Numerik von Differentialgleichungen
Sommersemester 2022
Prof. Dr. Bastian von Harrach
M. Sc. Andrej Brojatsch

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

7. Übungsblatt - Lösungsvorschlag

Aufgabe 7.1 (Votieraufgabe)
Betrachten Sie das AWP

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0, \quad x_0 \in \mathbb{R}, y_0 \in \mathbb{C}^d, \quad f \in GV. \tag{1}$$

Zeigen Sie, dass Runge-Kutta-Verfahren invariant unter Koordinatentransformationen sind. D.h., ist $T \in \mathbb{C}^{d\times d}$ eine reguläre Transformationsmatrix, y_1 die erste Iterierte eines RKVs angewandt auf das AWP (1) und $\hat{y}_1$ jene desselben RKVs angewandt auf das transformierte AWP

$$\hat{y}'(x) = \hat{f}(x, \hat{y}(x)), \quad \hat{y}(x_0) = \hat{y}_0, \quad x_0 \in \mathbb{R}, \hat{y}_0 \in \mathbb{C}^d,$$

mit $\hat{f} : \mathbb{R} \times \mathbb{C}^d \to \mathbb{C}^d$, $\hat{f} : (x, y) \mapsto Tf(x, T^{-1}y)$ und $\hat{y}_0 := Ty_0$, dann gilt:

$$y_1 = T^{-1}\hat{y}_1$$

für h_0 klein genug.

Lösungsvorschlag:

Seien $(\hat{\eta}_j)$ die Zwischenstellen des RKV angewandt auf . Dann gilt

$$\begin{aligned}
\hat{\eta}_j &= \hat{y}_0 + h_0 \sum_l a_{jl} \hat{f}(x_0 + c_l h_0, \hat{\eta}_l) \\
&= Ty_0 + h_0 \sum_l a_{jl} Tf(x_0 + c_l h_0, T^{-1}\hat{\eta}_l) \\
\Longrightarrow T^{-1}\hat{\eta}_j &= y_0 + h_0 \sum_l a_{jl} f(x_0 + c_l h_0, T^{-1}\hat{\eta}_l).
\end{aligned}$$

Da f die GV erfüllt, gilt dass $T^{-1}\hat{\eta}_j = \eta_j$ die Zwischenstellen des RKV angewandt auf 1 sind, für h_0 klein genug. Es folgt

$$\begin{aligned}
\hat{y}_1 &= \hat{y}_0 + h_0 \sum_j b_j \hat{f}(x_0 + c_j h_0, \hat{\eta}_j) = Ty_0 + h_0 \sum_j b_j Tf(x_0 + c_j h_0, T^{-1}\hat{\eta}_j) \\
&= T(y_0 + h_0 \sum_j b_j f(x_0 + c_j h_0, \eta_j)) = Ty_1.
\end{aligned}$$

Aufgabe 7.2 (schriftliche Aufgabe)[6 Punkte]

Gegeben sei das AWP

$$y'(x) = Ay + \varphi(x), \; y(0) = y_0 \in \mathbb{R}^d, \tag{2}$$

mit symmetrischer Matrix $A \in \mathbb{R}^{d\times d}$, sowie $\varphi : \mathbb{R} \to \mathbb{R}^d$ unendlich oft stetig differenzierbar. Seien $(\lambda_k, v_k),\ k = 1, ..., d$ die Eigenpaare von A, wobei die (v_k) eine ONB bilden. Zeigen Sie, dass $y^{(1)} = \sum_{k=1}^d \xi_k^{(1)} v_k$, wobei $y^{(1)}, \xi_k^{(1)}, k = 1, ..., d$ die jeweils ersten Iterierten eines RKVs, mit Schrittweite h_0 klein genug, angewandt auf (2) und

$$\xi'(x) = \lambda_k \xi(x) + (\varphi(x), v_k),\ \xi(0) = (y_0, v_k),\ k = 1, ..., d$$

sind.

<u>Lösungsvorschlag:</u>

Mit

$$D = \begin{pmatrix} \lambda_1 & & \\ & ... & \\ & & \lambda_d \end{pmatrix} \quad \text{und} \quad T = \begin{pmatrix} v_1 & ... & v_d \end{pmatrix},$$

gilt $A = TDT^{-1}$ und $T^{-1} = T^t$. Betrachte das AWP

$$\xi'(x) = D\xi(x) + T^{-1}\varphi(x) =: f(x, \xi(x)),\ \xi(0) = T^{-1}y_0. \tag{3}$$

f ist C^∞ und erfüllt die Voraussetzungen vom Satz von Picard-Lindelöff (globale Version). Damit folgt für die erste Iterierte angewandt auf (3), dass $\xi^{(1)} = \begin{pmatrix} \xi_1^{(1)} & ... & \xi_d^{(1)} \end{pmatrix}^t$ für h_0 klein genug. Weiterhin gilt

$$y'(x) = Ay(x) + \varphi(x) = T(DT^{-1}y(x) + T^{-1}\varphi(x)) = Tf(x, T^{-1}y(x))$$

und mit Teil (a) folgt

$$y^{(1)} = T\xi^{(1)} = \sum_k \xi_k^{(1)} v_k.$$

Aufgabe 7.3 (Programmieraufgabe)[6 Punkte]

(a) Schreiben Sie analog zu Aufgabe 5.4 eine `MATLAB`-Funktion

```
function [ti,yi] = impl_Runge_Kutta_Verfahren(t0,y0,h,f,fy,T,A,b,c),
```

für implizite Runge-Kutta-Verfahren, d.h. A ist keine linke untere Dreiecksmatrix. Verwenden Sie dazu das Newton-Verfahren[1] aus Aufgabe 6.1, mit Abbruchkriterium $\|F(x_k)\| \leq 10^{-8}$. Zusätzlich zu dem Parameter **f** (wie beim expl. Verfahren) soll mit dem Parameter **fy** die Ableitung $f_y(x, y)$ übergeben werden, welche für das Newton-Verfahren benötigt wird.

(b) Erstellen Sie mithilfe Ihrer `MATLAB`-Funktion für alle impliziten RKVs aus Abschnitt 1.3.4 Fehlerplots, wie in Teil (b) von Aufgabe 5.4 beschrieben.

Nebenrechnung: (Analog zum Beweis von 1.26)

(a) Betrachten Sie das Gleichungssystem

$$\eta_j = y_i + h_i \sum_{l=1}^s a_{jl} f(x_i + c_l h_i, \eta_l), \quad j = 1, \ldots, s,$$

[1] Es bietet sich hier an den `MATLAB`\-Operator und die MATLAB-Funktion **kron** zu verwenden.

aus der ersten Formulierung eines allgemeinen RKVs für ein skalares AWP, d.h. $f\colon \mathbb{R}\times\mathbb{R}\to\mathbb{R}$, $f\colon (x,y)\mapsto f(x,y)$ ist eine skalare Funktion.

Entwickeln Sie für dieses Gleichungssystem eine möglichst vereinfachte Iterationsvorschrift des Newton-Verfahrens zur Berechnung des Vektors

$$\eta = (\eta_1, \eta_2, \ldots \eta_s)^T \in \mathbb{R}^s,$$

wobei $\eta^{(0)} = (y_i, y_i, \ldots, y_i)^T \in \mathbb{R}^s$ der Startvektor dieser Iteration sei.

(b) Lösen Sie Teil (a) für ein allgemeines (nicht notwendigerweise skalares) AWP, d.h. $y'(x) = f(x, y(x))$ mit $f : \mathbb{R}\times\mathbb{R}^d \to \mathbb{R}^d$ für $d \in \mathbb{N}_0$.

Lösungsvorschlag:

(a)
- Der folgende Lösungsvorschlag ist so aufgeschrieben, dass man damit schon die Struktur für eine entsprechende Lösung sieht, die auch für d-dim. AWPs gilt. Tatsächlich beinhaltet der Vorschlag nun auch in rot Versionen der entsprechenden Ausdrücke für die d-dimensionale Version des AWPs (Vorischt: ist die Lösung von Teil b). Dabei wird die **Kronecker-Multiplikation** verwendet: Es seinen $M = (m_{ij}) \in \mathbb{R}^{a\times b}$ und $W \in \mathbb{R}^{c\times d}$, dann sei $M \otimes W := (m_{ij}W) \in \mathbb{R}^{ac\times bd}$.
- Zudem ist das Newton-Verfahren so aufgeschrieben, dass es sich in MATLAB besonders einfach und effizient implementieren lässt.
- Wir verwenden folgende Notationen:
  Es sei $1_{a\times b}$ bzw. $0_{a\times b}$ die $(a\times b)$-Matrix die ausschließlich Einsen bzw. Nullen als Einträge hat. Ferner sei $E_{a\times a}$ die $(a\times a)$-Einheitsmatrix.

Nun geht es los: Die Komponente von $\eta = (\eta_1, \eta_2, \ldots, \eta_s)^T$ lösen genau dann die Gleichungen des RKV, wenn η die Nullstelle von $N(\eta)$ (wohldef. nach Satz 1.21) mit

$$\begin{aligned}
N(\eta) &:= \eta - 1_{s\times 1}\cdot y_i - h_i A \cdot \begin{pmatrix} f(x_i + c_1 h_i, \eta_1) \\ f(x_i + c_2 h_i, \eta_2) \\ \vdots \\ f(x_i + c_s h_i, \eta_s) \end{pmatrix} \\
&= \eta - \underbrace{\left(1_{s\times 1}\cdot y_i + h_i A \cdot E_{1\times 1} \cdot \begin{pmatrix} f(x_i + c_1 h_i, \eta_1) \\ f(x_i + c_2 h_i, \eta_2) \\ \vdots \\ f(x_i + c_s h_i, \eta_s) \end{pmatrix} \right)}_{=\Phi(\eta)\ \text{aus dem Beweis zu Satz 1.21}} \\
&= \eta - \left(1_{s\times 1}\otimes y_i + h_i A \otimes E_{d\times d} \cdot \begin{pmatrix} f(x_i + c_1 h_i, \eta_1) \\ f(x_i + c_2 h_i, \eta_2) \\ \vdots \\ f(x_i + c_s h_i, \eta_s) \end{pmatrix} \right)
\end{aligned}$$

ist.

Die Iterationsvorschrift des Newton-Verfahrens hat damit die Gestalt:

$$\eta^{(i+1)} = \eta^{(i)} - (N'(\eta^{(i)}))^{-1} N(\eta^{(i)}), \quad \eta^{(0)} = 1_{s\times 1}\cdot y_i, \quad i = 0, 1, 2, \ldots$$

Berechnung der Jacobimatrix $N'(\eta)$:

$$\begin{aligned}
N'(\eta) &= E_{s\times s} - \left(0_{s\times s} + h_i \sum_{l=1}^{s} \begin{pmatrix} a_{1l}\frac{\mathrm{d}}{\mathrm{d}\eta} f(x_i + c_l h_i, \eta_l) \\ a_{2l}\frac{\mathrm{d}}{\mathrm{d}\eta} f(x_i + c_l h_i, \eta_l) \\ \vdots \\ a_{sl}\frac{\mathrm{d}}{\mathrm{d}\eta} f(x_i + c_l h_i, \eta_l) \end{pmatrix} \right) \\
&= E_{s\times s} - \sum_{l=1}^{s} \underbrace{\begin{pmatrix} 0_{1\times 1} & \cdots & 0_{1\times 1} & a_{1l} f_y(x_i + c_l h_i, \eta_l) & 0_{1\times 1} & \cdots & 0_{1\times 1} \\ 0_{1\times 1} & \cdots & 0_{1\times 1} & a_{2l} f_y(x_i + c_l h_i, \eta_l) & 0_{1\times 1} & \cdots & 0_{1\times 1} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0_{1\times 1} & \cdots & 0_{1\times 1} & a_{sl} f_y(x_i + c_l h_i, \eta_l) & 0_{1\times 1} & \cdots & 0_{1\times 1} \end{pmatrix}}_{\text{Die } l\text{-te Spalte ist die mit den “}f_y\text{”-Termen.}} \\
&= E_{s\times s} - \underbrace{h_i \left[a_{jl} f_y(x_i + c_l h_i, \eta_l) \right]_{j,l=1}^{s}}_{=\Phi'(\eta) \text{ aus dem Beweis zu Satz 1.21}} \\
&= E_{s\times s} - h_i \left[A_1 \cdot f_y(x_i + c_1 h_i, \eta_1), A_2 \cdot f_y(x_i + c_2 h_i, \eta_2), \dots, A_s \cdot f_y(x_i + c_s h_i, \eta_s) \right] \\
&= E_{ds\times ds} - h_i \left[A_1 \otimes f_y(x_i + c_1 h_i, \eta_1), A_2 \otimes f_y(x_i + c_2 h_i, \eta_2), \dots, A_s \otimes f_y(x_i + c_s h_i, \eta_s) \right],
\end{aligned}$$

wobei A_i ($= A(:, i)$) die i-te Spalte von A sei.

(b) Wir verwenden ein RKV um y_{i+1} zu bestimmen. Dabei wissen wir, dass jedes $\eta_j \in \mathbb{R}^d$ für $j = 1, \dots, s$ seine entsprechende Gleichung lösen muss

$$\eta_j = y_i + h_i \sum_{l=1}^{s} a_{jl} f(x_i + c_l h_i, \eta_l).$$

Sei $\eta := (\eta_1, \eta_2, \dots, \eta_s)^T \in R^{ds}$, dann löst η

$$\eta = 1_{s\times 1} \otimes y_i + h_i \left(A \otimes E_{d\times d} \right) \begin{pmatrix} f(x_i + c_1 h_i, \eta_1) \\ f(x_i + c_2 h_i, \eta_2) \\ \vdots \\ f(x_i + c_s h_i, \eta_s) \end{pmatrix},$$

mit $A := (a_{ij})_{1 \le i,j \le s}$.

Wir verwenden die Newton Iteration angewandt auf $N(\eta) = 0$ um η zu bestimmen. Es ist

$$N(\eta) := \eta - \underbrace{\left(1_{s\times 1} \otimes y_i + h_i \left(A \otimes E_{d\times d} \right) \begin{pmatrix} f(x_i + c_1 h_i, \eta_1) \\ f(x_i + c_2 h_i, \eta_2) \\ \vdots \\ f(x_i + c_s h_i, \eta_s) \end{pmatrix} \right)}_{=\Phi(\eta) \text{ aus dem Beweis zu Satz 1.19}}.$$

Die Iterationsvorschrift des Newton-Verfahrens hat damit die Gestalt:

$$\eta^{(0)} = 1_{s\times 1} \otimes y_i$$

und

$$\eta^{(i+1)} = \eta^{(i)} - \left(N'(\eta^{(i)}) \right)^{-1} N(\eta^{(i)}), \quad i = 0, 1, 2, \dots$$

Berechnung der Jacobimatrix $N'(\eta)$:

$$N'(\eta) \quad = \quad E_{ds\times ds} - h_i\,(A\otimes E_{d\times d})\begin{pmatrix} \frac{\mathrm{d}}{\mathrm{d}\eta} f(x_i+c_1h_i,\eta_1) \\ \frac{\mathrm{d}}{\mathrm{d}\eta} f(x_i+c_2h_i,\eta_2) \\ \vdots \\ \frac{\mathrm{d}}{\mathrm{d}\eta} f(x_i+c_sh_i,\eta_s) \end{pmatrix}.$$

Sei $f := (f_1, f_2, \ldots, f_d)^\top \in \mathbb{R}^d$. Da $f : (t,z) \in \mathbb{R}\times\mathbb{R}^d \mapsto f(t,z) \in \mathbb{R}^d$, gilt

$$\frac{\mathrm{d}}{\mathrm{d}\eta} f(x_i+c_1h_i,\eta_1) = \begin{pmatrix} f_z(x_i+c_1h_i,\eta_1) & 0_{d\times(s-1)d}\end{pmatrix} \in \mathbb{R}^{d\times ds},$$

mit $f_z(x_i+c_1h_i,\eta_1) := \frac{\partial f}{\partial z}(x_i+c_1h_i,\eta_1) = \left(\frac{\partial f_i}{\partial z_j}(x_i+c_1h_i,\eta_1)\right)_{i,j=1}^{d} \in \mathbb{R}^{d\times d}$.

Ähnlich erhält man $\frac{\mathrm{d}}{\mathrm{d}\eta} f(x_i+c_lh_i,\eta_l), l=2,\ldots,s$. Damit ist

$$N'(\eta) \quad = \quad E_{ds\times ds} - h_i\left[A_1\otimes f_z(x_i+c_1h_i,\eta_1), A_2\otimes f_z(x_i+c_2h_i,\eta_2),\ldots, A_s\otimes f_z(x_i+c_sh_i,\eta_s)\right],$$

wobei A_i $(= A(:,i))$ die i-te Spalte von A sei.

Definition der Kronecker-Multiplikation: Es seinen $M = (m_{ij}) \in \mathbb{R}^{a\times b}$ und $W \in \mathbb{R}^{d\times c}$, dann sei $M\otimes W := (m_{ij}W) \in \mathbb{R}^{ad\times bc}$. Bzw. siehe Bemerkung 2.6 der Vorlesung.

Siehe m-files.

Aufgabe 7.4 (Votieraufgabe)
Sei $R(\zeta) := 1+\zeta b^T(I-\zeta A)^{-1}\mathbb{1} \in \mathbb{C}$ die Stabilitätsfunktion einer allgemeinen Runge-Kutta Methode. Bestimmen Sie die Stabilitätsfunktionen der folgenden Verfahren: Verfahren von Runge, Verfahren von Heun und implizite Mittelpunktsregel. Plotten Sie das jeweilige Stabilitätsgebiet (Definition 1.38) dieser Methoden in `MATLAB`.

Lösungsvorschlag:

Verfahren von Runge:

$$\begin{aligned} y_{i+\frac{1}{2}} &= y_i + \frac{h_i}{2} f(x_i, y_i), \\ y_{i+1} &= y_i + h_i f\left(\frac{x_{i+1}+x_i}{2}, y_{i+\frac{1}{2}}\right); \end{aligned}$$

Das Butcher-Tableau:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}.$$

Die Stabilitätsfunktion:

$$\begin{aligned} R(\zeta) &= 1+\zeta b^T(I-\zeta A)^{-1}\mathbb{1} \\ &= 1+\zeta\begin{pmatrix}0 & 1\end{pmatrix}\left[\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} - \zeta\begin{pmatrix}0 & 0\\ \frac{1}{2} & 0\end{pmatrix}\right]^{-1}\begin{pmatrix}1\\1\end{pmatrix} \\ &= 1+\zeta\begin{pmatrix}0 & 1\end{pmatrix}\begin{pmatrix}1 & 0\\ \frac{\zeta}{2} & 1\end{pmatrix}\begin{pmatrix}1\\1\end{pmatrix} \end{aligned}$$

$$= \quad 1+\zeta\left(\frac{\zeta}{2}+1\right)=\frac{\zeta^2}{2}+\zeta+1.$$

Verfahren von Heun:

$$\begin{aligned}\eta &= y_i + h_i f(x_i, y_i)\\ y_{i+1} &= y_i + h_i\frac{f(x_i,y_i)+f(x_{i+1},\eta)}{2}.\end{aligned}$$

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}.$$

Die Stabilitätsfunktion:

$$\begin{aligned} R(\zeta) &= 1+\zeta b^T(I-\zeta A)^{-1}\mathbb{1}\\ &= 1+\zeta\begin{pmatrix}1/2 & 1/2\end{pmatrix}\left[\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}-\zeta\begin{pmatrix}0 & 0\\ 1 & 0\end{pmatrix}\right]^{-1}\begin{pmatrix}1\\1\end{pmatrix}\\ &= 1+\zeta\begin{pmatrix}1/2 & 1/2\end{pmatrix}\begin{pmatrix}1 & 0\\ \zeta & 1\end{pmatrix}\begin{pmatrix}1\\1\end{pmatrix}\\ &= 1+\zeta\left(\frac{\zeta}{2}+1\right)=\frac{\zeta^2}{2}+\zeta+1. \end{aligned}$$

impl. Mittelpunktregel: Achtung: wurde schon in der Vorlesung besprochen.

$$\begin{aligned} y_{i+\frac{1}{2}} &= y_i+\frac{h_i}{2}f\left(\frac{x_{i+1}+x_i}{2}, y_{i+\frac{1}{2}}\right),\\ y_{i+1} &= y_i+h_i f\left(\frac{x_{i+1}+x_i}{2}, y_{i+\frac{1}{2}}\right); \end{aligned}$$

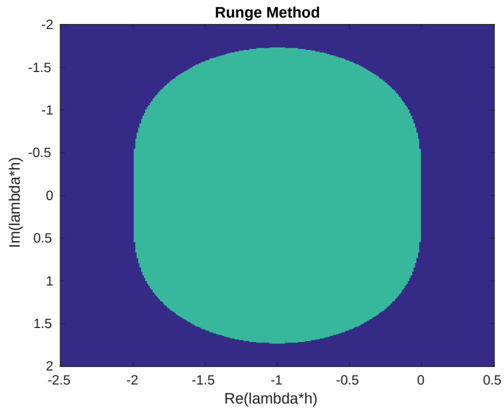
Das Butcher-Tableau:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}.$$

Die Stabilitätsfunktion:

$$\begin{aligned} R(\zeta) &= 1+\zeta b^T(I-\zeta A)^{-1}\mathbb{1}\\ &= 1+\zeta 1(1-\zeta/2)^{-1}1\\ &= \frac{1+\zeta/2}{1-\zeta/2}. \end{aligned}$$

Das Stabilitätsgebiet ist der grüne Bereich in Figur 1, die Stabilitätsfunktionen stimmen überein.


Runge Method
Im(lambda*h)
Re(lambda*h)
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
-2.5
-2
-1.5
-1
-0.5
0
0.5