Prof. Dr. Stefan Funken
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

WiSe 2016/2017
Blatt 10
11.01.2017

Angewandte Numerik 2

Besprechung in den Tutorien in der Woche vom 16.01.2017 bis 20.01.2017

Für dieses Übungsblatt gibt es 19 Theorie- und 22 Matlab-Punkte, sowie 10 Theorie- und 13 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 60-Prozent-Grenzen liegen aktuell (inklusive Blatt 10) bei 90,0 Theoriepunkten und 95,4 Matlabpunkten.

Aufgabe 31 *(Programmieraufgabe: Weitere Einschrittverfahren)* (6M+2M+3M+2M*+2M*+5T Punkte)

In dieser Aufgabe wollen wir den Vergleich verschiedener Einschritt-Verfahren aus Aufgabe 30 des letzten Übungsblattes um zwei Runge-Kutta-Verfahren erweitern.

a) Schreiben Sie analog zu Aufgabe 30 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 30: `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 $\alpha = (\alpha_1, \dots, \alpha_m)$ des Runge-Kutta-Verfahrens,
iii) `bt.beta` für die Matrix $\beta = (\beta_{i,j})$ $(i, j = 1, \dots, m)$ des Runge-Kutta-Verfahrens und
iv) `bt.gamma` für den Zeilenvektor $\gamma = (\gamma_1, \dots, \gamma_m)$ 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 33 (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 30 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 30. Zeichnen Sie diese Näherungslösungen und die exakte Lösung in ein neues Schaubild ein, und ergänzen Sie Ihr Schaubild mit den Diskretisierungsfehlern um die Diskretisierungsfehler 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) zurück geben.

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 Diskretisierungsfehlern um die Diskretisierungsfehler der neuen Lösungen.

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

Aufgabe 32 *(Programmieraufgabe: Konvergenzordnung verschiedener ESV)* (4M+4M+3M+4T Punkte)

In dieser Aufgabe untersuchen wir jeweils den maximalen Wert $\max_k |e_k| = \max_k |y(t_k) - y_k|$ des globalen Diskretisierungsfehler der Verfahren aus Aufgabe 30 vom letzten Übungsblatt und der Verfahren aus Aufgabe 31 für verschiedene Schrittweiten.

a) Schreiben Sie ein Matlabskript `konvergenzESV`, das für das Anfangswertproblem aus Aufgabe 30 und für die Schrittweiten $0.5^4, \dots, 0.5^{14}$ jeweils den Maximalwert des globalen Diskretisierungsfehlers des expliziten Euler-Verfahrens `eulerExplizit` berechnet und doppelt logarithmisch über den Schrittweiten plottet.

b) Ergänzen Sie Ihr Matlabskript `konvergenzESV` und Ihr Schaubild um die Maximalwerte der Diskretisierungsfehler der anderen Verfahren aus den Aufgaben 30 und 31 (`eulerImplizit` sowie die drei Runge-Kutta-Verfahren mit den Butcher-Tableaus `bt = rKverbEuler()`, `bt = rKHeun3()` und `bt = rK4()`).

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

d) Interpretieren Sie das Schaubild. Was geben die Steigungen der geplotteten Diskretisierungsfehler an?

Aufgabe 33 *(Programmieraufgabe: Zeeman'sches Herzschlagmodell)* (5M*+4M*+5T* Punkte)

Zeemanns Herzschlagmodell beschreibt die Funktionsweise des Herzens. Gesuchte Größen sind die Länge der Herzmuskelfaser $l(t)$ und das elektrochemische Potential $p(t)$. Das Modell wird beschrieben durch das System

$$\frac{d}{dt}\begin{pmatrix} l(t) \\ p(t) \end{pmatrix} = \begin{pmatrix} -(l(t)^3 - \alpha l(t) + p(t)) \\ \beta l(t) \end{pmatrix},$$

wobei α die Vorspannung der Muskelfaser und β der Rückkopplungsparameter sind.

a) Berechnen Sie in einem Matlabskript `zeemann` mit Ihrer Matlabfunktion `eulerExplizit` eine Näherungslösung im Zeitintervall $[0, 100]$ mit Schrittweite $h = 0.05$. Verwenden Sie $\alpha = 3$ und $\beta = 0.1$ und die Anfangswerte $l(0) = 1$ und $p(0) = 0$. Stellen Sie Ihre Näherungslösung grafisch dar.

b) Berechnen Sie auch mit den anderen Einschrittverfahren aus den Aufgaben 30 und 31 Näherungslösungen.

c) Decken sich die berechneten Ergebnisse in der graphischen Darstellung? Wie passt Ihre Beobachtung in dieser Aufgabe zu Ihren Ergebnissen aus Aufgabe 31? Welches Verfahren liefert (bzw. welche Verfahren liefern) vermutlich die genaueren Ergebnisse? Begründen Sie Ihre Vermutung.

Aufgabe 34 *(lokaler Diskretisierungsfehler, Konsistenzordnung)* (5T+5T+5T* Punkte)

Wir betrachten die folgenden Einschrittverfahren zur Lösung der Anfangswertaufgabe $y' = f(t,y),\ y(t_0) = y_0$ auf einem äquidistanten Gitter mit der Schrittweite h:

a) Das explizite Euler-Verfahren mit Schrittweite h: $v_{j+1} = v_j + hf(t_j, v_j)$.

b) Ein explizites Verfahren bestehend aus zwei Euler-Schritten je mit halber Schrittweite:

$$w_{j+\frac{1}{2}} = w_j + \frac{h}{2} f(t_j, w_j), \quad w_{j+1} = w_{j+\frac{1}{2}} + \frac{h}{2} f(t_j + \frac{h}{2}, w_{j+\frac{1}{2}}).$$

c) Ein extrapoliertes Verfahren: $u_{j+1} = \alpha\, v_{j+1} + (1-\alpha)\, w_{j+1}$.

Bestimmen Sie jeweils den führenden Term des lokalen Diskretisierungsfehlers. Kann α so gewählt werden, dass der lokale Diskretisierungsfehler des extrapolierten Verfahrens von der Ordnung 3 ist?

Hinweise:
Die Programmieraufgaben sind in Matlab zu erstellen. Der Source Code muss strukturiert und dokumentiert sein. Senden Sie spätestens am Tag vor Ihrem Tutorium **bis 18:00 Uhr** alle Matlab-Files und alle Ergebnisse in einer E-mail mit dem Betreff **Loesung-Blatt10** an **angewandte.numerik@uni-ulm.de**