Prof. Dr. Karsten Urban
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

WiSe 2017/2018
Blatt 07
01.12.2017

Angewandte Numerik 2

Besprechung in den Tutorien in der Woche vom 04.12.2017 bis 08.12.2017

Für dieses Übungsblatt gibt es 16 Theorie- und 20 Matlab-Punkte, sowie 4 Theorie- und 4 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 60-Prozent-Grenzen liegen aktuell (inklusive Blatt 07) bei 77,4 Theoriepunkten und 79,2 Matlabpunkten.

Aufgabe 27 *(Explizite Runge-Kutta-Verfahren)* (3T+5T+2T+4T* Punkte)

a) Zeigen Sie, dass das Verfahren von Heun (zweiter Ordnung) aus Aufgabe 24 vom letzten Übungsblatt ein zweistufiges Runge-Kutta-Verfahren ist. Geben Sie die Funktionen $k_l(t, y)$, die Werte für die Parameter α_r, β_{rl} und γ_l sowie die Verfahrensfunktion $F(f, t, h, y)$ an. Stellen Sie dieses Verfahren durch ein Butcher-Tableau dar.

b) Leiten Sie das Verfahren von Heun (dritter Ordnung) aus Beispiel 3.4.7 über den Ansatz auf Seite 78 des Skripts her. Geben Sie dazu die Funktionen $k_1(t, y)$, $k_2(t, y)$ und $k_3(t, y)$ sowie die Verfahrensfunktion F an. Welche Quadraturformeln werden bei $r = 2$ und $r = 3$ zur näherungsweisen Berechnung der $y(s_r)$ in $f(s_r, y(s_r)) \approx k_r(t, y)$ verwendet?

c) Zeigen Sie mit Satz 3.4.5, dass

   i) das Verfahren von Heun zweiter Ordnung genau die Konsistenzordnung 2 und

   ii) das Verfahren von Heun dritter Ordnung genau die Konsistenzordnung 3

   hat.

Aufgabe 28 *(Auch Runge-Kutta-Verfahren sind Einschrittverfahren)*
(5M+2M+2M+2M*+2M*+3T Punkte)

In dieser Aufgabe wollen wir den Vergleich verschiedener Einschritt-Verfahren aus Aufgabe 24 des letzten Übungsblattes um einige Runge-Kutta-Verfahren erweitern.

a) Schreiben Sie analog zu Aufgabe 24 eine Matlabfunktion `yk = rungeKutta(f, y0, tk, bt)`, die analog zu `eulerImplizit` für einen gegebenen Startwert $y^0 \in \mathbb{R}^n$, $n \in \mathbb{N}$, eine Lösung $y : \mathbb{R} \to \mathbb{R}^n$, $n \in \mathbb{N}$ der Anfangswertaufgabe

$$y' = f(t, y), \qquad y(t_0) = y_0$$

mit einem Runge-Kutta-Verfahren berechnet. Die Parameter sind dabei wie in Aufgabe 24: `f` ist die Funktion f als *function handle*, `y0` ist der Startwert $y_0 \in \mathbb{R}^n$ und `tk` ist ein Gitter mit den diskreten Zeitpunkten t_k. Der Rückgabewert `yk` ist der Vektor der einzelnen Näherungswerte y^k.

Ihre Matlabfunktion `yk = rungeKutta(f, y0, tk, bt)` soll den Algorithmus eines expliziten Runge-Kutta-Verfahrens unabhängig von den konkreten Werten für α, β und γ realisieren. Die konkreten Werte für α, β und γ, also das Butcher-Tableau, soll ihre Matlabfunktion `yk = rungeKutta(f, y0, tk, bt)` über den Parameter `bt` erhalten. `bt` soll dabei eine Struktur mit den Komponenten

i) `bt.m` für die Stufenanzahl des Runge-Kutta-Verfahrens,

ii) `bt.alpha` für den Spaltenvektor $\alpha = (\alpha_1, \ldots, \alpha_m)$ des Runge-Kutta-Verfahrens,

iii) `bt.beta` für die Matrix $\beta = (\beta_{i,j})$ $(i, j = 1, \ldots, m)$ des Runge-Kutta-Verfahrens und

iv) `bt.gamma` für den Zeilenvektor $\gamma = (\gamma_1, \ldots, \gamma_m)$ des Runge-Kutta-Verfahrens

sein. Die Matrix β soll dabei auf und oberhalb der Diagonalen nur Werte $\beta_{i,j} = 0$ enthalten.

Achten Sie darauf, dass Ihre Matlabfunktion `yk = rungeKutta(f, y0, tk, bt)` auch für Systeme von Differentialgleichungen geeignet ist. Diese Eigenschaft werden Sie für die Aufgabe 30 (Zeeman'sches Herzschlagmodell) benötigen.

b) Schreiben Sie eine Matlabfunktion `bt = rKverbEuler()`, die das Butcher-Tableau des verbesserten Euler-Verfahrens zurück gibt. Der Rückgabewert `bt` soll also vom Typ der oben beschriebenen Struktur mit den Komponenten `bt.m`, `bt.alpha`, `bt.beta` und `bt.gamma` sein.

c) Erweitern Sie Ihr Matlabskript `vergleichESV` zum Vergleich verschiedener Einschrittverfahren aus Aufgabe 24 des letzten Übungsblattes. Bestimmen Sie mit `yk = rungeKutta(f, y0, tk, bt)` und `bt = rKverbEuler()` sowie den Schrittweiten $h = \frac{1}{20}$ und $h = \frac{1}{40}$ zwei weitere Näherungslösungen der Anfangswertaufgabe aus Aufgabe 24. Zeichnen Sie diese Näherungslösungen und die exakte Lösung in ein neues Schaubild ein, und ergänzen Sie Ihr Schaubild mit den Gitterfehlern um die Gitterfehler der beiden neuen Lösungen.

d) Schreiben Sie weitere Matlabfunktionen `bt = rKHeun3()` und `bt = rK4()`, die die Butcher-Tableaus des Verfahrens von Heun (der Ordnung 3) und des klassischen Runge-Kutta-Verfahrens (Simpson Regel) zurückgeben.

e) Bestimmen Sie auch mit `yk = rungeKutta(f, y0, tk, bt)`, den Butcher-Tableaus `bt = rKHeun3()` und `bt = rK4()` sowie den Schrittweiten aus Aufgabenteil c) je zwei Näherungslösungen der Anfangswertaufgabe, die Sie zusammen mit der exakten Lösung jeweils in ein neues Schaubild eintragen. Ergänzen Sie Ihr Schaubild mit den Gitterfehlern um die Gitterfehler der beiden neuen Lösungen.

f) Erläutern Sie die Schaubilder. Vergleichen Sie alle sieben Verfahren.

Aufgabe 29 *(Programmieraufgabe: Fortsetzung Konvergenzordnung von ESV)* (3M+2M+3T Punkte)
Ergänzen Sie auch Ihr Matlabskript `konvergenzESV` aus Aufgabe 25 von Blatt 6 um das verbesserte Euler-Verfahren, das Verfahren von Heun (der Ordnung 3) und das klassische Runge-Kutta-Verfahren.

a) Plotten Sie wieder für das Anfangswertproblem vom letzten Übungsblatt und für die Schrittweiten $0.5^4, \ldots, 0.5^{14}$ jeweils den Diskretisierungsfehler doppelt logarithmisch über den Schrittweiten.

b) Zeichnen Sie in Ihr Schaubild wieder Geraden ein, mit deren Hilfe Sie die Steigungen der geplotteten Diskretisierungsfehler abschätzen können.

c) Interpretieren Sie das Schaubild. Bestätigen Ihre numerischen Ergebnisse die theoretischen Überlegungen zu den verwendeten Runge-Kutta-Verfahren?

Aufgabe 30 *(Programmieraufgabe: Explizite Runge-Kutta-Verfahren und Zeemann's Herzschlagmodell)* (6M Punkte)
Haben Sie in der obigen Aufgabe 28 darauf geachtet, dass Ihre Matlabfunktion `yk = rungeKutta(f, y0, tk, bt)` auch für Systeme von Differentialgleichungen geeignet ist? Testen Sie dies mit dem klassischen Runge-Kutta-Verfahren der Ordnung 4 am Zeemann'schen Herzschlagmodell aus Aufgabe 26 von Blatt 6.

Hinweise:
Die Programmieraufgaben sind in Matlab zu erstellen. Der Source Code muss strukturiert und dokumentiert sein. Senden Sie **spätestens 24 Stunden vor Ihrem Tutorium** alle Matlab-Files und alle Ergebnisse in einer E-mail mit dem Betreff **Loesung-Blatt07** an **angewandte.numerik@uni-ulm.de**.