Numerik von Differentialgleichungen
Sommersemester 2018
Prof. Dr. Bastian von Harrach
M. Sc. Tim Jahn

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Kombiniertes 5. und 6. Übungsblatt (erschienen am 10.05.2017)

Aufgabe 5.1 (schriftliche Aufgabe)[1+3 Punkte]

(a) Bestimmen Sie die maximale Konsistenzordnung für die Verfahren aus Abschnitt 1.3.4 der Vorlesung.

(b) Prüfen Sie für die drei RKVs (geg. durch die Butcher-Tableaus)

$$\begin{array}{c|c} c_1 & A_1 \\ \hline & {b_1}^T \end{array} := \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 2/6 & 2/6 & 1/6 \end{array}, \quad \begin{array}{c|c} c_2 & A_2 \\ \hline & {b_2}^T \end{array} := \begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1/6 & 2/6 & 2/6 & 1/6 & 0 \\ \hline & 1/6 & 2/6 & 2/6 & 0 & 1/6 \end{array}$$

und $\begin{array}{c|c} c_2 & A_2 \\ \hline & \left({b_1}^T\, 0\right) \end{array}$, ob sie mindestens die Konsistenzordnung 3 besitzen.

Aufgabe 5.2 (Programmieraufgabe zum Votieren)

Bei der Implementierung eines Einschrittverfahrens ist es sinnvoll die Schrittweite pro Schritt gerade so klein zu wählen, dass eine gewünschte Genauigkeit noch eingehalten wird. Effizient lässt sich dies für geeignete RKVs mithilfe eines eingebetteten Kontrollverfahrens realisieren.

Vorgehensweise: Zu einem gegebenen AWP werden die Näherungen y_{i+1} bzw. $\hat{y}_{i+1}$ an die Lösung y mit einem s-stufigen RKV (geg. durch $A, b1, c$) mit Konsistenzordnung p bzw. mit einem s-stufigen RKV (geg. durch $A, b2, c$) mit Konsistenzordnung $\hat{p} = p + 1$ berechnet. Da beide Verfahren mit dem gleichen A und c arbeiten, können sie weitestgehend simultan berechnet werden, so dass der Rechenaufwand nicht sonderlich erhöht wird. Man spricht dann von eingebetteten RKVs, die sich mit dem erweiterten Butcher-Tableau

$$\begin{array}{c|c} c & A \\ \hline & b1^T \\ \hline & b2^T \end{array}$$

beschreiben lassen. Nach einem Schritt[1]

$$y_{i+1} = \hat{y}_i + h_i \sum_{j=1}^{s} b1_j f(x_i + c_j h_i, \eta_j), \quad \hat{y}_{i+1} = \hat{y}_i + h_i \sum_{j=1}^{s} b2_j f(x_i + c_j h_i, \eta_j)$$

wird mithilfe des Kontrollverfahrens der Fehlerschätzer $\delta_{i+1} = \|y_{i+1} - \hat{y}_{i+1}\|$ berechnet. Abhängig von einer (selbst vorgegebenen) Fehlertoleranz `tol` > 0 und einem Parameter τ (der etwas kleiner als 1

[1] Bei der Berechnung von y_{i+1} "startet" man bewusst bei $\hat{y}_i$.

gewählt werden sollte) wird die Schrittweitengröße[2]

$$h = \tau \left(\frac{\texttt{tol}}{\delta_{i+1}}\right)^{1/(p+1)} h_i$$

angepasst. Gilt nun $\delta_{i+1} < \texttt{tol}$, dann wird mit $h_{i+1} = h$ der nächste Schritt begonnen, andernfalls wird der letzte Schritt mit der verkleinerten Schrittweite $h_i = h$ wiederholt.

(a) Schreiben Sie eine MATLAB-Funktion

```
[xi,hat_yi] = RKV_expl_embedded(A,b1,b2,c,f,y0,x0,T,h0,tol,tau,p),
```

die nach obigem Schema (adaptive Schrittweitensteuerung mit $h_0 = \text{h0}, \text{tau} = \tau$) für ein explizites eingebettetes RKV den diskreten Näherungsgraphen $[\texttt{xi}, \texttt{hat\_yi}]$ an den Lösungsgraphen des AWPs $y'(x) = f(x, y(x)),\ y(\text{x0}) = \text{y0} \in \mathbb{R}^d$ auf [x0,T] berechnet.

(b) Testen Sie ihr Programm anhand des AWPs

$$\dot{y}_1(x) = 1 + y_1(x)^2 y_2(x) - 4y_1(x),\ \dot{y}_2(x) = 3y_1(x) - y_1(x)^2 y_2(x), \quad y_1(0) = 1.01,\ y_2(0) = 3$$

auf dem Intervall $[0, 20]$ mit dem eingebetteten RKV $\begin{array}{c|c} c_2 & A_2 \\ \hline & b_2^T \\ \hline & \begin{pmatrix} b_1^T & 0 \end{pmatrix} \end{array}$,

wobei A_2, b_1, b_2, c_1 wie in Aufgabe 5/6.1 (b) angegeben und $\texttt{h0} = 1$, $\texttt{tol} = 10^{-5}$, $\tau = 0.9$ sowie $p = 3$ seien. Plotten Sie

(i) die Näherungsgraphen zu $[\texttt{xi}, \texttt{hat\_yi}(1,:)]$ und $[\texttt{xi}, \texttt{hat\_yi}(2,:)]$,

(ii) die Schrittweiten $\texttt{hi}$ gegen $\texttt{xi}$ mit $\texttt{hi} = \texttt{xi}(2:\texttt{end}) - \texttt{xi}(1:\texttt{end}-1)$.

Aufgabe 6.1 (Votieraufgabe)
Gegeben Sei ein autonomes AWP

$$y : \mathbb{R} \to \mathbb{R}^d,\ y'(x) = f(y(x)), \quad y(0) = y_0.$$

Betrachten Sie das zugehörige linearisierte AWP

$$y : \mathbb{R} \to \mathbb{R}^d,\ y'(x) = f(y_0) + f'(y_0)(y(x) - y_0),\ y(0) = y_0. \tag{1}$$

Nun soll angenommen werden, dass $f'(y_0) \in \mathbb{R}^{d\times d}$ regulär und diagonalisierbar ist, d.h. es existiert eine reguläre Diagonalmatrix $D \in \mathbb{C}^{d\times d}$ und eine reguläre Matrix $T \in \mathbb{C}^{d\times d}$, so dass $f'(y_0) = TDT^{-1}$ ist. Bestimmen Sie $v \in \mathbb{C}^d$ und $z_0 \in \mathbb{C}^d$, so dass für die Lösung des AWP

$$z' = Dz, \quad z(0) = z_0$$

gilt, dass $y(x) = Tz(x) + v$ das linearisierte AWP (1) löst.

[2] Überlegen Sie sich, warum diese Formel Sinn macht und wie man den Fall $\delta_{i+1} = 0$ einfach abfangen kann.

Aufgabe 6.2 (Votieraufgabe)
Betrachten Sie das AWP

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0, \quad x_0 \in \mathbb{R}, y_0 \in \mathbb{C}^d, \quad f : \mathbb{R} \times \mathbb{C}^d \to \mathbb{C}^d. \tag{2}$$

Zeigen Sie, dass Runge-Kutta-Verfahren invariant unter Koordinatentransformationen sind, d.h., ist $T \in \mathbb{C}^{d\times d}$ eine reguläre Transformationsmatrix, $y_0, y_1, y_2, \ldots, y_N$ die Iterierten eines RKVs angewandt auf das AWP (2) und $\hat{y}_0, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N$ die Iterierten desselben RKVs angewandt auf das transformierte AWP

$$\hat{y}'(x) = \hat{f}(x, \hat{y}(x)), \quad \hat{y}(x_0) = \hat{y}_0, \quad x_0 \in \mathbb{R}, \hat{y}_0 \in \mathbb{C}^d \tag{3}$$

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:

$$y_i = T^{-1}\hat{y}_i$$

für $i = 0, 1, 2, \ldots, N$ und $0 < h_i < h_0$, $\forall i$ und h_0 klein genug.

Aufgabe 6.3 (Programmieraufgabe)[5 Punkte]

(a) Schreiben Sie analog zu Teil (a) von Aufgabe 4.5 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[3] aus Aufgabe 4.4. 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 Fehlerplots wie in Teil (b) von Aufgabe 4.5 beschrieben für alle impliziten RKVs aus Abschnitt 1.3.4.

Hinweise zur Übungsblattbearbeitung:

- Zu **schriftlichen Aufgaben** soll eine Ausarbeitung/Lösung angefertigt werden, die bis zum 24.05.2017 um 10:00 Uhr in das Fach 42 im 3. Stock der Robert-Mayer-Str. 6-8 abzugeben ist. Sollte ein Übungstermin nicht wahrgenommen werden können, so kann die Abgabe der schriftlichen Aufgabe auch bis zum obigen Zeitpunkt an ihren Übungleiter geschickt werden.
- Zu **Programmieraufgaben** soll bis zum 24.05.2017 um 10:00 Uhr eine **kommentierte** Ausarbeitung in `MATLAB`-Code an ihren Übungleiter geschickt werden. Bitte beginnen Sie die Betreffzeile Ihrer E-Mail mit **"DGL5/6_2018_Gruppennummer:"** (wenn Sie z.B. in Gruppe 2 sind, so soll die Betreffzeile mit "DGL5/6_2018_2:"beginnen).
- Zu **Votieraufgaben** wird keine schriftliche Abgabe verlangt. Die Lösung wird in der Übung besprochen.
- Eine **Programmieraufgabe zum Votieren** soll wie eine normale Votieraufgabe sinnvoll bearbeitet/gelöst werden und wird in der Übung besprochen.

[3] Es bietet sich hier an den `MATLAB`\-Operator und die MATLAB-Funktion **kron** zu verwenden. Die Wahl einer geeigneten Abbruchbedingung für das Newton-Verfahren steht Ihnen frei.