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

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

6. Übungsblatt (erschienen am 18.05.2022)

Aufgabe 6.1 (Votieraufgabe)
Betrachten Sie das nicht autonome d-dimensionale AWP

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

und das dazugehörige autonome (d+1)-dimensionale AWP

$$Y'(x) = \begin{pmatrix} Y_1'(x) \\ Y_2'(x) \end{pmatrix} = F(Y(x)) = \begin{pmatrix} 1 \\ f(Y_1(x), Y_2(x)) \end{pmatrix}, \quad Y(a) = \begin{pmatrix} a \\ y_0 \end{pmatrix} \in \mathbb{R}^{d+1}, \tag{2}$$

wobei $f \in \mathcal{C}^\infty([a,b] \times \mathbb{R}^d)$.

(a) Beweisen Sie, dass AWP (1) und AWP (2) äquivalent sind, d.h. $y(x)$ ist genau dann die Lösung vom AWP (1), wenn $Y(x) = \begin{pmatrix} x \\ y(x) \end{pmatrix}$ die Lösung vom AWP (2) ist.

(b) Es seien $y^{(1)} \approx y(a+h_0)$ und $Y^{(1)} \approx Y(a+h_0)$ die Approximationen an die Lösungen der AWPs, die man nach einem Schritt mit einem beliebigen aber festen RKV für die Schrittweite $h_0 > 0$ klein genug erhält. Zeigen Sie, dass $\begin{pmatrix} a+h_0 \\ y^{(1)} \end{pmatrix} = Y^{(1)}$ gilt.

Aufgabe 6.2 (Schriftliche Aufgabe)[3 Punkte]
Bestimmen Sie die die maximale Konsistenzordnung der Verfahren aus Abschnitt 1.3.4.

Aufgabe 6.3 (Schriftliche Aufgabe)[3 Punkte]
Zeigen Sie, dass die implizite Mittelpunktsregel das einzige 1-stufige Runge-Kutta-Verfahren mit Konsistenzordnung 2 ist.

Aufgabe 6.4 (Programmieraufgabe)[6 Punkte]
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, sodass der Rechenaufwand nicht sonderlich erhöht wird. Man spricht dann von eingebetteten RKVs, die sich mit dem erweiterten Butcher-Tableau

c	A
	$b1^T$
	$b2^T$

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 Kotrollverfahrens der Fehlerschätzer $\delta_{i+1} = \|y_{i+1} - \hat{y}_{i+1}\|$ berechnet. Abhängig von einer (selbst vorgegebenen) Fehlertoleranz $\texttt{tol} > 0$ und einem Parameter τ (der etwas kleiner als 1 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 obigen Schema (adaptive Schrittweitensteuerung mit $h_0 = \text{h0}, \text{tau} = \tau$) für ein explizites eingebettetes RKV den diskreten Näherungsgraphen [xi, 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

$$\begin{array}{c|c} c & A \\ \hline & b1^T \\ \hline & b2^T \end{array} := \begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1/6 & 2/6 & 2/6 & 1/6 & 0 \\ \hline & 1/6 & 2/6 & 2/6 & 0 & 1/6 \\ \hline & 1/6 & 2/6 & 2/6 & 1/6 & 0 \end{array}$$

und $\texttt{h0} = 1$, $\texttt{tol} = 10^{-5}$, $\tau = 0.9$ sowie $p = 3$ seien. Plotten Sie

(i) die Näherungsgraphen zu [xi, hat_yi(1, :)] und [xi, hat_yi(2, :)],

(ii) die Schrittweiten hi gegen xi mit hi = xi(2 : end) − xi(1 : end − 1).

- Zu den **schriftlichen Aufgaben*** und **Programmieraufgaben** * soll eine Ausarbeitung/Lösung angefertigt werden, die bis zum 27.05.2022 um 12:00 Uhr in Fach 17 im 3. Stock der Robert-Mayer-Str. 6-8 abzugeben ist.

- Zu **Programmieraufgaben** ist ein **kommentierter** MATLAB-Quellcode zu schreiben, welcher zusammen mit den damit erstellten Plots ausgedruckt werden soll. Der Code ist nicht per Mail einzureichen.

- Zu **Votieraufgaben** wird keine schriftliche Abgabe verlangt. Die Lösung wird in der Übung besprochen.

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

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

*__Die Abgabe und Bearbeitung darf in Zweiergruppen erfolgen.__