta L k \sum_{j=0}^{i-1} \|Z_j\|_\infty + \frac{h}{\alpha_k}\beta L \sum_{j=1}^{i} \|Z_j\|_\infty + \frac{b-a}{\alpha_k} \max_{j=0,\dots,N-k} \|\delta_N(t_{j+k})\|_\infty \\ &\le \frac{h}{\alpha_k}\beta L(k+1) \sum_{j=0}^{i-1} \|Z_j\|_\infty + \frac{h}{\alpha_k}\beta L\|Z_i\|_\infty + \frac{b-a}{\alpha_k}\|\delta_N\|_\infty.\end{aligned}$$

Einsetzen in (3.9) liefert

$$\|Z_i\|_\infty \le C\left(\|Z_0\|_\infty + \frac{b-a}{\alpha_k}\|\delta_N\|_\infty + \frac{h}{\alpha_k}\beta L(k+1)\sum_{j=0}^{i-1}\|Z_j\|_\infty + \frac{h}{\alpha_k}\beta L\|Z_i\|_\infty\right).$$

Auflösen nach $\|Z_i\|_\infty$ mit $0 < h < h_0 := \frac{\alpha_k}{C\beta L}$ führt auf die Abschätzung

$$\begin{aligned}\|Z_i\|_\infty &\le \frac{C}{1-\frac{hC\beta L}{\alpha_k}}\left(\|Z_0\|_\infty + \frac{b-a}{\alpha_k}\|\delta_N\|_\infty + \frac{h}{\alpha_k}\beta L(k+1)\sum_{j=0}^{i-1}\|Z_j\|_\infty\right) \\ &\le C_1\|\delta_N\|_\infty + C_2 h\sum_{j=0}^{i-1}\|Z_j\|_\infty\end{aligned}$$

mit $C_1 = \frac{C}{1-\frac{h_0 C\beta L}{\alpha_k}}\left(1+\frac{b-a}{\alpha_k}\right)$ und $C_2 = \frac{C\beta L(k+1)}{\alpha_k - h_0 C\beta L}$. Anwendung von Hilfsatz 3.1.19 mit $\alpha = C_1\|\delta_N\|_\infty$ und $\beta = C_2$ liefert

$$\|Z_i\|_\infty \le C_1\|\delta_N\|_\infty \exp(ihC_2) \le C_1\|\delta_N\|_\infty \exp(NhC_2) = C_1\|\delta_N\|_\infty \exp((b-a)C_2),$$

so daß

$$\|Z_i\|_\infty \le S\|\delta_N\|_\infty, \qquad S := C_1\exp((b-a)C_2),$$

für alle $i = 0,\dots,N-k+1$ gilt. Aus der Definition von Z_i folgt schließlich

$$\|y_N - x_N\|_\infty \le S\|\delta_N\|_\infty,$$

was gerade die Stabilität des Mehrschrittverfahrens ist. □

Im Beweis haben wir das folgende Resultat verwendet.

Hilfsatz 3.1.19

Für Zahlen $\alpha, h, \beta \ge 0$ und $a_i \ge 0$, $i = 0,\dots,N$ gelte

$$a_0 \le \alpha, \qquad a_i \le \alpha + h\beta\sum_{j=0}^{i-1} a_j, \quad i = 1,2,\dots,N.$$

Dann gilt

$$a_i \leq \alpha \exp(\beta i h), \qquad i = 0, 1, \ldots, N.$$

Beweis: Definiere

$$w_i := \alpha + h\beta \sum_{j=0}^{i-1} a_j, \quad i = 0, \ldots, N.$$

Nach Voraussetzung gilt $a_i \leq w_i$ für alle $i = 1, \ldots, N$, sowie $a_0 \leq \alpha = w_0$.
Dann gilt für $0 \leq i < N$

$$w_{i+1} - w_i = h\beta a_i \leq h\beta w_i$$

bzw.

$$w_{i+1} \leq (1 + h\beta) w_i.$$

Induktiv ergibt sich daraus

$$w_i \leq (1 + h\beta)^i w_0 = \alpha(1 + h\beta)^i.$$

Weiter folgt

$$a_i \leq w_i \leq \alpha(1 + h\beta)^i \leq \alpha \exp(\beta h)^i = \alpha \exp(\beta i h).$$

□

Die Sätze 3.1.12 und 3.1.18 liefern zusammen den folgenden Konvergenzsatz für lineare Mehrschrittverfahren, der in Kurzform

$$\text{Konsistenz} + \text{Stabilität} \Rightarrow \text{Konvergenz}$$

lautet.

Satz 3.1.20 (Konvergenzssatz für lineare Mehrschrittverfahren)
$f : [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$ erfülle die Lipschitzbedingung

$$\|f(t, x) - f(t, y)\| \leq L\|x - y\| \qquad \forall x, y \in \mathbb{R}^n$$

mit einer Konstanten $L \geq 0$.
Das k-stufige lineare Mehrschrittverfahren (3.3) sei nullstabil und konsistent in einer Lösung $\hat{x}$ des AWP (2.1)-(2.2). Darüber hinaus gelte

$$\lim_{h \to 0} \|x_N(t_i) - \hat{x}(t_i)\| = 0 \qquad \forall i = 0, \ldots, k - 1.$$

Dann ist das Mehrschrittverfahren konvergent.
Besitzt das Mehrschrittverfahren darüber hinaus die Konsistenzordnung p in $\hat{x}$ und gilt

$$\|x_N(t_i) - \hat{x}(t_i)\| = \mathcal{O}(h^p) \qquad \forall i = 0, \ldots, k - 1,$$

so besitzt es die Konvergenzordnung p.

Bemerkung 3.1.21

Die globale Lipschitzbedingung für f im Konvergenzsatz kann durch eine lokale Lipschitzbedingung, die in der Lösung $\hat{x}$ gelten muß, abgeschwächt werden.

3.1.4 Prädiktor-Korrektor-Verfahren

Gegeben sei ein implizites k-stufiges lineares Mehrschrittverfahren in der Form

$$x_{i+k} + \frac{1}{\alpha_k} \sum_{j=0}^{k-1} \alpha_j x_{i+j} = h \frac{1}{\alpha_k} \sum_{j=0}^{k} \beta_j f(t_{i+j}, x_{i+j}) \tag{3.10}$$

mit $\alpha_k \neq 0$ und $\beta_k \neq 0$. Ein Ansatz zur Lösung dieser nichtlinearen Gleichung für x_{i+k} ist die Fixpunktiteration

$$x_{i+k}^{(\ell+1)} + \frac{1}{\alpha_k} \sum_{j=0}^{k-1} \alpha_j x_{i+j} = h \frac{1}{\alpha_k} \left(\beta_k f(t_{i+k}, x_{i+k}^{(\ell)}) + \sum_{j=0}^{k-1} \beta_j f(t_{i+j}, x_{i+j}) \right), \ \ell = 0, 1, 2, \ldots, \tag{3.11}$$

mit geeignetem Startwert $x_{i+k}^{(0)}$. Man kann zeigen (Übung!), daß die Fixpunktiteration konvergiert, falls

$$h \left| \frac{\beta_k}{\alpha_k} \right| L < 1$$

gilt, wobei L eine Lipschitzkonstante von f ist. Für hinreichend kleines h ist diese Bedingung stets erfüllbar, wobei die Schrittweite für sehr großes L sehr klein gewählt werden muss.

Um den Aufwand bei der Durchführung des Verfahrens zu reduzieren, wird die nichtlineare Gleichung zur Bestimmung der Näherung x_{i+k} nicht exakt (im Sinne der Rechengenauigkeit) gelöst, sondern es wird ein sogenannter **Prädiktorschritt** mit einem expliziten k-stufigen linearen Mehrschrittverfahren der Form

$$x_{i+k} + \frac{1}{\alpha_k^P} \sum_{j=0}^{k-1} \alpha_j^P x_{i+j} = h \frac{1}{\alpha_k^P} \sum_{j=0}^{k-1} \beta_j^P f(t_{i+j}, x_{i+j}) \tag{3.12}$$

mit $\alpha_k^P \neq 0$ durchgeführt, welches die Approximation $x_{i+k}^{(0)}$ liefert. Anschließend werden ein oder mehrere **Korrektorschritte** für das implizite Verfahren (3.10) im Sinne einer Fixpunktiteration (3.11) durchgeführt. Hierbei wird diese Iteration in der Regel jedoch nicht bis zur Konvergenz durchgeführt, sondern aus Effizienzgründen nur $M \in \mathbb{N}$ Mal angewendet. Konkret lautet das Prädiktor-Korrektor-Verfahren wie folgt:

Algorithmus 3.1.22 (Prädiktor-Korrektor-Verfahren)

(0) Gegeben seien $M \in \mathbb{N}$, $k \in \mathbb{N}$, $h = (b-a)/N$, $N \in \mathbb{N}$ und Startwerte $x_0, \dots, x_{k-1}$.

(1) Für $i = 0, 1, \dots, N-k$:

Berechne Prädiktor

$$x_{i+k}^{(0)} + \frac{1}{\alpha_k^P} \sum_{j=0}^{k-1} \alpha_j^P x_{i+j} = h \frac{1}{\alpha_k^P} \sum_{j=0}^{k-1} \beta_j^P f(t_{i+j}, x_{i+j}).$$

Für $\ell = 0, 1, \dots, M-1$:

Berechne $f_{i+k}^{(\ell)} = f(t_{i+k}, x_{i+k}^{(\ell)})$.

Berechne Korrektor

$$x_{i+k}^{(\ell+1)} + \frac{1}{\alpha_k} \sum_{j=0}^{k-1} \alpha_j x_{i+j} = h \frac{1}{\alpha_k} \left(\beta_k f_{i+k}^{(\ell)} + \sum_{j=0}^{k-1} \beta_j f(t_{i+j}, x_{i+j}) \right).$$

end

Setze $x_{i+k} = x_{i+k}^{(M)}$.

end

Konvergenzaussagen zum Prädiktor-Korrektor-Verfahren liefert folgender Satz.

Satz 3.1.23 (Strehmel und Weiner [SW95, Satz 4.6.1])

Sei f hinreichend oft stetig differenzierbar. Der Prädiktor habe die Konsistenzordnung $p^* \geq 1$ und der Korrektor die Konsistenzordnung $p \geq 1$. Dann besitzt das Prädiktor-Korrektor-Verfahren die Konsistenzordnung $p = \min\{p^* + M, p\}$, falls die Startwerte von dieser Ordnung sind. Ist das Korrektorverfahren nullstabil, so konvergiert das Prädiktor-Korrektor-Verfahren von der Ordnung p.

Kapitel 4

Steife Differentialgleichungen

In den vorangegangenen Kapiteln haben wir die Konvergenz von Ein- und Mehrschrittverfahren untersucht. Diese konnte für konsistente und stabile Diskretisierungen gezeigt werden. In der Praxis spielen neben dem in der Konvergenzanalyse verwendeten Stabilitätsbegriff jedoch noch andere Stabilitätsbegriffe eine große Rolle. Im folgenden werden wir die sogenannte A-Stabilität im Zusammenhang mit steifen Differentialgleichungen untersuchen. Zur Motivation betrachten wir das folgende Beispiel.
Betrachte für $\lambda_1 < 0$ und $\lambda_2 < 0$ das Anfangswertproblem

$$x'(t) = Ax(t), \qquad x(0) = x_a, \tag{4.1}$$

mit

$$A = \begin{pmatrix} \frac{\lambda_1+\lambda_2}{2} & \frac{\lambda_1-\lambda_2}{2} \\ \frac{\lambda_1-\lambda_2}{2} & \frac{\lambda_1+\lambda_2}{2} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad x_a = \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

Die Matrix A besitzt die Eigenwerte λ_1 und λ_2 und die zugehörigen Eigenvektoren $v_1 = (1,1)^\top$ und $v_2 = (1,-1)^\top$, sowie die Lösung

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \exp(\lambda_1 t) + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \exp(\lambda_2 t). \tag{4.2}$$

Es gilt

$$\lim_{t\to\infty} x(t) = 0.$$

Anwendung des expliziten Eulerverfahrens mit Schrittweite $h > 0$ liefert

$$x_{i+1} = (I + hA)x_i, \qquad i = 0, 1, 2, \ldots,$$

bzw.

$$x_i = (I + hA)^i x_a, \qquad i = 0, 1, 2, \ldots,$$

Die Matrix $I + hA$ läßt sich diagonalisieren gemäß

$$T^{-1}(I + hA)T = diag(1 + h\lambda_1, 1 + h\lambda_2) =: I + h\Lambda, \qquad T = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Einsetzen liefert

$$T^{-1}x_i = (I + h\Lambda)^i T^{-1}x_a, \qquad i = 0, 1, 2, \ldots.$$

Die numerische Lösung x_i, $i = 0, 1, 2, \ldots$, zeigt genau dann dasselbe qualitative Verhalten wie die exakte Lösung in (4.2), also

$$\lim_{i \to \infty} x_i = 0,$$

wenn die Bedingungen

$$|1 + h\lambda_1| < 1, \qquad |1 + h\lambda_2| < 1$$

bzw.

$$h < \frac{2}{|\lambda_1|}, \qquad h < \frac{2}{|\lambda_2|}$$

erfüllt sind. Sei nun etwa speziell $\lambda_1 = -2$ und $\lambda_2 = -2 \cdot 10^5$. Dann muß h mit $h < 10^{-5}$ sehr klein gewählt werden, um dasselbe qualitative Lösungsverhalten zu erhalten, wobei diese Einschränkung der Schrittweite durch den Eigenwert λ_2 bestimmt wird. Allerdings geht dieser Eigenwert λ_2 lediglich durch den Term $\exp(\lambda_2 t)$ in die exakte Lösung in (4.2) ein und hat – ausser am Anfang – nahezu keinen Einfluß auf die exakte Lösung. Mit anderen Worten: Ein vernachlässigbarer Lösungsanteil ist verantwortlich für eine starke Einschränkung der Schrittweite, wodurch das explizite Eulerverfahren sehr ineffizient wird, vgl. Abbildung 4.1. Dieses Phänomen wird **Steifheit** genannt. Es tritt auch bei expliziten Runge-Kutta-Verfahren höherer Ordnung auf.

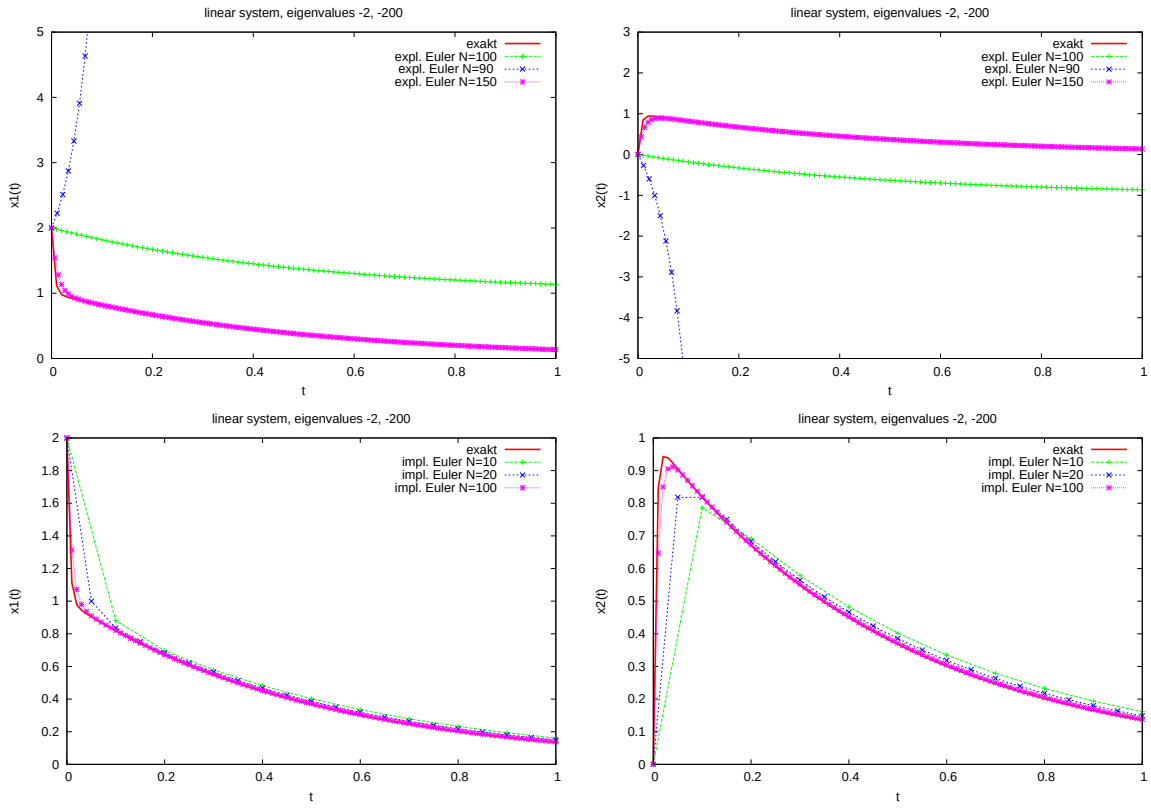


Abbildung 4.1: Vergleich des expliziten (oben) und impliziten Eulerverfahrens (unten) für $\lambda_1 = -2$ und $\lambda_2 = -200$: Das implizite Verfahren liefert unabhängig von der Schrittweite $h = 1/N$ gute Approximationen. Das explizite Verfahren liefert erst für sehr kleine Schrittweiten gute Näherungen. Beide Verfahren konvergieren mit Ordnung h.

Es gibt in der Literatur keine einheitliche Definition der Steifheit. Für lineare Systeme $x' = Ax$ wird häufig der Wert

$$K := \max_{i=1,\dots,k} \{|Re(\lambda_i)| \mid Re(\lambda_i) < 0\}$$

als Mass für die Steifheit verwendet. Häufig wird auch noch die Länge des Zeitintervalls berücksichtigt und $(b-a)K$ als Mass der Steifheit verwendet, da beobachtet wird, daß Systeme mit großem K, aber kurzem Zeitintervall $b-a$ sehr wohl effizient mit expliziten Verfahren gelöst werden können und daher kein steifes Verhalten zeigen, während Systeme mit kleinem K, aber langem Zeitintervall $b-a$ sehr wohl steifes Verhalten zeigen können. Man versucht nun, diese Definition durch Betrachtung der Jacobimatrix f'_x auf nichtlineare Systeme zu erweitern, wobei Steifheit im nichtlinearen Fall i.a. von der Stelle (t, x) abhängt und sich somit steife und nichtsteife Bereiche für dieselbe Diffferentialgleichung ergeben können.

Definition 4.0.1 **(Steifheit)**

Die Differentialgleichung

$$x'(t) = f(t, x(t))$$

heißt **steif in** $(t,x) \in \mathbb{R}^{n+1}$*, wenn für die Eigenwerte* λ_i*,* $i = 1, \ldots, k$*, von* $f'_x(t,x)$ *folgendes gilt:*

$$K := \max_{i=1,\ldots,k} \{|Re(\lambda_i)| \mid Re(\lambda_i) < 0\} \gg 1$$

Bemerkung 4.0.2

In der Literatur werden Systeme alternativ als steif bezeichnet, falls eine der folgenden Bedingungen erfüllt ist:

- $$\frac{\max\limits_{i=1,\ldots,k} \{|Re(\lambda_i)| \mid Re(\lambda_i) < 0\}}{\min\limits_{i=1,\ldots,k} \{|Re(\lambda_i)| \mid Re(\lambda_i) < 0\}} \gg 1.$$
- $\|f'_x\| \gg 1$
- *Erfüllt* f *eine Lipschitzbedingung* $\|f(t,x) - f(t,y)\| \leq L\|x - y\|$*, so gilt für die Lipschitzkonstante* $L \gg 1$*.*

Anschaulich spricht man von Steifheit, wenn die Lösung eines AWP Lösungsanteile besitzt, die ein stark unterschiedliches Zeitverhalten aufweisen, also etwa sehr schnelle und sehr langsame Anteile. Dies tritt beispielsweise in chemischen Reaktionen auf, in denen einige Reaktionspartner sehr schnell reagieren und in einen Sättigungsbereich laufen, während andere Stoffe relativ langsam reagieren. Ein anderer Bereich, in dem steife Differentialgleichungen auftreten, ist die Mechanik, bei der etwa sehr steife Feder- oder Dämpferelemente in einem technischen System vorhanden sind.
Es ist wichtig zu bemerken, daß Steifheit ein rein numerisches Problem ist. Nur die Realteile ungleich Null der Eigenwerte spielen hierbei eine Rolle. Die Imaginärteile sind für (stark) oszillierende Lösungskomponenten verantwortlich (man mache sich dies für lineare Dgln. klar) und erfordern aus Genauigkeitsgründen kleine Schrittweiten. Die Eigenwerte mit positivem Realteil können ebenfalls kleine Schrittweiten erzwingen, aber in diesem Fall sind kleine Schrittweiten notwendig, um die dominierenden Lösungsanteile gut zu approximieren. Es wird sich herausstellen, daß explizite Runge-Kutta-Verfahren für steife Differentialgleichungen nicht geeignet sind, während implizite Runge-Kutta-Verfahren sehr wohl geeignet sind. Wir illustrieren dies stellvertretend am impliziten Eulerverfahren für das zuvor diskutierte Problem.
Das implizite Eulerverfahren angewendet auf (4.1) führt auf die Vorschrift

$$x_{i+1} = (I - hA)^{-1} x_i, \qquad i = 0, 1, 2, \ldots,$$

bzw.

$$x_i = (I - hA)^{-i} x_a, \qquad i = 0, 1, 2, \ldots.$$

Die Matrix $(I - hA)^{-1}$ kann wiederum diagonalisiert werden:

$$T^{-1}(I - hA)^{-1}T = diag\left(\frac{1}{1 - h\lambda_1}, \frac{1}{1 - h\lambda_2}\right) =: (I - h\Lambda)^{-1}, \qquad T = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Einsetzen liefert

$$T^{-1}x_i = (I - h\Lambda)^{-i} T^{-1} x_a, \qquad i = 0, 1, 2, \ldots.$$

Die numerische Lösung x_i, $i = 0, 1, 2, \ldots$, zeigt genau dann dasselbe qualitative Verhalten wie die exakte Lösung in (4.2), also

$$\lim_{i \to \infty} x_i = 0,$$

wenn die Bedingungen

$$\frac{1}{|1 - h\lambda_1|} < 1, \qquad \frac{1}{|1 - h\lambda_2|} < 1$$

gelten. Diese Bedingungen sind für alle $h > 0$ erfüllt, da $\lambda_1 < 0$ und $\lambda_2 < 0$ vorausgesetzt war. Im Gegensatz zum expliziten Eulerverfahren schränken die Eigenwerte die Schrittweite nicht ein, egal wie klein sie sind, vgl. Abbildung 4.1.

4.1 A-Stabilität

Das Phänomen der Steifheit untersucht man häufig an Hand der **Testgleichung**

$$x'(t) = \lambda x(t), \qquad \lambda \in \mathbb{C},\ Re(\lambda) \le 0. \tag{4.3}$$

Dies ist für lineare Differentialgleichungen

$$x'(t) = Ax(t)$$

mit einer diagonalisierbaren Matrix $A \in \mathbb{R}^{n \times n}$ keine Einschränkung, da die diagonalisierbare Matrix A zerlegt werden kann in

$$T^{-1}AT = \Lambda = diag(\lambda_1, \ldots, \lambda_n),$$

wobei T regulär und Λ Diagonalmatrix mit den Eigenwerten von A ist. Einsetzen in die lineare Differentialgleichung liefert mit $y := T^{-1}x$ die Differentialgleichung

$$y'(t) = \Lambda y(t) \quad \Leftrightarrow \quad y_i'(t) = \lambda_i y_i(t),\ i = 1, \ldots, n,$$

welche entkoppelt ist und somit komponentenweise betrachtet werden kann, was schließlich auf die Testgleichung (4.3) führt.

Nichtlineare Differentialgleichungen können ebenfalls lokal um eine Referenzlösung x linearisiert werden. Sei dazu f hinreichend glatt, x eine Lösung der Differentialgleichung

$$x'(t) = f(t, x(t)), \qquad t \in [a, b].$$

und $y(t)$ eine Lösung der Differentialgleichung

$$y'(t) = f(t, y(t)), \qquad t \in [a, b],$$

die benachbart zu x sein soll. Dann gilt für $\delta(t) := y(t) - x(t)$

$$\delta'(t) = y'(t) - x'(t) = f(t, y(t)) - f(t, x(t)) \approx f'_x(t, x(t))(y(t) - x(t)) = f'_x(t, x(t))\delta(t).$$

Mit $A(t) := f'_x(t, x(t))$ entsteht eine lineare Differentialgleichung $\delta'(t) = A(t)\delta(t)$, die allerdings i.a. zeitabhängige Koeffizienten besitzt. Hängt f jedoch nicht explizit von t ab und ist x eine Gleichgewichtslösung mit $x'(t) \equiv 0$, so ist A zeitlich konstant.
Insofern beschränken wir uns im Folgenden auf die Betrachtung der Testgleichung (4.3). Diese besitzt die Lösung

$$\hat{x}(t) = c \exp(\lambda t) = c \exp(Re(\lambda)t) \left(\cos(Im(\lambda)t) + i \sin(Im(\lambda)t)\right) \tag{4.4}$$

mit einer Konstanten $c \in \mathbb{C}$. Für $Re(\lambda) \leq 0$ gilt offenbar $\lim\limits_{t\to\infty} |\hat{x}(t)| \leq M$ mit einer Konstanten M.
Wendet man nun das explizite Eulerverfahren auf die Testgleichung (4.3) an, so erhält man

$$x_{i+1} = x_i + h\lambda x_i = (1 + h\lambda)x_i =: R(\lambda h)x_i$$

mit der Funktion $R(z) = 1 + z$. Die Funktion R heißt **Stabilitätsfunktion** des expliziten Eulerverfahrens. Induktiv ergibt sich

$$x_i = R(\lambda h)^i x_0$$

und die numerische Lösung x_i, $i = 0, 1, 2, \ldots$, bleibt für jeden Anfangswert x_0 genau dann beschränkt, wenn

$$|R(\lambda h)| \leq 1$$

gilt. Die Schrittweite $h > 0$ muß dabei so gewählt werden, daß λh im **Stabilitätsgebiet**

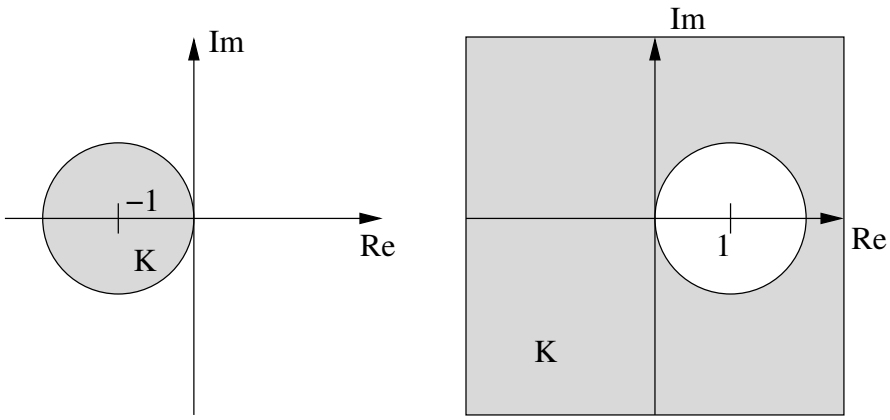
$$S := \{z \in \mathbb{C} \mid |R(z)| \leq 1\}$$

des expliziten Eulerverfahrens liegt. Das Stabilitätsgebiet des expliziten Eulerverfahrens ist ein Kreis mit Radius 1 um den Mittelpunkt -1, vgl. Abbildung 4.2. Da $Re(\lambda) \leq 0$ vorausgesetzt war, muß h hinreichend klein gewählt werden.

Dieselbe Prozedur für das implizite Eulerverfahren liefert

$$x_{i+1} = R(\lambda h)x_i \quad \text{mit} \quad R(z) = \frac{1}{1-z}.$$

Schrittweite $h > 0$ muß wieder so gewählt werden, daß λh im Stabilitätsgebiet des impliziten Eulerverfahrens verbleibt. Das Stabilitätsgebiet des impliziten Eulerverfahrens ist der ganze Raum $\mathbb{C}$ mit Ausnhame des Kreises mit Radius 1 um den Mittelpunkt 1, vgl. Abbildung 4.2. Da $Re(\lambda) \leq 0$ vorausgesetzt war, liegt λh für $h > 0$ stets im Stabilitätsgebiet des impliziten Eulerverfahrens.



Stabilitätsgebiet expliziter Euler Stabilitätsgebiet impliziter Euler

Abbildung 4.2: Vergleich der Stabilitätsgebiete des expliziten (links) und impliziten (rechts) Eulerverfahrens. Das explizite Verfahren ist nicht A-stabil, das implizite ist A-stabil.

Nahezu alle gängigen Einschrittverfahren führen bei Anwendung auf die Testgleichung auf die Vorschrift

$$x_{i+1} = R(\lambda h)x_i$$

mit der Stabilitätsfunktion $R(z)$.

Beispiel 4.1.1

Die Stabilitätsfunktion des expliziten Eulerverfahrens lautet

$$R(z) = 1 + z.$$

Die Stabilitätsfunktion des impliziten Eulerverfahrens lautet

$$R(z) = \frac{1}{1-z}.$$

Die Stabilitätsfunktion des Heun-Verfahrens, vgl. Algorithmus 2.3.4, lautet

$$R(z) = 1 + z + \frac{1}{2}z^2.$$

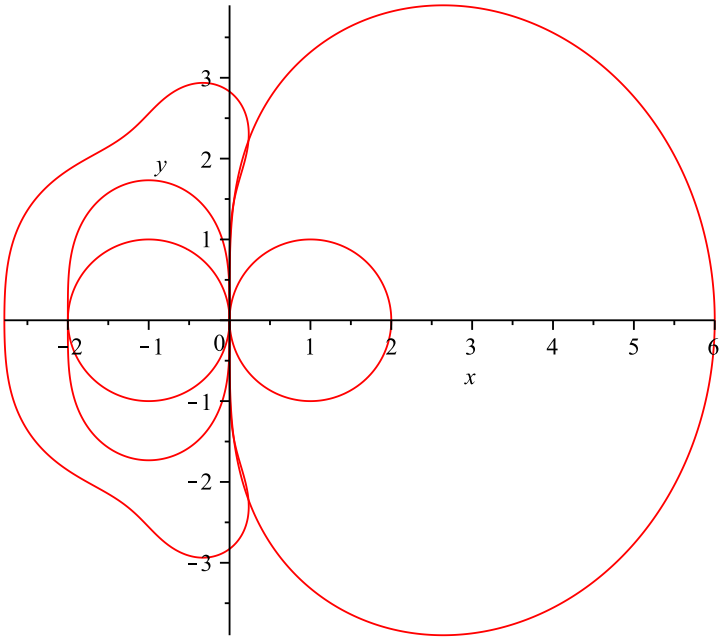
Die Stabilitätsfunktion des klassischen Runge-Kutta-Verfahrens, vgl. Beispiel 2.3.7, lautet

$$R(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4.$$

Die Stabilitätsfunktion des impliziten Radau-IIA-Verfahrens, vgl. Beispiel 2.3.8, lautet

$$R(z) = \frac{2(z+3)}{z^2 - 4z + 6}.$$

Die folgende Abbildung zeigt die Stabilitätsgebiete $\{z \in \mathbb{C} \mid |R(z)| \leq 1\}$*:*



Die Stabilitätsgebiete der expliziten Verfahren sind links zu erkennen, die der impliziten Verfahren sind durch die ganze Ebene mit Ausnahme der konvexen Bereiche rechts von der y-Achse gegeben.

Das Beispiel läßt vermuten, daß die Stabilitätsfunktionen expliziter Runge-Kutta-Verfahren Polynome sind, während die impliziter Runge-Kutta-Verfahren rationale Funktionen sind.

Satz 4.1.2

Ein s-stufiges Runge-Kutta-Verfahren mit den Daten $c, b \in \mathbb{R}^s$, $A \in \mathbb{R}^{s\times s}$ besitzt die Stabilitätsfunktion

$$R(z) = 1 + zb^\top (I - zA)^{-1} e, \qquad e = (1, \dots, 1)^\top \in \mathbb{R}^s.$$

Für explizite Runge-Kutta-Verfahren ist R ein Polynom vom Höchstgrad s.
Für implizite Runge-Kutta-Verfahren ist R eine rationale Funktion, deren Zähler- und Nennerpolynom Höchstgrad s besitzen. Insbesondere ist $I - zA$ genau dann invertierbar, wenn $1/z$ kein Eigenwert von A ist.

Beweis:

(i) Für das Runge-Kutta-Verfahren angewendet auf die Testgleichung ergibt sich

$$k_j = \lambda(x_i + h \sum_{\ell=1}^{s} a_{j\ell} k_\ell)$$

bzw.

$$k_j - z \sum_{\ell=1}^{s} a_{j\ell} k_\ell = \lambda x_i$$

für $j = 1, \dots, s$ und $z = \lambda h$. In Matrixschreibweise lautet dies

$$k - zAk = (I - zA)k = \lambda x_i e$$

wobei $k = (k_1, \dots, k_s)^\top$ und $e = (1, \dots, 1)^\top \mathbb{R}^s$ seien. Die Matrix $I - zA$ ist invertierbar, falls $1/z$ kein Eigenwert von A ist, und es folgt

$$k = \lambda x_i (I - zA)^{-1} e.$$

Einsetzen in die Inkrementfunktion des Runge-Kutta-Verfahrens liefert

$$x_{i+1} = x_i + h \sum_{j=1}^{s} b_j k_j = x_i + hb^\top k = (1 + zb^\top (I - zA)^{-1} e) x_i = R(z) x_i.$$

(ii) Für explizite Runge-Kutta-Verfahren ist A eine linke untere Dreiecksmatrix und $I - zA$ ist invertierbar für alle z. Desweiteren gilt $A^s = 0$ und mit Hilfe der Neumannschen Reihe folgt

$$(I - zA)^{-1} = I + zA + (zA)^2 + \dots + (zA)^{s-1}.$$

Damit ist R ein Polynom vom Höchstgrad s.

(iii) Betrachte das lineare Gleichungssystem

$$(I - zA)w = e.$$

Die Cramersche Regel liefert

$$w_i = \frac{\det(I - zA)_i}{\det(I - zA)}, \qquad i = 1, \dots, s,$$

wobei $(I - zA)_i$ die Matrix $I - zA$ bezeichnet in der die i-te Spalte durch e ersetzt wird. Die Determinanten sind jeweils Polynome vom Höchstgrad s. Wegen

$$\det(I - zA) = \det\left(z(\frac{1}{z}I - A)\right) = z^s \det(\frac{1}{z}I - A)$$

ist $I - zA$ genau dann singulär, wenn $1/z$ Eigenwert von A ist.

□

Wir bezeichnen nun Einschrittverfahren als A-stabil (oder absolut stabil), wenn die Bedingung $|R(z)| \leq 1$ mit $z = \lambda h$ für $Re(\lambda) \leq 0$ zu **keiner Einschränkung** der Schrittweite $h > 0$ führt. A-stabile Verfahren sind daher für steife Differentialgleichungen geeignet, während nicht A-stabile Verfahren für steife Differentialgleichungen ineffizient sind.

Definition 4.1.3 **(A-Stabilität)**

Sei R die Stabilitätsfunktion eines Einschrittverfahrens zur Lösung der Testgleichung (4.3). Die Menge

$$S := \{z \in \mathbb{C} \mid |R(z)| \leq 1\}$$

heißt **Stabilitätsgebiet des Einschrittverfahrens**. *Das Einschrittverfahren heißt* **A-stabil** *(oder* **absolut-stabil***), wenn*

$$\mathbb{C}_- \subseteq S,$$

wobei $\mathbb{C}_- := \{z \in \mathbb{C} \mid Re(z) \leq 0\}$ die linke Hälfte der komplexen Ebene bezeichnet.

A-Stabilität bedeutet also, daß $|R(z)| \leq 1$ für alle $z \in \mathbb{C}$ mit $Re(z) \leq 0$ gilt. Das implizite Eulerverfahren und das Radau-IIA-Verfahren sind A-stabil. Da die Stabilitätsfunktionen expliziter Runge-Kutta-Verfahren nach Satz 4.1.2 Polynome sind und daher $\lim_{|z|\to\infty} |R(z)| = \infty$ gilt, sind die zugehörigen Stabilitätsgebiete expliziter Runge-Kutta-Verfahren stets beschränkt. Somit sind explizite Runge-Kutta-Verfahren niemals A-stabil.

Bemerkung 4.1.4

Die Problematik der Steifheit läßt sich analog auch lineare Mehrschrittverfahren untersuchen, wobei die Analyse komplizierter ist. Dabei stellt sich heraus, daß explizite lineare

Mehrschrittverfahren nicht A-stabil sind. Darüber hinaus gibt es auch kein A-stabiles implizites lineares Mehrschrittverfahren mit Konsistenzordnung $p > 2$. Die implizite Trapezregel

$$x_{i+1} = x_i + \frac{1}{h}\left(f_{i+1} + f_i\right)$$

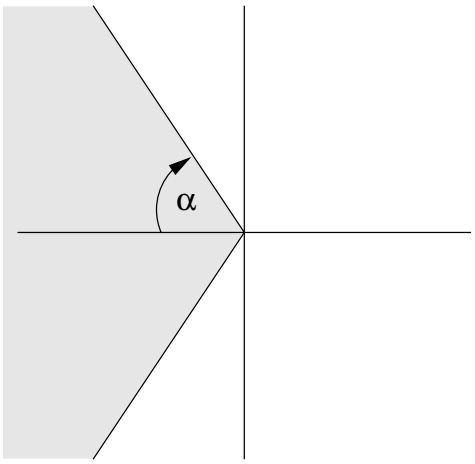
ist A-stabil und besitzt Konsistenzordnung $p = 2$, besitzt also die höchstmögliche Ordnung unter den A-stabilen lineare Mehrschrittverfahren.

Bemerkung 4.1.5

BDF-Verfahren sind nicht A-stabil, aber immerhin nullstabil für $k \leq 6$. Es stellt sich jedoch heraus, daß BDF-Verfahren eine Abschwächung der A-Stabilität, die sogenannte $A(\alpha)$-Stabilität, erfüllen. Ein Verfahren heißt hierbei $A(\alpha)$-stabil, wenn für das Stabilitätsgebiet S des Verfahrens die Bedingung

$$\{z \in \mathbb{C} \mid |arg(z) - \pi| \leq \alpha\} \subseteq S$$

für ein $\alpha \in [0, \pi/2]$ erfüllt. Hierbei ist $z = |z| \exp(iarg(z))$ mit $arg(z) \in [0, 2\pi)$. Die Menge auf der linken Seite sieht wie folgt aus:



Für BDF-Verfahren ergeben sich folgende Winkel:

k	*1*	*2*	*3*	*4*	*5*	*6*
α	90°	90°	86°	73.3°	51.8°	17.8°

Kapitel 5

Randwertprobleme

Wir betrachten Randwertprobleme der folgenden Form.

Problem 5.0.1 **(Zweipunkt-Randwertproblem)**
Seien $f : [a,b] \times \mathbb{R}^n \to \mathbb{R}^n$ und $r : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ gegeben. Gesucht ist eine Lösung x des Randwertproblems

$$\begin{aligned} x'(t) &= f(t, x(t)), \\ r(x(a), x(b)) &= 0 \end{aligned}$$

im Intervall $[a,b]$.

Derartige Randwertprobleme treten z.B. bei der Lösung von Variationsproblemen oder Optimalsteuerungsproblemen auf.

Bemerkung 5.0.2

- *Existenz- und Eindeutigkeitsresultate für nichtlineare Randwertprobleme finden sich in Ascher et al. [AMR95] in Kapitel 3. Ein Randwertproblem kann keine, genau eine oder unendlich viele Lösungen besitzen.*
- *Es kann passieren, daß die Lösung eines Randwertproblems auch für weniger als n Randwerte bereits eindeutig bestimmt ist. Betrachte z.B. das Randwertproblem*

$$x'(t) = f(t, x(t)), \qquad r(x(0)) = x_1(0)^2 + \ldots + x_n(0)^2 = 0.$$

Diese Randbedingung legt alle Randwerte eindeutig fest: $x_i(0) = 0$, $i = 1, \ldots, n$.

5.1 Sensitivitätsanalyse

Wir untersuchen die Abhängigkeit der Lösung einer Anfangswertaufgabe von Parametern.

Problem 5.1.1 **(Parameterabhängiges Anfangswertproblem)**
Für gegebene Funktionen $f : [a,b] \times \mathbb{R}^n \times \mathbb{R}^{n_p} \to \mathbb{R}^n$ und $x_a : \mathbb{R}^{n_p} \to \mathbb{R}^n$ und gegebenem

Parameter $p \in \mathbb{R}^{n_p}$ ist eine Lösung des Anfangswertproblems

$$x'(t) = f(t, x(t), p), \qquad x(a) = x_a(p). \tag{5.1}$$

gesucht.

Um anzudeuten, daß die Lösung des Anfangswertproblems 5.1.1 von p abhängt, schreiben wir auch $x(t; p)$, $t \in [a, b]$. Wir interessieren uns hier für die folgenden Fragestellungen:

- Unter welchen Bedingungen hängt die Lösung des Anfangswertproblems stetig oder sogar stetig differenzierbar vom Parameter p ab?
- Wie können die **Sensitivitäten** $S(t) := \frac{\partial x(t;p)}{\partial p}$ berechnet werden?

Wir benötigen zunächst noch ein Hilfsresultat.

Lemma 5.1.2 **(Gronwall)**
Für stetige Funktionen $u, z : [a, b] \to \mathbb{R}$ gelte

$$u(t) \le c + L \int_a^t u(\tau) d\tau + z(t)$$

für alle $t \in [a, b]$ mit Konstanten $c, L \ge 0$. Dann gilt

$$u(t) \le \left(c + \max_{a \le \tau \le t} |z(\tau)| \right) \exp(L(t - a))$$

für alle $t \in [a, b]$.

Beweis: Definiere

$$v(t) := \int_a^t u(\tau) d\tau.$$

Dann gilt $v(a) = 0$ und

$$v'(t) = u(t) \le c + L \int_a^t u(\tau) d\tau + z(t) = c + Lv(t) + z(t).$$

Desweiteren gilt

$$\frac{d}{dt} \left(v(t) \exp(-L(t - a)) \right) = (v'(t) - Lv(t)) \exp(-L(t - a)) \le (c + z(t)) \exp(-L(t - a)).$$

Integration liefert

$$\begin{aligned} v(t) \exp(-L(t - a)) &\le \int_a^t (c + z(\tau)) \exp(-L(\tau - a)) d\tau \\ &\le \left(c + \max_{a \le \tau \le t} |z(\tau)| \right) \frac{1 - \exp(-L(t - a))}{L}. \end{aligned}$$

Wir erhalten

$$v(t) \le \left(c + \max_{a\le\tau\le t} |z(\tau)|\right) \frac{\exp(L(t-a)) - 1}{L}.$$

Mit dieser Abschätzung erhalten wir

$$\begin{aligned} u(t) &\le c + Lv(t) + z(t) \\ &\le c + \max_{a\le\tau\le t} |z(\tau)| + \left(c + \max_{a\le\tau\le t} |z(\tau)|\right) (\exp(L(t-a)) - 1) \\ &= \left(c + \max_{a\le\tau\le t} |z(\tau)|\right) \exp(L(t-a)). \end{aligned}$$

□

Wir untersuchen nun die stetige Abhängigkeit der Lösung vom Parameter p und schreiben die Lösung x zum Parameter p als

$$x(t;p) = x_0(p) + \int_a^t f(t, x(t), p)dt.$$

Es gelte die Lipschitzbedingung

$$\|f(t, x_1, p_1) - f(t, x_2, p_2)\| \le L\left(\|x_1 - x_2\| + \|p_1 - p_2\|\right) \tag{5.2}$$

für alle $t \in [a, b], x_1, x_2 \in \mathbb{R}^n, p_1, p_2 \in \mathbb{R}^{n_p}$. Damit ist insbesondere auch die globale Existenz von Lösungen in $[a, b]$ für alle Parameter $p \in \mathbb{R}^{n_p}$ gesichert.
Seien nun zwei Parameter p_1 und p_2 und zugehörige Lösungen $x(t; p_1)$ und $x(t; p_2)$ gegeben. Mit Hilfe des Lemmas von Gronwall folgt für alle $t \in [a, b]$

$$\begin{aligned} \|x(t;p_1) - x(t;p_2)\| &\le \|x_a(p_1) - x_a(p_2)\| + \int_a^t \|f(\tau, x(\tau;p_1), p_1) - f(\tau, x(\tau;p_2), p_2)\|d\tau \\ &\le \|x_a(p_1) - x_a(p_2)\| + L\int_a^t \|x(\tau;p_1) - x(\tau;p_2)\|d\tau \\ &\qquad + L(t-a)\|p_1 - p_2\| \\ &\le \left(\|x_a(p_1) - x_a(p_2)\| + L(t-a)\|p_1 - p_2\|\right) \exp(L(t-a)). \end{aligned} \tag{5.3}$$

Diese Abschätzung beweist

Satz 5.1.3 (Stetige Abhängigkeit von Parametern)

Es gelte (5.2) für Problem 5.1.1 und die Funktion $x_a : \mathbb{R}^{n_p} \to \mathbb{R}^n$ sei stetig. Dann hängt die Lösung $x(t; p)$ für alle $t \in [a, b]$ stetig von p ab und es gilt

$$\lim_{p\to\hat{p}} x(t;p) = x(t;\hat{p}) \qquad \forall t \in [a, b],\ \hat{p} \in \mathbb{R}^{n_p}.$$

Ist x_a sogar lipschitzstetig, so gibt es eine Konstante S mit

$$\|x(t;p_1) - x(t;p_2)\| \le S\|p_1 - p_2\| \qquad \forall t \in [a, b],\ p_1, p_2 \in \mathbb{R}^{n_p}.$$

Beweis: Die erste Aussage folgt direkt aus (5.3). Die Lipschitzstetigkeit folgt mit $S := (L_x + L(b-a))\exp(L(b-a))$, wobei L_x die Lipschitzkonstante von x_a bezeichnet. □

Abhängigkeit vom Anfangswert

Die Abhängigkeit der Lösung vom Anfangswert x_a ergibt sich als Spezialfall mit $x_a(p) = p$ und $f = f(t,x)$. Es gilt dann

$$\|x(t;p_1) - x(t;p_2)\| \le \|p_1 - p_2\| \exp(L(t-a)).$$

Berechnung von Sensitivitäten

Wir untersuchen nun, wie die Sensitivitäten $S(t) = \frac{\partial x(t;p)}{\partial p} \in \mathbb{R}^{n\times n_p}$ berechnet werden können. Wir stellen fest, daß das parameterabhängige Anfangswertproblem (5.1) eine Identität in p darstellt. Totale Differentiation des Problems nach p liefert formal

$$\begin{aligned} \frac{\partial x(a;p)}{\partial p} &= x_a'(p), \\ \frac{\partial}{\partial p}\left(\frac{d}{dt}x(t;p)\right) &= f_x'(t,x(t;p),p)\cdot\frac{\partial x(t;p)}{\partial p} + f_p'(t,x(t;p),p). \end{aligned}$$

Unter der Annahme, daß

$$\frac{\partial}{\partial p}\left(\frac{d}{dt}x(t;p)\right) = \frac{d}{dt}\left(\frac{\partial}{\partial p}x(t;p)\right)$$

gilt, erfüllt die Sensitivitätsmatrix $S(t) = \frac{\partial x(t;p)}{\partial p}$ in $[a,b]$ das folgende Matrix-Anfangswertproblem:

Sensitivitäts-Differentialgleichung

$$\begin{aligned} S(a) &= x_a'(p), \\ S'(t) &= f_x'(t,x(t;p),p)\cdot S(t) + f_p'(t,x(t;p),p). \end{aligned}$$

Fazit: Durch gleichzeitiges numerisches Lösen des Anfangswertproblems (5.1) und der Sensitivitäts-Differentialgleichung erhalten wir die Ableitungen $\frac{\partial x(t;p)}{\partial p}$. Ist man nur an der Abhängigkeit vom Anfangswert interessiert, so gilt $x_a(p) = p$ und folglich $x_a'(p) = I$. Demailly [Dem91] zeigt in Abschnitt 11.1.3, daß die hier motivierte Vorgehensweise tatsächlich gerechtfertigt ist, wenn f und x_a stetig sind und stetige partielle Ableitungen bzgl. x und p besitzen.

5.2 Schießverfahren

Wir betrachten das Zweipunkt-Randwertproblem 5.0.1. Die Idee des Einfachschießverfahrens (engl. single shooting) basiert auf dem in Abbildung 5.1 dargestellten Ansatz.

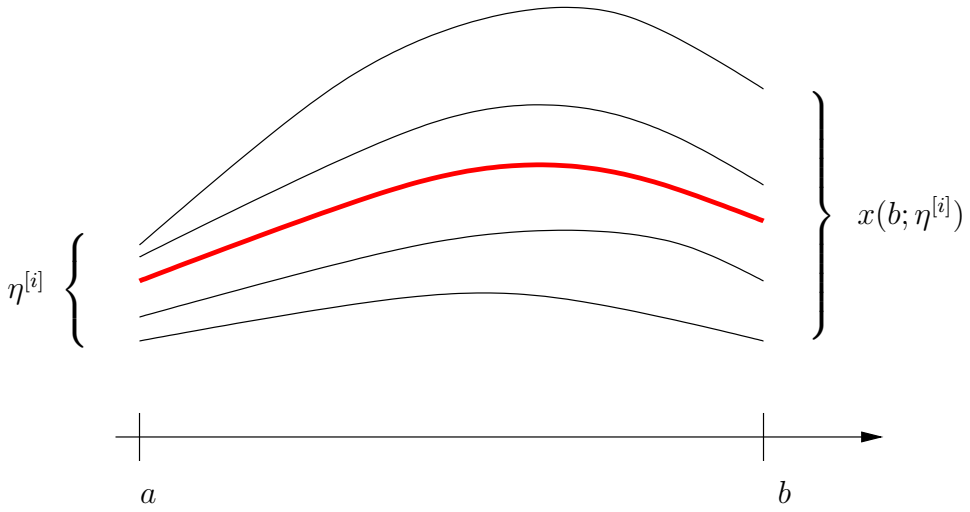


Abbildung 5.1: Idee des Einfachschießverfahrens (single shooting): Iterative Lösung von Anfangswertproblemen mit variierendem Anfangswert. Die Lösung des Randwertproblems ist rot eingezeichnet.

Für eine gegebene Startschätzung η des Anfangswerts $x(a)$ besitze das Anfangswertproblem

$$x'(t) = f(t, x(t)), \qquad x(a) = \eta$$

die Lösung $x(t;\eta)$ auf $[a,b]$. Damit $x(t;\eta)$ auch die Randbedingung erfüllt, muß

$$F(\eta) := r(x(a;\eta), x(b;\eta)) = r(\eta, x(b;\eta)) = 0 \tag{5.4}$$

gelten. Gleichung (5.4) ist also ein **nichtlineares Gleichungssystem** für die Funktion F. Anwendung des Newtonverfahrens führt auf das sogenannte Einfachschießverfahren:

Algorithmus: Einfachschießverfahren

(0) Wähle Startschätzung $\eta^{[0]} \in \mathbb{R}^n$ und setze $i = 0$.

(1) Löse das Anfangswertproblem

$$x'(t) = f(t, x(t)), \qquad x(a) = \eta^{[i]}, \qquad a \le t \le b,$$

zur Berechnung von $F(\eta^{[i]})$ und berechne die Jacobimatrix

$$F'(\eta^{[i]}) = r'_{x_a}(\eta^{[i]}, x(b; \eta^{[i]})) + r'_{x_b}(\eta^{[i]}, x(b; \eta^{[i]})) \cdot S(b),$$

wobei S Lösung der Sensitivitäts-Differentialgleichung

$$S'(t) = f'_x(t, x(t; \eta^{[i]})) \cdot S(t), \qquad S(a) = I$$

ist.

(2) Ist $F(\eta^{[i]}) = 0$ (oder ist ein anderes Abbruchkriterium) erfüllt, STOP.

(3) Berechne die Newton-Richtung $d^{[i]}$ als Lösung des linearen Gleichungssystems

$$F'(\eta^{[i]})d = -F(\eta^{[i]}).$$

(4) Setze $\eta^{[i+1]} = \eta^{[i]} + d^{[i]}$ und $i = i + 1$ und gehe zu (1).

Bemerkung 5.2.1

Die Ableitung $F'(\eta^{[i]})$ in Schritt (2) des Einfachschießverfahrens kann alternativ durch finite Differenzen approximiert werden:

$$\frac{\partial}{\partial \eta_j} F(\eta) \approx \frac{F(\eta + he_j) - F(\eta)}{h}, \qquad j = 1, \dots, n,$$

$e_j = j-ter$ Einheitsvektor. Dieser Ansatz erfordert das Lösen von n Anfangswertproblemen!

Da das Einfachschießverfahren im Wesentlichen ein Newtonverfahren ist, gelten unter entsprechenden Voraussetzungen auch alle Konvergenzaussagen für Newtonverfahren und man kann mit lokal superlinearer bzw. sogar lokal quadratischer Konvergenz rechnen. Die Jacobimatrix $F'(\eta^{[i]})$ in Schritt (2) ist invertierbar, wenn die Matrix

$$r'_{x_a}(\eta^{[i]}, x(b; \eta^{[i]})) \cdot S(a) + r'_{x_b}(\eta^{[i]}, x(b; \eta^{[i]})) \cdot S(b)$$

invertierbar ist.

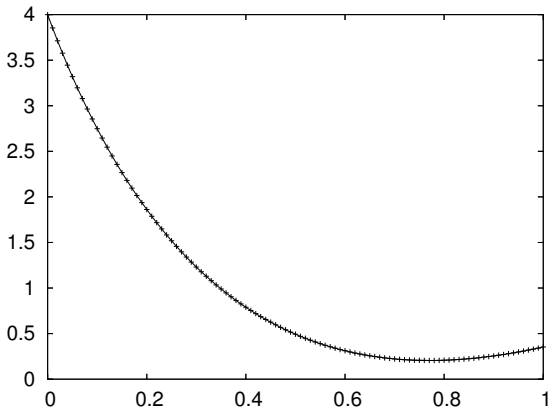
Beispiel 5.2.2

Betrachte das Randwertproblem

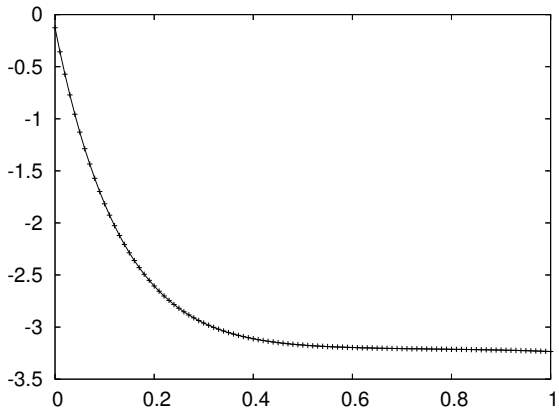
$$\begin{aligned} x_1'(t) &= -x_2(t) - r(t), \\ x_2'(t) &= -\frac{3}{2}x_1(t)^2, \\ x(0) - 4 &= 0, \\ x_2(1) - 5(x_1(1) - 1) &= 0, \end{aligned}$$

mit $r(t) = 15\exp(-2t)$. *Wir lösen das Randwertproblem mit dem Einfachschießverfahren mit Startwert* $\eta^{[0]} = (4, -5)^\top$ *und erhalten folgende Lösung:*

Zustand x_1*:*



Zustand x_2*:*




```
ITER      L2 NORM OF RESIDUALS
-------------------------------
 0        0.1365880904004427E+03
 1        0.4463547386262193E+02
 2        0.2131159392149783E+02
 3        0.7013694175663481E+01
 4        0.1805812047209367E+01
 5        0.2290634278529377E+00
 6        0.9082950012230806E-02
 7        0.4847021565884679E-04
 8        0.1033707347497526E-07
 9        0.9420242363944453E-13
10        0.8326672684688674E-14

     FINAL L2 NORM OF THE RESIDUALS        0.8326672684688674E-14
     EXIT PARAMETER                         1
     FINAL APPROXIMATE SOLUTION
            0.4000000000000000E+01       -0.1260067289470882E+00
```

Ein Problem des Einfachschießverfahrens ist mitunter die Durchführbarkeit bzw. die Stabilität des Verfahrens. Wie wir im Abschnitt über die Abhängigkeit eines Anfangswertproblems von Parametern gesehen haben, gilt die Abschätzung

$$\|x(t;\eta_1) - x(t;\eta_2)\| \le \|\eta_1 - \eta_2\| \exp(L(t-a)).$$

Für große Lipschitzkonstanten L und große Intervalllängen $b - a$ können Lösungen für

unterschiedliche Startwerte drastisch voneinander abweichen. Im Extremfall gelingt es mitunter nicht, das Anfangswertproblem zum Startwert $\eta^{[i]}$ auf dem ganzen Intervall $[a,b]$ zu lösen. Diesen Umstand vermeidet das sogenannte **Mehrfachschießverfahren** (engl. multiple shooting) durch Einführung zusätzlicher **Mehrzielknoten**

$$a = t_0 < t_1 < \cdots < t_{N-1} < t_N = b.$$

Anzahl und Lage der benötigten Mehrzielknoten hängen vom konkreten Problem ab.

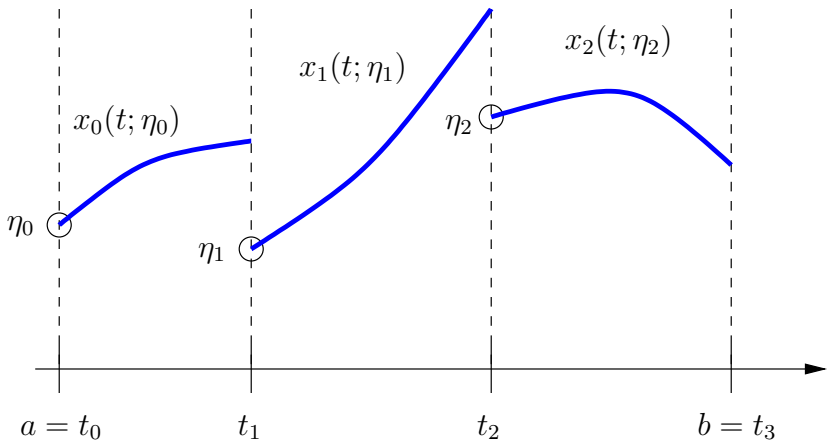


Abbildung 5.2: Idee des Mehrfachschießverfahrens (multiple shooting): Iterative Lösung von Anfangswertproblemen auf Teilintervallen.

Wie in Abbildung 5.2 angedeutet, wird in jedem Teilintervall $[t_j, t_{j+1})$, $j = 0, \dots, N-1$ ausgehend von dem Startwert η_j das Anfangswertproblem

$$x'(t) = f(t, x(t)), \qquad x(t_j) = \eta_j$$

gelöst und man erhält in $[t_j, t_{j+1})$ die Lösung $x_j(t;\eta_j)$. Damit die zusammengesetzte Funktion

$$x(t;\eta_0,\dots,\eta_{N-1}) := \begin{cases} x_j(t;\eta_j), & \text{falls } t_j \le t < t_{j+1},\ j = 0,\dots,N-1, \\ x_{N-1}(t_N;\eta_{N-1}), & \text{falls } t = b \end{cases}$$

die Randbedingungen erfüllt und eine Lösung des Randwertproblems ist, müssen die folgenden Stetigkeits- und Randbedingungen erfüllt sein:

$$F(\eta) := F(\eta_0,\dots,\eta_{N-1}) := \begin{pmatrix} x_0(t_1;\eta_0) - \eta_1 \\ x_1(t_2;\eta_1) - \eta_2 \\ \vdots \\ x_{N-2}(t_{N-1};\eta_{N-2}) - \eta_{N-1} \\ r(\eta_0, x_{N-1}(t_N;\eta_{N-1})) \end{pmatrix} = 0. \tag{5.5}$$

Gleichung (5.5) ist also wieder ein **nichtlineares Gleichungssystem** für die Funktion F in den Variablen $\eta := (\eta_0, \dots, \eta_{N-1})^\top \in \mathbb{R}^{N \cdot n}$.
Als Spezialfall des Mehrfachschießverfahrens entsteht für $N = 1$ das Einfachschießverfahren. Abhängig von der Anzahl der Mehrzielknoten, wächst die Dimension des nichtlinearen Gleichungssystems (5.5) an. Allerdings besitzt die Jacobimatrix $F'(\eta)$ eine dünn besetzte Struktur, die bei der Lösung von linearen Gleichungssystemen ausgenutzt werden kann:

$$F'(\eta) = \begin{pmatrix} S_0 & -I & & & \\ & S_1 & -I & & \\ & & \ddots & \ddots & \\ & & & S_{N-2} & -I \\ A & & & & B \cdot S_{N-1} \end{pmatrix} \tag{5.6}$$

mit

$$S_j := \frac{\partial}{\partial \eta_j} x_j(t_{j+1}; \eta_j), \qquad A := r'_{x_a}(\eta_0, x_{N-1}(t_N; \eta_{N-1})), \qquad B := r'_{x_b}(\eta_0, x_{N-1}(t_N; \eta_{N-1})).$$

Anwendung des Newtonverfahrens auf das nichtlineare Gleichungssystem (5.5) liefert den folgenden Algorithmus.

Algorithmus: Mehrfachschießverfahren

(0) Wähle Startschätzung $\eta^{[0]} = (\eta_0^{[0]}, \dots, \eta_{N-1}^{[0]})^\top \in \mathbb{R}^{N \cdot n}$ und setze $i = 0$.

(1) Für $j = 0, \dots, N-1$ löse die Anfangswertprobleme

$$x'(t) = f(t, x(t)), \qquad x(t_j) = \eta_j^{[i]}, \qquad t_j \le t \le t_{j+1},$$

zur Berechnung von $F(\eta^{[i]})$ und berechne die Sensitivitätsmatrizen $S_j = S(t_{j+1})$, wobei S Lösung der Sensitivitäts-Differentialgleichung

$$S'(t) = f'_x(t, x(t; \eta_j^{[i]})) \cdot S(t), \qquad S(t_j) = I, \quad t_j \le t \le t_{j+1}$$

ist. Berechne damit $F'(\eta^{[i]})$ gemäß (5.6).

(2) Ist $F(\eta^{[i]}) = 0$ (oder ist ein anderes Abbruchkriterium) erfüllt, STOP.

(3) Berechne die Newton-Richtung $d^{[i]}$ als Lösung des linearen Gleichungssystems

$$F'(\eta^{[i]}) d = -F(\eta^{[i]}).$$

(4) Setze $\eta^{[i+1]} = \eta^{[i]} + d^{[i]}$ und $i = i + 1$ und gehe zu (1).

Detailliertere Darstellungen von Mehrfachschießverfahren erfolgen in Stoer und Bulirsch [SB90], Abschnitt 7.3.5 und in Ascher et al. [AMR95], Kapitel 4.

Bemerkung 5.2.3

- *Zur Globalisierung des lokalen Newtonverfahrens wird das gedämpfte Newtonverfahren verwendet, bei dem sich die neue Iterierte unter Verwendung einer Schrittweite α_i berechnet zu*

$$x^{[i+1]} = x^{[i]} + \alpha_i d^{[i]}, \qquad i = 0, 1, 2, \ldots.$$

  Die Schrittweite α_i kann dabei durch eindimensionale Minimierung der Funktion

$$\varphi(\alpha) := \frac{1}{2}\|F(x^{[i]} + \alpha d^{[i]})\|_2^2$$

  (z.B. mit dem Armijo-Verfahren) ermittelt werden.

- *Anstatt mit der exakten Jacobimatrix $F'(\eta^{[i]})$ zu rechnen (die Berechnung derselben ist sehr teuer, da sie die Lösung der Sensitivitäts-Differentialgleichung erfordert!), kann sie analog zu den Quasi-Newton-Verfahren in der Optimierung durch Update-Formeln ersetzt werden, etwa durch die Rang-1-Update Formel von Broyden:*

$$J_+ = J + \frac{(z - Jd)d^\top}{d^\top d}, \qquad d = \eta^+ - \eta, \quad z = F(\eta^+) - F(\eta).$$

- *Ein wesentliches und nicht zu unterschätzendes Problem bei der numerischen Lösung von Randwertproblemen, die etwa im Rahmen von Optimalsteuerungsprobleme entstehen, ist die Beschaffung einer guten Startschärtzung für die Lösung. Hier gibt es leider kein Patentrezept. Mögliche Ansätze sind Homotopieverfahren (es werden benachbarte, einfach zu lösende Probleme gelöst und deren Lösung wird als Startschätzung verwendet).*

5.3 Kollokationsverfahren

Kollokationsverfahren basieren auf einer Approximation durch stückweise definierte Polynome. Wir motivieren das Verfahren für das Zweipunkt-Randwertproblem 5.0.1. Das Intervall $[a, b]$ wird in Teilintervalle $[t_i, t_{i+1}]$, $i = 0, \ldots, N-1$ mit Maschenweite h unterteilt. Es seien nun $0 \le \rho_1 < \ldots < \rho_k \le 1$ feste Werte. In jedem Teilintervall $[t_i, t_{i+1}]$ sind durch ρ_j, $j = 1, \ldots, k$ sogenannte **Kollokationsstellen**

$$t_{ij} = t_i + \rho_j(t_{i+1} - t_i), \qquad j = 1, \ldots, k$$

festgelegt. Es wird nun eine sogenannte **Kollokationslösung** x_h auf $[a, b]$ gesucht. x_h heißt Kollokationslösung, wenn folgende Bedingungen erfüllt sind:

- x_h ist stetig auf $[a,b]$;
- In jedem Intervall $[t_i, t_{i+1}]$, $i = 0, \dots, N-1$ ist $x_h|_{[t_i,t_{i+1}]}$ ein Polynom vom Grad k;
- x_h erfüllt die **Kollokationsbedingungen**

$$x_h'(t_{ij}) = f(t_{ij}, x_h(t_{ij})), \qquad i = 0, \dots, N-1,\ j = 1, \dots, k$$

  und die Randbedingung

$$r(x_h(a), x_h(b)) = 0.$$

Zur Bestimmung einer Kollokationslösung x_h muß ein i.a. nichtlineares Gleichungssystem der Dimension $N(k+1)n$ gelöst werden, welches aus den $N \cdot k \cdot n$ Kollokationsbedingungen, den n Randbedingungen und $(N-1)n$ Stetigkeitsbedingungen in t_i, $i = 1, \dots, N-1$ besteht. Es kann gezeigt werden, daß eine Äquivalenz zwischen Kollokationsverfahren und bestimmten Runge-Kutta-Verfahren besteht, vgl. Ascher et al. [AMR95], Theorem 5.73, S. 219.

5.4 Weitere Verfahren

Für bestimmte Klassen von Problemen eignen sich Finite-Element-Verfahren zur Lösung von Randwertproblemen, welche sich insbesondere bei der Lösung von partiellen Differentialgleichungen sehr grosser Beliebtheit erfreuen.
Neben Finite-Element-Verfahren werden häufig auch Finite-Differenzen-Verfahren zur Diskretisierung von Randwertproblemen verwendet.

Kapitel A

Software

Available software in the world wide web:

- Scilab: A Free Scientific Software Package; http://www.scilab.org
- GNU Octave: A high-level language, primarily intended for numerical computations; http://www.octave.org/octave.html
- GAMS: Guide to Available Mathematical Software; http://gams.nist.gov/
- Netlib: collection of mathematical software, papers, and databases; http://www.netlib.org/
- NEOS Guide: www-fp.mcs.anl.gov/otc/Guide

Empfohlene Literatur für Numerische Mathematik II:

- Deuflhard, P. und Bornemann, F. *Numerische Mathematik II*. de Gruyter, Berlin, 1994.
- Stoer, J. und Bulirsch, R. *Numerische Mathematik 2*. Springer, Berlin-Heidelberg-New York, 1990, 3. Auflage.
- Kincaid, D. und Cheney, W. *Numerical Analysis: Mathematics of Scientific Computing*. Brooks/Cole Series in Advanced Mathematics, Thomson Learning, 3rd Edition, 2002.
- Lempio, F. *Numerische Mathematik I und II*. Bayreuther Mathematische Schriften, 51 und 55, 1997 und 1998.
- Strehmel, K. und Weiner, R. *Numerik gewöhnlicher Differentialgleichungen*, Teubner, Stuttgart, 1995
- Hairer, E., Norsett, S. P. und Wanner, G. *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer Series in Computational Mathematics, Vol. 8, Berlin-Heidelberg-New York, 2nd edition, 1993.
- Hairer, E. und Wanner, G. *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, Springer Series in Computational Mathematics, Vol. 14, Berlin-Heidelberg-New York, 2nd edition, 1996
- Ascher, U. M., Mattheij, R. M. M. und Russell, R. D. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, SIAM, Philadelphia, Classics In Applied Mathematics, Vol. 13, 1995.

Literaturverzeichnis

[AMR95] Ascher, U. M., Mattheij, R. M. and Russell, R. D. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, volume 13 of *Classics In Applied Mathematics*. SIAM, Philadelphia, 1995.

[CH52] Curtiss, C. F. and Hirschfelder, J. O. *Integration of stiff equations*. Proceedings of the National Academy of Sciences of the United States of America, 38; 235–243, 1952.

[DB94] Deuflhard, P. and Bornemann, F. *Numerische Mathematik II*. de Gruyter, Berlin, 1994.

[DB02] Deuflhard, P. and Bornemann, F. *Scientific Computing with Ordinary Differential Equations*. volume 42 of *Texts in Applied Mathematics*. Springer-Verlag New York, New York, 2002.

[Dem91] Demailly, J.-P. *Gewöhnliche Differentialgleichungen*. Vieweg, Braunschweig, 1991.

[Gea71] Gear, C. W. *Simultaneous Numerical Solution of Differential-Algebraic Equations*. IEEE Transactions on Circuit Theory, 18 (1); 89–95, 1971.

[Ger05] Gerdts, M. *Solving Mixed-Integer Optimal Control Problems by Branch&Bound: A Case Study from Automobile Test-Driving with Gear Shift*. Optimal Control, Applications and Methods, 26 (1); 1–18, 2005.

[HNW93] Hairer, E., Norsett, S. P. and Wanner, G. *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 8. Springer Series in Computational Mathematics, Berlin-Heidelberg-New York, 2nd edition, 1993.

[HW96] Hairer, E. and Wanner, G. *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, volume 14. Springer Series in Computational Mathematics, Berlin-Heidelberg-New York, 2nd edition, 1996.

[Kan08] Kanzow, C. *Numerische Mathematik II*. Vorlesungsskript, Institut für Mathematik, Universität Würzburg, 2008.

[KC02] Kincaid, D. and Cheney, W. *Numerical Analysis: Mathematics of Scientific Computing*. Brooks/Cole–Thomson Learning, Pacific Grove, CA, 3rd edition, 2002.

[Lem97] Lempio, F. *Numerische Mathematik I – Methoden der linearen Algebra*. volume 51 of *Bayreuther Mathematische Schriften*. Bayreuth, 1997.

[Lem98] Lempio, F. *Numerische Mathematik II – Methoden der Analysis*. volume 55 of *Bayreuther Mathematische Schriften*. Bayreuth, 1998.

[Mur93] Murray, J. D. *Mathematical Biology*. Biomathematics Texts. Springer-Verlag, 2nd edition, 1993.

[PB93] Pacejka, H. and Bakker, E. *The Magic Formula Tyre Model*. Vehicle System Dynamics, 21 supplement; 1–18, 1993.

[SB90] Stoer, J. and Bulirsch, R. *Numerische Mathematik II*. Springer, Berlin-Heidelberg-New York, 3rd edition, 1990.

[Ste73] Stetter, H. J. *Analysis of Discretization Methods for Ordinary Differential Equations*. volume 23 of *Springer Tracts in Natural Philosophy*. Springer-Verlag Berlin Heidelberg New York, 1973.

[SW95] Strehmel, K. and Weiner, R. *Numerik gewöhnlicher Differentialgleichungen*. Teubner, Stuttgart, 1995.

[Wal90] Walter, W. *Gewöhnliche Differentialgleichungen*. Springer, Berlin-Heidelberg-New York, 4th edition, 1990.

[Wlo71] Wloka, J. *Funktionalanalysis und Anwendungen*. de Gruyter, Berlin, 1971.