Numerik von Differentialgleichungen
Sommersemester 2019
Dr. Sarah Eberle
M. Sc. Tim Jahn

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

5. Übungsblatt (erschienen am 14.05.2019)

Aufgabe 5.1 (schriftliche Aufgabe)[1+3+1 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.

(c) Zeigen Sie, dass es kein von b_1 verschiedenes $\hat{b}$ gibt, sodass das RKV $\begin{array}{c|c} c_1 & A_1 \\ \hline & \hat{b}^T \end{array}$ Konsistenzordnung 3 besitzt.

Aufgabe 5.2 (Programmieraufgabe zum Votieren)

Erstellen Sie einen Fehlerplot wie in Aufgabe 4.5 (b) (für die gleiche Problemstellung und die gleichen Schrittweiten) für die Verfahren aus Aufgabe 5.1 (b) und schließen Sie von der Steigung der Graphen auf die Konsistenzordnung der Verfahren.

Aufgabe 5.3 (Programmieraufgabe)[5 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

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

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 Kotrollverfahrens 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 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 $h_0 = \mathrm{h0}, \mathrm{tau} = \tau$) für ein explizites eingebettetes RKV den diskreten Näherungsgraphen [`xi`, `hat_yi`] an den Lösungsgraphen des AWPs $y'(x) = f(x, y(x)),\ y(\mathrm{x0}) = \mathrm{y0} \in \mathbb{R}^d$ 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 $\begin{array}{c|c} c_2 & A_2 \\ \hline & b_2^T \\ \hline & (b_1^T\ 0) \end{array}$,

wobei A_2, b_1, b_2, c_1 wie in Aufgabe 5.2 (b) angegeben und `h0` $= 1$, `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)`.

Hinweise zur Übungsblattbearbeitung:

- Zu **schriftlichen Aufgaben** soll eine Ausarbeitung/Lösung angefertigt werden, welche am 21.05.2019 in der Übung abzugeben ist.

- Zu **Programmieraufgaben** soll bis zum 21.05.2019 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_2019:"**

- 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. Diese wird wie eine normale Votieraufgabe in den Übungen besprochen und diskutiert.

[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.