Numerik II - Mitschrift

Stephan Weller

15. Juli 2005

Inhaltsverzeichnis

1. Klassische Iterationsverfahren

1.1. Einführung

Beispiel: Stationäre Wärmeleitung, zunächst eindimensional.
Eine Funktion u soll eine Temperaturverteilung z.B. in einem Stab beschreiben. Es ist also gesucht:

Beispiel 1.1

$$\begin{aligned} u : [0,1] &\to \mathbb{R} \\ \text{mit } -u''(x) &= f(x) \forall x \in]0,1[\\ \text{und } u(0) &= u(1) = 0 \end{aligned}$$

Diskretisierung von 1.1: Wähle ein Gitter $\Sigma = \{x = ih, i = 0, \ldots, n, h = \frac{1}{n}\}$ und bestimme eine Näherung von $u(x)$ durch $u_i \approx u(x_i)$.

Definition: Vorwärts- und Rückwärsdifferenzenquotient

$$\partial_h^+ v_i := \frac{v_{i+1} - v_i}{h}$$
$$\partial_h^- v_i := \frac{v_i - v_{i-1}}{h}$$

mit $v : \sigma \to \mathbb{R}$ und $i = 1, \ldots, n$.
Nun ist

$$\begin{aligned} u''(x_i) &\approx \partial_h^+ \partial_h^- v_i \\ &= \partial_h^+ \frac{v_i - v_{i-i}}{h} \\ &= \frac{\frac{u_{i+1}-u_i}{h} - \frac{u_i - u_{i-1}}{h}}{h} \\ &= \frac{1}{h^2}(u_{i+1} - 2u_i + u_{i-1}) \end{aligned}$$

Wir diskretisieren nun 1.1:

Diskretisierung 1.2 *Finde $u : \sigma \to \mathbb{R}$ mit*

$$\frac{-u_{i+1} + 2u_i - u_{i-1}}{h^2} = f_i := f(u_i)$$

mit $i = 1, \ldots, n$ und $u_0 = u_n = 0$.

1.2 ist äquivalent zu folgendem

Gleichungssystem 1.3

$$Au = f$$

mit $u = (u_0, \ldots, u_n)^T, f = (0, f_1, \ldots, f_{n-1}, 0)^T$ *und*

$$A = \frac{1}{h^2} \begin{pmatrix} 1 & 0 & & & \ldots & 0 \\ -1 & 2 & -1 & 0 & \ldots & \\ 0 & -1 & 2 & -1 & \ldots & \vdots \\ \vdots & & \ddots & & & \\ & & & -1 & 2 & -1 \\ 0 & & \ldots & & 0 & 1 \end{pmatrix}$$

Es ist $A \in GL(N+1)$.
Verallgemeinert betrachten wir jetzt die Wärmeleitung in einem homogenen Körper $\Omega \subset \mathbb{R}^d$ mit der Vorgabe $u = 0$ auf ganz $\partial\Omega$. Innerhalb des Körpers soll gelten:

Bedingung 1.4 $-\triangle u = 0$

Dabei ist $\triangle$ der Laplace-Operator $\triangle u(x) = \sum_{i=1}^{d} \frac{\partial^2 u(x)}{\partial x_i} = trHess(u) = \nabla\nabla u(x) = \nabla^2 u(x)$.
Der Einfachheit halber sei im folgenden $\Omega =]0,1[^d \subset \mathbb{R}^d$. Wir definieren wieder ein Gitter $\Sigma = \{x_\alpha = h\alpha | \alpha \in \mathbb{N}^d, 0 \leq \alpha_i \leq N, i = 1, \ldots, d\}$ mit $h = \frac{1}{N}, N$ gegeben.
Wir definieren die Menge der inneren Knoten $\mathring{\Sigma} = \Sigma \cap \Omega$ und die Menge der Randknoten $\partial\Sigma = \Sigma \cap \partial\Omega = \Sigma \backslash \mathring{\Sigma}$.
Wir suchen jetzt also eine Approximation $(u_\alpha)_{x_\alpha \in \Sigma} : \Sigma \to \mathbb{R}, u_\alpha \approx u(x_\alpha)$.
Die Approximation geschieht durch die partiellen Ableitungen $\partial_i := \frac{\partial}{\partial x_i}$. Wir definieren:

$$\begin{aligned} \partial_i^{+h} u_\alpha &:= \frac{u_{\alpha+e_i} - u_\alpha}{h} \\ \partial_i^{-h} u_\alpha &:= \frac{u_\alpha - u_{\alpha+e_i}}{h} \\ \triangle u(x_\alpha) &\approx \triangle_h u_\alpha := \sum_{i=1}^{d} \partial_i^{+h} \partial_i^{-h} u_\alpha \end{aligned}$$

Wir erhalten also als Approximation von 1.4 die

Finite-Differenzen-Approximation 1.5 *Gesucht ist:*

$$(u_\alpha)_{x_\alpha \in \Sigma} \text{ mit } -\triangle_n u_\alpha = f_\alpha := f(x_\alpha) \text{ für } x_\alpha \in \mathring{\Sigma} \text{ und } u_\alpha = 0 \text{ für } x_\alpha \in \partial\Sigma.$$

Man erhält eine Matrix mit d Bändern der Breite 1. Behandelt man diese Matrix als Bandmatrix der Bandbreite m mit der Gauß-Elimination, so ist der Speicheraufwand $O(nm)$. Für $A \in \mathbb{R}^{n\times n}$ ist $n = m = (N+1)^d$. Der Speicheraufwand ist also im $\mathbb{R}$ $O(n)$, im $\mathbb{R}^2$ $O(n^{\frac{3}{2}})$ und im $\mathbb{R}^3$ $O(n^{\frac{5}{3}})$. Z.B. ist also im $\mathbb{R}^3$ mit $N = 99$, also $n = 10^6$ $n^{\frac{5}{3}} = 10^{10}$, die Aufgabe mit dem Gauß-Algorithmus also nicht ohne weiteres lösbar. Als Alternative zum Gauß-Algorithmus bieten sich hier Iterationsverfahren an.

Ansatz:

Sei A regulär. Zu Lösen ist $Ax = b$. Wir zerlegen $A = M - N$, wobei M regulär und möglichst nah bei A ist. Dann folgt:

$$\begin{aligned} Ax = b &\iff Mx = Nx + b \\ &\iff x = M^{-1}Nx + M^{-1}b \\ &\iff x = (I - M^{-1}A)x + M^{-1}b \end{aligned}$$

Wir definieren nun also ein

Iterationsverfahren 1.6 *Sei $x_0 \in \mathbb{R}^n$ gegeben.*

$$x_{k+1} = Tx_k + M^{-1}b$$

mit $T = (I - M^{-1}A)$ (Iterationsmatrix)

Es ergeben sich folgende Forderungen:

1. M muss einfach zu invertieren sein
2. Das Verfahren muss konvergieren

Um die Konvergenz zu untersuchen, sei $\|\cdot\|$ eine Norm auf $\mathbb{R}^n$. Der Banachsche Fixpunktsatz liefert nun ein hinreichendes Kriterium für die Konvergenz: $\Phi(x) = Tx + M^{-1}b$ muss kontrahierend sein, d.h.

$$\|\Phi(x) - \Phi(y)\| \overset{!}{\leq} q\|x - y\|$$

mit $q < 1$.
Es ist:

$$\begin{aligned} \|\Phi(x) - \Phi(y)\| &= \|T(x-y)\| \\ &= \|(I - M^{-1}A)(x-y)\| \\ &\overset{!}{\leq} q\|x-y\| \forall x, y \\ \iff \|I - M^{-1}A\| := \sup_{z\neq 0} \frac{\|(I - M^{-1}A)z\|}{\|z\|} &\leq q \end{aligned}$$

Fragen:

1. Handelt es sich um eine notwendige Bedingung?
2. Welche Norm ist anzuwenden?
3. Wann ist $\|I - M^{-1}A\| < 1$?

1.2. Ein Allgemeiner Konvergenzsatz

Definition: Spektrum und Spektralradius

Für $T \in \mathbb{R}^{n\times n}$ ist $\sigma(T) := \{\lambda \in \mathbb{C} | \lambda \text{ ist Eigenwert von } T\}$ das *Spektrum* von T und $\rho(T) = \max_{\lambda\in\sigma(T)} |\lambda|$ der *Spektralradius* von T.

Satz 1.1

1. *Gegeben eine Norm $\|\cdot\|$ auf $\mathbb{R}^n$ und die zugehörige Operatornorm $\|\cdot\|$. Dann gilt:*
$$\rho(T) \leq \|T\|$$
für alle $T \in \mathbb{R}^{n\times n}$
2. *Für alle $T \in \mathbb{R}^{n\times n}, \varepsilon > 0$ existiert eine Norm auf $\mathbb{R}^n$, so dass für die zugehörige Operatornorm gilt: $\|T\| \leq \rho(T) + \varepsilon$.*

Beweis 1.1.1

1. *Sei $0 \neq e \in \mathbb{R}^n$ mit $Te = \lambda e$ und $|\lambda| = \rho(T)$. Dann ist:*
$$\|T\| = \sup_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Tx\|}{\|x\|} \geq \frac{\|Te\|}{\|e\|} = |\lambda| = \rho(T)$$
2. *Zu $T \in \mathbb{R}^{n\times n}$ existiert ein $S \in GL(n, \mathbb{C})$ mit $S^{-1}TS = J$ und $J = \begin{pmatrix} J_1 & & 0 \\ & \ddots & \\ 0 & & J_p \end{pmatrix}$ und $J_i \in \mathbb{C}^{m_i\times m_i}$ Jordanblöcke, $J_i = \begin{pmatrix} \lambda_i & 1 & 0 \\ & \ddots & 1 \\ 0 & & \lambda_i \end{pmatrix}$. Definiere nun $D := diag(1, \varepsilon, \ldots, \varepsilon^{n-1}) \in \mathbb{R}^{n\times n}$ und $\tilde{J} = D^{-1}JD$. Dann ist $\tilde{J}_{jk} = \frac{\varepsilon^{k-1}}{\varepsilon^{j-1}} J_{jk}$ und also $\tilde{J} = \begin{pmatrix} J_1 & & 0 \\ & \ddots & \\ 0 & & \tilde{J}_p \end{pmatrix}$ mit $J_i = \begin{pmatrix} \lambda_i & \varepsilon & 0 \\ & \ddots & \varepsilon \\ 0 & & \lambda_i \end{pmatrix}$.*

Jetzt ist $U := (SD)^{-1} \in GL(n, \mathbb{C})$ *und wir definieren* $\|\cdot\|$ *auf* $\mathbb{C}^n$ *durch* $\|x\| := \|Ux\|_\infty$. *Dann gilt:*

$$\begin{aligned}
\|T\|_{\mathbb{C}^n} &:= \sup_{x\in\mathbb{C}^n\setminus\{0\}} \frac{\|Tx\|}{\|x\|} \\
&= \sup_{x\in\mathbb{C}^n\setminus\{0\}} \frac{\|UTx\|_\infty}{\|Ux\|_\infty} \\
&= \sup_{y=Ux, x\in\mathbb{C}^n\setminus\{0\}} \frac{\|UTU^{-1}y\|_\infty}{\|y\|_\infty} \\
&= \sup_{y=Ux, x\in\mathbb{C}^n\setminus\{0\}} \frac{\|D^{-1}S^{-1}TSDy\|_\infty}{\|y\|_\infty} \\
&= \sup_{y=Ux, x\in\mathbb{C}^n\setminus\{0\}} \frac{\|\tilde{J}y\|_\infty}{\|y\|_\infty} \\
&\leq \rho(T) + \varepsilon
\end{aligned}$$

Beachte nun: $\|T\|_{\mathbb{R}^n} := \sup_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Tx\|}{\|x\|} \leq \sup_{x\in\mathbb{C}^n\setminus\{0\}} \frac{\|Tx\|}{\|x\|} \leq \rho(T) + \varepsilon$.

Satz 1.2 *Sei* $A \in \mathbb{R}^{n\times n}, A = M - N, M \in GL(n)$. *Dann gilt:*

1. *Die Iteration* $x_0 \in \mathbb{R}^n, Mx_{k+1} = Nx_k + b$ *konvergiert für jedes* $x_0 \in \mathbb{R}^n$ *genau dann, wenn gilt:* $\rho(I - M^{-1}A) < 1$.
2. *Sei* $\|\cdot\|$ *eine submultiplikative Norm auf* $\mathbb{R}^n$ *mit* $\|I - M^{-1}A\| = q < 1$. *Dann gilt:*
   - $\|x_k - x\| \leq \frac{q^k}{1-q}\|x_1 - x_0\|$
   - $\|x_{k+1} - x\| \leq \frac{q}{1-q}\|x_{k+1} - x_k\|$

   wobei $Ax = b$ *ist.*

Beweis 1.2.1 *Sei* $T := I - M^{-1}A, Mx_{k+1} = Nx_k + b, Mx = Nx + b$. *Dann ist* $M(x - x_{k+1}) = N(x - x_k)$, *kurz* $Me_{k+1} = Ne_k$, *also* $e_{k+1} = M^{-1}Ne_k = Te_k$.

1. *„⇒“:*

   Sei $\rho(T) = 1 \geq 1$. *Dann gilt:* $\exists e \neq 0$ *mit* $Te = \lambda e, |\lambda| = q \geq 1$. *Wir wählen nun* $x_0 = e + x$, *dann ist* $e_1 = e$. *Es gilt also* $e_k = T^k e = \lambda^k e$ *und deshalb* $\|e_k\| = q^k\|e\|$ *und das Verfahren konvergiert nicht.*

   „⇐“:

   Folgt aus Satz 1.1 und dem Banach'schen Fixpunktsatz.

2. *Folgt jetzt aus dem Banach'schen Fixpunktsatz.*

1.3. Gesamtschrittverfahren, Einzelschrittverfahren, Relaxation

Sei $A \in GL(n)$, zu Lösen: $Ax = b$. Sei weiter $A = A_D + A_L + A_R$ (Diagonale, linke untere Hälfte, rechte obere Hälfte).

Verfahren 1.3 *(Gesamtschrittverfahren)*
Sei $M := A_D, N := -(A_L + A_R)$. Vorausgesetzt sei $A_D \in GL(n)$.
Die Iteration 1.6 lautet dann:

Gesamtschrittverfahren, Jacobiverfahren 1.7

$$x_i^{k+1} = \frac{1}{a_i i}(b_i - \sum_{j \neq i} a_{ij} x_j^k), i = 1, \ldots, n, k \in \mathbb{N}$$

Bemerkungen:

1. Zu Lösen ist $Ax = b$, also $\sum_{j=1}^n a_{ij} x_j = b_i$, d.h. $a_{ii} x_i + \sum_{j \neq i} a_{ij} x_j = b_i \iff x_i = \frac{1}{a_{ii}}(b_i - \sum_{j \neq i} a_{ij} x_j)$
2. Der Speicheraufwand beträgt O(Speicher für A).

Verfahren 1.4 *(Einzelschrittverfahren, ESV, Gauß-Seidel)*

Einzelschrittverfahren 1.8

$$x_i^{k+1} = \frac{1}{a_{ii}}(b_i - \sum_{j<i} a_{ij} x_j^{k+1} - \sum_{j>i} a_{ij} x_j^k)$$

Bemerkung*: Das Verfahren hängt von der Nummerierung ab!*

In Vektorschreibweise:
$A_D x^{k+1} = b - A_L x^{k+1} - A_R x^k$
$\iff (A_D + A_L) x^{k+1} = b - A_R x^k$
also $M = (A_D + A_L), N = -A_R$
Damit ist:

$$T_{ESV} = (I - M^{-1}A) = I - (A_D + A_L)^{-1}(A_D + A_L + A_R) = -(A_D + A_L)^{-1} A_R$$

Idee 1.5 *Führe ein $\omega \neq 0$ ein, mit dem man die Wirkung von ESV oder GSV verstärken bzw. abschwächen kann.*

a) GSV
Sei x_i^k gegeben. Berechne dann z_i^{k+1} wie in 1.7: $z_1^{k+1} = \frac{1}{a_{ii}}(b_i - \sum_{j \neq i} a_{ij} x_j^k)$ und weiter:

Gesamtschrittverfahren + Relaxation 1.9

$$x_i^{k+1} := (1-\omega)x_i^k + \omega z_i^{k+1}$$

(im Fall $\omega = 1$ ergibt sich wieder das Gesamtschrittverfahren 1.7)

Es ist nun (in Vektorschreibweise):

$$\begin{aligned} z^{k+1} &= -A_D^{-1}(A_L + A_R)x_k + A_D^{-1}b \\ x^{k+1} &= (1-\omega)x^k + \omega z^{k+1} \\ &= T_{GSV}(\omega)x^k + \omega A_D^{-1}b \end{aligned}$$

mit $T_{GSV}(\omega) := (1-\omega)I - \omega A_D^{-1}(A_L + A_R)$, also:

Gesamtschrittverfahren + Relaxation (Vektornotation) 1.10

$$T_{GSV}(\omega) = I - \omega(I + A_D^{-1}A - I) = I - \omega A_D^{-1}A$$

b) ESV (mit Relaxation, auch SOR = Successive Over Relaxation)

Einzelschrittverfahren + Relaxation 1.11
Sei x^k gegeben, berechne für $i = 1, \ldots, n$:
$z_i^{k+1} = \frac{1}{a_{ii}}(b_i - \sum_{j<i} a_{ij}x_j^{k+1} - \sum_{j>i} a_{ij}x_j^k)$
$x_i^{k+1} = (1-\omega)x_i^k + \omega z_i^{k+1}$

Es ergibt sich in Vektornotation:
$A_D z^{k+1} = -A_L x^{k+1} - A_R x^k + b$
$x^{k+1} = (1-\omega)x^k + \omega z^{k+1}$
$x^{k+1} = (1-\omega)x^k + \omega A_D^{-1}(-A_L x^{k+1} - A_R x^k + b)$
$\iff (A_D + \omega A_L)x^{k+1} = (1-\omega)A_D x^k - \omega A_R x^k + \omega b$
und damit:

Einzelschrittverfahren + Relaxation (Vektornotation) 1.12

$$\begin{aligned} T_{ESV}(\omega) &= (A_D + \omega A_L)^{-1}\left((1-\omega)A_D - \omega A_R\right) \\ &= I - (\frac{1}{\omega}A_D + A_L)^{-1}A \end{aligned}$$

1.4. Konvergenzaussagen für ESV, GSV, Relaxation

Definition 1.6
Sei $A \in \mathbb{R}^{n\times n}$

i) A erfüllt das starke Zeilensummenkriterium*, A ist* diagonal dominant*, gdw.* $\sum_{j\neq i} |a_{ij}| < |a_{ii}| \forall i = 1,\ldots,n.$

ii) A erfüllt das schwache Zeilensummenkriterium*, gdw.* $\sum_{j\neq i} |a_{ij}| \leq |a_{ii}| \forall i = 1,\ldots,n$ ***und*** $\sum_{j\neq i_0} |a_{i_0 j}| < |a_{i_0 i_0}|$ *für mindestens ein* i_0*.*

iii) A heißt reduzibel *(zerlegbar), gdw. es* N_1, N_2 *gibt,* $\emptyset \neq N_1, N_2 \subset \{1,\ldots,n\}$*, so dass* $N_1 \cap N_2 = \emptyset, N_1 \cup N_2 = \{1,\ldots,n\}$ *und* $a_{ij} = 0$ *für alle* $i \in N_1, j \in N_2$ *gilt.*

iv) A heißt irreduzibel*, falls A nicht reduzibel ist.*

Veranschaulichung: Sei A reduzibel, mit N_1, N_2. Wir permutieren die Nummerierung, so dass O.E. gilt: $N_1 = \{1,\ldots,p\}, N_2 = \{p+1,\ldots,n\}$. Dann hat A die Form:

$$A = \begin{pmatrix} \in \mathbb{R}^{p\times p} & 0 \\ (*) & \in \mathbb{R}^{n-p\times n-p} \end{pmatrix}$$

Lemma 1.7 *Sei* $T \in \mathbb{R}^{n\times n}$*. Dann gilt:*

$$\begin{aligned} \|T\|_\infty &:= \sup_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Tx\|_\infty}{\|x\|_\infty} \\ &= \max_{1\leq i\leq n} \sum_{j=1}^{n} |T_{ij}| \text{ (Zeilensummennorm)} \\ \|T\|_1 &:= \sup_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Tx\|_1}{\|x\|_1} \\ &= \max_{1\leq j\leq n} \sum_{i=1}^{n} |T_{ij}| \text{ (Spaltensummennorm)} \end{aligned}$$

Beweis:

$$\begin{aligned} \frac{\|Tx\|_\infty}{\|x\|_\infty} &= \max_i \frac{|\sum_{j=1}^n T_{ij}x_j|}{\|x\|_{infty}} \\ &\leq \max_i \sum_{j=1}^{n} |T_{ij}| \end{aligned}$$

und für $\geq$: Finde ein geeignetes $x \in \mathbb{R}^n$.
Für $\|T\|_1$ analog.

Satz 1.8 *Sei $A \in \mathbb{R}^{n\times n}$.*

i) A erfülle das starke Zeilensummenkriterium. Dann ist das GSV wohldefiniert und es gilt $\|T_{GSV}\|_\infty < 1$.

ii) A erfülle das schwache Zeilensummenkriterium und sei irreduzibel. Dann ist das GSV wohldefiniert und es gilt $\rho(T_{GSV}) < 1$.

Beweis 1.8.1

i) Aus dem starken Zeilensummenkriterium folgt: $a_{ii} \neq 0$. Folglich ist das GSV wohldefiniert. Es gilt:

$$(T_{GSV})_{ij} = \begin{cases} -\frac{1}{a_{ii}} a_{ij} & \text{für } i \neq j \\ 0 & \text{für } i = j \end{cases}$$

Aus Lemma 1.7 folgt dann: $\|T_{GSV}\|_\infty = \max_i \sum_j |T_{ij}| = \max_i (\frac{1}{|a_{ii}|} \sum_{j\neq i} |a_{ij}|) < 1$.

ii)

1. Schritt: *Für alle i ist $a_{ii} \neq 0$ und damit ist das GSV wohldefiniert:*
Annahme: Es sei $a_{ii} = 0$ für ein i. Sei dann $N_1 := \{i|a_{ii} = 0\} \neq \emptyset$ und $N_2 := \{i|a_{ii} \neq 0\} \neq \emptyset$ (da n.V. ein i_0 existiert mit $\sum_{j\neq i_0} |a_{i_0 j}| < |a_{i_0 i_0}|, i_0 \in N_2$). Weiter ist $N_1 \dot\cup N_2 = \{1, \ldots, n\}$.
Für $i \in N_1$ gilt:
$\sum_{j\neq i} |a_{ij}| \leq |a_{ii}| = 0$, also $a_{ij} = 0$ für $j = 1, \ldots, n$. Insbesondere ist $a_{ij} = 0$ für ein $j \in N_2$ und A damit zerlegbar. $\lightning$

2. Schritt: *Wie unter i) folgt:*
$1 \geq \|T_{GSV}\|_\infty \geq \rho(T_{GSV})$
Annahme: Es existieren (λ, e), so dass $Te = \lambda e$ und $|\lambda| = 1, \|e\|_\infty = 1$. Sei dann $M_1 := \{i||e_i| = 1\} \neq \emptyset$.
Für alle $i = 1, \ldots, n$ gilt:
$|e_i| = |\lambda||e_i| = |\lambda e_i| = |(T_{GSV} e)_i| = \frac{1}{|a_{ii}|} |\sum_{j\neq i} a_{ij} e_j| \leq \frac{1}{|a_{ii}|} \sum_{j\neq i} |a_{ij}| \leq 1$
Für $i \in M_1$ gilt in dieser Ungleichung Gleichheit.
Sei nun $M_2 := \{1, \ldots, n\} \backslash M_1$. Es ist $M_2 \neq \emptyset$, denn aus der Existenz eines i_0 mit $\sum_{j\neq i_0} |a_{i_0 j}| < |a_{i_0 i_0}|$ folgt $|e_{i_0}| < 1$.
Weiter ist A unzerlegbar, d.h. $\exists (k, l) \in M_1 \times M_2 : a_{kl} \neq 0$. Es folgt:

$$1 = |\lambda||e_k| = |(T_{GSV}e)_k| = \tfrac{1}{|a_{kk}|}\sum_{j\neq k}|a_{kj}||e_j|$$
$$= \tfrac{1}{|a_{kk}|}\big(\sum_{j\neq k, j\in M_1}|a_{kj}|\underbrace{|e_j|}_{=1} + \sum_{j\neq k, j\in M_2}|a_{kj}|\underbrace{|e_j|}_{<1}\big) < \tfrac{1}{|a_{kk}|}\sum_{j\neq k}|a_{kj}| \leq 1$$
n.V. ↯

Satz 1.9 (Optimale Wahl von ω beim relaxierten GSV) *Sei $A \in \mathbb{R}^{n\times n}$.*
Es gelte $\sigma(T_{GSV}) = \{\lambda_1 \leq \cdots \leq \lambda_n\} \subset \mathbb{R}$ mit $\lambda_n < 1$.
Dann gilt:

$$\min_{\omega\in\mathbb{R}} \rho(T_{GSV}(\omega)) = \rho(T_{GSV}(\omega_{opt})) = \frac{\lambda_n - \lambda_1}{2 - \lambda_1 - \lambda_n} = 1 - 2\frac{1-\lambda_n}{2-\lambda_1-\lambda_n}$$

mit $\omega_{opt} = \frac{2}{2-\lambda_1-\lambda_n}$.

Beweis 1.9.1 *Es ist $\sigma(T_{GSV}(\omega)) = \{1-\omega+\omega\lambda_1 | \lambda_i \in \sigma(T_{GSV})\}$, also*

$$\min_{\omega\in\mathbb{R}} \rho(T_{GSV}(\omega)) = \min_{\omega\in\mathbb{R}} \max_{\lambda_i\in\sigma(T_{GSV})} |1-\omega+\omega\lambda_i|$$

Sei nun $f_i(\omega) := |1-\omega+\omega\lambda_i| = |1-(1-\lambda_i)\omega|$. Dann ist ω_{opt} gegeben durch $f_1(\omega_{opt}) = f_n(\omega_{opt})$, also $2 = ((1-\lambda_n)+(1-\lambda_1))\omega_{opt}$.

SOR (Successive Over Relaxation)

(dies enthält auch den Fall $\omega = 1$, also ESV)
SOR ist oft besser als das GSV, aber die Analysis ist komplizierter, daher wird hier nur ansatzweise darauf eingegangen. Genaueres findet sich in Stör/Bulisch: *Numerik II.*
Zuerst betrachten wir die „Grenzen“ des SOR-Verfahrens.

Satz 1.10 *Sei $A \in \mathbb{R}^{n\times n}$ mit $diag(a_{ii}) \in GL(n)$.*
Dann gilt: $\rho(T_{SOR}(\omega)) \geq |\omega - 1|$.
Eine notwendige Bedingung für die Konvergenz des SOR-Verfahrens ist also $\omega \in]0,2[$.

Beweis 1.10.1
Es ist $T_{SOR}(\omega) = (A_D + \omega A_L)^{-1}((1-\omega)A_D - \omega A_R)$. Weiter gilt (allgemein): $det(M) = \prod_{\lambda_i\in\sigma(M)} \lambda_i$. Daher folgt:

$$\begin{aligned} |det(T_{SOR}(\omega))| &= \frac{1}{det(diag(a_{ii}))}\prod_{i=1}^{n}(1-\omega)a_{ii} \\ &= (1-\omega)^n \\ &= \prod_{\lambda_i\in\sigma(M)} \lambda_i \end{aligned}$$

und damit:

$$\rho(T_{SOR}(\omega))^n \geq |\prod_{i=1}^{n} \lambda_i| = |1-\omega|^n$$

Satz 1.11 *Sei $A \in \mathbb{R}^{n\times n}$ mit $diag(a_{ii}) \in GL(n)$. A erfülle das starke Zeilensummenkriterium. Dann gilt für $\omega \in]0,1]$:*

$$\|T_{SOR}(\omega)\|_\infty \leq 1-\omega+\omega \underbrace{\|T_{GSV}\|_\infty}_{<1} < 1$$

Beweis 1.11.1 *siehe Werner:* Numerik I

Satz 1.12 *Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und positiv definit.*
Dann gilt:

i) $diag(a_{ii}) \in GL(n)$

ii) Das SOR-Verfahren konvergiert für alle $\omega \in]0,2[$

iii) „In Spezialfällen ist die Bestimmung eines optimalen ω möglich "

Beweis 1.12.1

i) Sei $e^{(i)}$ der i-te Einheitsvektor. Dann ist $0 < \langle e^{(i)}, Ae^{(i)}\rangle = a_{ii}$

ii) $T = T_{SOR}(\omega) = I - \underbrace{(\frac{1}{\omega}A_D + A_L)^{-1}}_{=:B(\omega)} A$

Sei nun $(\lambda, e) \in \mathbb{C}\times\mathbb{C}^n$ ein Eigenpaar von T, d.h. $(I - B(\omega)^{-1}A)e = \lambda e$, also $Ae = (1-\lambda)B(\omega)e$.

Es ist $\lambda \neq 1$, da $A \in GL(n)$.

Es folgt:

$$
\begin{aligned}
\langle Ae, e\rangle &= (1-\lambda)\langle B(\omega)e, e\rangle \\
\iff \frac{1}{1-\lambda} &= \frac{\langle B(\omega)e, e\rangle}{\langle Ae, e\rangle} \\
\iff \frac{1}{1-\bar{\lambda}} &= \frac{\overline{\langle B(\omega)e, e\rangle}}{\overline{\langle Ae, \rangle}} \\
&= \frac{\langle B^T(\omega)e, e\rangle}{\langle Ae, e\rangle} \\
\text{Denn : } \overline{\langle Ae, e\rangle} &= \langle e, Ae\rangle \\
&= \langle A^H e, e\rangle \\
&= \langle Ae, e\rangle \\
\text{und } \overline{\langle B(\omega)e, e\rangle} &= \langle e, B(\omega)e\rangle \\
&= \langle B^H(\omega)e, e\rangle \\
&= \langle B^T(\omega)e, e\rangle
\end{aligned}
$$

Es folgt jetzt:

$$
\begin{aligned}
2Re(\frac{1}{1-\lambda}) &= \frac{\langle (B(\omega)+B^T(\omega))e, e\rangle}{\langle Ae, e\rangle} \\
&= \frac{\langle (A+(\frac{2}{\omega}-1)A_D)e, e\rangle}{\langle Ae, e\rangle} \\
&= 1 + \underbrace{(\frac{2}{\omega}-1)}_{>0}\underbrace{\frac{\langle A_D e, e\rangle}{\langle Ae, e\rangle}}_{>0} \\
&> 1
\end{aligned}
$$

Denn $B(\omega)+B^T(\omega) = \frac{2}{\omega}A_D + A_L + A_L^T = A + (\frac{2}{\omega}-1)A_D$.
Sei $\lambda = \alpha + i\beta$. *Dann folgt:*
$\frac{1}{1-\lambda} = \frac{1}{(1-\alpha)-i\beta} = \frac{1-\alpha}{(1-\alpha)^2+\beta^2} + i\frac{\beta}{(1-\alpha)^2+\beta^2}$

$$
\begin{aligned}
&\Rightarrow \quad 2Re(\frac{1}{1-\lambda}) = 2\frac{1-\alpha}{(1-\alpha)^2+\beta^2} > 1 \\
&\Rightarrow \quad 2(1-\alpha) > (1-\alpha)^2 + \beta^2 \\
&\Rightarrow \quad |\lambda|^2 = \alpha^2 + \beta^2 < 1
\end{aligned}
$$

1.5. Anwendung auf das Modellproblem

Wir betrachten die Wärmeleitung (eindimensional, höhere Dimensionen analog)

Wir betrachten: $A \begin{bmatrix} a_1 \\ \vdots \\ a_{N-1} \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_{N-1} \end{bmatrix}$

Dabei ist $A = \frac{1}{h^2} \begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & & & \ddots & & & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n \times n}, n = N-1$

(Beachte: Hier ist $u_0 = u_N = 0$))

A erfüllt das schwache Zeilensummenkriterium und ist unzerlegbar.

Satz 1.13 (Eigenwerte von Tridiagonalmatrizen) *$A \in \mathbb{R}^{n \times n}$ sei eine Tridiagonalmatrix der Form*

$$A = \begin{bmatrix} \gamma & \beta & & 0 \\ \alpha & \ddots & \ddots & \\ & \ddots & \ddots & \beta \\ 0 & & \alpha & \gamma \end{bmatrix}, \alpha, \beta, \gamma \in \mathbb{R}, \alpha \cdot \beta \geq 0$$

Dann gilt: $\sigma(A) = \{\lambda_k = \gamma + 2\sqrt{ab}\ sgn(\alpha) \cos(\frac{k\pi}{n+1})\big|_{k=1}^{n}\}$

Beweis 1.13.1 *Ausnutzen von Additionstheoremen*

Wir betrachten nun die Wärmeleitung (eindimensional).

$$\mathbb{R}^{n \times n} \ni A = \frac{1}{h} \begin{bmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & & & \ddots & & & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix}$$

Die Eigenwerte berechnen sich nach 1.13 mit $\gamma = 2, \alpha = \beta = 1$:

$$\begin{aligned} \lambda_k &= \frac{1}{h^2}(2 - 2\cos(\frac{k\pi}{N})), k = 1, \ldots, n \\ \lambda_1 := \lambda_{min} &= \frac{2}{h^2}(1 - \cos(\pi k)) \\ \lambda_n := \lambda_{max} &= \frac{2}{h^2}(1 - \cos(n\pi h)) \end{aligned}$$

Die Taylor-Entwicklung von cos liefert:

$$\begin{aligned}\cos(\pi h) &= 1 - \frac{(\pi h)^2}{2} + \dots \\ \cos(\pi n h) &= -1 + \frac{(\pi h)^2}{2} - \dots\end{aligned}$$

Also:

$$\lambda_{min} = \frac{2}{h^2}(1 - 1 + \frac{(\pi h)^2}{2} - \dots) = \pi^2 + \dots$$

(Näherung gilt für kleine h)
Für λ_{max} gilt:

FiXme: *Hier stimmt was nicht...*

$$\begin{aligned}\lambda_{max} &= \frac{2}{h^2}(1 + 1 - \frac{(\pi h)^2}{2} + \dots) \\ \lambda_{max} &= O(\frac{1}{h^2})\end{aligned}$$

Lemma 1.14
$\lambda_{min} \approx \pi^2, \lambda_{max} = O(\frac{1}{h^2})(\ bei\ h \to 0)$

Anwendung von GSV:

$$T_{GSV} = -A_D^{-1}(A_L + A_R) = \begin{pmatrix} 0 & \frac{1}{2} & & \\ \frac{1}{2} & 0 & \frac{1}{2} & \\ & \ddots & \ddots & \\ & & \frac{1}{2} & 0 \end{pmatrix}$$

Es folgt:
$\sigma(T_{GSV} = \{lambda_k = \cos(k\pi h) | k = 1, \dots, n\}$
Also:
$\rho(T_{GSV} = \cos(\pi k) = -\cos(n\pi k) = 1 - \frac{(n\pi)^2}{2} + \dots$
Satz 1.9 liefert ein optimales ω_{opt} für die Relaxation:
$\rho(T_{GSV}(\omega_{opt})) = 1 - 2\frac{1-\cos(\pi k)}{2-\cos(\pi k)-\cos(n\pi k)} = 1 - 2\frac{\frac{(\pi k)^2}{2}}{1+\frac{(\pi k)^2}{2}+1-\frac{(\pi k)^2}{2}} + \dots = 1 - \frac{(\pi k)^2}{2} + \dots$
(für $h \to 0$)

Fazit:

Für das Modellproblem ergibt sich beim GSV kein Vorteil durch Relaxation.

Anwendung von SOR (ohne Beweis):

$$\rho(T_{ESV}) = 1 - (\pi k)^2 + \dots \text{ für } h \to 0$$

$$\rho(T_{SOR}(\omega_{opt})) = 1 - O(h) \text{ für } h \to 0$$

Wir betrachten die Anzahl benötigter Iterationen:
Sei $\|\cdot\|$ eine Norm auf $\mathbb{R}^n$ und auch die zugehörige Operatornorm. Dann gilt:

$$\|x_k - x^*\| \leq q^k \|x_1 - x^*\|$$

(Banachscher Fixpunktsatz)
mit $q = \|I - M^{-1}A\| < 1$.
Wir nehmen nun an, dass $\|x_k - x^*\| \approx q^k \|x_1 - x^*\|$ ist.
Weiteres Vorgehen: Wir wählen als Abbruchkriterium TOL:
$TOL \approx \|x_k - x^*\| \approx q^n \|x_1 - x^*\|$
Es folt:

Lemma 1.15 $\Rightarrow k \approx \frac{|\ln(\frac{TOL}{\|x_1 - x^*\|}|}{|\ln q|}$

Wir ersetzen jetzt $\|\cdot\|$ durch den Spektralradius und erhalten:

- für GSV:

  $k \approx c_0 \frac{1}{|\ln(1 - \frac{(\pi h)^2}{2})|}$

  mit $c_0 = |\ln(\frac{TOL}{\|x_1 - x^*\|}|$

  Nun ist $\ln(1 - x) \approx \ln(1) - x = -x$ und es folgt:

  $$k \approx c_0 \frac{2}{\pi^2} \frac{1}{h^2}$$

- für ESV:

  $k \approx c_0 \frac{1}{\pi^2} \frac{1}{h^2}$, ist also ca. doppelt so schnell wie das GSV bei gleichem Aufwand.

- für SOR mit ω_{opt}:

  $k \approx O(\frac{1}{h})$, also eine Ordnung schneller als GSV (bei gleichem Aufwand).

Warnung:

Ein „optimales“Verfahren würde $k = O(1)$ Iterationen benötigen. (zur Erreichbarkeit vgl. nächstes Semester)

2. Warnung:

ω_{opt} für das SOR-Verfahren ist i.A. nur schwer zu schätzen.

Bemerkung:

Diese asymptotischen Resultate sind im Prinzip übertragbar auf die FD-Approximation von $\triangle$ auf $\Omega \subset \mathbb{R}^n, d \in \mathbb{N}$.

Wir betrachten jetzt die FD-Approximation von $\triangle$ auf $\Omega \subset \mathbb{R}^d$. Für die Anzahl der Knoten gilt: $n = O(h^{-d})$. Der numerische Gesamtaufwand zum Lösen ist dann:
Aufwand = $\underbrace{\text{Anzahl der Iterationen}}_{k} * \underbrace{\text{Aufwand pro Iteration}}_{O(n) \text{ bei GSV, ESV, SOR}}$
Bei GSV und ESV ist $k = O(\frac{1}{h^2})$, der Gesamtaufwand ist also:

$$O(\frac{1}{h^2})O(n) = O(h^{-(d+2)} = O(n^{1+\frac{2}{d}})$$

$$\text{also } \begin{cases} O(n^3) & \text{in 1d} \\ O(n^2) & \text{in 2d} \\ O(n^{\frac{5}{3}}) & \text{in 3d} \end{cases}$$

Für das SOR-Verfahren mit ω_{opt} ist $h = O(\frac{1}{h})$ und der Aufwand ist

$$kO(n) = O(h^{-(d+1)}) = O(n^{1+\frac{1}{d}}$$

$$\text{also } \begin{cases} O(n^2) & \text{in 1d} \\ O(n^{\frac{3}{2}}) & \text{in 2d} \\ O(n^{\frac{4}{3}}) & \text{in 3d} \end{cases}$$

Als Alternative betrachten wir die Gauß-Elimination (in der Version als Bandmatrizenlöser):
Aufwand: $O(nm^2)$ (mit Bandbreite m)
Für die Bandbreite gilt: $m = O(h^{-d+1})$, der Aufwand ist also insgesamt $O(n^{3-\frac{2}{d}})$,

$$\text{also } \begin{cases} O(n) & \text{in 1d} \\ O(n^2) & \text{in 2d} \\ O(n^{\frac{7}{3}}) & \text{in 3d} \end{cases}$$

Warnung:

Ein optimales Verfahren würde den Aufwand $O(n)$ haben.

2. Numerische Lösung von Eigenwert- und -vektorproblemen

Wir betrachten Probleme der Gestalt: $A \in \mathbb{K}^{n\times n}$,$\mathbb{K} = \mathbb{R}, \mathbb{C}$. Gesucht: Eigenpaar $(\lambda, e) \in \mathbb{K} \times \mathbb{K}^m \backslash \{0\}$ mit $Ae = \lambda e$.

2.1. Einführung

Gegeben sei ein Gebiet $\Omega \subset \mathbb{R}^2$. Wir wollen das Schwingen einer dünnen Membran betrachten (z.B. gespannte Trommel). Wir betrachten kleine Auslenkungen v.
Es sei also $(x_1, x_2, v(t, x_1, x_2)) \in \mathbb{R}^3$ die Position der Membran zum Zeitpunkt t am Aufpunkt x.
Die Membran sei eingespannt am Rand $\partial\Omega$. v wird beschrieben durch die

Gleichung 2.1

$$\begin{aligned} \frac{\partial^2 v}{\partial t^2}(t,x) &= \triangle v(t,x), t > 0, x \in \Omega \\ v(t,x) &= 0 \text{ für } t > 0, x \in \partial\Omega \text{ (Randbedingung)} \\ v(0,x) &= v_0(x) \text{ für } x \in \Omega \\ \frac{\partial v}{\partial t}(v,x) &= v_1(x) \text{ (Anfangswerte)} \end{aligned}$$

Einfacher Fall: Separationsansatz

Wir nehmen an, dass v zerlegbar ist in: $v(t, x) = \psi(t)u(x)$.
Einsetzen in 2.1 liefert:
$\ddot{\psi}(t)u(x) = \psi(t)\triangle u(x)$ für alle $t > 0, x \in \Omega$.
Es folgt:

Gleichung 2.2

$$\frac{\ddot{\psi}(t)}{\psi(t)} = \frac{\triangle u(x)}{u(x)} \text{ für alle t mit } \psi(t) \neq 0 \text{ und x mit } u(x) \neq 0$$

Da die linke Seite nicht von x und die rechte Seite nicht von t abhängt, folgt:

$$\exists \lambda \in \mathbb{R} : \frac{\ddot{\psi}(t)}{\psi(t)} = -\lambda = \frac{\triangle u(x)}{u(x)} \text{ für alle t mit } \psi(t) \neq 0 \text{ und x mit } u(x) \neq 0$$

Die führt auf ein Eigenwertproblem für u:

Gleichung 2.3

$$\begin{aligned} -\triangle u(x) &= \lambda u \text{ für } x \in \Omega \\ u(x) &= 0 \text{ für } x \in \partial\Omega \end{aligned}$$

(Wir werden später sehen: $\lambda > 0$)
Sei also u Lösung von 2.3. Dann ist ψ gegeben durch:

Gleichung 2.4

$$\psi(t) = a\sin(\sqrt{\lambda}t) + b\cos(\sqrt{\lambda}t)$$

(a und b werden aus den Anfangsbedingungen bestimmt)

Es folgt dann für v:

$$v(t,x) = (a\sin(\sqrt{\lambda}t) + b\cos(\sqrt{\lambda}t))u(x)$$

Dies ergibt eine „stehende Welle“.
Das wesentliche Problem ist also die Lösung von 2.3, dem Eigenwertproblem für $-\triangle$. Dies ist nur in Spezialfällen explizit lösbar, z.B. wenn $\Omega =]0,1[^d$, dann folgt:
$u(x_1,\ldots,x_d) = \sin(k_1\pi x_1)\ldots\sin(k_d\pi x_d)$ mit $k_1,\ldots,k_d \in \mathbb{N}$.
In diesem Fall ist $\lambda = \pi^2(k_1^2 + \cdots + k_d^2)$.
Wir diskretisieren $-\triangle$, z.B. durch FD-Approximation, $-\triangle \rightsquigarrow A \in \mathbb{R}^{n\times n}$. Dies führt auf ein endlichdimensionales EW-Problem $Ae = \lambda e$. „Hoffnung“: Die Eigenwerte von A approximieren die Eigenwerte von $-\triangle$.

Achtung:

Nur die „kleinen“Eigenwerte von A sind i.d.R. gute Approximationen der Eigenwerte von $-\triangle$.

Allgemeine Anwendungen von Eigenwert-Proglemen

- Alle „schwingungsfähigen“Strukturen (Fahrzeuge, Flugzeuge, Vibrationsverhalten von Bauteilen etc.)
- Kenntnis von Eigenwerten dient dem Verständnis linearer Systeme

Literatur:

- *Golub / Loan:* Matrix Computation (Numerische Lineare Algebra)
- *Werner*: Numerische Mathematik II
- *Deuflhard / Hohmann*

2.2. Theoretische Grundlagen

Wir betrachten $\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$.
Es gilt: $\lambda \in \sigma(A) \iff \exists e \neq 0 : Ae = \lambda e \iff \lambda$ ist eine Nullstelle des charakteristischen Polynoms χ_A
Es ist $\chi_A(\lambda) := \det(A - \lambda I) \in \mathcal{P}_n = \prod_{l=1}^{k} (\lambda_e - \lambda)^{m_e}$ mit $\lambda_i \neq \lambda_j$ für $i \neq j$
Dabei ist m_e die algebraische Vielfachheit von λ_e.
Für $\lambda \in \sigma(A)$ ist $dim\ ker(A - \lambda I)$ die geometrische Vielfachheit.

Warnung:

Die Bestimmung von λ über das charakteristische Polynom ist i.A. überhaupt keine gute Idee für $n \geq 5$. (Aufwand ist $n!$).

Jordan-Form:

Zu $A \in \mathbb{C}^n$ existieren $S \in Gl(n, \mathbb{C})$ mit $J = S^{-1}AS, J = \begin{bmatrix} J_i & & 0 \\ & \ddots & \\ 0 & & J_p \end{bmatrix}$,

$$J_i = \begin{pmatrix} \lambda_i & 1 & & 0 \\ & \lambda_i & 1 & \\ & & \ddots & \ddots \\ 0 & & & \lambda_i \end{pmatrix}$$

Definition: $A \in \mathbb{K}^{n \times n}$ heißt *normal*, falls $A^H A = A A^H$.
A ist normal $\iff \exists Q \in \mathbb{K}^{n \times n}$ unitär, d.h. $Q^{-1} = Q^H$, und $Q^H A Q = diag(\lambda_i)$. Die Spalten von Q sind dann die Eigenvektoren von A.
Ist A hermitesch (d.h. $A^H = A$), so gilt $\sigma(A) \subset \mathbb{R}$.

Satz 2.5 *Komplexe Schur-Zerlegung Sei $A \in \mathbb{C}^{n \times n}$. Dann existiert eine unitäre Matrix $U \in \mathbb{C}^{n \times n}$ mit $U^H A U = \begin{pmatrix} \lambda_1 & & * \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$ eine rechte obere Dreiecksmatrix.*

Beweis 2.5.1 *Induktion nach n.*

Satz 2.6 *Reelle Schur-Zerlegung $A \in \mathbb{R}^{n \times n}$. Dann existiert eine orthogonale Matrix $Q \in \mathbb{R}^{n \times n}$ mit $Q^T A Q = \begin{pmatrix} R_{11} & R_{12} & \dots & R_{1m} \\ & \ddots & \ddots & \\ & & \ddots & \\ 0 & & & R_{mm} \end{pmatrix}$ und die R_{ij} sind entweder 1×1-Matrizen oder 2×2-Matrizen mit einem Paar komplex konjugierter Eigenwerte (nicht reell).*

Falls $\sigma(A) \subset \mathbb{R}$ ist, treten nur 1×1-Blöcke auf, d.h. es handelt sich um eine rechte obere Dreiecksmatrix.

Beweis 2.6.1 *Werner: Numerische Mathematik II*

Zur Stabilität der Eigenwertberechnung:

Satz 2.7 *Die Eigenwerte einer Matrix $A \in \mathbb{K}^{n \times n}$ hängen stetig von den Einträgen A_{ij} ab.*

Beweis 2.7.1 *z.B. Werner II, liegt daran, dass Nullstellen von Polynomen stetig von den Koeffizienten abhängen und die Eigenwerte die Nullstellen des charakteristischen Polynoms sind.*

Satz 2.8 *(Gerschgorein)*
$A \in \mathbb{C}^{n \times n}$. Für $\lambda = 1, \ldots, n$ definieren wir
$G_i := \{z \in \mathbb{C} | |z - a_{ii}| \leq r_k\} = \overline{B_{a_i i}(r_i)}$ mit $r_{ii} := \sum_{j \neq i} |a_{ii}|$.
Dann gilt:

i) $\sigma(A) \subset \bigcup_{i=1}^{m} G_i$

ii) *$i_1, \ldots, i_n$ Permutation von $1, \ldots, n$, $m < n$. Falls $(G_{i_1} \cup \cdots \cup G_{i_m}) \cap (G_{i_{m+1}} \cup \cdots \cup G_{i_n})$, dann enthält $G_{i_1} \cup \cdots \cup G_{i_m}$ genau m Eigenwerte (jeder Eigenwert wird entsprechen d seiner algebraischen Vielfachheit gezählt).*

Beispiel: $A = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} + \varepsilon E, E_{ii} = 0, |E_{ii}| \geq 1, 0 < \varepsilon < 0.05$

Sei $\lambda \in \sigma(A) \subset \mathbb{C}$, dann gilt: $|\lambda - 0.9| \leq 2\varepsilon$ oder $|\lambda - 0.4| \leq 2\varepsilon$ oder $|\lambda - 0.2| \leq 2\varepsilon$. Der Kreis $\overline{B_{x_0}(2\varepsilon)}$ enthält genau einen Eigenwert, falls $\varepsilon < 0.05$ für $x_0 \in \{0.9, 0.4, 0.2\}$.

Beweis 2.8.1 *a) Sei $\lambda \in \sigma(A)$, OE sei $\lambda \neq a_{ii}$ (sonst ist die Aussage trivial) Definiere $D := diag(a_i i)$, dann ist $\lambda_I - D \in Gl(n, \mathbb{C})$. Es ist $1 \in \sigma((\lambda I - D)^{-1}(A - D))$.*

Sei e Eigenvektor von A, d.h. $Ae = \lambda e$, also $(A - D)e = \lambda e - De = (\lambda I - D)e$, d.h. $(\lambda_I - D)^{-1}(A - D)e = e$, also $1 \in \sigma()$.

Sei $\| \cdot \|$ eine beliebige Operatornorm, dann ist $1 \leq \rho((\lambda I - D)^{-1}(A - D)) \leq \|(\lambda I - D)^{-1}(A - D)\|$.

Nun gilt: $((\lambda I - D)^{-1}(A - D))_{ij} = \frac{a_{ij}}{\lambda - a_{ii}}(1 - \delta_{ij})$ für $1 \leq i, j \leq n$.

Wähle nun $\|\cdot\| = \|\cdot\|_\infty$. *Dann folgt:*

$$\begin{aligned} 1 &\leq \|(\lambda I - D)^{-1}(A-D)\|_\infty \\ &= \max_i \sum_j |((\lambda I - D)^{-1}(A-D))_{ij}| \\ &= \max_i \sum_{j\neq i} \frac{|a_{ij}|}{|\lambda - a_{ii}|} = \max_i \frac{r_i}{|\lambda - a_{ii}|} \end{aligned}$$

Also existiert ein i_0 *mit* $1 \leq \frac{r_{i_0}}{|\lambda - a_{i_0 i_0}|}$, *also* $|lambda - a_{i_0 i_0}| \leq r_{i_0}$ *und damit* $\lambda \in G_{i_0}$.

b) **FiXme**: *Fehlt noch*

Definition 2.9

i) Sei $x \in \mathbb{K}^n$. *Wir definieren* $|x| \in \mathbb{K}^n$ *durch* $|x|_i = |x_i|$, *d.h.* $|x| = (|x_1|, \ldots, |x_n|)^T$.

ii) Für $x, y \in \mathbb{R}^n$ *sei* $x \leq y :\iff x_i \leq y_i \forall i = 1, \ldots, n$ *(Halbordnung)*

iii) Eine Norm auf $\mathbb{K}^n$ *heißt* monoton, *falls gilt:* $|x| \leq |y| \Rightarrow \|x\| \leq \|y\|$

iv) Eine Norm auf $\mathbb{K}^n$ *heißt* absolut, *falls gilt:* $\||x|\| = \|x\|$.

Beispiel: Sei $1 \leq p \leq \infty, \|x\|_p := (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}, \|x\|_\infty := \max_i |x_i|$. Dann sind $\|\cdot\|_p$ und $\|\cdot\|_\infty$ monoton und absolut.

Lemma 2.10
Gegeben eine Norm auf $\mathbb{K}^n$, *dann sind äquivalent:*

(i) Die Norm ist monoton.

(ii) Die Norm ist absolut.

(iii) Für jede Diagonalmatrix $D = diag(d_i) \in \mathbb{K}^{n\times n}$ *gilt:* $\|D\| = \max_i |d_i|$ *(dabei ist* $\|\cdot\|$ *die zugehörige Operatornorm).*

Beweis 2.10.1 *Übungsaufgabe*

Satz 2.11 *(Bauer-Fike) Sei eine absolute Norm* $\|\cdot\|$ *auf dem* $\mathbb{C}^n$ *gegeben. Sei* $A \in \mathbb{C}^{n\times n}$ *diagonalisierbar, d.h.* $\exists S \in GL(n, \mathbb{C}) : S^{-1}AS = diag(\lambda_i) =: D$. *Sei weiter* $E \in \mathbb{C}^{n\times n}$ *eine „Störung“.*
Dann gilt für $\lambda \in \sigma(A + E)$:

$$\begin{aligned} \min_{\lambda_j \in \sigma(A)} |\lambda - \lambda_j| &\leq \|S^{-1}ES\| \\ &\leq cond(S)\|E\|. \end{aligned}$$

Dabei ist $\|\cdot\|$ *die zugehörige Operatornorm und* $cond(S) := \|S\|\|S^{-1}\|$.
Beachte dabei: $S = [e_1, \ldots, e_n], e_i$ *Eigenvektoren von* A.

Beweis 2.11.1 *OE sei $\lambda \notin \sigma(A)$ (sonst ist die Aussage trivial). Sei e Eigenvektor zum Eigenwert λ von $(A+E)$, d.h. $(A+E)e = \lambda e$. Dann folgt:*

$$\begin{aligned}
Ee &= (\lambda I - A)e \\
&= (\lambda I - SDS^{-1})e \\
&= S(\lambda I - D)S^{-1}e \\
\iff (\lambda I - D)^{-1}S^{-1}Ee &= S^{-1}e \\
\iff S^{-1}e &= (\lambda I - D)^{-1}(S^{-1}ES)S^{-1}e \\
\Rightarrow \|S^{-1}e\| &\leq \underbrace{\|(\lambda I - D)^{-1}\|}_{\max_j(\frac{1}{|\lambda-\lambda_j|}) \text{ nach Lemma}} \|S^{-1}ES\|\|S^{-1}e\| \\
\Rightarrow 1 &\leq \max_j(\frac{1}{|\lambda - \lambda_j|})\|S^{-1}ES\| \\
\Rightarrow \frac{1}{\max_j(\frac{1}{|\lambda-\lambda_j|})} &\leq \|S^{-1}ES\| \\
\iff \min_j(|\lambda - \lambda_j|) &\leq \|S^{-1}ES\| \leq cond(S)\|E\|
\end{aligned}$$

Folgerungen / Erläuterungen 2.12 *i) Satz 2.11 liefert eine Abschätzung für die Kondition der Eigenwert-Berechnung:*

$\kappa_{abs} \leq cond(S)$

Im Allgemeinen kann $cond(S)$ mehr oder weniger groß sein!

ii) Falls A hermitesch ist, gilt $\kappa_{abs} = 1$ bezüglich der 2-Norm, da S dann unitär ist.

Beispiel 2.13 *(A nicht hermitesch)*
Sei $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, E = \begin{pmatrix} 0 & 0 \\ \delta & 0 \end{pmatrix}, \delta > 0.$
Dann ist $\sigma(A) = \{0\}, A + E = \begin{pmatrix} 0 & 1 \\ \delta & 0 \end{pmatrix}, \sigma(A+E) = \{\pm\sqrt{\delta}\}.$
Sei $\lambda := \sqrt{\delta}$.
Dann folgt: $\min_{\lambda_j \in \sigma(A)} |\lambda_i - \lambda| = \sqrt{\delta}$, *also:*

$$\kappa_{aps} \geq \underbrace{\frac{\sqrt{\delta}}{\|E\|_\infty}}_{=\delta} = \frac{1}{\sqrt{\delta}} \overset{\delta \to 0}{\to} \infty$$

Einige Eigenschaften hermitescher Matrizen

Satz 2.14 (Satz von Rayleigh)
Sei $A \in \mathbb{K}^{n\times n}$ hermitesch, Eigenwerte $\lambda_n \leq \cdots \leq \lambda_1$ und $e_1, \ldots, e_n$ zugehörige Orthonormalbasis von Eigenvektoren, d.h. $Ae_i = \lambda e_i, \langle e_i, e_j \rangle = \delta_{ij}$ für $i = 1, \ldots, n$.

Sei $M_j := \{x \in \mathbb{K}^n : \langle e_j, x\rangle = 0, i = 1, \ldots, j-1\} = span\{e_i, \ldots, e_{j-1}\}^\perp = span\{e_j, \ldots, e_n\}$ *(für* $j = 2, \ldots, n$, $M_1 := \mathbb{K}^n$*). Es ist* $dim M_j = n + 1 - j$.
Jetzt gilt:

$$\lambda_j = \max_{x \in M_j \setminus \{0\}} \underbrace{\frac{\langle x, Ax\rangle}{\|x\|_2^2}}_{\text{Rayleigh-Quotient}} \qquad \text{für } j = 1, \ldots, n$$

Beweis 2.14.1 *Sei* $x \in \mathbb{K}^n, x = \sum_{j=1}^n \alpha_i e_i, \alpha_i = \langle x, e_i\rangle$. *Es ist* $x \in M_j \iff \alpha_i = \cdots = \alpha_{j-1} = 0$.
Für $x \in M_j \setminus \{0\}$ *gilt:* $x = \sum_{i=j}^n \alpha_i e_i$.

$$\begin{aligned} \|x\|_2^2 &= \langle x, x\rangle \\ &= \langle \sum_{i=j}^n \alpha_i e_i, \sum_{i=j}^n \alpha_k e_k\rangle \\ &= \sum_{i,j=k}^n \alpha_i \overline{\alpha_k} \underbrace{\langle e_i, e_k\rangle}_{\delta_{ik}} \\ &= \sum_{i=j}^n |\alpha_i|^2 \end{aligned}$$

Ähnlich folgt:

$$\begin{aligned} \langle x, Ax\rangle &= \langle \sum_{i=j}^n \alpha_i e_i, \sum_{k=j}^n \alpha_k \underbrace{Ae_k}_{\lambda_k e_k}\rangle \\ &= \sum_{i=j}^n |\alpha_i|^2 \lambda_i \ (\lambda_i \text{ reell}) \end{aligned}$$

Es ist also:

$$\frac{\langle x, Ax\rangle}{\|x\|_2^2} = \frac{\sum_{i=j}^n |\alpha_i|^2 \lambda_i}{\sum_{i=j}^n |\alpha_i|^2} \leq \lambda_j \frac{\sum_{i=j}^n |\alpha_i|^2}{\sum_{i=j}^n |\alpha_i|^2} = \lambda_j$$

und damit:

$$\max_{x \in M_j \setminus \{0\}} \frac{\langle x, Ax\rangle}{\|x\|_2^2} \leq \lambda_j$$

Andererseits folgt aber mit $x = e_j \in M_j$*:*

$$\frac{\langle x, Ax\rangle}{\|x\|_2^2} = \lambda_j$$

und damit die Aussage.

Satz 2.15 (Satz von Courant)
Sei $A \in \mathbb{K}^{n\times n}$ hermitesch mit Eigenwerten $\lambda_n \leq \cdots \leq \lambda_1$. Für $1 \leq j \leq n$ gilt dann:

$$\lambda_j = \min_{\mathcal{N}\subset\mathbb{K}^n,\mathcal{N}\text{ UR},dim(\mathcal{N})=n+i-j} \max_{0\neq x\in\mathcal{N}} \frac{\langle x, Ax\rangle}{\|x\|_2^2}$$

Beweis 2.15.1 *Übungsaufgabe (ähnlich wie 2.14)*

2.3. Vektoriteration (power method)

Sei $A \in \mathbb{K}^{n\times n}$ diagonalisierbar mit Eigenpaaren $(\lambda_i, e_i) \in \mathbb{K} \times \mathbb{K}^n$, d.h. $\exists S \in GL(n,\mathbb{K})$: $S^{-1}AS = diag(\lambda_i)$.
Weitere Voraussetzung: $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$, λ_1 ist also einfacher Eigenwert.
Sei $y^{(0)} = \sum_{j=1}^n \alpha_j e_j$ ein „Startwert“.
Dann gilt:

Gleichung 2.16

$$\begin{aligned} A^k y^{(0)} &= \sum_{j=1}^n \alpha_j \lambda_j^k e_j \\ &= \lambda_1^k \left[\alpha_1 e_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k e_2 + \cdots + \alpha_n \left(\frac{\lambda_n}{\lambda_1}\right)^k e_n\right] \end{aligned}$$

Wir nehmen an, dass $\alpha_1 \neq 0$ gilt. Es ist also

$$A^k y^{(0)} = \lambda_1^k \left[\alpha_1 e_1 + O(|\frac{\lambda_2}{\lambda_1}|^k)\right]$$

Sei nun $1 \leq l \leq n$, so dass $(e_1)_l \neq 0$. Dann folgt:

Gleichung 2.17

$$\frac{(A^{k+1}y^{(0)})_l}{(A^k y^{(0)})_l} = \frac{\lambda_1^{k+1}[\alpha_1 e_1 + O(|\frac{\lambda_2}{\lambda_1}|^{k+1})]_l}{\lambda_1^k[\alpha_1 e_1 + O(|\frac{\lambda_2}{\lambda_1}|^k)]_l} \overset{k\to\infty}{\to} \lambda_1 \frac{(\alpha_1 e_1)_l}{(\alpha_1 e_1)_l} = \lambda_1$$

Für $x^{(k)} := \frac{A^k y^{(0)}}{\|A^k y^{(0)}\|}$ gilt dann:

$$dist(x^{(\alpha)}, span\{e_1\}) = O(|\frac{\lambda_2}{\lambda_1}|^n)$$

(wobei $dist(x,U) := \inf_{u\in U} \|x-u\|$)

Algorithmus 2.18 (Vektoriteration)

$$\begin{aligned} y^{(0)} &= \sum_j \alpha_j e_j, \alpha_1 \neq 0, 1 \leq l \leq n \text{ mit } (e_1)_l \neq 0 \\ x^{(0)} &:= \frac{y^{(0)}}{\|y^{(0)}\|} \end{aligned}$$

für $k \geq 1$ iteriere :

$$\begin{aligned} y^{(k)} &= Ax^{(k-1)} \\ \lambda^{(k)} &:= \frac{y_l^{(k)}}{x_l^{(k-1)}} \\ x^{(k)} &= \frac{y^{(k)}}{\|y^{(k)}\|} \end{aligned}$$

Lemma 2.19 *Für die $x^{(k)}$ aus Algorithmus 2.18 gilt:*

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

und damit:

$$\begin{aligned} dist(x^{(k)}, span\{e_1\}) &= O(|\frac{\lambda_2}{\lambda_1}|^n) \\ |\lambda^{(k)} - \lambda_1| &= O(|\frac{\lambda_2}{\lambda_1}|^k) \end{aligned}$$

Beweis 2.19.1 *1. Aussage durch Induktion, der Rest ist klar mit 2.16 und 2.17.*

Bemerkung 2.20

i) Die Konvergenz der Vektoriteration wird bestimmt durch $q := |\frac{\lambda_2}{\lambda_1}| < 1$.

ii) Die Implementierung ist trivial, die einzige wichtige Operation ist die Matrix-Multiplikation Ax.

iii) Zur Wahl von l: Wir erwarten $x^{(k)} \to e_1$, eine „gute Wahl“ für l wäre also:

Wähle l so, dass $|x_l^{(k)}| = \|y^{(k)}\|_\infty$.

iv) Zur Annahme $\alpha_1 \neq 0$:

Dies ist eine der wenigen Gelegenheiten, bei denen Rundungsfehler helfen. Die Annahme, dass zumindest $|\alpha_1| \geq$ eps ist, ist gerechtfertigt.

Modifikation der Vektoriteration

Sei λ eine „gute“ Näherung für λ_j, d.h.

Gleichung 2.21

$$|\lambda - \lambda_j| << |\lambda - \lambda_i| \forall i \neq j$$

O.E. sei $\lambda \neq \lambda_j$ (sonst ist die Lösung trivial). Dann ist
$(A - \lambda I) \in GL(n, \mathbb{K})$ und außerdem $\sigma((A - \lambda I)^{-1}) = \{\frac{1}{\lambda_i - \lambda}, i = 1, \ldots, n, \lambda_i \in \sigma(A)\}$.
Damit folgt

Gleichung 2.22

$$\frac{1}{|\lambda - \lambda_i|} << \frac{1}{|\lambda - \lambda_j|} \forall i \neq j$$

Die Idee ist jetzt: Wende den Algorithmus 2.18 auf $(A - \lambda I)^{-1}$ an. Es folgt:

Algorithmus 2.23 (Inverse Vektoriteration)
$\lambda_j \neq \lambda$ erfülle 2.21, $y^{(0)}$ sei gegeben und l so, dass $(e_j)_l \neq 0$. Sei $x^{(0)} := \frac{y^{(0)}}{\|y^{(0)}\|}$.
Iteriere jetzt für $k \geq 1$:

$$\begin{aligned} y^{(k)} &:= (A - \lambda I)^{-1} x^{(k-1)} \\ \lambda^{(k)} &:= \lambda + \frac{x_l^{(k-1)}}{y_l^{(k)}} \\ x^{(k)} &:= \frac{y^{(k)}}{\|y^{(k)}\|} \end{aligned}$$

Bemerkung 2.24 *i) Spezialfall $\lambda = 0$ liefert die Vektoriteration für A^{-1}, also ein Verfahren für λ_n, falls $|\frac{\lambda_n}{\lambda_{n-1}}| < 1$ (wichtig für partielle Differentialgleichungen (PDG))*

ii) Auf keinen Fall *$(A - \lambda I)^{-1}$ direkt ausrechnen! Bilde stattdessen LR-Zerlegung von $(A - \lambda I) = LR$ und löse in jedem Schritt das gestaffelte System:*

$$\begin{aligned} Rv &= x^{(k-1)} \\ Ly^{(k)} &= v \\ (\iff LRy^{(k)} &= x^{(k-1)}) \end{aligned}$$

Der Aufwand ist dabei: $O(n^3)$ für die LR-Zerlegung und dann in jedem Schritt $O(n^2)$ für die gestaffelten Gleichungssysteme.

Wir verallgemeinern jetzt das Verfahren auf höhere Dimension der Eigenräume.
Die Vektoriteration lässt sich auch so schreiben:

$$y^{(k)} = Ax^{(k-1)}$$
$$x^{(k)} r_k = y^{(k)}$$

Dabei ist r_k so gewählt, dass $\|x^{(k)}\|_2 = 1$.
Statt $x^{(k-1)}$ seien jetzt Vektoren $q_1^{(k-1)}, \dots, q_p^{(k-1)}, 1 \leq p \leq n$ gegeben mit $\langle q_i^{(k-1)}, q_j^{(k-1)} \rangle = \delta_{ij}$.

Algorithmus 2.25 (Orthogonale Iteration)
Seien $q_1^{(0)}, \dots, q_p^{(0)}$ gegeben mit $\langle q_i^{(0)}, q_j^{(0)} \rangle = \delta_{ij}$. Sei $Q_0 = [q_1^{(0)}, \dots, q_p^{(0)}] \in \mathbb{K}^{n \times p}$.
Für $k \geq 1$ iteriere jetzt

$$Z_k = AQ_{k-1}$$

und bestimme Q_k, R_k mit $Q_k = [q_1^{(k)}, \dots, q_p^{(k)}], \langle q_1^{(k)}, q_j^{(k)} \rangle = \delta_{ij}$ und $Q_k \underbrace{R_k}_{\in \mathbb{K}^{p \times p}} = Z_k \in \mathbb{K}^{n \times p}$ mit einer rechten oberen Dreiecksmatrix R_k.
Zur Bestimmung von $Q_k R_k$ mit $Q_k R_k = Z_k$ siehe Kapitel 5.3 aus Numerik I.
Aufwand: *Bestimmung von Q_k, R_k: $O(np^2)$*
Details: *Übungsaufgabe*

Warum soll das funktionieren?

Sei $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$. Die Schur-Zerlegung nach Satz 2.5 liefert:

$$Q^H A Q = \underbrace{T}_{\text{rechte obere Dreiecks-Matrix}} = diag(\lambda_i) + N$$

Nun sei $Q = [\underbrace{Q_\alpha}_{\in \mathbb{K}^{n \times p}} \underbrace{Q_\beta}_{\in \mathbb{K}^{n \times n-p}}]$,
$T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}$ mit $T_{11} \in \mathbb{K}^{p \times p}$.
Es gilt:

$$\begin{aligned} Q^H AQ &= T \\ \iff AQ &= QT \\ \iff A[Q_\alpha Q_\beta] &= [Q_\alpha Q_\beta] \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \end{aligned}$$

Es ist $Arg([Q_\alpha Q_\beta]) \subset rg([Q_\alpha Q_\beta])$, d.h. $rg([Q_\alpha 0])$ ist A-invariant.
Genauer gilt: $rg([Q_\alpha 0]) = span\{e_1, \dots, e_p\}$, wobei $e_1, \dots, e_p$ die Eigenvektoren zu $\lambda_1, \dots, \lambda_p$ sind. Definiere $D_p(A) := rg([Q_\alpha 0]) = span\{e_1, \dots, e_p\}$.

Satz 2.26
Seien die Bezeichnungen wie oben gewählt und zusätzlich $|\lambda_p| > |\lambda_{p+1}|$. *Es sei* $Q^H A Q = T$, *wobei* $Q_0 = [q_1^{(0)}, \dots, q_p^{(0)}]$ *mit* $\langle q_i^{(0)}, q_j^{(0)} \rangle = \delta_{ij}$ *und* $dist(D_p(A^H), rg([Q_0, 0])) < 1$.
Dann gilt:
Für Q_k *aus Algorithmus 2.25 gilt:*

FiXme: *Stimmt da was mit den Q-Indizes nicht?*

$$dist(D_p(A), rg([Q_k, 0])) = O(|\frac{\lambda_{p+1}}{\lambda_p}|^k)$$

Beweis 2.26.1 *ziemlich technisch, vgl. z.B. Golub / Loan*

Orthogonale Iteration für „kleine“ p ist dann von Bedeutung, wenn nur wenige Eigenwerte interessieren (z.B. bei PDG; arbeite hierbei mit A^{-1}).

2.4. QR-Verfahren

Ziel:

Bestimmung *aller* Eigenwerte von $A \in \mathbb{K}^{n \times n}$ (der Einfachheit halber sei hier $\mathbb{K} = \mathbb{R}$).
Ansatz: Im Prinzip Algorithmus 2.25, die orthogonale Iteration mit $p = n$.
Aber: Der Aufwand in jedem QR-Schritt ist dann $O(np^2) = O(n^3)$.
Stattdessen: In einem Schritt wird A mit Ähnlichkeitstransformationen auf eine „einfachere“ Gestalt überführt.

Definition 2.27 (Hessenberg-Matrix)
$A \in \mathbb{K}^{n \times n}$ *heißt* Hessenberg-Matrix, *falls* $A_{ij} = 0$ *für alle* $1 \leq j \leq i - 1$, A *also bis auf eine Nebendiagonale eine rechte obere Dreiecksmatrix ist.*
Eine Hessenberg-Matrix A *heißt* unreduziert, *falls alle* $A_{i+1,i}$ *wirklich* $\neq 0$ *sind.*

Klar ist: A Hessenberg und symmetrisch $\Rightarrow$ A tridiagonal.

Die grundlegende Idee des QR-Verfahren ist also: Die Matrix A wird mit Ähnlichkeitstransformationen auf Hessenberg-Form gebracht (in endlich vielen Schritten). Dies geschieht mittle Householder-Reflexion. Dann wird die QR-Zerlegung auf die Hessenberg-Matrix angewandt, dies erfordert nur noch den Aufwand $O(n^2)$.

Erinnerung: Householder-Reflexion

Sei $0 \neq v \in \mathbb{R}^n$. Definiere:

Gleichung 2.28

$$Q(v) = I - 2\frac{vv^T}{\|v\|_2^2}$$

Beachte: $vv^T \in \mathbb{R}^{n \times n}$.
Es ist nun $vv^T x = \langle v, x \rangle v$, für $x \in \mathbb{R}^n$. Es ist $rank(vv^T) = 1$.

Lemma 5.10 aus Wintersemester

i) $QQ^T = QQ^T = I$

ii) $Q^T = Q$

iii) $Q^2 = I$

Sei nun $y \in \mathbb{R}^n, v := y + sgn(y_1)\|y\|e_1$. Dann gilt $Q(v)y = -sgn(y_1)\|y\|e_1$, also:

Gleichung 2.29

$$\begin{aligned} \|v\|^2 &= \langle v, v\rangle \\ &= \|y\|^2 + \|y\|^2 + 2\langle y, sgn(y_1)\|y\|e_1\rangle \\ &= 2\|y\|^2 + 2|y_1|\|y\| \\ &= 2\|y\|(|y_1| + \|y\|) \end{aligned}$$

Also: $\frac{2}{\|v\|^2} = \frac{1}{\|y\|(|y|+\|y\|)}$

Sei nun A gegeben. Wir transformieren nun A mit Ähnlichkeitstransformationen $A_{k+1} = Q_k A Q_k$. Dabei hat A stets die Form $A_k = \begin{pmatrix} H_k & & B_k \\ 0 & a_k & C_k \end{pmatrix}$, wobei $H_k \in \mathbb{R}^{k\times k}$ Hessenberg, $B_k \in \mathbb{R}^{k\times(n-k)}, C_k \in \mathbb{R}^{(n-k)\times(n-k)}, a_k \in \mathbb{R}^{n-k}$.
Für $k = 1$ ist diese Form offensichtlich immer erfüllt, für $k = n-2$ entspricht sie der Hessenberg-Form von A. Wir zeigen, dass die Gestalt bei entsprechender Wahl von Q_k erhalten bleibt:

Wähle $Q_k = \begin{pmatrix} I_k & 0 \\ 0 & \bar{Q}_k \end{pmatrix}$ mit $\bar{Q}_k$ so, dass $\bar{Q}_k a_k = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix}$. Damit bleibt genau die Form erhalten.

Algorithmus 2.30 (Transformation auf Hessenberg-Form)
Sei $A \in \mathbb{R}^{n\times n}$ gegeben.

```
for k = 1,...,n:
  if
    a_k := (a_{k+1,k}, ..., a_{n,k})^T  ≠  0
  then
    v_k  :=  a_k  +  ( sgn(a_{k+1,k}) ||a_k||_2 )
    β_k  :=  1 / (||a_k|| (|a_{k+1,k}| + ||a_k||))
```

$\bar{Q}_k = I_{n-k} - \beta_k \, v_k \, v_k^T$
$Q_k = \begin{pmatrix} I_k & 0 \\ 0 & \bar{Q}_k \end{pmatrix}$
`else`
$Q_K = I_n$
$A := Q_k \; A \; Q_k$

Input von 2.30: $A \in \mathbb{R}^{n \times n}$
Output: $Q = Q_1 \dots Q_{n-2}$ *orthogonal,* $Q^T AQ$ *Hessenberg nach Konstruktion, gespeichert als* A*.*

Satz 2.31 *Sei* $A \in \mathbb{R}^{n \times n}$ *beliebig. Dann existieren Householder-matrizen* $Q_1, \dots, Q_{n-2}$ *mit* $Q^T AQ$ *Hessenberg für* $Q = Q_1 \dots Q_{n-2}$*.*

Beweis 2.31.1 *s.o.*

Klar ist nach Konstruktion: $H := Q^T AQ$ und A haben die selben Eigenwerte. $(\bar{e}, \lambda)$ ist ein Eigenpaar von H gdw. (e, λ) ein Eigenpaar von A ist, mit $e := Q_1 \dots Q_{n-2}\bar{e}$.
Zur Berechnung der Eigenvektoren von A werden also die Q_k benötigt.
Da $Q_k = \begin{pmatrix} I_k & 0 \\ 0 & \bar{Q}_k \end{pmatrix}$ und $\bar{Q}_k = I_{n-k} - \beta_k v_k v_k^T$, reicht es, v_k und β_k zu speichern.
Da bei der Berechnung von H die Plätze unterhalb der Nebendiagonalen mit Nullen aufgefüllt werden, speichert man dort die v_i, die erste Komponente speichert man separat. Auch die β_i werden separat gespeichert. Insgesamt ergibt sich konkret folgender

Algorithmus 2.32 (Reduktion auf Hessenberg-Form)
Sei $A \in \mathbb{R}^{n \times n}$ *gegeben.*

`for` $k = 1, \dots, n-2$
$\tau := \max_{i \geq k+1} |a_{ik}|$
`if`
$\tau \leq \|A\|$`eps`
`then`
$d_k := 0$
$\beta_k := 0$
`else`
`/* Berechnung von` v_k `*/`
$\alpha := 0$
`for` $i = 0, \dots, n$
$\alpha := \alpha + a_{ik}^2$
$\alpha := \sqrt{\alpha}$
$\beta_k := \frac{1}{\alpha + |a_{ik}|}$

$d_k := -sgn(a_{k+1,k})\alpha$
$a_{k+1,k} := a_{k+1,k} + sgn(a_{k+1,k})\alpha$
`/* Berechnung von` $\overline{Q_k}C$ `*/`
`for` $j = k+1, \ldots, n$
  $s := \beta_k \sum_{i=k+1}^{n} a_{ik}a_{ij}$
  `for` $i = k+1, \ldots, n$
    $a_{ij} := a_{ij} - sa_{ik}$
`/* Berechnung von` $\overline{Q_k}C\overline{Q_k}$ `und` $B\overline{Q_k}$ `*/`
`for` $i = 1, \ldots, n$
  $s := \beta_k \sum_{j\geq k+1} a_{ij}a_{jk}$
  `for` $j = k+1, \ldots, n$
    $a_{ij} = a_{ij} - sa_{jk}$
Vertausche $a_{k+1,k}$ ***und*** d_k

Der Aufwand dieses Algorithmus verhält sich etwa wie:

$$\sum_{k=1}^{n}[(n-k) + \sum_{j=k+1}^{n}(n-k) + \sum_{i=1}^{n}(n-k)] \approx \sum_{k=1}^{n} n(n-k) = O(n^3)$$

Das weitere Vorgehen ist jetzt wie folgt:
Sei $A = H$ Hessenberg. Iteriere dann mit $A_1 = A$ für $k \geq 1$:
$Q_k R_k = A_k$
$A_{k+1} = R_k Q_k$
Damit dies Sinn macht, muss zumindest A_{k+1} Hessenberg sein.

Definition und Lemma 2.33 (Givens-Operationen)
Seien $s, c \in \mathbb{R}, s^2 + c^2 = 1$.
Durch

$$G_{ij} := \begin{pmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ 0 & & & & & & 1 \end{pmatrix}$$

ist eine orthogonale Matrix $G_{ij} \in \mathbb{R}^{n\times n}$ *gegeben, eine sogenannte Givens-Rotation.*
Es gilt: Für $x \in \mathbb{R}^n$ *und* $y = G_{ik}x$ *gilt:*

i) $y = \begin{pmatrix} x_1 \\ cx_i + sy_k \\ cx_k - sx_i \\ x_n \end{pmatrix}$

ii) Für $A = (a_{ij})_{ij} \in \mathbb{R}^{n\times n}$ *gilt:*

$$(G_{ij}A)_{mj} = \begin{cases} ca_{ij} + sa_{ij} & , \text{ falls } m = i \\ -s_{aj} + ca_{kj} & , \text{ falls } m = k \\ a_{mj} & \text{sonst} \end{cases}$$

Es werden also nur die i-te und die k-te Zeile verändert.

Beweis 2.33.1 *Nachrechnen*

Geometrische Interpretation: Es handelt sich um eine Drehung um den Winkel ϑ mit $s = \sin(\vartheta)$ und $c = \cos(\vartheta)$. $G_{ik}x$ rotiert den Vektor x um den Winkel ϑ in der $(x_i - x_k)$-Ebene.
Zu $x \in \mathbb{R}^n$ existieren also c, s mit $c^2+s^2 = 1$ und $(G_{ij}x)_k = 0$, alle anderen Komponenten außer i und k bleiben unverändert. Die

Gleichung 2.34

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} x_i \\ x_k \end{pmatrix} = \begin{pmatrix} \gamma \\ 0 \end{pmatrix}$$

ist also lösbar mit

Gleichung 2.35

$$\begin{aligned} c &= \pm \frac{x_i}{\sqrt{x_i^2 + x_k^2}} \\ s &= \pm \frac{x_k}{\sqrt{x_i^2 + x_k^2}} \\ \gamma &= \pm \sqrt{x_i^2 + x_k^2} \end{aligned}$$

Die Idee ist jetzt: Verwende Givens-Rotationen, um A in eine QR-Zerlegung zu überführen, d.h. $QR = A$ mit $Q = G_{n-1,n} \dots G_{1,2}$.
Genauer: Bestimme induktiv Givens-Rotationen $G_{j,j+1}$ mit $A^{(k-1)} := G_{k-1,k} \dots G_{1,2}A = \begin{pmatrix} R_{k-1} & * \\ 0 & H_{k-1} \end{pmatrix}$, wobei R_{k-1} obere Dreiecksmatrix ist und H_{k-1} und A Hessenberg-Matrizen.
Dies ist offensichtlich möglich für $k = 1$:

$H_0 = A, G_{0,1} = I$

Wir zeigen jetzt den Induktionsschritt. Habe also $A^{(k-1)}$ die Form $A^{(k-1)} = \begin{pmatrix} R_{k-1} & * \\ 0 & H_{k-1} \end{pmatrix}$.

Wir suchen jetzt ein $G_{k,k+1}$, so dass $(G_{k,k+1}A^{(k-1)})_{k+1,k} = 0$.
Sei $a_k = (a_{1k}, \ldots, a_{nk})^T$. Bestimme jetzt $G_{k,k+1}$ so, dass $(G_{k,k+q}a_k)_{k+1} = 0$ (dies geht nach 2.35).
Beachte: $G_{k,k+1}A^{(k-1)}$ verändert nur die k-te und die $k+1$-te Zeile. Zudem sind die neuen Zeilen Linearkombinationen voneinander, es ergeben sich also gerade die passenden 0-Zeilen.

Aufwand: Zur Berechnung von $G_{i,k}A^{(k-1)}$: $O(n-k)$. Daher erfordert die Berechnung von $A^{(n-1)} = R = G_{n-1,n}, \ldots, G_{1,2}A$ die Operationen:
$O(\sum_{k=1}^{n-1}(n-k)) = O(n^2)$.
Zum Vergleich: Die Householder-Transformation braucht hierzu $O(n^3)$ Operationen.

Noch zu klären ist jetzt:

i) Wie berechnet man effektiv $\tilde{A} = RQ$?

ii) Ist $\tilde{A}$ Hessenberg?

Zu ii): Es ist $Q^T = G_{n-1,n} \ldots G_{1,2}$, also $RQ = RG_{1,2}^T \ldots G_{n-1,n}^T$.
Annahme: $RG_{1,2}^T \ldots G_{k-1,k}^T = \begin{pmatrix} H_k & * \\ 0 & R_k \end{pmatrix}$
Für $k = 1$ ist dies erfüllt mit $G_{0,1}^T = I$. Da die Givens-Rotation $G_{k,k+1}$ nur die k-te und $k+1$-te Spalte verändert und diese Linearkombinationen der alten Spalten sind (insbesondere bleiben Nullen in beiden Spalten erhalten), bleibt diese Gestalt erhalten. Induktiv folgt dann, dass $\tilde{A} = RQ$ eine Hessenberg-Matrix ist.

Zu i): Wir betrachten einen Zwischenschritt in der QR-Zerlegung: $A^{(1)} = A \rightsquigarrow A^{(2)} = G_{2,3}G_{1,2}A$. Nun verändert $A^{(2)}G_{1,2}^T$ nur die erste und zweite Spalte, die Multiplikation von links verändert nur die 3. und 4. Zeile. Folglich bleiben die ersten beiden Spalten unverändert.
Induktiv folgt folgendes Verfahren:
Sei $G_{k-1,k} \ldots G_{1,2}AG_{1,2}^T \ldots G_{k-2,k-1}^T$ gegeben. Bestimme dann $G_{k,k+1}$ und bilde $G_{k,k+1} \ldots G_{1,2}AG_{1,2}^T \ldots G_{k-2}^T$
und multipliziere anschließend von rechts mit $G_{k-1,k}^T$.
Ganz am Ende muss dann noch mit $G_{k-1,k}^T$ (von rechts) multipliziert werden.

Satz 2.36
Sei $A \in \mathbb{R}^{n\times n}$ Hessenberg. Es existieren dann $n-1$ Givens-Rotationen $G_{k,k+1}, k = 1, \ldots, n-1$, so dass für $Q = G_{1,2}^T \ldots G_{n-1,n}T$ gilt:

i) $Q^T A = R$ *ist obere Dreiecksmatrix*

ii) $\tilde{A} = RQ$ *ist Hessenberg-Matrix*

iii) $\tilde{A}$ *kann mit einem Aufwand von* $O(n^2)$ *berechnet werden.*

Algorithmus 2.37 (ROT)

$$\begin{aligned} (c, s, \gamma) &= ROT(\alpha, \beta) \\ (c, s, \gamma)\text{erfüllen :} & \\ \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} &= \begin{pmatrix} \gamma \\ 0 \end{pmatrix} \end{aligned}$$

c, s, γ *ergeben sich nach 2.35*

Der folgende Algorithmus realisiert nun: Für A in Hessenberg-Form, bestimme $QR = A$ und bilde $\tilde{A} = RQ = Q^T AQ$.

Algorithmus 2.38
Sei $A \in \mathbb{R}^{n \times n}$ *Hessenberg.*

```
for k = 1, ..., n
  if k < n
    (c_neu, s_neu, a_kk) = ROT(a_kk, a_{k+1,k})
    a_{k+1,k} = 0
    for j = k+1, ..., n
      ( a_kj     )   (  c_neu   s_neu )( a_kj     )
      ( a_{k+1,j} ) = ( -s_neu   c_neu )( a_{k+1,j} )
    end
  endif
  if k > 1
    for i = 1, ..., k
      (a_{i,k+1}  a_ik) = (a_{i,k+1}  a_ik) (  c_alt   -s_alt )
                                            (  s_alt    c_alt )
    end
  endif
  c_alt = c_neu
  s_alt = s_neu
end
```

Dieser Algorithmus realisiert einen QR-Schritt. Input: $A^{(k)}$*, Output:* $A^{(k+1)} = Q^T A^{(k)} Q$.

Zur Konvergenz des QR-Verfahrens

Im optimalen Fall konvergiert das Verfahren gegen die Schur-Zerlegung von A. Dies setzt allerdings voraus, dass alle Eigenwerte von A reell sind. Wir hoffen: $A^{(k)} \overset{k\to\infty}{\to} Q_\infty^T A Q_\infty = R$ mit orthogonalem Q_∞ und dann $\sigma(A) = \{r_{ii}, i = 1, \dots, n\}$. Die Konvergenz in dieser Situation setzt voraus, dass A diagonalisierbar ist (und alle λ_i reell).

Satz 2.39
Sei $A \in \mathbb{R}^{n\times n}$ diagonalisierbar, alle Eigenwerte von A reell und paarweise disjunkt, $|\lambda_1| > \dots > |\lambda_n| > 0$. Es gelte:

$$span\{e^{(1)}, \dots, e^{(m)}\} \cap span\{e_{m+1}, \dots, e_n\} = \{0\}$$

für alle $m = 1, \dots, n-1$.
Dabei sind e_i die Eigenvektoren von A und $e^{(i)}$ die i-ten Einheitsvektoren.
Dann gilt für $A^{(k)}$ aus 2.38:

$$A_{ij}^{(k)} \overset{k\to\infty}{\to} 0 (1 \le j < i \le n)$$
$$A_{ii}^{(k)} \overset{k\to\infty}{\to} \lambda_i (1 = 1, \dots, n)$$

Bemerkung 2.40

i) A sei Hessenberg. Die Bedingung über den span ist dann erfüllt, falls A unreduziert ist.

Ist A nicht unreduziert, gilt also $A = \begin{pmatrix} A' & B \\ 0 & A'' \end{pmatrix}$, so reicht es, die Eigenwertprobleme für A' und A'' zu betrachten, wegen Lemma 2.41.

ii) Der Fall $A \notin GL(n)$:

Nach i) ist A o.B.d.A unreduzierte Hessenberg-Matrix, d.h. $a_1, \dots, a_{n-1}$ sind in jedem Fall linear unabhängig. Daraus ergibt sich im ersten Schritt dann eine Nullzeile in der letzten Zeile von R und $A^{(2)}$ hat dann auch eine letzte Nullzeile, was einen Spezialfall von i) mit $A'' = 0 \in \mathbb{R}^{1\times 1}$ darstellt.

Lemma 2.41
Sei $A \in \mathbb{K}^{n\times n}$ von der Form $A = \begin{pmatrix} A' & B \\ 0 & A'' \end{pmatrix}, A' \in \mathbb{K}^{l\times l}, A'' \in \mathbb{K}^{n-l\times n-l}$.
A habe paarweise verschiedene Eigenwerte $\lambda_1, \dots, \lambda_n$. Dann gilt: $\sigma(A) = \sigma(A') \dot\cup \sigma(A'')$.
Außerdem gilt:
Sei e Eigenvektor von A, dann folgt: $e'' = (e_{l+1}, \dots, e_n)^T = 0$ oder e'' ist ein Eigenvektor von A''.

Beweis 2.41.1
Sei $e' := (e_1, \ldots, e_l)^T, e'' := (e_{l+1}, \ldots, e_n)^T$ *und* (λ_i, e_i) *sei Eigenpaar von* A*, also* $Ae_i = \lambda_i e_i$*.*
Aus der Struktur von A *folgt jetzt:* $A'e_i' + Be_i'' = \lambda_i e_i'$ *und* $A''e_i'' = \lambda_i e_i''$*. Ist nun* $e_i'' \neq 0$*, so folgt sofort, dass* (λ_i, e_i'') *Eigenpaar von* A'' *ist. Ist aber* $e_i'' = 0$*, so ist* $Be_i'' = 0$ *und damit* (λ_i, e_i') *Eigenpaar von* A'*.*
Sei nun $\mathcal{N} := \{i : e_i'' \neq 0\}$ *und* $\mathcal{M} := \{i : e_i'' = 0\}$*. Dann ist* $\mathcal{N} \cap \mathcal{M} = \emptyset$ *und* $\mathcal{N} \cup \mathcal{M} = \{1, \ldots, n\}$*. Außerdem ist* $\#\mathcal{N} + \#\mathcal{M} = n$*. Es muss aber* $\#\mathcal{N} \leq n - l$ *sein, sonst hätte* A'' *mehr als* $n - l$ *Eigenwerte, und* $\#\mathcal{M} \leq l$*, denn sonst hätte* A' *mehr als* l *Eigenwerte. Insgesamt folgt also* $\#\mathcal{N} = n - l$ *und* $\#\mathcal{M} = l$ *und damit die Aussage.*

Shift-Strategien zur Konvergenzbeschleunigung

Ziel: Konvergenzverbesserung des QR-Verfahrens.

Lemma 2.42 *Sei* $A \in \mathbb{R}^{n \times n}$ *eine unreduzierte Hessenberg-Matrix und* $\lambda \in \mathbb{R}$ *Eigenwert von* A*. Sei* $QR = A - \lambda I$ *und* $\tilde{A} = RQ + \lambda I$*.*
Dan gilt:

$$\tilde{A} = RQ + \lambda I = Q^T(A - \lambda I)Q + \lambda I = Q^T A Q$$

und

$$(\tilde{A})_{n,j} = 0 \text{ für } j = 1, \ldots, n-1, (\tilde{A})_{n,n} = \lambda$$

Beweis 2.42.1 *Sei* $\bar{A} := A - \lambda I = [\bar{a}_i, \ldots, \bar{a}_n]$*. Da* A *unreduziert ist, muss dies auch für* $\bar{A}$ *gelten und damit sind* $\bar{a}_1, \ldots, \bar{a}_{n-1}$ *linear unabhängig und deswegen* $r_{ii} \neq 0$ *für* $i = 1, \ldots, n-1$*.*
Andererseits ist $\bar{A}$ *nicht regulär, also muss* $r_{nn} = 0$ *sein und damit* $(RQ)_{nj} = 0$ *für* $j = 1, \ldots, n$*.* $\tilde{A} = RQ + \lambda I$ *erfüllt also die Bedingung aus dem Satz.*

Die Idee ist jetzt: Verwende Näherung an λ als Shift.
$A(k)$ sei eine iterierte aus dem QR-Verfahren, A Hessenberg. Der 2×2-Block ganz rechts unten in A habe die Form $\begin{pmatrix} * & * \\ \varepsilon & a_{nn} \end{pmatrix}$. Wir nehmen an, dass $|\varepsilon|$ deutlich kleiner als $|a_{nn}|$ ist. Wir führen jetzt einen QR-Schritt mit $\sigma = a_{nn}$ als Shift durch, d.h. $QR = A - \sigma I, \tilde{A} = RQ + \sigma I$.
Die QR-Zerlegung per Givens-Rotation liefert:

$$G_{n-1,n} \ldots G_{1,2}(A - \sigma I) = \begin{pmatrix} R_{n-2} & & * \\ & a|a| & |a|b \\ 0 & 0 & -sgn(a)b\varepsilon \end{pmatrix}$$

Berechnet man weiter $RQ = RG_{12}^T \dots G_{n-1,n}^T$, so folgt:

$$RQ = \begin{pmatrix} H_{n-2} & & * & \\ & * & * & * \\ 0 & 0 & sr_{nn} & cr_{nn} \end{pmatrix}$$

Dabei ist $sr_{nn} = \frac{-\varepsilon^2 b}{a^2+\varepsilon^2}$.
Ist also $|\varepsilon|$ deutlich kleiner als $|a|$ und b nicht zu groß, so konvergiert $(\tilde{A})_{n,n-1}$ mit $O(A_{n,n-1}^2)$.
Die Voraussetzung ist also, dass $\varepsilon << a$, wobei

$$\begin{aligned} \varepsilon &= A_{n,n-1}^{(k)} \\ a &= ((G_{n-2,n-1} \dots G_{1,2})(A - \sigma I))_{n-1,n-1} \end{aligned}$$

Es ergibt sich:

Algorithmus 2.43 (QR-Verfahren mit Shift)
$A^{(1)} = A$ sei eine unreduzierte Hessenberg-Matrix. Für $k \geq 1$ iteriere:

`if` $|(A^{(k)})_{n,n-1}| > TOL(|A_{n,n}^{(k)}| + |A_{n-1,n-1}^{(k)}|)$
  $\sigma = (A^{(k)})_{n,n}$
  $Q_k R_k = A^{(k)} - \sigma\ I$
  $A^{(k+1)} = R_k\ Q_k + \sigma\ I$
`else`
  `Streiche letzte Zeile u. Spalte, erhalte`
  $\tilde{A}^{(k)} \in \mathbb{R}^{(n-1)\times(n-1)}$
  $A^{(k)} = \tilde{A}^{(k)}$

Bemerkungen

i) Falls ε nicht hinreichend klein ist, betrachte die beiden Eigenwerte σ, τ des rechten unteren 2×2-Blocks von $A^{(k)}$ und führe einen QR-Doppelschritt durch, jeweils mit Shift σ und dann τ. Falls σ und τ nicht reell sind, kann die Berechnung so organisiert werden, dass alles reell bleibt (vgl. Werner II).

ii) Algorithmus 2.38 lässt sich durch zwei Zeilen so verändern, dass Shifts berücksichtigt werden (vgl. Werner II).

2.5. Eigenwertprobleme für symmetrische Matrizen

Wir betrachten nun eine symmetrische Matrix $A \in \mathbb{R}^{n\times n}$. Es existiert dann immer ein $U \in \mathbb{R}^{n\times n}$, U orthogonal mit $A = U^T D U, D = diag(\lambda_i), \lambda_i \in \mathbb{R}$. Dabei ist $U = [u_1, \dots, u_n]$ mit $\langle u_i, u_j \rangle = \delta_{ij}$ eine Orthonormalbasis aus Eigenvektoren.

Der Algorithmus 2.30 liefert jetzt, dass Q^TAQ Hessenberg ist, und weil $(Q^TAQ)^T = Q^TA^TQ = Q^TAQ$ Hessenberg ist, muss also Q^TAQ tridiagonal sein.
Außerdem ist mit $QR = A$ und $\tilde{A} = RQ = Q^TAQ$ bei symmetrischem, tridiagonalen A die Matrix $\tilde{A}$ wieder tridiagonal.
Der QR-Algorithmus vereinfacht sich also erheblich.

Für den Fall, dass A groß, aber schwach besetzt ist, d.h. $n >> 0, \#\{(i,j) : A_{ij} \neq 0\} << n^2$ (typisch für die Diskretisierung partieller Differentialgleichungen), ist die Reduktion auf Tridiagonalgestalt oft zu teuer ($O(n^3)$ Operationen).
In diesem Fall hilft der Ansatz: Bestimmung der λ_i über den Rayleigh-Quotienten $r(x) = \frac{\langle x,Ax\rangle}{\langle x,x\rangle} (x \in \mathbb{R}^n \backslash \{0\})$.
Es ist $\lambda_{min} = \min\limits_{x\neq 0} r(x)$ und $\lambda_{max} = \max\limits_{x\neq 0} r(x)$ (folgt z.B. aus Satz 2.14 oder 2.15).
Wir approximieren nun z.B. λ_{max} durch

$$\lambda_{max}^{(k)} = \max_{y\in V_k\backslash\{0\}} r(y)$$

wobei $V_k = span\{x_0, Ax_0, \ldots, A^{k-1}x_0\}$ mit einem Startvektor $x_0 \neq 0$ ist (Krylov-Raum).

λ_{max} ergibt sich nun als größter Eigenwert von $Q_k^TAQ_k \in \mathbb{R}^{k\times k}$ mit $Q_k = [q_1, \ldots, q_n] \in \mathbb{R}^{n\times k}, \langle q_i, q_j\rangle = \delta_{ij}$ und $V_k = span\{q_1, \ldots, q_k\}$.
Die q_i können leicht rekursiv berechnet weren und $Q_k^TAQ_k$ ist tridiagonal:

$$Q_k^TAQ_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \end{pmatrix}$$

Die Bestimmung der q_i sowie der α_i und β_i erfolgt mit dem

Algorithmus 2.44 (Lanczos-Verfahren) *Setze $q_0 = 0, q_1 = \frac{x_0}{\|x_0\|_2}$ und iteriere dann:*

$$\begin{aligned} \alpha_k &= \langle q_k, Aq_k\rangle \\ \omega_{k+1} &= Aq_k - \alpha_k q_k - \beta_k q_{k-1} \\ \beta_{k+1} &= \|\omega_k\| \\ q_{k+1} &= \frac{1}{\beta_{k+1}}\omega_{k+1}, \text{ falls } \beta_{k+1} \neq 0, \text{ sonst Abbruch} \end{aligned}$$

β_1 kann dabei beliebig gewählt sein, da $q_0 = 0$ gilt.

3. Gewöhnliche Differentialgleichungen

3.1. Einleitung / Beispiele

i) Bakterienwachstum
Wir betrachten eine Population von Bakterien, die Anzahl der Individuen der Population zum Zeitpunkt t sei $n(t)$.

0. Modell
Die Wahrscheinlichkeit der Zellteilung im Zeitintervall $[t, t+\Delta t]$ ist proportional zur Länge des Intervalls: $p\Delta t$.
Dies führt auf das Modell:

$$\underbrace{n(t+\Delta t) - n(t)}_{\Delta n} = p\Delta t n(t)$$

Betrachtet man $\Delta t \to 0$ und wählt n so gross, dass nach einer evtl. Normierung n als „kontinuierlich“ angenommen werden kann, so folgt:

$$\frac{n(t+\Delta t) - n(t)}{\Delta t} = pn(t) \overset{\Delta t \to 0}{\to} \dot{n}(t) = pn(t), \dot{n} := \frac{\partial n}{\partial t}$$

Die Lösung ist dann $n(t) = n(0)e^{pt}$. Damit wächst $n(t)$ exponentiell, also auch unbeschränkt.

1. Modell
Bakterien sterben u.a. aufgrund mangelnden Nahrungsangebots. Die Wahrscheinlichkeit des Sterbens im Zeitintevall Δt sei $kn(t)\Delta t$, also:

$$\Delta n(t) = p\Delta t n(t) - kn(t)\Delta t n(t) \overset{\Delta t \to 0}{\to} \dot{n}(t) = pn(t) - kn^2(t) = n(t)(p - kn(t))$$

(*Logistische Gleichung*) Die einzigen *stationären* Lösungen, d.h. solche mit $\dot{n}(t) = 0$ sing gegeben durch $pn = kn^2$, also entweder $n = 0$ oder $n = \frac{p}{k}$ für $p, k > 0$.

ii) Räuber Beute-Modell
Wir betrachten das Vorkommen von Luchsen und Schneehasen in einem bestimmten Gebiet und nehmen an, dass die Population nicht von weiteren Einflüssen abhängt. Sei $u(t)$ die Anzahl der Hasen zum Zeitpunkt t und $v(t)$ die Anzahl der Luchse zum Zeitpunkt t.

Setzen wir alle Konstanten auf 1, so gilt dann:

$$\dot{u}(t) = u(t) - u(t)v(t)$$
$$\dot{v}(t) = u(t)v(t) - v(t)$$

iii) Newton'sche Bewegungsgleichung
Wir beschreiben die Bewegung einer „Punktmasse“ im Kraftfeld einer Kraft F. $x(t) \in \mathbb{R}^3$ beschreibe den Ort des Massenpunktes zum Zeitpunkt t.

$\dot{x}(t) = \frac{\partial x}{\partial t}(t)$ ist dann die Geschwindigkeit und

$\ddot{x}(t)$ die Beschleunigung.

Nach Newton gilt jetzt: $m\ddot{x}(t) = F(t, x(t))$.

Spezialfälle:

a) $F \equiv 0$
Hieraus folgt $\ddot{x}(t) = 0$ und damit $\dot{x} = const.$

b) F ist das Gravitationsfeld einer Masse M im Punkt 0, $M \gg m$
Es ist $F(t, x(t)) = -\frac{GmM}{\|x\|^2}\frac{x}{\|x\|}$, wobei $\|\cdot\|$ die euklidische Norm und G die Gravitationskonstante ist ($G = 6,67 \cdot 10^{-11}\frac{Nm^2}{kg^2}$). Es folgt:

$$\ddot{x} = -\frac{GM}{\|x\|^2}\frac{x}{\|x\|}$$

Laut Kepler sind periodische Lösungen dieser Gleichung Ellipsen, die in einer Ebene liegen.

c) Harmonischer Oszillator
Wir betrachten ein an einer Feder aufgehängtes Gewicht. $x(t)$ beschreibe die Auslenkung des Gewichts zum Zeitpunkt t ($x(t) = 0$ sei die Ruhelage). Die Beschleunigung des Gewichts ist dann $\ddot{x}$. Die rücktreibende Federkraft ist proportional zur Auslenkung $x(t)$, also gilt (mit einer Federkonstante D):

$$m\ddot{x} = F(x) = -Dx \Rightarrow \ddot{x} = -\frac{D}{m}x$$

Die Lösung ist in diesem Fall $x = A\sin(\sqrt{\frac{D}{m}}t) + B\cos(\sqrt{\frac{D}{m}}t)$ mit zu bestimmenden Konstanten A, B (freie Parameter). Sie werden durch Anfangswerte bestimmt: $x(0), \dot{x}(0)$.

Allgemein gilt: $m\ddot{x} = F(t, x)$. Falls $F(t, x) = F(x) = -\nabla J(x)$ mit $J : \mathbb{R}^3 \to \mathbb{R}$, so heißt die Kraft konservativ und J das Potential.

Wir definieren die *mechanische Gesamtenergie*:

$$E = \underbrace{\frac{1}{2}m\|\dot{x}\|^2}_{\text{kinetische Energie}} + \underbrace{J(x)}_{\text{potentielle Energie}}$$

Falls F konservativ ist, gilt:

$$\begin{aligned}\frac{\partial}{\partial t}E &= m\langle\dot{x},\ddot{x}\rangle + \langle\nabla J,\dot{x}\rangle \\ &= m\langle\dot{x},\ddot{x}\rangle + \langle -F(x),\dot{x}\rangle \\ &= \langle\dot{x},\underbrace{m\ddot{x} - F(x)}_{=0}\rangle = 0\end{aligned}$$

Die Gesamtenergie ändert sich also nicht!

Zu den Beispielen:

a) $F \equiv 0 \Rightarrow F$ ist konservativ.

b) $F = -\frac{GmM}{\|x\|^2}\frac{x}{\|x\|}$
$J(x) = \frac{GmM}{\|x\|}$
$\nabla J = -F$

c) $F(x) = -Dx$
$J(x) = \frac{1}{2}D\|x\|^2$

Definition 3.1 *Sei $\emptyset \neq I \subset \mathbb{R}$ ein Intervall, $t_0 \in I, \Omega \subset R^d$ offen, $y_0 \in \Omega$ und $f : I \times \Omega \to \mathbb{R}^d$ stetig.*

Das Problem: Finde $y \in C^1(I,\omega), y = \begin{pmatrix} y_1 \\ \vdots \\ y_d \end{pmatrix}$ mit

$$\begin{aligned}\dot{y}(t) &= f(t,y(t)) \text{ für alle } t \in I \\ y(t_0) &= y_0\end{aligned}$$

heißt Anfangswertproblem (AWP) eines Systems von Differentialgleichungen 1.Ordnung, d.h. die Ableitungen von y treten nur zur Ordnung 1 auf und es ist $y(t) \in \mathbb{R}^d$ (daher System).

Zu den Beispielen:

i) $\dot{n} = n - n^2$
$d = 1, f(t,n) = n - n^2$

ii) Räuber-Beute-Modell
$d = 2, y = \begin{pmatrix} u \\ v \end{pmatrix}, \dot{y} = \begin{pmatrix} u - uv \\ uv - v \end{pmatrix} = \begin{pmatrix} y_1 - y_1y_2 \\ y_1y_2 - y_2 \end{pmatrix} = F(y)$

iii) $\ddot{x} = \frac{1}{m}F(t,x)$ ist eine Differentialgleichung 2. Ordnung.

Definition 3.2 *Sei* $\emptyset \neq I \subset \mathbb{R}$ *ein Intervall,* $t_0 \in I, m \in \mathbb{N}, y_0^{(0)}, \ldots, y_0^{(m-1)} \in \mathbb{R}, f : I \times \mathbb{R}^m \to \mathbb{R}$.
Das Problem: Finde ein $y \in C^m(I, \mathbb{R})$, *so dass* $y^{(m)}(t) = f(t, y(t), \dot{y}(t), \ldots, y^{(m-1)}(t))$ *für alle* $t \in I$ *und* $y^{(j)}(t) = y_0^{(j)}$ *für alle* $j = 0, \ldots, m-1$, *heißt ein AWP für eine Differentialgleichung m-ter Ordnung.*
Die Übertragung auf Systeme, d.h. $y(t) \in \mathbb{R}^d$, *geschieht analog.*

Beispiel: $\ddot{x} = F(x)$
$m = 2, f(t, y, \dot{y}) = F(x), y(t) = x(t)$
$y(0) = x(0) = x_0$
$\dot{y}(0) = \dot{x}(0)$ vorgeschrieben.

Satz 3.3 *Die Bezeichnungen seien wie in der Definition gewählt. Dann ist das AWP für eine DGL m-ter Ordnung äquivalent zu einem AWP für ein System 1. Ordnung.*
Genauer: Sei $y(t)$ *Lösung des AWPs m-ter Ordnung, dann ist* $z(t)$ *mit*

$$\begin{aligned} z_1(t) &= y(t) \\ z_2(t) &= \dot{y}(t) \\ &\vdots \\ z_m(t) &= y^{(m-1)}(t) \end{aligned}$$

Lösung der DGL mit $z_0 = \begin{pmatrix} y_0^{(0)} \\ \vdots \\ y_0^{(m-1)} \end{pmatrix}$.

Beweis 3.3.1 *Definiere* $\tilde{f} : I \times \mathbb{R}^m \to \mathbb{R}^m$ *mit*

$$\begin{aligned} \tilde{f}_1(t, z) &= z_2 \\ \tilde{f}_2(t, z) &= z_3 \\ &\vdots \\ \tilde{f}_{m-1}(t, z) &= z_m \\ \tilde{f}_m(t, z) &= f(t, z_1, \ldots, z_m) \end{aligned}$$

Damit gilt:

$$\dot{z} = \begin{pmatrix} \dot{y} \\ \ddot{y} \\ \vdots \\ y^{(m)} \end{pmatrix} = \begin{pmatrix} z_2 \\ \vdots \\ z_m \\ f(t, z_1, \ldots, z_m) \end{pmatrix} = \tilde{f}(t, z)$$

Auch die Anfangswerte sind gerade erfüllt.

3.2. Theorie

Satz 3.4 (Picard-Lindelöf)
$I = [t_0 - a, t_0 + a]$ sei ein Intervall, $a > 0, y_0 \in \mathbb{R}^d, \Omega = B_r(y_0), r > 0$ und $f : I \times \Omega \to \mathbb{R}^d$ stetig und Lipschit-stetig im 2. Argument, d.h. $\exists L \geq 0 : \|f(t, y_1) - f(t, y_2)\| \leq L\|y_1 - y_2\| \forall y_1, y_2 \in \Omega, t \in I$.
Dann existiert genau eine Lösung $y \in C^1(J, \Omega)$ des AWPs $\dot{y}(t) = f(t, y(t)) \forall t \in J, y(t_0) = y_0$. Dabei ist $J = (t_0 - \alpha, t_0 + \alpha) \subset I$ mit $\alpha = \min\{a, \frac{r}{M_f}\}, M_f = \sup_{(t,y)} \|f(t, y)\|$.

Beweis 3.4.1 *In allen Büchern über gewöhnliche Differentialgleichungen.*

Bemerkung 3.5

i) Die Einschränkung des Intervalls I auf J kann notwendig sein, z.B. existiert für $\dot{y} = y^2, y(0) = y_0$ nur eine Lösung bis zu einem t^*.

ii) Der Beweis benutzt, dass das AWP äquivalent ist zur Volterra-Gleichung:

$$y(t) = y_0 + \int_{t_0}^{t} f(s, y(s))ds$$

Die Volterra-Gleichung wird als Fixpunktgleichung in $C(\bar{J}, \mathbb{R}^d)$ mit $\|\cdot\|_\infty$ betrachtet und mit dem Banach'schen Fixpunktsatz gelöst.

iii) Hinreichend für die Lipschitz-Stetigkeit im zweiten Argument ist: Die $\frac{\partial f_i}{\partial y_j}$ existieren und sind auf $I \times \Omega$ beschränkt.

iv) (Lokale) Existenz und Eindeutigkeit gilt auch, falls f nur lokal Lipschitz-stetig ist, d.h. $\forall (\bar{t}, \bar{y}) \in I \times \Omega \exists U = U(\bar{t}, \bar{y}), L = L(\bar{t}, \bar{y}) : \|f(t_1, y_1) - f(t_2, y_2)\| \leq L\|y_1 - y_2\| \forall (t_1, y_1), (t_2, y_2) \in U$.

v) Falls $f \in C^k(I \times \Omega)$, ist $y \in C^{k+1}(J)$.
Beweisidee (für $d = 1$):
$\dot{y} = f(t, y(t)) \Rightarrow \frac{d}{dt} f(t, y(t)) = \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t))\dot{y}(t) \in C$, falls $f \in C^1$
Also: $\dot{y} \in C^1 \iff y \in C^2$ etc.

vi) Weiterer Existenzsatz: Peano

f braucht nur stetig zu sein, dies liefert aber nur Existenz und keine Eindeutigkeit. Z.B. ist $f(y) = \sqrt{|y|}$ *lösbar durch* $y(t) = \begin{cases} 0 & \text{für } 0 \leq t \leq t^* \\ \frac{1}{4}(t-t^*)^2 & \text{für } t \geq t^* \end{cases}$.

Dabei ist $t^* \geq 0$ *beliebig!*

Definition 3.6 *I sei ein Intervall,* $\Omega \subset \mathbb{R}^s$ *offen und* $f : I \times \Omega \to \mathbb{R}^d$*. Das AWP*

$$\dot{y}(t) = f(t, y(t)), y(t_0) = y_0$$

habe für alle $t_0 \in I, y_0 \in \Omega$ *eine eindeutige Lösung auf ganz I. Dann bezeichne zu* $y_0 \in \Omega$

$$\Phi^{t,t_0} y_0 = y(t)$$

diese Lösung. Φ *heißt kontinuierlicher Fluss, Evolution.*

Satz 3.7 (Lemma von Gronwall)
Sei $I = [t_0, t_0 + a], z, h : I \to \mathbb{R}$ *stetig,* $z(s), h(s) \geq 0$*. Sei* $\alpha \geq 0$ *und z erfülle die folgende Ungleichung:*

$$0 \leq z(t) \leq \alpha + \int_{t_0}^{t} h(s)z(s)ds \forall t \in I$$

Dann gilt: $z(t) \leq \alpha \exp(\int_{t_0}^{t} h(s)ds)$

Beweis 3.7.1
Nehmen wir zunächst an, dass $\alpha > 0$ *ist. Definiere* $\omega(t) := \alpha + \int_{t_0}^{t} h(s)z(s)ds$*.*
Nun ist $0 \leq z(t) \leq \omega(t)$ *und* $\dot{\omega}(t) = h(t)z(t), \omega(t) \geq \alpha > 0$ *und damit* $\dot{\omega}(t) = h(t)z(t) \leq h(t)\omega(t)$*. Es folt:*
$\frac{\dot{\omega}(t)}{\omega(t)} \leq h(t)$*, also* $\frac{d}{dt} ln(\omega(t)) \leq h(t)$ *und damit* $ln(\omega(t)) - ln(\omega(t_0)) \leq \int_{t_0}^{t} h(s)ds$*.*
Es folgt $\frac{\omega(t)}{\omega(t_0)} \leq \exp(\int_{t_0}^{t} h(s)ds)$ *und wegen* $\omega(t_0) = \alpha$ *folgt:* $z(t) \leq \alpha \exp(\int_{t_0}^{t} h(s)ds)$*.*
Ist nun $\alpha = 0$*, so gilt obige Ungleichung für ein entsprechend definiertes* ω *mit* $\bar{\alpha} > 0$*. Durch Grenzübergang* $\bar{\alpha} \to 0$ *folgt dann der Beweis.*

Satz 3.8 (Diskrete Gronwall-Ungleichung)
Gegeben sei $z_k, a_k \geq 0, k = 0, \ldots, n, \alpha \geq 0$*. Es sei* $z_k \leq \alpha + \sum_{j=0}^{k-1} a_j z_j$ *für* $k = 1, \ldots, n$*.*
Dann gilt: $z_k \leq (\alpha + a_0 z_0) \exp(\sum_{j=1}^{k-1} a_j)$ *für* $k = 1, \ldots, n$*.*

Beweis 3.8.1
Wir definieren $\omega_k := \alpha + \sum_{j=0}^{k-1} a_j z_j$*.*
Dann ist $\omega_{k+1} - \omega_k = a_k z_k \leq a_k \omega_k$ *und es folt* $\omega_{k+1} \leq (1 + a_k)\omega_k$*.*
Induktiv folgt dann:
$\omega_k \leq (1 + a_{k-1}) \ldots (1 + a_k)\omega_1 \leq \omega_1 \prod_{j=1}^{k-1} \exp(a_j) = \omega_1 \exp(\sum_{j=1}^{k-1} a_j)$
und damit: $z_k \leq \omega_k \leq \omega_1 \exp(\sum_{j=1}^{k-1} a_j) = (\alpha + a_0 z_0) \exp(\sum_{j=1}^{k-1} a_j)$*.*

Korollar 3.9 (Stetige Abhängigkeit von den Anfangswerten)
$f : I \times \Omega \to \mathbb{R}^d$ sei stetig und Lipschitz-stetig im zweiten Argument, $t_0 \in I$. Für $z_1, z_1 \in \Omega$ seien Φ_{z_1}, Φ_{z_2} die eindeutigen Lösungen in I.
Dann ist $\|\Phi^{t,t_0} z_1 - \Phi^{t,t_0} z_2\| \leq \exp(L(t-t_0))\|z_1 - z_2\| \forall t \in I$.

Beweis 3.9.1 *Für $i = 1, 2$ ist $\Phi^{t,t_0} z_i = z_i + \int_{t_0}^{t} f(s, \Phi^{s,t_0} z_i) ds$.*
Es folgt:

$$\begin{aligned}
\|\Phi^{t,t_0} z_1 - \Phi^{t,t_0} z_2\| &= \|z_1 - z_2 + \int_{t_0}^{t} f(s, \Phi^{s,t_0} z_1) - f(s, \Phi^{s,t_0} z_2) ds\| \\
&\leq \|z_1 - z_2\| + \int_{t_0}^{t} \|f(s, \Phi^{s,t_0} z_1) - f(s, \Phi^{s,t_0} z_2)\| ds \\
&\leq \|z_1 - z_2\| + \int_{t_0}^{t} L\|\Phi^{s,t_0} z_1 - \Phi^{s,t_0} z_2\| ds
\end{aligned}$$

Die Anwendung des Lemma von Gronwall mit $z(t) = \|\Phi^{t,t_0} z_1 - \Phi^{t,t_0} z_2\|, h(t) = L, \alpha = \|z_1 - z_2\|$ liefert jetzt den Beweis.

3.3. Stabilität von Fixpunkten

Definition 3.10 *Eine Differentialgleichung $\dot{y}(t) = f(t, y(t))$ heißt* autonom, *falls $f(t, y(t)) = f(y(t))$.*

Bemerkung 3.11 *i) DGL Autonom $\Rightarrow \Phi^{t,t_0}$ hängt nur noch von $t - t_0$ ab.*

ii) Ist eine DGL nicht autonom, so kann man zu $\dot{y} = f(t, y(t))$ wie folgt ein z definieren:
$z(t) := \begin{pmatrix} y(t) \\ t \end{pmatrix} \in \mathbb{R}^{d+1}, F(z) = \begin{pmatrix} f(t,y) \\ 1 \end{pmatrix}$
$\dot{z}(t) = F(z)$ ist dann autonom.

Definition 3.12 (Fixpunkt)
Gegeben eine autonome Differentialgleichung $\dot{y} = F(y)$. $y^* \in \mathbb{R}^d$ heißt Fixpunkt *der zugehörigen Evolution Φ, falls $\Phi^{t,t_0} y^* = y^*$ für alle t, t_0, d.h. $f(y^*) = 0$.*

Definition 3.13
Gegeben eine autonome Differentialgleichung $\dot{y} = f(y)$ mit Fixpunkt $y^* \in \mathbb{R}^d$.

i) y^* heißt stabiler *Fixpunkt, falls für alle $\varepsilon > 0$ ein $\delta > 0$ existiert, so dass für alle y_0 mit $\|y^* - y_0\| \leq \delta$ gilt: Φ^{t,t_0} existiert für alle $t > 0$ und $\|\Phi^t y_0 - y^*\| \leq \varepsilon \forall t \geq 0$.*

ii) y^* *heißt* asymptotisch *stabiler Fixpunkt, falls es stabil ist und außerdem ein* δ_0 *existiert, so dass* $\lim_{t\to\infty} \|\Phi^t y_0 - y^*\| = 0 \forall y_0, \|y_0 - y^*\| \leq \delta_0$.

iii) y^* *heißt instabil, falls es nicht stabil ist.*

Beispiel: $\dot{y} = \lambda y, \lambda \in \mathbb{C}, y(0) = y_0$
Die Lösung ist hier $y(t) = y_0 e^{\lambda t} = y_0 e^{tRe\lambda} e^{tiIm\lambda}$.
Der einzige Fixpunkt ist $y^* = 0$.
Es ist $|y(t)| = |y_0 e^{t\lambda}| = |y_0| e^{tRe\lambda}$. Also gilt:

stabil	$Re\lambda \leq 0$
y^* ist asymptotisch stabil, falls	$Re\lambda < 0$
instabil	$Re\lambda > 0$

Zur Erinnerung: $A \in \mathbb{R}^{d\times d}, A^j \in \mathbb{R}^{d\times d}$
Sei $p \in \mathcal{P}_k, p(x) = \sum_{j=0}^k a_j x^j$. Dann ist $p(A) := \sum_{j=0}^k a_j A^j \in \mathbb{R}^{d\times d}$.

Definition und Satz 3.14
Sei $A \in \mathbb{R}^{d\times d}$. *Dann konvergiert die Reihe* $\exp(A) := e^A := \sum_{j=0}^\infty \frac{1}{j!} A^j$ *in dem Sinne, dass* $S_N(A) = \sum_{j=0}^N \frac{1}{j!} A^j$ *einen Grenzwert besitzt.*
Falls $\|\cdot\|$ *sublmultiplikativ auf* $\mathbb{R}^{d\times d}$ *ist, so gilt:*

$$\|\exp(A)\| \leq \exp(\|A\|)$$

Beweis 3.14.1 *Die Konvergenz ist unabhängig von der Wahl der Norm, da* $dim(\mathbb{R}^{d\times d}) < \infty$. *Wähle also* $\|\cdot\|$ *submultiplikativ.*
Für $M > N$ *gilt jetzt:* $\|S_N(A) - S_M(A)\| = \|\sum_{j=N+1}^M \frac{1}{j!} A^j\| \leq \sum_{j=N+1}^M \frac{1}{j!} \|A\|^j \to 0$ *für* $N, M \to \infty$, *da* $\exp(x)$ *für jedes* $x \in \mathbb{R}$ *konvergiert.*
Also ist $S_N(A)$ *eine Cauchy-Folge auf* $\mathbb{R}^{d\times d}$. *Daher existiert* $\lim_{N\to\infty} S_N(A) =: \exp(A)$.
Außerdem folgt wegen des Grenzübergangs $N \to \infty$ *aus der Ungleichung:* $\|\exp(A)\| \leq \exp(\|A\|)$.

Satz 3.15 (Eigenschaften von exp**)**
Sei $A \in \mathbb{R}^{d\times d}$. *Dann gilt:*

i) $t \mapsto \exp(tA)$ *ist differenzierbar auf* $\mathbb{R}$ *und* $\frac{d}{dt}\exp(ta) = A\exp(tA) = \exp(tA)A$.

ii) $y(t) = \exp(tA)y_0$ *ist die eindeutige Lösung des Anfangswertproblems* $\dot{y} = Ay, y(0) = y_0$.

iii) Ist $f \in C(\mathbb{R}, \mathbb{R}^d)$, *so ist* $y(t) = \exp(tA)y_0 + \int_a^t \exp((t-s)A) f(s) ds$ *die eindeutige Lösung des Anfangswertproblems* $\dot{y}(t) = Ay(t) + f(t)$.

iv) *Falls* $A, B \in \mathbb{R}^{d\times d}$ *mit* $AB = BA$, *so gilt* $\exp(A)\exp(B) = \exp(A+B)$. *Insbesondere gilt:* $\exp(A) \in GL(d)$ *für alle* $A \in \mathbb{R}^{d\times d}$.

v) *Für alle regulären* $S \in GL(d)$ *gilt:* $\exp(SAS^{-1}) = S\exp(A)S^{-1}$

vi) *Für* $\Lambda \in \mathbb{R}^{d\times d}$ *mit* $\Lambda = \begin{pmatrix} \Lambda_1 & & 0 \\ & \ddots & \\ 0 & & \Lambda_s \end{pmatrix}, \Lambda_i \in \mathbb{R}^{d_i\times d_i}, \sum_{i=1}^{s} d_i = d$ *gilt:*

$$\exp(\Lambda) = \begin{pmatrix} \exp(\Lambda_1) & & \\ & \ddots & \\ & & \exp(\Lambda_s) \end{pmatrix}$$

Beweis 3.15.1

i) *Definiere:* $\varphi_N(t) := S_N(tA)$. *Dann gilt* $\varphi_N(t) \to \exp(tA)$ *bei* $N \to \infty$ *punktweise für alle* t. *Außerdem gilt:*

$$\begin{aligned}
\varphi_N'(t) &= (\sum_{j=0}^{N} \frac{1}{j!}(tA)^j)' \\
&= \sum_{j=0}^{N} \frac{1}{j!}(t^j A^j)' \\
&= \sum_{j=1}^{N} \frac{1}{j!} j t^{j-1} A^j \\
&= A \sum_{j=0}^{N-1} \frac{1}{j!}(tA)^{j-1} \\
&= AS_{N-1}(tA) \\
&= S_{N-1}(tA)A
\end{aligned}$$

Ohne Einschränkung sei $\|\cdot\|$ *submultiplikativ. Dann folt weiter:*
$\|\varphi_N'(t) - A\exp(tA)\| = \|A\sum_{j=N}^{\infty} \frac{1}{j!}(tA)^j\| \le \|A\| \sum_{j=N}^{\infty} \frac{1}{j!}|t|^j\|A\|^j$.
Nun konvergiert aber $\exp(x)$ *lokal gleichmäßig in* $x \in \mathbb{R}$, *also konvergiert auch* $\|\varphi_N'(t) - A\exp(tA)\|$ *lokal gleichmäßig.*
Es gilt also

$$\begin{aligned}
\varphi_N(t) &\overset{N\to\infty}{\to} \exp(tA) && \text{punktweise in } t \\
\varphi_N'(t) &\overset{N\to\infty}{\to} A\exp(tA) && \text{lokal gleichmäßig}
\end{aligned}$$

Also ist $\varphi_\infty(t) := \lim_{N\to\infty}$ *differenzierbar und* $\varphi'_\infty = \lim_{N\to\infty} \varphi'_N(t)$.
Wegen $\exp(tA) = \varphi_\infty(t)$ *folgt jetzt* $\exp(tA)' = A\exp(tA)$ *und ebenso* $\exp(tA)' = \exp(tA)A$.

ii) und

iii) sind klar durch Nachrechnen, mit Anwendung von i).

iv) Wir zeigen zunächst: $\exp(A)\exp(-A) = id$. *Definiere dazu:* $\varphi(t) = \exp(tA)\exp(-tA)$.
Klar ist: $\varphi(0) = id$. *Nun gilt aber:*

$$\begin{aligned}\varphi'(t) &= (\exp(tA))'\exp(-tA) + \exp(tA)(\exp(-tA))' \\ &= A\exp(tA)\exp(-tA) + A\exp(tA)(-\exp(-tA)) \\ &= A(\exp(tA)\exp(-tA) - \exp(tA)\exp(-tA) \\ &= 0\end{aligned}$$

und damit ist $\varphi(t) = id$ *für alle* t *und bei* $t = 1$ *folgt* $\exp(A)\exp(-A) = id$.
Definiere jetzt $\psi(t) = \exp(tA)\exp(tB)\exp(-t(A+B))$. *Wieder ist* $\psi(0) = id$ *und es ergibt sich:* $\psi'(t) = \exp(tA)\exp(tB)\exp(-t(A+B))(A+B-(A+B)) = 0$. *Wie eben folgt also* $\exp(A)\exp(B) = \exp(A+B)$.

v)

$$\begin{aligned}\exp(SAS^{-1}) &= \sum_{j=0}^{\infty} \frac{1}{j!}(SAS^{-1})^j \\ &= \sum_{j=0}^{\infty} \frac{1}{j!} SA^jS^{-1} \\ &= S\exp(A)S^{-1}\end{aligned}$$

vi) $\Lambda^j = \begin{pmatrix}\Lambda_1 & & \\ & \ddots & \\ & & \Lambda_s\end{pmatrix}^j = \begin{pmatrix}\Lambda_1^j & & \\ & \ddots & \\ & & \Lambda_s^j\end{pmatrix}$

Erinnerung: Jordan'sche Normalform Sei $A \in \mathbb{C}^{d\times d}$, dann existiert ein $S \in GL(d,\mathbb{C}), J \in \mathbb{C}^{d\times d}$ mit $A = S^{-1}JS, J = \begin{pmatrix}J_1 & & \\ & \ddots & \\ & & J_m\end{pmatrix}, J_i \in \mathbb{C}^{d_i\times d_i}, \sum_{i=1}^m d_i = d.$
Dabei ist J_i ein Jordan-Block zum Eigenwert λ_i:

$$J_i = \begin{pmatrix} \lambda_1 & 1 & & & \\ & \lambda_i & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ & & & & \lambda_i \end{pmatrix}$$

Sei nun $N_j := (J_j - \lambda_j I) \in \mathbb{C}^{d_j \times d_j}$. Dann ist N_j nilpotent. Es gilt: $N_j^k \neq 0$ für $k < d_j$ und $N_j^k = 0$ für $k \geq d_j$.

Definition 3.16
Sei $A \in \mathbb{C}^{d\times d}$.

i) $\nu(A) = \max_{\lambda\in\sigma(A)} Re\lambda$ heißt die Spektralabszisse

ii) Zu $\lambda \in \sigma(A)$ bezeichne $\iota(\lambda)$ den Index von λ, d.h. die maximale Dimension eines Jordanblocks zu λ, d.h. $\iota(\lambda) = \max\{d_j | \lambda = \lambda_j\}$.

Lemma 3.17
Sei $A \in \mathbb{R}^{d\times d}, \varepsilon > 0$.
Dann existiert ein $C_\varepsilon > 0$ mit $\|\exp(tA)\| \leq C_\varepsilon \exp(t(\nu(A) + \varepsilon))\forall t \geq 0$.
Gilt außerdem $\iota(\lambda) = 1$ für alle $\lambda \in \sigma(A)$ mit $Re\lambda = \nu(A)$, so gilt außerdem:
$\|\exp(tA)\| \leq C \exp(t\nu(A))\forall t \geq 0$.

Beweis 3.17.1
Da alle Normen auf $\mathbb{R}^d$ äquivalent sind, wähle o.E. eine submultiplikative Norm.
Sei $S \in GL(n)$ so, dass $A = S^{-1}JS$ für die Jordan-Matrix J zu A. Es gilt nun:

$$\|\exp(tA)\| = \|S^{-1}\exp(tJ)S\| \leq \|S^{-1}\|\|S\|\|\exp(tJ)\|$$

$$\exp(tJ) = \begin{pmatrix} \exp(tJ_1) & & \\ & \ddots & \\ & & \exp(tJ_m) \end{pmatrix}$$

$$\begin{aligned} \exp(tJ_i) &= \exp(t(N_j + \lambda_j I)) \\ &= \exp(t\lambda_j I)\exp(tN_j) \\ &= e^{t\lambda_j}\exp(tN_j) \\ \|\exp(tJ_j)\| &= |e^{tRe\lambda_j}|\|\exp(tN_j)\| \\ &= e^{tRe\lambda_j}\|\exp(tN_j)\|\|\|\exp(tN_j)\| \qquad \leq \sum_{n=0}^{m_j-1} \frac{1}{k!} t^k \|N_j\|^k \\ &= p(t), p \in \mathcal{P}_{m_j-1} \end{aligned}$$

Also ergibt sich:

$$\begin{aligned}\|\exp(tJ_j)\| &\leq e^{tRe\lambda_j}p(t)\\ &\leq C_\varepsilon e^{tRe\lambda_j}e^{t\varepsilon}\\ &\leq C_\varepsilon e^{t(\nu(A)+\varepsilon)}\end{aligned}$$

Zusammensetzen und eventuelles Anpassen der Norm liefert jetzt das Ergebnis.

Beweis der Zusatzbehauptung: *Wir wiederholen das Argument wie eben, falls* $Re\lambda_j < \nu(A)$*, wählen wir* $\varepsilon > 0$ *so, dass* $Re\lambda_j + \varepsilon < \nu(A)$ *ist. Dann folgt:* $e^{t(Re\lambda_j+\varepsilon)} \leq e^{t\nu(A)}$*. Für* $Re\lambda_j = \nu(A)$ *gilt n.V.* $m_j = 1$*, also* $J_j = \lambda_j$*, d.h.* $\|\exp(t\lambda_j)\| = e^{tRe\lambda_j} = e^{t\nu(A)}$*.*

Satz 3.18
Sei $0 \neq A \in \mathbb{R}^{d\times d}$*. Dann ist* $y^* = 0$ *Fixpunkt der Differentialgleichung* $\dot{y} = Ay$ *und es gilt:*

i) $y^* = 0$ *ist stabiler Fixpunkt gdw.* $\nu(A) \leq 0$ *und für alle* $\lambda_j \in \sigma(A)$ *mit* $Re\lambda_j = 0$ *gilt:* $m_j = 1$*.*

ii) $y^* = 0$ *ist asymptotisch stabil, falls* $\nu(A) < 0$*.*

Beweis 3.18.1

i) „⇐“
Für $y_0 \in \mathbb{R}^d$ *ist die Lösung des AWPs gegeben durch*

$$y(t) = \Phi^t y_0 = \exp(tA)y_0$$

Also gilt:

$$\begin{aligned}\|y(t)\| &\leq \|\exp(tA)\|\|y_0\|\\ &\leq C\exp(t\nu(A))\|y_0\|\\ &\leq C\|y_0|, \text{ da } \nu(A) \leq 0\end{aligned}$$

„⇒“
Annahme: $\nu(A) > 0$*. Wähle dann* $\lambda \in \sigma(A)$ *mit* $Re\lambda = \nu(A)$ *und den zugehörigen Eigenvektor* e*. Dann ist* $Ae = \lambda e$ *und* $\Phi^t e = e^{t\lambda}e$*, denn* $\frac{d}{dt}\underbrace{(e^{t\lambda}e)}_{v(t)} = \lambda e^{t\lambda}e =$

$A\underbrace{(e^{t\lambda}e)}_{v(t)}$.
Es folgt:

$\|v(t)\| = |e^{t\lambda}|\|e\| = e^{tRe\lambda}\|e\| \overset{t\to\infty}{\to} \infty \lightning$

Also muss $y^* = 0$ *staibler Fixpunkt sein, also* $\nu(A) \leq 0$.

Wir nehmen also jetzt an, dass $\nu(A) = 0$ *ist,* $\lambda_j \in \sigma(A)$ *mit* $Re(\lambda_j) = 0$, *aber* $m_j > 1$. *Ohne Einschränkung sei* $j = 1$. *Es folgt:*

$$A = S^{-1}JS, J = \begin{pmatrix} J_1 & & \\ & J_2 & \\ & & \ddots \end{pmatrix}, J_1 = \begin{pmatrix} \lambda_1 & 1 & \\ & \ddots & \ddots \\ & & \lambda_1 \end{pmatrix}.$$

Setze nun $y_0 := S^{-1}e_{m_j}$ *(*e_{m_j} *ist der* m_j*-te Einheitsvektor).*

Nun gilt:

$$\begin{aligned}
\Phi^t y_0 &= \exp(tA)y_0 \\
&= S^{-1}\exp(tJ)e_{m_j} \\
&= S^{-1}\begin{pmatrix} \exp(tJ_1) & & \\ & \exp(tJ_2) & \\ & & \ddots \end{pmatrix} e_{m_j} \\
&= S^{-1}\begin{pmatrix} \exp(tJ_1)\begin{pmatrix} 0 \\ \vdots \\ 1 \end{pmatrix} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\
&= S^{-1}e^{t\lambda_1}\begin{pmatrix} \exp(tN_1)\begin{pmatrix} 0 \\ \vdots \\ 1 \end{pmatrix} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\
&= S^{-1}(\sum_{k=0}^{m_j-1}\frac{1}{k!}t^k k N_1^k \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix})
\end{aligned}$$

Wähle jetzt o.E. $\|\cdot\| = \|\cdot\|_2$ *und benutze, dass* $|e^{t\lambda_1}| = 1$ *gelten muss, dann folgt:*

$$\begin{aligned}\|\exp(tJ_1)\begin{pmatrix}0\\\vdots\\1\end{pmatrix}\|_2 &= |e^{t\lambda_1}|\|\exp(tN_1)\begin{pmatrix}0\\\vdots\\1\end{pmatrix}\| \\ &\geq |(exp(tN_1)\begin{pmatrix}0\\\vdots\\1\end{pmatrix})_{m_1-2}| \\ &\geq t \\ &\overset{t\to\infty}{\to} \infty\end{aligned}$$

Also folgt $\|S^{-1}t\| \overset{t\to\infty}{\to} \infty$ *und damit* $\|\Phi^t y_0\| \overset{t\to\infty}{\to} \infty$ *↯.*

ii) Ist klar mit Satz 3.17. Wähle hier $\varepsilon > 0$ *so, dass* $\nu(A) + \varepsilon < 0$*, dann gilt:*

$$\|\Phi^t y_0\| \leq C_\varepsilon \exp(t(\nu(A) + \varepsilon)) \overset{t\to\infty}{\to} 0$$

Beispiel 3.19

i) Harmonischer Oszillator:

$$\ddot{x} = -\frac{D}{m}x, D, m > 0$$

Schreibe dies also System:

$$y = \begin{pmatrix}y_1\\y_2\end{pmatrix} = \begin{pmatrix}x\\\dot{x}\end{pmatrix}$$

$$\dot{y} = Ay = \begin{pmatrix}0 & 1\\-\frac{D}{m} & 0\end{pmatrix} y$$

Der einzige Fixpunkt ist hier $y^* = 0$*. Es ist* $\sigma(A) = \{\pm i\sqrt{\frac{D}{m}}\}, \nu(A) = 0, \iota(\lambda) = 1$*.* $y^* = 0$ *ist also stabiler Fixpunkt, aber nicht asymptotisch stabil.*

ii) Harmonischer Oszillator mit Dämpfung (einfachstes Modell):

$$\ddot{x} = -\frac{D}{m}x - \frac{\sigma}{m}\dot{x}, \sigma > 0$$

$$\dot{y} = \begin{pmatrix}0 & 1\\-\frac{D}{m} & -\frac{\sigma}{m}\end{pmatrix} y$$

$$\sigma(A) = \{-\frac{\sigma}{2m}(1 \pm \sqrt{1 - \frac{Dm}{\sigma^2}})\}$$

Nun ist entweder $1 - \frac{Dm}{\sigma^2} \geq 0$*, dann ist* $\sigma(A) \subset \mathbb{R}, \sigma(A) \subset (-\infty, 0), \nu(A) < 0$*, oder es ist* $1 - \frac{Dm}{\sigma^2} < 0$*, dann ist* $\sqrt{1 - \frac{Dm}{\sigma^2}} = \pm i\sqrt{\frac{Dm}{\sigma^2} - 1}$ *und also* $\nu(A) < 0$*.*

Insgesamt ist also $y^* = 0$ *auf jeden Fall asymptotisch stabiler Fixpunkt.*

iii) Harmonischer Oszillator „verkehrt herum“:

$$\ddot{x} = -\frac{D}{m}x$$

$$y = \underbrace{\begin{pmatrix} 0 & 1 \\ \frac{D}{m} & 0 \end{pmatrix}}_{A} y$$

$$\sigma(A) = \{\pm\sqrt{\frac{D}{m}}\}$$

$y^* = 0$ *ist also instabil (die Lösung ist* $x(t) = B_1 e^{\sqrt{\frac{D}{m}}t} + B_2 e^{-\sqrt{\frac{D}{m}}t}$*).*

Wir betrachten jetzt allgemeine autonome Systeme, also Systeme der Form $\dot{y} = f(y)$.

Satz 3.20
$f : \mathbb{R}^d \to \mathbb{R}^d$ *sei hinreichend glatt.* y^* *sei Fixpunkt, d.h.* $f(y^*) = 0$*. Es gelte* $\nu(Df(y^*)) < 0$*. Dann ist* y^* *asymptotisch stabil.*

Beweis 3.20.1 *O.E. sei* $y^* = 0$*, sonst betrachte* $\tilde{f}(y) = f(y + y^*)$*.*

$$\begin{aligned} f(y) &= \underbrace{f(0)}_{=0} + Df(0)y + R(y), \|R(y)\| = o(\|y\|) \\ &= \underbrace{Df(0)}_{A} y + R(y) \end{aligned}$$

Es folgt die Differentialgleichung $\dot{y} = Ay + R(y)$ *und mit der Variation der Konstanten folgt:*

$$\Phi^t y_0 = \exp(tA)y_0 + \int_0^t \exp((t-s)A)R(\Phi^s y_0)ds$$

Wähle jetzt $\varepsilon > 0$ *so, dass* $q = -\nu(A) - \varepsilon = -\nu(Df(0)) - \varepsilon > 0$*. Da* $\|R(y)\| = o(\|y\|)$*, existiert nun ein* $\delta > 0$*, so dass* $\|R(\tilde{y})\| \leq \frac{q}{2}\|\tilde{y}\|$ *für alle* $\tilde{y}$ *mit* $\|\tilde{y}\| \leq \delta$*.*

Wir wenden nun 3.17 an:
$\|\exp(tA)\| \leq C_\varepsilon \exp(t(\nu(A)+\varepsilon)) = c\exp(-tq)$.
Damit ist $\|\Phi^t y_0\| \leq \|\exp(tA)\|\|y_0\| + \int_0^t \|\exp(t-s)A\|\frac{q}{2}\|\Phi^s y_0\|ds$ *für alle* y_0 *mit* $\|y_0\| \leq \delta$.
Also gilt:

$$\|\Phi^t y_0\| \leq ce^{-ta}\|y_0\| + \int_0^t e^{-(t-s)q}\frac{q}{2}\|\Phi^s y_0\|ds$$

Sei nun $G(t) := e^{tq}\|\Phi^t y_0\|$. *Dann folgt:*

$$G(t) \leq cy_0 + \int_0^t G(s)\frac{q}{2}ds$$

und mit dem Lemma von Gronwall:

$$\begin{aligned} G(t) &\leq c\|y_0\|e^{\frac{tq}{2}} \\ \Rightarrow e^{tq}\|\Phi^t y_0\| &\leq c\|y_0\|e^{\frac{tq}{2}} \\ \Rightarrow \|\Phi^t y_0\| &\leq c\|y_0\|e^{\frac{-tq}{2}} \end{aligned}$$

Bemerkung:

i) δ *muss eventuell verkleinert werden, damit* $c\|y_0\| \leq \delta$ *gilt.*

ii) $\Phi^t y_0$ *existiert zunächst nur für* $t \in [0,\alpha], \alpha > 0$. *Man kann aber zeigen: Falls die Lösungen beschränkt bleiben, existieren sie für alle* $t > 0$.

Insgesamt folgt:
$y(t) = \Phi^t y_0 \overset{t\to\infty}{\to} 0$, *falls* $\|y_0\|$ *klein genug. Es handelt sich also um einen asymptotisch stabilen Fixpunkt.*

Bemerkung 3.21
Sei $f : \mathbb{R}^d \to \mathbb{R}^d$ *hinreichend glatt und* y^* *Fixpunkt,* $f(y^*) = 0$.
Falls $\nu(Df(y^*)) = 0$, *kann man* nicht *wie beim linearen Problem so einfach Aussagen über die Stabilität treffen.*
Beispiel: $f : \mathbb{R}^2 \to \mathbb{R}^2, f(y) = \begin{pmatrix} -y_2 + y_1^3 \\ y_1 \end{pmatrix}$. *Einziger Fixpunkt der DGL* $\dot{y} = f(y)$ *ist* $y^* = 0$.
$Df(y) = \begin{pmatrix} 3y_1^2 & -1 \\ 1 & 0 \end{pmatrix}, A = Df(0) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$.
Es ist $\sigma(A) = \{\pm i\}, \nu(A) = 0$. *Beide Eigenwerte haben den Index* 1, *also ist das linarisierte Problem* $\dot{y} = Ay$ *nach Satz 3.18 stabil. Das nichtlineare Problem ist aber instabil.*

3.4. Einschrittverfahren

Im folgenden sei $I = [t_0, t_0 + a], \Omega \subset \mathbb{R}^d, f : I \times \Omega \to \mathbb{R}^d$ stetig und Lipschitz-stetig im zweiten Argument. Der Fluss Φ sei wohldefiniert auf I. Wir zerlegen nun I wie folgt: $t_0 < t_2 < \cdots < t_n = t_0 + a$.
Wir betrachten das Anfangswertproblem $\dot{y} = f(t, y(t))$ mit $y(t_0) = y_0 \in \Omega$.
Ansatz:

$$\dot{y}(t_i) \approx \frac{y(t_{i+1}) - y(t_i)}{h_i}, h_i = t_{i+1} - t_i$$

Nähere jetzt $y(t_i)$ durch $u_i \approx y(t_i)$ an.
Wir definieren das *explizite Euler-Verfahren*:

$$\begin{aligned} u_0 &= y_0 \\ \underbrace{\frac{u_{i+1} - u_i}{h_i}}_{\approx \dot{y}(t_i)} &:= \underbrace{f(t_i, u_i)}_{\approx f(t_i, y(t_i))} \\ u_{i+1} &= u_i + h_i f(t_i, u_i) \end{aligned}$$

Allgemeiner kann man definieren:

Definition 3.22
Ein explizites Einschrittverfahren ist ein Verfahren zur Berechnung von u_i der Form

$$\begin{aligned} u_0 &= y_0 \\ u_{i+1} &= u_i + h_i \varphi(t_i, u_i, h_i; f) \end{aligned}$$

mit der Inkrementfunktion φ, φ stetig in den ersten drei Argumenten und Lipschitz-stetig bezüglich u_i.

Das Euler-Verfahren ist also ein explizites Einzelschrittverfahren mit $\varphi(t, u, h; f) = f(t, u)$.

Im folgenden bezeichne Ψ die *diskrete Evolution*

$$\Psi^{t+h,t} u = u + \phi(t, u, h; f)$$

Wir definieren:

Definition 3.23 (Globaler Diskretisierungsfehler)

i) Für eine gegebene Zerlegung $\Sigma = \{t_0, \dots, t_n\}$ von I heißt $(e_i)_{i=0}^n, e_i = y(t_i) - u_i = \Phi^{t,t_i} y_0 - \Psi^{t_i,t_{i+1}} u_{i+1}$ der globale Diskretisierungsfehler.

ii) Ein Verfahren heißt konvergent, falls $\|e\|_\infty := \max_{0\le i\le n} \|e_i\| \to 0$ für $h := \max_i h_i \to 0$, für hinreichend glattes f.

iii) Ein Verfahren heißt konvergent von der Ordnung $p \in \mathbb{N}$, falls $\|e\|_\infty = O(n^p)$, für hinreichend glatte f.

Definition 3.24 (Lokaler Diskretisierungsfehler)

i) Sei $t \in I, z \in \Omega$.

$$\tau(t,z,h;f) := \begin{cases} \frac{\Phi^{t+h,t}z - z}{h} - \varphi(t,z,h;f) & , \text{ falls } h > 0 \\ f(t,z) - \varphi(t,z,0;f) & , \text{ falls } h = 0 \end{cases}$$

heißt lokaler Diskretisierungsfehler.

ii) Ein Verfahren heißt konsistent, falls $\lim_{n\to\infty} \tau(t,z,h;f) = 0$.

iii) Ein Verfahren heißt konsistent der Ordnung $p \in \mathbb{N}$, falls $\tau(t,z,h;f) = O(n^p)$ für hinreichend glatte f.

Bemerkung 3.25 *i) Für den kontinuierlichen Fluss gilt: $\Phi^{t+h,t}z = \underbrace{z + h\varphi(t,z,h;f)}_{\Psi^{t+h,t}z} + h\tau(t,h,z;f)$.*

ii) τ misst, wie gross der Fehler ist, wenn man die kontinuierliche Gleichung in die diskrete einsetzt.

iii) Ein Verfahren ist konsistent gdw. $\varphi(t,z,0;f) = f(t,z)$.

Beispiel: Explizites Euler-Verfahren:

Dieses Verfahren ist konsistent der Ordnung $p = 1$. Es gilt:

$$\begin{aligned} y(s) &:= \Phi^{t+s,t}z \\ \Phi^{t+h,t}z = y(h) &= y(0) + y'(0) + \frac{1}{2}y''(\theta)h^2 \text{ mit } \theta \in [0,h] \\ &= z + f(t,z) \\ y_j''(s) &= \frac{d}{ds}y'(s)_j)\frac{d}{ds}f_j(t+s,y(s)), j = 1,\dots,d \\ &= \partial_t f(t+s,y(s)) + \sum_{k=1}^{d} \partial_{y_k} f_j(t+s,y) + f_k(t+s,y(s)) \\ &= \partial_t f(t+s,y(s)) + [Df(t+s,y(s))f(t+sy(s))]_j \end{aligned}$$

Also ergibt sich:

$$\begin{aligned}\tau(t,z,h;f) &= \frac{y(h)-z}{h} - \varphi(t,z,h;f)\\ &= f(t,z) + \frac{1}{2}h(\partial_t f(\theta,y(\theta)) + Df(\theta,y(\theta))) - f(t,z)\\ &= O(h^1)\end{aligned}$$

Lemma 3.26
Es seien $z_k, a_k, b_k \geq 0, k = 0,\ldots,n$ und z_k erfülle: $z_k \leq (q + a_{k-1})z_{k-1} + b_{k-1} (k = 1,\ldots,n)$.
Dann gilt: $z_k \leq (\sum_{j=0}^{k-1} b_j + z_0)\exp(\sum_{j=0}^{k-1} a_j), k = 1,\ldots,n.$

Beweis 3.26.1

$$\begin{aligned}z_j - z_{j-1} &\leq (1 + a_{j-1})z_{j-1} + b_{j-1} - z_{j-1}\\ &= a_j z_{j-1} + b_{j-1}\end{aligned}$$

Also:

$$\begin{aligned}z_k - z_0 &= \sum_{j=1}^{k}(z_j - z_{j-1})\\ &\leq \sum_{j=1}^{k} a_{j-1}z_{j-1} + \sum_{j=1}^{k} b_{j-1}\\ &= \sum_{j=0}^{k-1} a_j z_j + \sum_{j=0}^{k-1} b_j\end{aligned}$$

Daraus folgt:

$$z_k \leq \underbrace{z_0 + \sum_{j=0}^{k-1} b_j}_{=:\alpha} + \sum_{j=0}^{k-1} a_j z_j$$

und mit der diskreten Gronwall-Ungleichung:

$$\begin{aligned}z_k &\leq (z_k + \sum_{j=0}^{k-1} b_j + a_0 z_0)\exp(\sum_{j=1}^{k-1} a_j)\\ &\leq (z_0 + \sum_{j=0}^{k-1} b_j)\exp(\sum_{j=0}^{k-1} a_j)\end{aligned}$$

Satz 3.27 (Konvergenz des expliziten Einschrittverfahrens)
I, Ω seien so, dass für alle $t_0, t \in I, y_0 \in \Omega$ die Flüsse Φ, Ψ wohldefiniert sind.
$\Sigma = \{t_0, \dots, t_n\}$ sei eine Zerlegung von $I = [t_0, t_n]$.
Dann gilt: $\|e\|_\infty \leq (\|e_0\| + \|\tau\|_\infty (t_n - t_0)) \exp(L(t_n - t_0))$.
Dabei ist L die Lipschitz-Konstante von Φ bzgl. u und $\|\tau\|_\infty = \sup_{0 \leq i \leq n} \|\tau_i\|, \tau_i = \tau(t_i, \Phi^{t_i, t_0} y_0, h_i; f)$.

Beweis 3.27.1
Sei $y_k := y(t_k) := \Phi^{t_k, t_0} y_0$.

$$
\begin{aligned}
u_{k+1} &= u_k + h_k \varphi(t_k, u_k, h_k) \\
y_{k+1} &= y_k + h_k \varphi(t_k, y_k, h_k) + h_k \underbrace{\tau(t_k, y_k, h_k)}_{=:\tau_k} \\
e_{k+1} &= y_{k+1} - u_{k+1} = e_k + h_k(\varphi(t_k, y_k, h_k) - \varphi(t_k, u_k, h_k)) + h_k \tau_k \\
\|e_{k+1}\| &\leq \|e_k\| + h_k L \|y_k - u_k\| + h_k \|\tau_k\| \\
&= (1 + h_k L) \|e_k\| + h_k \|\tau_k\|
\end{aligned}
$$

Lemma 3.26 liefert jetzt:

$$
\begin{aligned}
\|e_k\| &\leq (\|e_0\| + \sum_{j=0}^{k-1} h_j \|\tau_j\|) \exp(\sum_{j=0}^{k-1} h_j L) \\
&= (\|e_0\| + \|\tau\|_\infty |t_n - t_0|) \exp(|t_n - t_0| L)
\end{aligned}
$$

da $\sum_{j=0}^{k-1} h_j = |t_k - t_0|$

Korollar 3.28 *Sei $e_0 = 0$. Dann folgt:*

i) Ein konsistentes Einschrittverfahren ist konvergent.

ii) Ein Einschrittverfahren der Konstistenzordnunc p ist konvergent von der Ordnung p.

Merkregel für Einschrittverfahren: Konsistenz = Konvergenz

3.5. Explizite Runge-Kutta-Verfahren

Ziel: Die Konstruktion von Verfahren von höherer Konvergenzordnung als 1.

1. Idee: Taylor-Entwicklung Es ist

$$\frac{1}{h}[\Phi^{t+h,t}z - z] = f(t,z) + \frac{h}{2}[\partial_t f(t,z) + Df(t,z)f(t,z)] + O(h^2) =: \varphi(t,z,h)$$

Denn:

$$\begin{aligned}
y(s) &:= \Phi^{t+s,t}z \\
y(0) &= z \\
\dot{y}(s) &= f(s+t, y(s)) \\
\ddot{y}(s) &= \frac{d}{ds}\dot{y} = \frac{d}{ds}f(s+t), y(s)) \\
y(h) - z &= y(h) - y(0) \\
&= \underbrace{\dot{y}(0)}_{f(t,z)} h + \frac{1}{2} \underbrace{\ddot{y}(0)}_{\partial_t f(t,z) + Df(t,z)f(t,z)} h^2 + O(h^3)
\end{aligned}$$

Definieren wir also φ wie oben, dann ist das zugehörige Verfahren von der Ordnung 2. Dieser Mechanismus funktioniert auch für beliebige $p \in \mathbb{N}$, allerdings hat er einen großen Nachteil: Die Ableitungen von f werden gebraucht.

2. Idee: Quadratur Wieder sei $y(s) := \Phi^{t+s,t}z$.
Nun gilt:

$$\begin{aligned}
\frac{1}{h}[y(h) - y(0)] &= \frac{1}{h}\int_0^h \dot{y}(s)ds \\
&= \frac{1}{h}\int_0^h f(t+s, y(s))ds \\
&= \frac{1}{h}[h\sum_{i=1}^{m} \omega_i f(t+s_i, y(s_i))] + O(h^p)
\end{aligned}$$

falls f glatt genug ist und die Quadraturformel gegeben durch ω_i, s_i von der Ordnung $p \in \mathbb{N}$.
Wir verwenden als Beispiel die Mittelpunktsformel (diese integriert lineare Funktionen exakt). Dann ist $m = 1, \omega_1 = 1$ und $s_1 = \frac{h}{2}$.
Damit folgt:

$$\frac{1}{h}[y(h) - z] = \omega_1 f(t + \frac{h}{2}, y(\frac{h}{2})) + O(h^2)$$

falls $f \in C^2$. Das Problem ist hier: $y(\frac{h}{2})$ ist unbekannt.
Wir ersetzen also $y(\frac{h}{2})$ durch eine Approximation, z.B. mit der expliziten Euler-Formel:

$$y(\frac{h}{2}) = z + \frac{h}{2} f(t,z) + O(h^2)$$

Damit folgt dann:

$$\begin{aligned} f(t+\frac{h}{2}, y(\frac{h}{2})) &= f(t+\frac{h}{2}, z+\frac{h}{2}f(t,z)+O(h^2)) \\ &= f(t+\frac{h}{2}, z+\frac{h}{2}f(t,z)) + O(h^2) \end{aligned}$$

falls f glatt genug.
Folglich haben wir:

$$\frac{1}{h}[y(h)-z] = f(t+\frac{h}{2}, z+\frac{h}{2}f(t,z)) + O(h^2)$$

und es ergibt sich ein Verfahren 2. Ordnung mit $\varphi(t,z,h) := f(t+\frac{h}{2}, z+\frac{h}{2}f(t,z))$.
Alternativ geschrieben haben wir:

$$\begin{aligned} k_1 &:= f(t,z) \\ s_2 &:= \frac{h}{2} \\ k_2 &:= f(t+s_2, z+\frac{h}{2}k_1) \\ \varphi(t,z,h) &:= \underbrace{\omega}_{=1} k_2 \end{aligned}$$

Hieraus ergibt sich ein wesentlich allgemeinerer Ansatz:

Definition 3.29 (m-stufiges, explizites Runge-Kutta-Verfahren)
Wähle $m \in \mathbb{N}, \alpha_i, \omega_i, \beta_{ij} \in \mathbb{R}, i = 1,\dots,m, j = 1,\dots,i-1$.

$$\begin{aligned} \alpha_i &= \sum_{j=1}^{i-1} \beta_{ij} \\ s_i &= \alpha_i h \end{aligned}$$

und definiere induktiv:

$$\begin{aligned} k_1 &:= f(t,z) \\ k_i &:= f(t+s_i, z+h\sum_{j=1}^{i-1}\beta_{ij}k_j) \\ \varphi(t,z,h) &:= \sum_{i=1}^{m} \omega_i k_i \end{aligned}$$

Bemerkungen:

i) Die k_i sind wohldefiniert, da die Summation nur von $j = 1$ bis $i - 1$ läuft

ii) Die k_i heißen „i-te Stufe“

iii) $z + h \sum_{j=1}^{i-1} \beta_{ij} k_j$ kann als Approximation von $y(s_i)$ interpretiert werden

iv) Der numerische Aufwand (in f-Auswertungen) beträgt etwa $comp \sim m$.

Ein Runge-Kutta-Verfahren ist durch die Angabe von $\alpha = (\alpha_1, \ldots, \alpha_m)^T, \omega = (\omega_1, \ldots, \omega_m)^T$ und $\beta \in \mathbb{R}^{m \times m}$ eindeutig festgelegt.
Eine gängige Notation für α, β und ω ist das *Butcher-Schema*:

α	β
	ω^T

(β ist dabei eine untere Dreiecksmatrix)
Als Beispiel verwenden wir wieder die Mittelpunktsregel, dabei ist dann:
$m = 2, \alpha_1 = 0, \alpha_2 = \frac{1}{2}, \beta_{11} = 0, \beta_{21} = \frac{1}{2}, \omega_1 = 0, \omega_2 = 0$

0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0
	0	1

Weitere Beispiele:

i) Modifiziertes Euler-Verfahren, Ordnung $p = 2$

0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0
	0	1

ii) Euler-Verfahren, Ordnung $p = 1$

0	0
	1

iii) Verfahren von Heun, Ordnung $p = 2$ (Quadratur basiert auf Trapezregel)

0	0	0
1	1	0
	$\frac{1}{2}$	$\frac{1}{2}$

iv) Klassisches Runge-Kutta-Verfahren, Ordnung $p = 4$ (Quadratur basiert auf Simpson-Regel)

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Beispielhaft zeigen wir die „Rückübersetzung“ aus dem Butcher-Schema für das klassische Runge-Kutta-Verfahren:

$$\begin{aligned}\alpha_1 = 0, s_1 = 0 \\ k_1 &= f(t,z) \\ k_2 &= f(t+\frac{h}{2}, z+\frac{h}{2}k_1) \\ k_3 &= f(t+\frac{h}{2}, z+\frac{h}{2}k_2) \\ k_4 &= f(t+h, z+hk_3) \\ \varphi(t,z,h) &= \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\end{aligned}$$

Numerischer Test der Konvergenzbeschleunigung

Wähle t_{end} fest, $t_{\text{end}} = t_N$ für ein $N \in \mathbb{N}$.
$t_{\text{end}} - t_0 = Nh, h > 0$ fest.
Erwartet wird:
$E(h) \approx Ch^p$ für ein $p \in \mathbb{N}(\mathbb{R}, p > 0)$.
Dabei ist $E(h) := \|e\| = \|u_n - y(t_(end))\|$.
Für $h_1, h_2 > 0, h_1 \neq h_2$ ist dann $E(h_1) \approx Ch_1^p, E(h_2) \approx Ch_2^p$.
Es folgt also:

$$\frac{E(h_1)}{E(h_2)} \approx (\frac{h_1}{h_2})^p$$

Definiere für $h_1 \neq h_2$:

$$p := \frac{\log(\frac{E(h_2)}{E(h_1)})}{\log(\frac{h_2}{h_1})}$$

Dies nennt man *EOC* (experimental order or convergence).

Konsistenzordnung von Runge-Kutta-Verfahren

Als Konsistenzordnung $p \in \mathbb{N}$ haben wir definiert: $\tau(z,h,f) = O(h^p)$ für alle (zulässigen) z und alle (glatten) f.
Zunächst machen wir eine Abschätzung nach oben:

Lemma 3.30
Gegeben ein m-stufiges, explizites Runge-Kutta-Verfahren der Ordnung $p \in \mathbb{N}$. Dann ist $p \leq m$.

Beweis 3.30.1
Betrachte $f(t,y) = y, z = 1$, also $\dot{y}(t) = y, y(0) = z = 1$. Dann ist $y(t) = e^t$. Wir betrachten:

$$y(h) = e^h = 1 + h + \frac{h^2}{2} + \cdots + \frac{h^{p+1}}{(p+1)!} + O(h^{p+2})$$
$$\frac{1}{h}(y(h) - z) = 1 + \frac{h}{2} + \cdots + \frac{h^p}{(p+1)!} + O(h^{p+1})$$

Wir beweisen per Induktion: Die k_i sind Polynome höchstens $(i-1)$-ten Grades in h.
$i = 1$:
$k_1 = f(t,z) = z = 1 \in \mathcal{P}_0$.
Sei also $k_i \in \mathcal{P}_{i-1}$.
$k_{i+1} = f(t + s_{i+1}, z + h\sum_{j=1}^{i} \beta_{ij} k_j) = z + h\sum_{j=1}^{i} \beta_{ij} k_j \in \mathcal{P}_i$.
Es ist jetzt aber

$$\begin{aligned}\tau &= \frac{1}{h}(y(h) - z) - \sum_{j=1}^{m} \omega_j k_j \\ &= 1 + \frac{h}{2} + \frac{h^2}{6} + \cdots + \frac{h^p}{p+1!} - \underbrace{\sum_{j=1}^{m} \omega_j k_j}_{=:q(h)\in\mathcal{P}_{m-1}} + O(h^{p+1})\end{aligned}$$

Nach Voraussetzung ist $\tau = O(h^p)$, es muss also gelten:

$$q(h) = 1 + \frac{h}{2} + \cdots + \frac{h^{p-1}}{p!}$$

für alle $h > 0$.
Also muss $m - 1 \geq p - 1$ und damit $p \leq m$ sein.

Um Abschätzungen nach unten zu erhalten, entwickeln wir jetzt die diskrete Evolution in h:

$$\Psi^{t+h,t} z = z + h \sum_{j=1}^{m} \omega_j k_j$$

(Zu diesem Abschnitt vergleiche auch die Anmerkungen in Appendix A)

0. Schritt:
   $k_i = f(t + \alpha_i h, z + h\sum_{j=1}^{i-1} \beta_{ij} k_j)$

Natürlich ist $\alpha, \beta, \omega = O(1)$ für $h \to \infty$.

Wir zeigen per Induktion, dass $k_i = O(1)$ ist. Für $k_1 = f(t,z) = O(1)$ ist dies klar. Seien also $k_1, \dots, k_{i-1} = O(1)$. Dann ist

$$k_i = f(\underbrace{t + \alpha_i h_i}_{O(1)}, \underbrace{z + h \sum_{j=1}^{i-1} \beta_{ij} k_j}_{O(1)})$$

für $h \to \infty$ und damit $k_1 = O(1)$ für $h \to \infty$.

1. Schritt:
Wir zeigen: $k_i = f(t,z) + O(h)$ für $h \to \infty$.

$$\begin{aligned} k_i &= f(t + \underbrace{\alpha_i h}_{O(h)}, z + h \underbrace{\sum_{j=1}^{m} \omega_j k_j}_{O(1)}) \\ &= f(t + \delta_t, z + \delta_z), \text{ mit } \delta_t = O(h), \delta_z = O(h) \\ &= f(t,z) + O(h), \text{ falls } f \text{ glatt genug} \end{aligned}$$

2. Schritt:
Im folgenden sei $f := f(t,z), f_t := \partial_t f(t,z)$.
Wir zeigen: $k_i = f + \alpha_i h (f_t + D_y f[f]) + O(h^2)$.

$$\begin{aligned} k_i &= f(t + \alpha_i h, z + h \sum_{j=1}^{m} \beta_{ij} k_j) \\ &= f(t + \alpha_i h, z + h \sum_{j=1}^{m} \beta_{ij} (k_j + O(h))) \\ &= f(t + \alpha_i h, z + h \sum_{j=1}^{m} \beta_{ij} k_j) + O(h^2) \\ &= f + \alpha_i h f_t + D_y f[h \sum_{j=1}^{m} \beta_{ij} k_j] + O(h^2) \\ &= f + \alpha_i j f_t + h \underbrace{\sum_{j=1}^{m}}_{=\alpha_i \text{ per Def.}} D_y f[k_j] + O(h^2) \\ &= f + \alpha_i h f_t + h \alpha_i D_y f[f] + O(h^2) \\ &= f + \alpha_i h (f_t + D_y f[f]) + O(h^2) \end{aligned}$$

3. Schritt:
(ohne Beweis)
$k_i = f + h\alpha_i(f_t + D_y f[f]) + h^2(\frac{1}{2}\alpha_i^2 f_{tt} + (\sum_j \beta_{ij}\alpha_j)(D_y f[f_t] + D_y f \circ D_y f[f]) + \frac{1}{2}\alpha_i^2 D^2 y f[f,f] + \alpha_i^2 D_y f_t[f]) + O(h^3)$

Wir betrachten jetzt den lokalen Diskretisierungsfehler:
$\tau = \frac{1}{h}(y(h) - z) - \sum_{j=1}^m \omega_j k_j$
$\frac{1}{h}(y(h) - z) = \frac{1}{h}(h\dot{y}(0) + O(h^2)) = f(t,z) + O(h)$
Damit Konsistenzordnung $p = 1$ erreicht wird, muss also gelten:

$$\begin{aligned}\tau = O(h) &= f(t,z) - \sum_{j=1}^m \omega_j k_j + O(h) \\ &= f(t,z) - \sum_{j=1}^m \omega_j (f(t,z) + O(h)) + O(h) \\ &\stackrel{!}{=} O(h) \text{ für alle } f, z\end{aligned}$$

Hinreichend und notwendig für die Konsistenzordnung $p = 1$ ist also: $\sum_{j=1}^m \omega_j = 1$.
Wir betrachten nun die Konsistenzordnung $p = 2$. Es gilt:

$$\begin{aligned}\frac{1}{h}(y(h) - z) &= \frac{1}{h}(h\dot{y}(0) + \frac{h^2}{2}\ddot{y}(0) + O(h^3)) \\ &= f(t,z) + \frac{h}{2}(f_t + D_y f[f]) + O(h^2)\end{aligned}$$

Außerdem gilt:

$$\begin{aligned}\tau &= \frac{1}{h}[y(h) - z] - \sum_{j=1}^m \omega_j k_j \\ &= f + \frac{h}{2}(f_t + D_y f[f]) - \sum_{j=1}^m \omega_j (f + \alpha_j h(f_t + D_y f[f])) + O(h^2)\end{aligned}$$

und, weil ja wie bei $p = 1$ $\sum_{j=1}^m \omega_j = 1$ gelten muss:

$$\tau = h(\frac{1}{2}(f_t + D_y f[f]) - \sum_{j=1}^m \omega_j \alpha_j (f_t + D_y f[f])) + O(h^2)$$

Also it $\tau = O(h^2)$ für alle f, z, gdw. $\sum_{j=1}^m \omega_j \alpha_j = \frac{1}{2}$ ist.

Für die Konsistenzordnung $p = 3$ ist (ohne Beweis) notwendig und hinreichend, dass gilt:

$$\sum_j \omega_j = 1$$
$$\sum_j \omega_j \alpha_j = \frac{1}{2}$$
$$\sum_j \omega_j \alpha_j^2 = \frac{1}{3}$$
$$\sum_{i,j} \omega_i \beta_{ij} \alpha_j = \frac{1}{6}$$

Wir erhalten also

Satz 3.31 *Ein m-stufiges, explizites Runge-Kutta-Verfahren mit Paramtern α, β und ω ist für alle hinreichend glatten f und alle z konsistent von Ordnung:*

$p = 1$ *, falls* $\sum_{j=1}^m \omega_j = 1$

$p = 2$ *, falls außerdem* $\sum_{j=1}^m \omega_j \alpha_j = \frac{1}{2}$

$p = 3$ *, falls außerdem* $\sum_{j=1}^m \omega_j \alpha_j^2 = \frac{1}{3}$ *und* $\sum_{i,j=1}^m \omega_i \beta_{ij} \alpha_j = \frac{1}{6}$

$p = 4$ *, falls außerdem* $\sum_{j=1}^m \omega_j \alpha_j^3 = \frac{1}{4}$ *und* $\sum_{i,j=1}^m \omega_i \alpha_i \beta_{ij} \alpha_j = \frac{1}{8}$ *und* $\sum_{i,j=1}^m \omega_i \beta_{ij} \alpha_j^2 = \frac{1}{12}$ *und* $\sum_{i,j,k=1}^m \omega_i \beta_{ij} \beta_{jk} \alpha_k = \frac{1}{24}$

Bemerkungen

i) Die Bedingungsgleichungen an die α, β, ω sind ab $p = 2$ nichtlinear.

ii) Eine allgemeine Formulierung der Bedingungsgleichung für beliebiges $p \in \mathbb{N}$ ist möglich (über "Wurzelbäume"), siehe Deuflhard/Bornemann

Wie konstruiert man nun entsprechende Runge-Kutta-Verfahren?

Beispiel: Konstruktion eines Runge-Kutta-Verfahrens ($p = 4$):
Nach Satz 3.30 muss $m \geq 4$ sein. Wir wählen $m = 4$. Die Anzahl der Freiheitsgrade ist 10 (4 für die ω_i und 6 für die β_{ij}, die α_i sind durch $\alpha_i = \sum_{j=1}^m \beta_{ij}$ festgelegt). Es gibt aber nur 8 Bedingungsgleichungen. Folglich erwarten wir mehr als nur eine Lösung.
Die ersten Bedingungen sind genau dann erfüllt, wenn die Quadraturformel exakt auf $\mathcal{P}_3$ ist. Es gibt 2 Newton-Cotes-Formeln mit maximal 4 Knoten, die dies erfüllen:

i) Newtonsche $\frac{3}{8}$-Regel
$\alpha = (0, \frac{1}{3}, \frac{2}{3}, 1), \omega = (\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8})$.

Aus den weiteren Bedingungen folgt dann:

$$\beta = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ -\frac{2}{3} & 1 & 0 & 0 \\ 1 & -1 & 1 & 0 \end{pmatrix}$$

ii) Simpson-Regel
Diese Regel hat nur 3 Knoten, ist aber trotzdem exakt auf $\mathcal{P}_3$. Wir verdoppeln daher den mittleren Knoten und erhalten:

$\alpha = (0, \frac{1}{2}, \frac{1}{2}, 1), \omega = (\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6})$.

Dieses Verfahren ist genau dann von der Ordnung $p = 4$, wenn gilt:

$$\beta = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Es handelt sich also um das klassische Runge-Kutta-Verfahren.

Im allgemeinen ist es sehr schwierig, Runge-Kutta-Verfahren beliebiger Ordnung zu konstruieren, denn es gilt:

p	1	2	3	4	5	6	...	10	...	20
$\#Bedingungen$	1	2	4	8	17	37	...	1205	...	20.247.374

(Quelle: Deuflhard/Bornemann)

3.6. Schrittweitensteuerung bei expliziten Einschritt-Verfahren

Bisher ist schon gezeigt worden, dass die Konsistenzordnung gleich der Konvergenzordnung ist, d.h. $\|e\|_\infty = O(h^p)$, genauer $\|e\|_\infty \leq Ch^p$ für $h \to \infty$.
Es ergeben sich folgende Fragen:

- Wie groß ist C?
- Für gegebenes h, wie groß ist $\|e\|_\infty$?
- Falls $\|e\|_\infty \leq TOL$, wie wählt man h?

Unser nächster Ansatz ist also, h variabel zu wählen, d.h. den temporären Situationen anzupassen.
Die Aufgabe ist also eine lokale, automatische Steuerung der Schrittweiten. Wir haben hier zwei entgegengesetzte Forderungen:

- h_i muss möglichst groß sein, um den Aufwand zu minimieren
- h_i muss klein genug sein, um den globalen Diskretisierungsfehler klein zu halten

Lokale Genauigkeitskontrolle Wir benütigen Informationen über die Differenz der exakten und der numerischen Lösung, also $e_i = y(t_i) - u_i$.
Es gilt::

$$\begin{aligned} e_{i+1} = & \underbrace{y_{i+1}}_{=:y(t_{i+1})} - u_{i+1} \\ & = \Phi^{t_{i+1},t_i} y_i - \Psi^{t_{i+1},t_i} u_i \\ & = \underbrace{\Phi^{t_{i+1},t_i} y_i - \Phi^{t_{i+1},t_i} u_i}_{\eta_{i+1}} + \underbrace{\Phi^{t_{i+1},t_i} - \Psi^{t_{i+1},t_i} u_i}_{\varepsilon_{i+1}} \end{aligned}$$

η_{i+1} beschreibt (in erster Näherung), wie der Fehler im Schritt $t_i \to t_{i+1}, e_i = y_i - u_i$ „transportiert“ wird. ε_{i+1} beschreibt den Fehler zwischen kontinuierlicher und diskreter Evolution im Schritt $t_i \to t_{i+1}$.

- η_{i+1} ist schwer zugänglich, es wird im folgenden ignoriert.
- ε_{i+1} enthält als „Daten“ numerisch bekannte Werte, nämlich die u_i.
- Das Ziel ist also, „Schätzungen“ für ε_{i+1} zu finden.

Sei nun das Verfahren von der Ordnung p.
Es ist $\varepsilon_{i+1} = \Phi^{t_{i+1},t_i} u_i - \Psi^{t_{i+1},t_i} u_i = h_i \tau(t_i, h_i, u_i)$. Damit gilt:

$$\begin{aligned} \|\varepsilon_{i+1}\| &= \|h_i \tau(t_i, h_i, u_i)\| \\ &= C(t_i, u_i) h_i^{p+1} + O(h_i^{p+2}) \end{aligned}$$

(falls f glatt genug ist).
Die Forderung der Schrittweitensteuerung muss nun sein:

$$\|\varepsilon_{i+1}\| \leq TOL$$

(TOL vorgegeben, eigentlich sogar $\|\varepsilon_{i+1} \approx TOL$, damit keine Rechenschritte verschenkt werden.
ε_{i+1} ist im Allgemeiner nicht exakt berechenbar. Stattdessen wird eine Näherung $\tilde{\varepsilon}_{i+1}$ berechnet werden.

Natürliche Forderung:

$$C_1\|\tilde{\varepsilon}_{i+1}\| \leq \|\varepsilon_{i+1}\| \leq \|C_2\|\tilde{\varepsilon}_{i+1}\|$$

mit $C_1, C_2 > 0$, möglichst dicht bei 1.
Wir nehmen zunächst an, der Algorithmus zur Berechnung von $\tilde{\varepsilon}_{i+1}$ sei schon bekannt (siehe nächster Abschnitt).

Generelle Vorgehensweise: Sei u_i berechnet, ebenso $\tilde{\varepsilon}_{i+1}$.
Falls $\|\tilde{\varepsilon}_{i+1}\| \leq TOL$, so wird der Zeitschritt akzeptiert und eine „optimale" Zeitschrittweite $\tilde{h}_i^*$ als Vorschlag für den Zeitschritt $t_{i+1} \to t_{i+2}$ berechnet, also $h_{i+1} := h_i^*$.
Falls $\|\tilde{\varepsilon}_{i+1}\| > TOL$, wird der Schritt $t_i \to t_{i+1}$ verworfen und mit der Schrittweite $h_i = h_i^*$ wiederholt.

Heuristische Herleitung einer „optimalen" Zeitschrittweite h_i^*: Wir wollen erreichen, dass $\|\varepsilon_{i+1}\| \approx \|\tilde{\varepsilon}_{i+1}\| \approx TOL$, bzw. $\|\varepsilon_{i+1}\| \approx \|\tilde{\varepsilon}_{i+1}\| = c(t_i, u_i)h_i^{p+1} + O(h_i^{p+2})$.
Die Bedingung an h_i^* soll so sein, dass $\varepsilon_{i+1}^* = \Phi^{t_i+h_i^*,t_i}u_i - \Psi^{t_i+h_i^*,t_i}u_i$.
Also:

$$\begin{aligned}
\|\varepsilon_{i+1}^*\| &\approx c(t_i, u_i)h_i^{p+1} + O(h_i^{p+2}) \\
&\approx c(t_i, u_i)h_i^{p+1} \\
\|\varepsilon_i^*\| &\approx c(t_i, u_i)(h_i^*)^{p+1} + O((h_i^*)^{p+2}) \\
&\approx c(t_i, u_i)(h_i^*)^{p+1} \\
&\Rightarrow \frac{TOL}{\|\tilde{\varepsilon}_{i+1}\|} \approx (\frac{h_i^*}{h_i})^{p+1}
\end{aligned}$$

Dies motiviert folgende Definition:

$$h_i^* := (\frac{\rho TOL}{\|\tilde{\varepsilon}_{i+1}\|})^{\frac{1}{p+1}} h_i$$

mit einem „Sicherheitsfaktor" ρ mit $0 < \rho < 1$.

Bemerkung:

i) Modifikationen an h_i^*:
Da $\frac{TOL}{\|\tilde{\varepsilon}_{i+1}\|}$ ggf. sehr groß werden kann, begrenzen wir $h_i^* \leq h_{\max}$. Die Schrittweite sollte außerdem in einem Schritt nur um einen Faktor $q > 1$ wachsen. Dies führt auf die veränderte Definition:

$$h_i^* := \min\{(\frac{\rho TOL}{\|\tilde{\varepsilon}_{i+1}\|})^{\frac{1}{p+1}} h_i, qh_i, h_{\max}\}$$

mit $q > 1, TOL, h_{\max}$ vorgegeben

ii) Die Bedingung $\|\tilde{\varepsilon}_{i+1}\| \leq TOL$ kann evtl. durch $\|\tilde{\varepsilon}_{i+1}\| \leq TOL_{\text{abs}} + \|u_i\| TOL_{\text{rel}}$ ersetzt werden.

Algorithmus 3.32 (Einschrittverfahren mit Schrittweitensteuerung)
Seien $i = 0, t_0 = 0, u_0 = y_0, h_0 > 0, t_{\text{end}}$ vorgegeben.

```
while (t_i < t_end) do
  t = t_i + h_i
  u = Ψ^{t,t_i} u_i
  Berechne ε̃
  h* := min{(ρ TOL / ‖ε̃‖)^{1/(p+1)}, q h_i, h_max}
  if
    ‖ε̃‖ ≤ TOL
  then
    t_{i+1} = t
    u_{i+1} = u
    h_{i+1} = h*
    i = i + 1
  else
    h_i := h_i*
end
```

Berechnung von $\tilde{\varepsilon}$
Die Forderungen an $\tilde{\varepsilon}$ sind:

- $C_1\|\tilde{\varepsilon}\| \leq \|\varepsilon\| \leq C_2\|\tilde{\varepsilon}\|$
- $\tilde{\varepsilon}$ muss einfach zu berechnen sein.

Ansatz: Wir bestimmen $\tilde{\varepsilon}$ durch Vergleich des Ergebnisses zweier Verfahren Ψ und $\hat{\Psi}$. Dabei soll Ψ das „genauere“ Verfahren sein.
Genauer gesagt:

$$\varepsilon = \Phi^{t+h,t}z - \Psi^{t+h,t}z$$
$$\hat{\varepsilon} = \Phi^{t+h,t}z - \hat{\Psi}^{t+h,t}z$$

soll erfüllen: $\Theta := \frac{\|\varepsilon\|}{\|\hat{\varepsilon}\|} < 1$.
Dann setzen wir $\tilde{\varepsilon} = \hat{\varepsilon} - \varepsilon = \hat{\Psi}^{t+h,t}z - \Psi^{t+h,h}z$. Diese Größe ist (im Gegensatz zu ε und $\hat{\varepsilon}$) berechenbar.

Beispiel:
Ψ sei ein Verfahren der Ordnung $p+1$, $\hat{\Psi}$ sei ein Verfahren der Ordnung p, aber nicht $p+1$.
Dann ist $\|\hat{\varepsilon}\| = c(t,z)h^{p+1} + O(h^{p+2})$, wobei $c(t,z) \neq 0$ im „generischen“ Fall gilt. Außerdem ist $\|\varepsilon\| = O(h^{p+2})$. Es folgt also:

$$\begin{aligned}\|\tilde{\varepsilon}\| &= \|\hat{\varepsilon} - \varepsilon\| \\ &= c(t,z)h^{p+1} + O(h^{p+2}) \\ &= \|\varepsilon\|(1 + O(h))\end{aligned}$$

Folglich ist in diesem Fall $\tilde{\varepsilon}$ eine gute Näherung an $\hat{\varepsilon}$.
Falls $c(t,z) \neq 0$, gilt:

$$\Theta = \frac{\|\varepsilon\|}{\|\hat{\varepsilon}\|} = \frac{O(h^{p+2})}{c(t,z)h^{p+1} + O(h^{p+2})} = O(h)$$

Lemma 3.33
Es gelte $\Theta := \frac{\|\varepsilon\|}{\|\hat{\varepsilon}\|} < 1$. *Dann folgt:*

$$\frac{1}{1+\Theta}\|\tilde{\varepsilon}\| \leq \|\hat{\varepsilon}\| \leq \frac{1}{1-\Theta}\|\tilde{\varepsilon}\|$$

d.h. unsere Forderung ist erfüllt mit $C_1 = \frac{1}{1+\Theta}, C_2 = \frac{1}{1-\Theta}$.

Beweis 3.33.1

i)

$$\begin{aligned}\|\tilde{\varepsilon}\| &= \|\hat{\varepsilon} - \varepsilon\| \\ &\leq \|\hat{\varepsilon}\| + \|\varepsilon\| \\ &= \|\hat{\varepsilon}\| + \Theta\|\hat{\varepsilon}\| \\ &= (1+\Theta)\|\hat{\varepsilon}\| \\ &\Rightarrow \frac{1}{1+\Theta}\|\tilde{\varepsilon}\| \leq \|\hat{\varepsilon}\|\end{aligned}$$

ii)

$$\begin{aligned}\|\hat{\varepsilon}\| &= \|\tilde{\varepsilon} + \varepsilon\| \\ &\leq \|\tilde{\varepsilon}\| + \|\varepsilon\| \\ &= \|\tilde{\varepsilon}\| + \Theta\|\hat{\varepsilon}\| \\ &\Rightarrow (1-\Theta)\|\hat{\varepsilon}\| \leq \|\tilde{\varepsilon}\| \\ &\Rightarrow \|\hat{\varepsilon}\| \leq \frac{1}{1-\Theta}\|\tilde{\varepsilon}\|, \, da \; \Theta < 1\end{aligned}$$

Spezialfall:
Gilt $\Theta \to 0$ bei $h \to 0$, so folgt $C_1 \to 1, C_2 \to 1$. Dann heißt $\tilde{\varepsilon}$ *asymptotisch exakt.*
Dies gilt z.B. bei dem Beispiel oben (im generischen Fall).

Bemerkung (Dilemma):
$\tilde{\varepsilon}$ ist eine gute Fehlerschätzung für $\hat{\varepsilon} = \Phi^{t+h,t}z - \hat{\Psi}^{t+h,t}z$, aber $\Psi^{t+h,t}z$ ist „besser“ also $\hat{\Psi}^{t+h,t}z$.
Als numerische Lösung wird daher $\Psi^{t+h,t}z$ genommen. Die Fehlerschätzung kann also zu pessimistisch sein.

Falls Ψ ein Runge-Kutta-Verfahren der Ordnung p und $\hat{\Psi}$ ein Runge-Kutta-Verfahren der Ordnung $\hat{p}$ ist, so bezeichnet $RK_p(\hat{p})$ die Vorgehensweise:

- Rechnung mit Ψ
- Fehlerschätzung mit $\hat{\Psi}^{t+h,t}z - \Psi^{t+h,t}z$

Eingebettete Runge-Kutta-Verfahren
Wir wollen nun erreichen, dass $p = \hat{p} + 1$ ist (damit die Fehlerschätzung nicht zu pessimistisch ist) und außerdem sollen zusätzliche f-Auswertungen möglichst vermieden werden. Dies kann erreicht werten durch $\alpha = \hat{\alpha}, \beta = \hat{\beta}$.

Es ergibt sich folgendes Butcher-Schema:

$\alpha = \hat{\alpha}$	$\beta = \hat{\beta}$
	ω^T
	$\hat{\omega}^T$

d.h. Ψ und $\hat{\Psi}$ unterscheiden sich nur in $\omega, \hat{\omega}$. Solche Verfahren heißen eingebettete Runge-Kutta-Verfahren.

Beispiele

i) $p = 3, m = 3, \hat{p} = 2$

0			
α_2	β_{21}		
α_3	β_{31}	β_{32}	
	ω_1	ω_2	ω_3

sei ein Runge-Kutta-Verfahren der Ordnung $p = 3$. Die Bedingungen für $\hat{p} = 2$ sind nun:

$$\sum_{j=1}^{3} \hat{\omega}_j = 1, \sum_{j=1}^{3} \hat{\omega}_j \alpha_j = \frac{1}{2}$$

Durch nachrechnen ergibt sich, dass $\alpha_2 \neq 0$ sein muss.

Wir setzen jetzt $\hat{\omega}_3 = 0, \hat{\omega}_2 = \frac{1}{2\alpha_2}, \hat{\omega}_1 = 1 - \frac{1}{2\alpha_2}$.

Dann ist $\sum_{j=1}^{3} \hat{\omega}_j = 0$ und $\sum_{j=1}^{3} \hat{\omega}_j \alpha_j = \frac{1}{2}$.

Es ist also $(\alpha, \beta, \hat{\omega})$ ein Runge-Kutta-Verfahren mit $\hat{m} = m = 3$ Stufen und der Ordnung $\hat{p} = 2$.

$\hat{\Psi}$ ist *nicht* von der Ordnung 3, denn es gilt:

$$\sum_{i,j=1}^{3} \hat{\omega}_j \beta_{ji} \alpha_i = 0 \neq \frac{1}{6}$$

Die Fehlerabschätzung liefert:

$$\tilde{\varepsilon} = \hat{\Psi}^{t+h,t} z - \Psi^{t+h,t} z = h(\sum_{j=1}^{3} k_k(\hat{\omega}_j - \omega_j))$$

ii) Ψ sei das klassische Runge-Kutta-Verfahren:

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Wir suchen jetzt ein 4-stufiges Runge-Kutta-Verfahren der Ordnung $\hat{p} = 3$ mit $\hat{\alpha} = \alpha, \hat{\beta} = \beta$.

Aus einer Übungsaufgabe folgt: $\hat{\omega} = \omega$, das heißt $\hat{\Psi} = \Psi$ und damit ist keine Fehlerabschätzung möglich.

Ein Ausweg: Wir erhöhen die Stufenzahl $\hat{m}$, um dadurch zusätzliche f-Auswertungen zu vermeiden.

Es muss also $\hat{m} = 5$ sein, wir schreiben also das klassische Runge-Kutta-Verfahren als Verfahren mit $m = 5$.
Dann wählen wir $\hat{\Psi}$ so, dass die letzte Stufe $\hat{k}_{\hat{m}}$ identisch mit der 1. Stufe k_1^* von Ψ im nächsten Schritt ist, also:

$$\begin{aligned}\hat{k}_{\hat{m}} &= f(t + \alpha_m b, z + h \sum_{j=1}^{\hat{m}-1} \beta_{\hat{m}_j} k_j) \\ &\stackrel{!}{=} f(t + \hat{\alpha}_m h, \Psi^{t+h,h} z) \\ &= f(t + h, z + h \sum_{j=1}^{m} \omega_j h_j)\end{aligned}$$

Dies ist erreichbar, falls $\hat{\alpha}_m = 1, \hat{\omega}_m = 0, \beta_{\hat{m}j} = \omega_j \forall j = 1, \ldots, \hat{m} - 1$.
Nach der Definition von Runge-Kutta-Verfahren muss gelten:

$$\alpha_{\hat{m}} = \sum_{j=1}^{\hat{m}-1} \beta_{\hat{m}j}$$

Dies ist erfüllt, da $\hat{\alpha}_m = 1 = \sum_{j=1}^{\hat{m}-1} \beta_{\hat{m}j}$.

Ausführung:

0					
$\frac{1}{2}$	$\frac{1}{2}$				
$\frac{1}{2}$	0	$\frac{1}{2}$			
1	0	0	1		
1	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	0

Die 5. Stufe ist ein „Dummy“.
Wir wählen nun $\hat{\alpha} = \alpha, \hat{\beta} = \beta$. Die Bedingungsgleichungen für $\hat{p} = 3$ ergeben jetzt:

$$\hat{\omega} = (\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6} - \lambda, \lambda)^T$$

für jedes $\lambda \in \mathbb{R}$. Wir setzen z.B. $\lambda = \frac{1}{6}$, dann ergibt sich:

$$\begin{aligned}\tilde{\varepsilon} &= h \sum_{j=1}^{\hat{m}} k_j (\hat{\omega}_j - \omega_j) \\ &= h \frac{k_4 - k_5}{6} \\ &= h \frac{k_4 - k_1^*}{6}\end{aligned}$$

Bei der Implementierung von $k_1, \ldots, k_4$ benutzt man dann:

$$u = z + h \sum_{j=1}^{4} \omega_j k_j, k_4 = k_1^*$$

Dann ist $\tilde{\varepsilon} = h\frac{k_4 - k_1^*}{6}$. Dies bezeichnet man auch als Fehlberg-Trick oder FSAL (First Same As Last).

3.7. Mehrschrittverfahren

Wir betrachten zunächst nochmal die Einschrittverfahren. Dabei war per Definition:

$$u_{i+1} = u_i + h\varphi(t_i, u_i, h_i; f)$$

Wir benutzen hierbei als Information von u also nur u_i, um u_{i+1} zu berechnen.
Die Idee zu Mehrschrittverfahren ist jetzt also: Benutze Informationen über y in der „Vergangenheit“, um u_{i+1} zu berechnen.

Beispiel:

i) $\dot{y} = f(t, y), y(t_0) = y_0, t_j = t_0 + jh, h > 0.$
Dann ist

$$y(t_{j+1}) - y(t_{j-1}) = \int_{t_{j-i}}^{t_{j+1}} f(s, y(s))ds$$

Wir approximieren dieses Integral mit der Mittelpunktsregel:

$$y(t_{j+1}) - y(t_j) \approx 2hf(t_j, y(t_j))$$

Dies motiviert folgendes Verfahren:

$$u_{j+1} - u_{j-1} = 2hf(t_j, u_j)$$

Falls u_0, u_1 vorgegeben oder bekannt, kann damit u_j für $j \geq 2$ berechneg werden:

$$u_{j+1} = u_{j-1} + 2hf(t_j, u_j)$$

Dies nennt man die explizite Mittelpunktsregel (es ist *kein* Einschrittverfahren).

ii) Simpson-Regel:

$$y(t_{j+1}) - y(t_{j-1}) \approx \frac{h}{3}(f(t_{j+1}, y(t_{j+1})) + 4f(t_j, y(t_j)) + f(t_{j-1}, y(t_{j-1})))$$

Dies führt auf:

$$u_{j+1} - u_{j-1} = \frac{h}{3}(f(t_{j+1}, u_{j+1}) + 4f(t_j, u_j) + f(t_{j-1}, u_{j-1}))$$

Dieses Verfahren heißt Milne-Simpson-Verfahren (es handelt sich um ein implizites Verfahren).

Allgemeiner definieren wir:

Definition 3.34
Es sei $t_j = t_0 + jh$, $\alpha_0, \dots, \alpha_k$ und $\beta_0, \dots, \beta_k \in \mathbb{R}$ seien gegeben, $\alpha_k \neq 0, \alpha_0\beta_0 \neq 0$.
Ein lineares k-Schritt-Verfahren ist eine Rekursion zur Bestimmung von $(u_l)_{l\in\mathbb{N}}$ der folgenden Form:
Seien $u_0, \dots, u_{k-1}$ vorgegeben. Für $j \in \mathbb{N}$ bestimme dann rekursiv u_{j+k} aus:

$$\alpha_k u_{j+k} + \cdots + \alpha_0 u_j = h(\beta_k f(t_{j+k}, u_{j+k}) + \cdots + \beta_0 f(t_j, u_j))$$

Für $\beta_k = 0$ heißt das verfahren explizit, sonst implizit.

Beispiel (explizite Mittelpunktsregel, $k = 2$):
Es ist $\alpha_0 = -1, \alpha_1 = 0, \alpha_2 = 1, \beta_0 = 0, \beta_1 = 2, \beta_2 = 0$.

Lemma 3.35
Sei $f \in C^0(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R}^d)$, Lipschitz-stetig in y mit einer Lipschitz-Konstante L. Sei $\beta_k \neq 0$.
Falls $h < \frac{|\alpha_k|}{|\beta_k|L}$, so existiert zu gegebenem $u_0, \dots, u_{k-1}$ eine eindeutige Folge $(u_l)_{l\in\mathbb{N}}$, die die Rekursionsformel erfüllt.

Beweis 3.35.1 *Per Induktion sei u_{l+k} eindeutig bestimmt für alle $l = -k, \dots, j-1$.*
Wir zeigen jetzt, dass u_{j+k} eindeutig bestimmt ist.
$u_{j+k} = u^*$ ist die Lösung der Fixpunktgleichung

$$u^* = \underbrace{-\frac{1}{\alpha_k}(\alpha_{k-1}u_{j+k-1} + \cdots + \alpha_0 u_j)}_{=:c_1} + \frac{h}{\alpha_k}\beta_k f(t_{j+k}, u^*) + \underbrace{\frac{h}{\alpha_k}(\beta_{k-1}f(t_{j+k-1}, u_{j+k-1}) + \cdots + \beta_0 f(t_j, u_j))}_{=:c_2}$$

d.h. $u^* = F(u^*) + g = \Phi(u^*)$ mit $g = c_1 + c_2 \in \mathbb{R}^d$, $F : \mathbb{R}^d \to \mathbb{R}^d, F(u) = h\frac{\beta_k}{\alpha_k}f(t_{j+k}, u)$.
Behauptung: Φ ist kontrahierend.

Beweis:

$$\begin{aligned}\|Phi(u) - \Phi(v)\| &= F(u) - F(v) \\ &= \frac{|\beta_k|}{|\alpha_k|} h \|f(t_{j+k}, u) - f(t_{j+k}, v)\| \\ &\leq \underbrace{L\frac{|\beta_k|}{|\alpha_k|}h}_{=:q<1} \|u - v\| \\ &= q\|u - v\|\end{aligned}$$

Nach dem Banach'schen Fixpunktsatz existiert genau ein Fixpunkt u^* von Φ.

Shift-Operator

Sei $u : \mathbb{N} \to \mathbb{R}^d, u = (u_l)_{l\in\mathbb{N}}$. Mit S bezeichnen wir den Shift-Operator $(Su)_j = u_{j+1}$.
Für $m \in \mathbb{N}$ ist dann $(S^m u)_j = u_{j+m}$.
Sei nun q ein Polynom k-ten Grades, $q(x) = \sum_{j=0}^{k} a_l x^l$. Dann bezeichnen wir

$$(q(S)u)_j = (\sum_{l=0}^{k} a_l S^l u)_j = \sum_{l=0}^{k} a_l u_{j+l}$$

Seien jetzt $\rho, \sigma \in \mathcal{P}_k$. Sei $\rho(x) = \alpha_k x^k + \cdots + \alpha_0$ und $\sigma(x) = \beta_k x^k + \cdots + \beta_0$.
Damit lässt sich die Rekursionsformel wie folgt schreiben:

$$(\rho(S)u)_j = (h\sigma(S)F)_j$$

mit $F_j = f(t_j, u_j)$.
Folglich ist jedes Mehrschrittverfahren eindeutig durch die Angabe von ρ und σ bestimmt.

Beispiele

i) Euler-Verfahren: $\rho(x) = x - 1, \sigma(x) = 1$

ii) Explizite Mittelpunktsregel: $\rho(x) = x^2 - 1, \sigma(x) = 2x$

iii) Milne-Simpson-Regel: $\rho(x) = x^2 - 1, \sigma(x) = \frac{1}{3}x^2 + \frac{4}{3}x + \frac{1}{3}$

Eine wichtige Beobachtung: Nach einer Anlaufrechnung, d.h. nach dem Auswerten von $f(t_0, u_0), \ldots, f(t_{k-1}, u_{k-1})$ wird für jeden Zeitschritt beim expliziten k-Schritt-Verfahren nur eine f-Auswertung benötigt.
Wir wollen jetzt die Konsistenz untersuchen.

Definition 3.36 (lokaler Diskretisierungsfehler)
Sei $y : \mathbb{R} \to \mathbb{R}^d$ glatt. Für $h > 0$ sei:

$$\tau(t, h, y) := \frac{1}{h}(\rho(S)y)(t) - (\sigma(S)y')(t)$$

Dabei ist (ρ, σ) ein k-Schritt-Verfahren und es ist

$$(Sy)(t) := y(t + h)$$
$$(\rho(S)y)(t) := \sum_{l=0}^{k} \alpha_l y(t + lh)$$

und σ entsprechend.
τ heißt lokaler Diskretisierungsfehler.

Bemerkung: Def. 3.36 ist eine Übertragung von Def. 3.24.
Wir betrachten hierzu das Euler-Verfahren:

$$\begin{aligned}
\tau(0, h, y) &= \frac{1}{h}(\rho(S)y)(0) - (\sigma(S)y')(0) \\
&= \frac{1}{h}(y(0 + h) - y(0)) - y'(0) \\
&= \frac{1}{h}(y(h) - z) - f(0, z) \\
&= \frac{1}{h}(\Phi^{h,0}z - z) - f(0, z) \\
&= \frac{1}{h}(\Phi^{h,0}z - z) - \varphi(0, h, f)
\end{aligned}$$

Dies entspricht genau der Definition des lokalen Diskretisierungsfehlers in 3.24. (Wir haben hier ausgenutzt, dass y trivialerweise das Anfangswertproblem $\dot{y} = \dot{y}, y(0) = y(0)$ löst)

Definition 3.37
Ein k-Schritt-Verfahren (ρ, σ) heißt konsistent, falls für alle glatten $y : \mathbb{R} \to \mathbb{R}^d$ gilt:

$$\lim_{h \to 0, h \neq 0} \tau(t, h, y) = 0 \text{ lokal gleichmäßig in } t$$

(ρ, σ) heißt konsistent von Ordnung $p \in \mathbb{N}$, falls $\tau(t, h, y) = O(h^p)$ lokal gleichmäßig in t gilt.

Lemma 3.38
Folgende Aussagen sind äquivalent:

i) (ρ, σ) *hat Konsistenzordnung* $p \in \mathbb{N}$

ii) $\tau(0, h, q) = 0$ *für alle* $q \in \mathcal{P}_p, h > 0$.

iii) $\tau(0, h, exp) = \frac{1}{h}\rho(e^h) - \sigma(e^h) = O(h^p)$

iv)

$$\sum_{l=0}^{k} \alpha_l = 0$$
$$\sum_{l=0}^{k} \alpha_l^j = j \sum_{l=0}^{k} \beta_l l^{j-1} \forall j = 1, \ldots, p$$

Dabei definieren wir $0^0 := 1$.

Beweis 3.38.1 *Wir zeigen* $i) \Rightarrow ii) \Rightarrow iii) \Rightarrow iv) \Rightarrow i)$.

$i) \Rightarrow ii)$
Sei q *ein Polynom vom Grad* p. *Dann ist*

$$(\rho(S)q)(0) = \sum_{l=0}^{k} \alpha_l q(lh)$$

ein Polynom in h, *ebenfalls vom Grade* p, $(\sigma(S)q)(0)$ *entsprechend. Folglich ist auch* $h\tau(0, h, q)$ *ein Polynom in* h *vom Grad* p. *Nach* $i)$ *gilt:*

$$h\tau(0, h, q) = O(h^{p+1})$$

Ein Polynom p*-ten Grades soll also für alle* $h > 0$ *in* $O(h^{p+1})$ *liegen. Dies ist aber nur möglich, wenn es konstant* 0 *ist, also muss* $\tau(0, h, q) = 0$ *für alle* $q \in \mathcal{P}_p, h > 0$ *gelten.*

$ii) \Rightarrow iii)$

Es gilt:

$$\begin{aligned}
\exp(h) &= \sum_{l=0}^{\infty} \frac{1}{l!} h^l \\
&= \sum_{l=0}^{p} \underbrace{\frac{1}{l!} h^l}_{=:q(h)\in\mathcal{P}_p} + \underbrace{r(h)}_{=O(h^{p+1}) \text{ und damit } r'(h)=O(h^p)} \\
\tau(o,h,\exp) &= \tau(0,h,q+r) \\
&= \underbrace{\tau(0,h,q)}_{=0 \text{ nach } ii)} + \underbrace{\tau(0,h,r)}_{=\frac{1}{h}((\rho(S)r)(0))-(\sigma(S)r')(0)} \\
(\rho(S)r)(0) &= \sum_{l=0}^{k} \alpha_l r(lh) = O(h^{p+1}) \\
&\Rightarrow \frac{1}{h}((\rho(S)r)(0)) = O(h^p) \\
&(\sigma(S)r')(0) = O(h^p) \\
&\tau(0,h,r) = O(h^p)
\end{aligned}$$

Wir zeigen noch das erste Gleichheitszeichen in iii):

$$\begin{aligned}
(\rho(S)\exp)(0) &= \sum_{l=0}^{k} \alpha_l \exp(lh) = \sum_{l=0}^{k} \alpha_l (e^h)^l = \rho(e^h) \\
(\sigma(S)\exp) &= \cdots = \sigma(e^h) \\
&\Rightarrow \tau(0,h,\exp) = \frac{1}{h}\rho(e^h) - \sigma(e^h)
\end{aligned}$$

iii) $\Rightarrow$ *iv)*

Es gilt:

$$\begin{aligned}
h\tau(0,h,\exp) &= \sum_{l=0}^{k} \alpha_l e^{lh} - \sum_{l=0}^{k} \beta_k h e^{lh} \\
&= \sum_{j=0}^{p} \frac{1}{j!} \sum_{l=0}^{k} \alpha_l (lh)^j - \sum_{j=0}^{p} \frac{1}{j!} \sum_{l=0}^{k} \beta_l l^j h^{j+1} + O(h^{p+1}) \quad \overset{iii)}{=} O(h^{p+1})
\end{aligned}$$

Und es muss also gelten:

$$\sum_{j=0}^{p} \frac{1}{j!} \sum_{l=0}^{k} \alpha_l (lh)^j = \sum_{j=0}^{p} \frac{1}{j!} \sum_{l=0}^{k} \beta_l l^j h^{j+1} \forall h > 0$$

Durch Ordnen nach Potenzen von h ergibt sich:

$$\begin{aligned} h^0 &: \sum_{l=0}^{k} \alpha_l = 0 \\ h^1 &: \sum_{l=0}^{k} \alpha_l l = \sum_{l=0}^{k} \beta_l \\ &\dots \\ h^j &: \sum_{l=0}^{k} \alpha_l l^j = j \sum_{l=0}^{k} \beta_l l^{j-1} \end{aligned}$$

iv) $\Rightarrow$ *i)*

Sei y hinreichend glatt. Die Taylor-Entwicklung liefert:

$$\begin{aligned} y(t+lh) &= \sum_{j=0}^{p} \frac{1}{j!} y^{(j)}(t)(lh)^j + O(h^{p+1}) \\ y'(t+lh) &= \sum_{j=0}^{p-1} \frac{1}{j!} y^{(j+1)}(t)(lh)^j + O(h^p) \end{aligned}$$

Nun folgt:

$$\begin{aligned} \tau(t,h,y) &= (\frac{1}{h}\rho(S)y - \sigma(S)y')(t) \\ &= \sum_{j=0}^{k} \alpha_l \frac{1}{h}(lh)^j y^{(j)}(t) + O(h^p) - \sum_{j=0}^{p-1} \frac{1}{j!} \sum_{l=0}^{k} \beta_l y^{(j+1)}(t)(lh)^j - O(h^p) \\ &= 0 + O(h^p) \ \textit{nach iv)} \end{aligned}$$

Bemerkung: Ein k-Schritt-Verfahren (ρ, σ) ist konsistent genau dann, wenn $\rho(1) = 0, \sigma'(1) = \rho(1)$.

Beweis: „⇒“

$$\begin{aligned} &\tau(0,h,\exp) = \frac{1}{h}\rho(e^h) - \sigma(e^h) \overset{h\to 0}{\to} 0 \\ &\lim_{h\to 0} \frac{1}{h}\rho(e^h) \overset{\text{l'Hospital}}{=} \rho'(1) \end{aligned}$$

und damit: $\tau(0,h,\exp) \overset{h\to 0}{\to} \rho'(1) - \sigma(1) = 0$
„$\Leftarrow$“analog.

Beispiele:

i) Explizites Euler-Verfahren ($\rho(x) = x - 1, \sigma(x) = 1$):

$$\sum_{l=0}^{k} \alpha_l = \alpha_1 + \alpha_0 = 1 - 1 = 0$$

$$\sum_{l=0}^{k} \alpha_l l = 1 = \sum_{l=0}^{k} \beta_l$$

aber:

$$\sum_{l=0}^{k} \alpha_l l^2 = 1 \neq 0 = \sum_{l=0}^{k} \beta_l l$$

Das Euler-Verfahren hat also Konsistenzordnung $p = 1$, aber nicht $p = 2$.

ii) Explizite Mittelpunktsregel ($\rho(x) = x^2 - 1, \sigma(x) = 2x$):

$$\sum_{l=0}^{k} \alpha_l = 0$$

$$\alpha_2 2 + \alpha_1 1 = 2 = \beta_2 + \beta_1 + \beta_0$$

$$\alpha_2 2^2 + \alpha_1 1^2 + \alpha_0 0^2 = 4 = 2(\beta_2 2 + \beta_1 1)$$

$$\alpha_2 2^3 + \alpha_1 1^3 = 8 \neq 6 = 3(\beta_2 2^2 + \beta_1 1^2)$$

Folglich hat die explizite Mittelpunktsregel die Konsistenzordnund $p = 2$, aber nicht $p = 3$.

Definition 3.39 (Konvergenz bei Mehrschrittverfahren)
(ρ, σ) sei ein k-Schritt-Verfahren. y sei die (glatte) Lösung von $y' = f(t, y(t)), y(t_0) = y_0$. Dann heißt (ρ, σ) konvergent gegen y, falls

$$\lim_{\substack{h\to 0 \\ \frac{t-t_0}{h}\in\mathbb{N}}} u_h(t) = y(t)$$

für alle Anfangswerte $u_0^h, \ldots, u_{k-1}^h$ mit $\lim_{h\to 0} u_l^h = y_0, l = 1, \ldots, k-1$ (dabei ist $u_h(t) = u_j$ gegeben durch das k-Schritt-Verfahren, mit $t = t_0 + hj$).
(ρ, σ) heißt konvergent, falls (ρ, σ) konvergent gegen y für alle hinreichend glatten f und alle y_0, t_0 ist.

Im Gegensatz zu Einschrittverfahren reicht bei Mehrschrittverfahren allerdings Konsistenz allein *nicht* für Konvergenz.

Beispiel 3.40

$$\rho(x) = x^2 + 4x - 5$$
$$\sigma(x) = 4x + 2$$

(ρ, σ) hat Konsistenzordnung $p = 3$.
Wir betrachten jetzt das Anfangswertproblem

$$y' = 0$$
$$y(0) = y_0 = 1$$

(Exakte Lösung: $y(t) = 1$ für alle t)
Wir wählen $u_0 := 1, u_1 := 1 + \varepsilon h$.
Das Verfahren $u_{j+2} + 4u_{j+1} - 5u_j = h \cdot 0$ führt jetzt auf folgende Gleichungen:

$$\begin{pmatrix} u_{j+1} \\ u_{j+2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 5 & -4 \end{pmatrix} \begin{pmatrix} u_j \\ u_{j+1} \end{pmatrix} (j \geq 1)$$
$$\begin{pmatrix} u_{l+1} \\ u_{l+2} \end{pmatrix} = A^l \begin{pmatrix} u_0 \\ u_1 \end{pmatrix}$$

A hat die Eigenwerte $\lambda_1 = 1, \lambda_2 = -5$ mit den zugehörigen Eigenvektoren $e_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, e_2 = \begin{pmatrix} -1 \\ 5 \end{pmatrix}$. Es gilt:

$$\begin{pmatrix} u_0 \\ u_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 + \varepsilon h \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \beta \begin{pmatrix} -1 \\ 5 \end{pmatrix}$$

Dies wird gelöst durch $\alpha = 1 + \frac{\varepsilon h}{6}, \beta = \frac{\varepsilon h}{6}$. Damit folgt dann:

$$\begin{aligned} \begin{pmatrix} u_{l+1} \\ u_l \end{pmatrix} &= A^l \begin{pmatrix} u_0 \\ u_1 \end{pmatrix} \\ &= A^l(\alpha e_1 + \beta e_2) \\ &= \lambda_1^l \alpha e_1 + \lambda_2^l \beta e_2 \\ &= (1 + \frac{\varepsilon h}{6}) \begin{pmatrix} 1 \\ 1 \end{pmatrix} + (-5)^l \frac{\varepsilon h}{6} \begin{pmatrix} -1 \\ 5 \end{pmatrix} \\ u_{l+1} &= 1 + \frac{\varepsilon h}{6}(1 - (-5)^l) \end{aligned}$$

Jetzt folgt mit $t_l = ln, l = \frac{t_l}{h}, 5^l = 5^{\frac{t_l}{h}} = \exp(\frac{t_l}{h}\log 5)$*:*

$$u_h(t_l) = u_l^h = 1 + \frac{\varepsilon h}{6}(1 - (-5)^l)$$
$$|u_h(t_l)| = |1 + \frac{\varepsilon h}{6}(1 \pm 5^l)|$$
$$|u_h(t_l)| = |1 + \frac{\varepsilon h}{6}(1 \pm \exp(\frac{t_l}{h}\log 5))|$$

Wir wählen jetzt ein h_l *mit* $h_l \overset{l\to\infty}{\to} 0$*, z.B.* $h_l = \frac{1}{l}$*. Dann ist* $t_l = 1$ *und es folgt:*

$$|u_{h_l}(1)| = |1 + \frac{\varepsilon h}{6}(1 \pm \exp(\frac{1}{h}\log 5))| \overset{l\to\infty, h_l\to 0}{\to} \infty$$

(ρ, σ) *konvergiert also nicht gegen* y*.*

Bei diesem Beispiel haben wir effektiv die Folge $(u_j)_{j\in\mathbb{N}}$ betrachtet, die durch $u_0, \ldots, u_{k-1}$ und für $j \geq 0$ rekursiv durch $(\rho(S)u)_j = 0$ gegeben ist. Eine solche Gleichung nennt man lineare, homogene Differenzengleichung.
Dies motiviert folgende Definition:

Definition 3.41

i) Eine lineare, homogene Differenzengleichung heißt stabil, falls es für alle Startwerte $u_0, \ldots, u_{k-1}$ *ein* $C > 0$ *gibt, so dass* $|u_j| \leq C$ *für alle* j*.*

ii) Ein k-Schritt-Verfahren (ρ, σ) *heißt stabil, falls die zugehörige homogene, lineare Differenzengleichung stabil ist.*

Lemma 3.42
Es sei $(u_j)_{j\in\mathbb{N}}$ *die Lösung einer linearen, homogenen Differenzengleichung zu* ρ*. Dann gilt:*

$$U^{j+1} = AU^j \text{ für } j \geq 0$$

mit

$$U^j = \begin{pmatrix} u_j \\ \vdots \\ u_{j+k-1} \end{pmatrix}, A = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -\alpha_0 & \ldots & \ldots & -\alpha_{k-1} \end{pmatrix}$$

Beweis 3.42.1
für $l < k$:

$$(AU^j)_l = u_{j+l} = U_l^{j+1}$$

für k:

$$(AU^j)_k = \sum_{l=0}^{k-1} \alpha_l u_{j+l} \underbrace{=}_{(\rho(S)u)m=0} u_{j+k} = (U^{j+1})_k$$

Lemma 3.43 *Sei $A \in \mathbb{R}^{k\times k}$ wie in Lemma 3.42. Dann gilt:*

i)
$$\chi_A(\lambda) = \det(\lambda I - A) = \lambda^k + \alpha_{k-1}\lambda^{k-1} + \cdots + \alpha_1\lambda + \alpha_0 = \rho(\lambda)$$

ii) Ist $\lambda \in \sigma(A)$, so ist $\dim \operatorname{eig}(\lambda) = 1$

iii) Zu jedem Eigenwert λ von A gibt es genau einen Jordan-Block.

Beweis 3.43.1

i) Induktion über k und Entwicklung von $\det(\lambda I - A)$ *nach der ersten Spalte*

ii) Sei λ Eigenwert von A zum Eigenvektor e, d.h. $Ae = \lambda e$. Dann ist für alle $l = 1, \ldots, k$ $(Ae)_l = e_{l+1} = \lambda e_l$.
Ist $e_1 = x$ vorgegeben, folgt daraus:

$$\operatorname{eig}(\lambda) = \operatorname{span}\{(x, \lambda x, \ldots, \lambda^{k-1}x)^T\}$$

iii) folgt direkt aus ii)

Satz 3.44 (Wurzelbedingung von Dahlquist)
Ein k-Schritt-Verfahren (ρ, σ) ist stabil genau dann, wenn für alle Nullstellen λ von ρ gilt:

i) $|\lambda| \leq 1$

ii) Falls $|\lambda| = 1$, so ist λ einfache Nullstelle von ρ.

Beweis 3.44.1 *Die Folge $(u_j)_{j\in\mathbb{N}}$ ist genau dann beschränkt, wenn die Folge $(U^j)_{j\in\mathbb{N}}$ beschränkt ist.*
Analog wie im Beweis von Satz 3.18 zeigt man:
$(U^j)_{j\in\mathbb{N}}$ ist beschränkt für alle Anfangswerte U^0 genau dann, wenn für alle $\lambda \in \sigma(A)$ gilt:

i) $|\lambda| \leq 1$

ii) Falls $|\lambda| = 1$*, so ist* $\iota(\lambda) = 1$*.*

Wir beobachten jetzt, dass λ *Nullstelle von* ρ *ist genau dann, wenn* λ *Eigenwert von* A *ist. Es bleibt also noch zu zeigen:*
$\iota(\lambda) = 1$ *genau dann, wenn* λ *einfache Nullstelle von* ρ *ist.*
Sei also $A = S^{-1}JS$ *mit* $S \in GL_k(\mathbb{C})$ *und* $J = \begin{pmatrix} J_1 & & \\ & J_2 & \\ & & \ddots \end{pmatrix}$ *und* $\chi_A(\lambda) = \prod_{\lambda \in \sigma(A)} (x - \lambda)^{n_\lambda}$*. Dabei ist* n_λ *die Dimension von* J_λ*.*
Nach Lemma 3.42 gilt jetzt: Die Eigenwerte zu verschiedenen Jordan-Blöcken sind verschieden. Daraus folgt:

$$\iota(\lambda) = 1 \iff n_\lambda = 1 \iff \lambda \text{ ist einfache Nullstelle von } \chi_A = \rho$$

Beispiel:

i) Für jedes Einschrittverfahren, dass sich als $k = 1$-Mehrschrittverfahren schreiben lässt, gilt: $\rho(x) = x - 1$ (z.B. Euler, $\sigma(x) = x - 1$). Damit ist (ρ, σ) stabil.

ii) Für die explizite Mittelpunktsregel und das Milne-Simpson-Verfahren gilt: $\rho(x) = x^2 - 1$. Die Nullstellen $\lambda = \pm 1$ sind einfach und die Verfahren damit stabil.

Satz 3.45
Ein konvergentes k*-Schritt-Verfahren ist stabil und konsistent (*$\rho(1) = 0, \rho'(1) = \sigma(1)$*). Zudem gilt:* $\rho'(1) = \sigma(1) \neq 0$*.*

Beweis 3.45.1

Konvergenz $\Rightarrow$ *Stabilität*
Annahme: $u_0, \ldots, u_{k-1}$ *seien so, dass für* $(u_j)_{j \in \mathbb{N}}$ *mit* $(\rho(S)u)_j = 0$ *gilt:*

$$\limsup_{j \to \infty} |u_j| = \infty$$

Dann gibt es eine Nullfolge $\varepsilon_j \overset{j \to \infty}{\to} 0$*, so dass* $\limsup_{j \to \infty} |\varepsilon_j u_j| = \infty$ *ist.*

Denn: Wir können eine Teilfolge j_l *auswählen, so dass* $|u_{j_l}| \overset{l \to \infty}{\to} \infty$ *ist und ohne Einschränkung annehmen, dass* $u_{j_l} \neq 0$ *für alle* l *ist.*

Dann definieren wir $\varepsilon_j := \begin{cases} \frac{1}{\sqrt{|u_{j_l}|}} & \text{, falls } j = j_l \\ 0 & \text{, sonst} \end{cases}$.

Dann gilt: $\varepsilon_j \overset{j\to\infty}{\to} 0$ *und außerdem:*

$$\begin{aligned}\limsup_{j\to\infty} |\varepsilon_j u_j| &\geq \limsup_{l\to\infty} |\varepsilon_{j_l} u_{j_l}| \\ &= \limsup_{j\to\infty} \sqrt{|u_{j_l}|} \\ &= \infty\end{aligned}$$

Wir betrachten jetzt das Anfangswertproblem $y' = 0, y(0) = y_0 = 0$.

Wir wählen $t > 0$ *fest und setzen* $h := \frac{t}{n}, n \in \mathbb{N}$.

Wir definieren: $\bar{u}_0 = \varepsilon_n u_0, \ldots, \bar{u}_{k-1} = \varepsilon_n u_{k-1}$.

Das k-Schritt-Verfahren liefert zu diesen Anfangswerten eine Folge $(\bar{u}_j)_{j\in\mathbb{N}} = \varepsilon_n (u_j)_{j\in\mathbb{N}}$. *Die* $\bar{u}_j$ *sind eine Approximation an* $y(jh)$.

Die Startwerte erfüllen: $\bar{u}_0, \ldots, \bar{u}_{k-1} \overset{h\to 0}{\to} 0 = y_0$.

Da (ρ, σ) *konvergent ist, folgt:* $\bar{u}_j \overset{h\to 0, jh=t}{\to} y(t) = 0$. *Jetzt ist aber* $\limsup \bar{u}_n = \limsup \varepsilon u_n = \infty$.

↯

Konvergenz $\Rightarrow \rho(1) = 0$

Wir betrachten das Anfangswertproblem $y' = 0, y(0) = y_0 = 1$ *mit der Lösung* $y(t) = 1$. *Seien* $u_0 = \cdots = u_{k-1} = y_0 = 1$.

Der erste Schritt des Verfahrens liefert:

$$\alpha_k u_k^h + \alpha_{k-1} + \cdots + \alpha_0 = h(\beta_k F_k + \cdots + \beta_0 F_0) = 0$$

Wegen der Konvergenz folgt: $u_k^h = u_h(kh) \to y(0) = 1$ *und damit* $u_k^h = 1 \forall h$. *Daraus folgt* $\alpha_k + \cdots + \alpha_0 = \rho(1) = 0$.

$\rho'(1) \neq 0$

Dies folgt direkt, denn wegen der Stabilität ist $\lambda = 1$ *einfach und damit* $\rho'(1) \neq 0$.

$\rho'(1) = \sigma(1)$

Wir definieren $\Theta = \frac{\sigma(1)}{\rho'(1)}$, *dies ist wohldefiniert, da* $\rho(1) \neq 0$ *ist. Wir wollen zeigen, dass* $\Theta = 1$ *gilt.*

Dazu betrachten wir das Anfangswertproblem $y' = 1, y(0) = y_0 = 0$ *mit der Lösung* $y(t) = t$.

Wir setzen die Startwerte

$$u_j^h = u_h(jh) := jh\Theta (j = 0, \ldots, k-1)$$

Die Konsistenzvoraussetzung $\lim_{h\to 0} u_j^h = \lim_{h\to 0} jh\Theta = 0$ *ist natürlich erfüllt.*
Es gilt einerseits:

$$\begin{aligned}
&\rho(x) = \alpha_k x^k + \cdots + \alpha_0\\
&\rho'(x) = k\alpha_k x^{k-1} + (k-1)\alpha_{k-1}x^{k-2} + \cdots + \alpha_1\\
&\rho'(1) = k\alpha_k + (k-1)\alpha_{k-1} + \cdots + \alpha_1\\
&\quad \Rightarrow \alpha_k(\Theta kh) + \alpha_{k-1}(\Theta(k-1)h) + \cdots + \alpha_1(\Theta h) + \alpha_0(\Theta 0h) = \Theta\rho'(1)h = h\sigma(1)
\end{aligned}$$

und andererseits wegen $\rho(1) = 0$*:*

$$\alpha_k(\Theta jh) + \alpha_{k-1}(\Theta jh) + \cdots + \alpha_0(\Theta jh) = 0 \forall j \in \mathbb{N}$$

Die Summe der beiden Gleichungen ergibt:

$$\alpha_k(\Theta(k+j)h) + \alpha_{k-1}(\Theta(k-1+j)h) + \cdots + \alpha_0(\Theta(jh)) = h\sigma(1)$$

Mit $w_j := \Theta jh$ *ergibt sich hieraus:*

$$(\rho(S)w)_j = h\sigma(1)$$

und da $F^j = f(t_j, \omega_j) = 1$ *ist:*

$$(\rho(S)w)_j = h(\sigma(S)F)_j$$

Mit anderen Worten, $(w_j)_{j\in\mathbb{N}}$ *ist eine numerische Lösung des Anfangswertproblems mit* $u_h(t_j) = w_j = \Theta jh$*.*
Da wir Konvergenz vorausgesetzt haben, folgt hieraus:

$$\lim_{\substack{h\to 0\\ \frac{t}{h}\in\mathbb{N}}} u_h(t) = y(t) = t$$

Andererseits ist aber $j = \frac{t}{h}$ *und deswegen* $\lim_{\substack{h\to 0\\ \frac{t}{h}\in\mathbb{N}}} \Theta jh = \Theta t$ *und es folgt* $\Theta t = t$ *und damit* $\Theta = 1$ *bzw.* $\rho'(1) = \sigma(1)$*.*

Lemma 3.46
Sei $\rho \in \mathcal{P}_k$ *so, dass die zugehörige homogene Differenzengleichung stabil ist.* $(b_j)_{j\in\mathbb{N}}$ *sei eine Folge, und* $(a_j)_{j\in\mathbb{N}}$ *löse die inhomogene Differenzengleichung*

$$(\rho(S)a)_j = b_j$$

für $j \in \mathbb{N}, a_j, b_j \in \mathbb{R}^d$*.* $a_0, \ldots, a_{k-1}$ *seien vorgegeben. Dann existiert ein* $C > 0$*, so dass*

$$\|a_{j+k}\| \leq C(\max_{0\leq l\leq k-1} \|a_l\| + \sum_{l=0}^{j} \|b_l\|) \forall j \in \mathbb{N}$$

Beweis 3.46.1
Ohne Einschränkung ist $\alpha_k = 1$. Wir definieren:

$$U^j = \begin{pmatrix} a_j \\ \vdots \\ a_{j+k-1} \end{pmatrix}, V^j = \begin{pmatrix} 0 \\ \vdots \\ b_j \end{pmatrix}, A = \begin{pmatrix} 0 & 1 & \\ & \ddots & \ddots \\ -\alpha_0 & \dots & -\alpha_{k-1} \end{pmatrix}$$

Damit lässt sich die Gleichung dann schreiben als

$$U^{j+1} = AU^j + V^j$$

Ohne Einschränkung sei jetzt $d = 1$, wir verwenden also den Betrag als Norm. Auf dem $\mathbb{R}^k$ wählen wir dann die Norm $\|U\| := \max_{1 \le l \le k} |u_l|$.
Aus der Stabilität folgt: Für jedes $W \in \mathbb{R}^k$ existiert ein $C > 0$, so dass $\|A^j W\| \le C$ für alle $j \in \mathbb{N}$ ist.
Es existiert sogar ein $C_1 > 0$ mit $\|A^j W\| \le C_1 \|W\|$ (folgt aus der Betrachtung von $\sup_{j \in \mathbb{N}} \sup_{\|W\| \le 1} \|A^j W\| = C_1$).
Es gilt für $j \in \mathbb{N}$:

$$\begin{aligned} U^j &= AU^{j-1} + V^{j-1} \\ &= A(AU^{j-1} + V^{j-2}) + V^{j-1} \\ &= \dots = A^j U^0 + \sum_{l=0}^{j-1} A^{j-l} V^l \end{aligned}$$

Und daher folgt:

$$\|U^j\| \le \|A^j U^0\| + \sum_{l=0}^{j-1} \|A^{j-l} V^l\| \le C_1 (\|U^0\| + \sum_{l=0}^{j-1} \|V^l\|)$$

und in einer einzelnen Komponente:

$$\begin{aligned} |a_{j+k}| &= |U_k^{j+1}| \\ &\le \|U^{j+1}\| \\ &\le C_1 (\|U^0\| + \sum_{l=0}^{j} \|V^l\|) \\ &= C_1 (\|U^0\| + \sum_{l=0}^{j} |b_l|) \\ &= C_1 (\max_{0 \le l \le k-1} |a_l| + \sum_{l=0}^{j} |b_l|) \end{aligned}$$

Satz 3.47 (Konvergenz von Mehrschrittverfahren)
Sei $f : I \times \mathbb{R}^d \to \mathbb{R}^d$ glatt (insbesondere Lipschitz-stetig im 2. Argument), $I = [t_0, t_0 + a]$, und y glatte Lösung von $\dot{y}(t) = f(t, y(t)), t \in I, y(t_0) = y_0 \in \mathbb{R}^d$.
(ρ, σ) sei ein stabiles Mehrschrittverfahren der Konsistenzordnung $p \in \mathbb{N}$.
Ist $h > 0$ klein genug (diese Bedingung spielt nur im Fall impliziter Verfahren eine Rolle), so gilt:
Ist $e_j := y(t_j) - u_j, t_j = jh + t_0, u_j$ die numerische Lösung, so ist für alle j mit $t_j \in I$

$$\|e_j\| \leq C(\max_{0 \leq l \leq k-1} \|e_l\| + h^p)$$

Fazit: Konsistenz + Stabilität = Konvergenz

Beweis 3.47.1
Falls h klein genug ist, existiert $(u_j)_{j \in \mathbb{N}}$ eindeutig (nur problematisch, falls $\beta_k \neq 0$, siehe Lemma 3.35). Sei $\tau_j := \tau(t_j, h, y)$. Es gilt:

$$\begin{aligned}(\rho(S)y)(t_j) - (h\sigma(S)y')(t_j) &= (\rho(S)y)t_j - h(\sigma(S)f(t_j, y(t_j))) = h\tau_j \\ (\rho(S)y)t_j - h\sigma(S)f(t_j, u_j) &= (\rho(S)u)_j - h(\sigma(S)F)_j = 0\end{aligned}$$

Wir substrahieren die beiden Gleichungen und erhalten:

$$\rho(S)\underbrace{(y_j - u_j)}_{=e_j} - h\sigma(S)(f(t_j, y_j) - F^j) = h\tau_j$$

und damit:

$$\rho(S)e_j = \underbrace{h(\tau_j + \sigma(S)(f(t_j, y_j) - F^j))}_{=:b_j}$$

Aus Lemma 3.46 folgt jetzt:

$$\|e_{j+k}\| \leq C(\max_{0 \leq l \leq k-1} \|e_l\| + \sum_{l=0}^{j} \|b_l\|)$$

Es bleibt also nur noch, $\|b_l\|$ abzuschätzen. Es gilt:

$$i|b_l\| \leq Ch^{p+1} + h \sum_{m=0}^{k} |\beta_m| \|f(t_{l+m}, y_{l+m}) - f(t_{l+m}, u_{l+m})\|$$

und wegen $\|f(t_{l+m}, y_{l+m}) - f(t_{l+m}, u_{l+m})\| \leq L\|y_{l+m} - u_{l+m}\| = L\|e_{l+m}\|$ folgt:

$$\|b_l\| \leq Ch^{p+1} + h \sum_{m=0}^{k} |\beta_m| L \|e_{l+m}\|$$

Weiter folgt:

$$\sum_{l=0}^{j} \|b_l\| \leq C(j+1)h^{p+1} + \sum_{l=0}^{j}\sum_{m=0}^{k} hL|\beta_m|\|e_{l+m}\|$$

Mit eventueller Vergrößerund der Konstanten folgt:

$$\sum_{l=0}^{j} \|b_l\| \leq Ch^p + \underbrace{\sum_{l=0}^{j}\sum_{m=0}^{k} hL|\beta_m|\|e_{l+m}\|}_{\leq \bar{C}h\sum_{l=0}^{j+k}\|e_l\|}$$

und schließlich:

$$\|e_{j+k}\| \leq C(\max_{0\leq l\leq k-1}\|e_l\| + h\sum_{l=0}^{j+k}\|e_l\|)$$

für alle $j \in \mathbb{N}$, mit $t_j \in I$.
Das diskrete Gronwall-Lemma liefert jetzt:

$$\begin{aligned}\|e_{j+k}\| &\leq C(\max_{0\leq l\leq k-1}\|e_l\| + h^p)\underbrace{\exp(C(j+k)h)}_{\leq C}\\ &\leq C(\max_{0\leq l\leq k-1}\|e_l\| + h^p)\end{aligned}$$

Bemerkung

Um optimale Konvergenz zu erreichen, muss $\|e_l\| = O(h^p)$ für $l = 0, \ldots, k-1$ sein. Die Wahl der $u_0, \ldots, u_{k-1}$ geschieht durch eine Anlaufrechnung.
Es reicht, $u_0, \ldots, u_{k-1}$ mit einem Verfahren der Ordnung $p-1$ zu bestimmen (z.B. einem Runge-Kutta-Verfahren).

(Explizite) Adams-Bashforth-Verfahren

Aufgabe: Finde $(\rho, \sigma) \in \mathcal{P}_k \times \mathcal{P}_{k-1}$ mit

1. Forderung: Stabilität
   Aus Konsistenzgründen fordern wir $\rho(1) = 0$, $\lambda = 1$ muss also einfache Nullstelle von ρ sein. Folglich gilt:
   $$\rho(x) = (x-1)\bar{\rho}(x)$$
   mit $\bar{\rho} \in \mathcal{P}_{k-1}, \bar{\rho}(1) \neq 0$ und $|\lambda| \leq 1$ für alle Nullstellen λ von $\bar{\rho}$.
   Wir wählen der Einfachheit halber alle Nullstellen $\lambda = 0$. Folglich ergibt sich (wegen der Normierung ist $\alpha_k = 1$):
   $$\rho(x) = (x-1)x^{k-1} = x^k - x^{k-1}$$

2. Forderung: Möglichst hohe Konvergenzordnung
   Die erste Bedingung $\sum_{l=0}^{k} = 0 = \rho(1)$ ist schon erfüllt. Weitere Konsistenzbedingungen:

$$\sum_{l=0}^{k-1} \beta_l l^{j-1} = \underbrace{\frac{1}{j} \sum_{l=0}^{k} \alpha_l l^j}_{\text{schon festgelegt}} \quad \text{für } j = 1, \ldots, p$$

   Dies ist ein lineares Gleichungssystem in den β_l der Form:

$$\begin{pmatrix} 1 & 1 & 1 & \ldots \\ 0 & 1 & 2 & \ldots \\ \vdots & \vdots & \vdots & \\ 0 & 1 & 2^{p-1} & \ldots \end{pmatrix} \begin{pmatrix} \beta_0 \\ \vdots \\ \\ \vdots \\ \beta_{k-1} \end{pmatrix}$$

   Es handelt sich um ein Vandermonde-System. Dieses ist eindeutig lösbar, falls $p = k$ ist.

   Fazit: Wir haben $\rho(x) = x^k - x^{k-1}$ und durch die obige Gleichung für $p = k$ eindeutig festgelegte $\beta_0, \ldots, \beta_{k-1}$.

   Diese Verfahren heißen explizite Adams-Bashforth-Verfahren.

Eine andere Sichtweise ergibt eine Quadraturformel:

$$\begin{aligned} O(h^{p+1}) &= h\tau(t_j, h, y) \\ &= y(t_j + k) - y(t_{j+k-1}) - h \sum_{l=0}^{k-1} \beta_l y'(t_{j+l}) \\ &= \int_{t_{j+k-1}}^{t_{j+k}} y'(s)ds - \underbrace{h \sum_{l=0}^{k-1} \beta_l y'(t_{j+l})}_{I_h y'} \end{aligned}$$

Dies gilt genau dann, wenn I_h exakt auf $\mathcal{P}_{p-1}$ ist.
Wir haben dabei die Stützstellen $t_j, \ldots, t_{j+k-1}$ verwendet.
Zu gegebenem $t_j, \ldots, t_{j+k-1}$ und Werten $F^j, \ldots, F^{j+k-1}$ existiert genau ein interpolierendes Polynom $q' \in \mathcal{P}_{k-1}$ mit

$$q'(t_{j+l}) = F^{j+l} \text{ für } l = 0, \ldots, k-1$$

Wir können also entweder die Vandermonde-Gleichung lösen oder aber nach geeigneten Quadraturformeln suchen.

Es ergibt sich folgende Tabelle:

$k = p = 1$	$\sigma(x) = 1$ (expl. Euler)
$k = p = 2$	$\sigma(x) = \frac{1}{2}(3x - 1)$
$k = p = 3$	$\sigma(x) = \frac{1}{12}(23x^2 - 16x + 5)$
$k = p = 4$	$\sigma(x) = \frac{1}{24}(55x^3 - 59x^2 + 37x - 9)$

Bemerkung

Geht man von k nach $k + 1$, in der Ordnung also von $p = k$ zu $p = k + 1$, so ist der Mehraufwand in f-Auswertungen Null.
Mehrschrittverfahren haben allerdings auch einen großen Nachteil: Eine Schrittweitensteuerung ist deutlich komplizierter als bei Einschrittverfahren.

Literatur

Peter Deuflhard and Folkmar Bornemann. *Numerische Mathematik II, Gewöhnliche Differentialgleichungen*. de Gruyter, 2002.

Peter Deuflhard and Andreas Hohmann. *Numerische Mathematik I, Eine algorithmisch orientierte Einführung*. de Gruyter, 2002.

Josef Stoer. *Numerische Mathematik 1*. Springer, 2004.

Josef Stoer and Roland Bulirsch. *Numerische Mathematik 2*. Springer, 2004.

Jochen Werner. *Numerische Mathematik 1 + 2*. Vieweg, 1991.

A. Höhere Ableitungen und multi-lineare Abbildungen

Wir betrachten $f : I \times \Omega \to \mathbb{R}^d, f \in C^{p+1}$. Dann ist für $k \leq p+1$:

$$D_y^k f : I \times \Omega \to \mathcal{ML}(\underbrace{\mathbb{R}^d \times \cdots \times \mathbb{R}^d}_{k\text{-mal}}, \mathbb{R}^d)$$

mit

$$D_y^k f(t,z)[y^{(1)}, \ldots, y^{(k)}] = \sum_{i_1,\ldots,i_k \in \{1,\ldots,d\}} \frac{\partial^k f}{\partial y_{i_1} \ldots \partial y_{i_k}}(t,z) y_{i_1}^{(1)} \ldots y_{i_k}^{(k)}$$

Dabei ist $\mathcal{ML}$ der Raum der Multi-Linear-Abbildungen.
Eine Linear-Abbildung ist dabei eine lineare Abbildung $A : \mathbb{R}^d \to \mathbb{R}^d$, eine Bilinear-Abbildung eine Abbildung $B : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d, B(x,y) = z$, die jeweils linear in x und y ist.
Eine k-lineare Abbildung ist dann eine Abbildung $L : \underbrace{\mathbb{R}^d \times \cdots \times \mathbb{R}^d}_{k-\text{mal}} \to \mathbb{R}^d$, die in jeder Komponente linear ist.

$D_y^k f(t,z)$ ist symmetrisch, d.h.

$$D_y^k f(t,z)[y_1, \ldots, y_k] = D_y^k f(t,z)[y_{\sigma(1)} \ldots y_{\sigma(k)}]$$

für alle Permutationen $\sigma : \{1, \ldots, k\} \to \{1, \ldots, k\}$.