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

WiSe 2017/2018
Blatt 09
16.12.2017

Angewandte Numerik 2

Besprechung in den Tutorien in der Woche vom 18.12.2017 bis 22.12.2017

Für dieses Übungsblatt gibt es 14 Theorie- und 17 Matlab-Punkte, sowie 5 Theorie- und 5 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 60-Prozent-Grenzen liegen aktuell (inklusive Blatt 09) bei 93,6 Theorie- und 104,4 Matlabpunkten.

Aufgabe 33 *(Stabilität)* (4T+4T+6T+3T*+2T* Punkte)

a) Berechnen und skizzieren Sie das Stabilitätsgebiet des expliziten Euler-Verfahrens.

b) Berechnen und skizzieren Sie das Stabilitätsgebiet des impliziten Euler-Verfahrens.

c) Berechnen und skizzieren Sie das Stabilitätsgebiet des aus der Trapezregel resultierenden impliziten Verfahrens $y_{k+1} = y_k + h\,\frac{1}{2}(f(t_k, y_k) + f(t_{k+1}, y_{k+1}))$ (vergleiche Beispiel 3.2.3 (c) des Skripts).

d) Welches dieser Verfahren ist A-stabil, d. h., dass das Stabilitätsgebiet die ganze negative Halbebene der komplexen Zahlenebene umfasst (siehe Definition 3.6.13 im Skript). Welches dieer Verfahren ist nicht A-stabil?

e) Was bedeutet das jeweils für Ihre Wahl der Schrittweite h?

Hinweise:

i) Wenden Sie zunächst die Verfahrensvorschrift des jeweiligen Verfahrens auf das skalare Modellproblem $y' = \lambda y,\ \ y(0) = 1$ an.

ii) Bringen Sie die so erhaltene Gleichung in die Form $y_h(t_{j+1}) = R(\lambda h) y_h(t_j)$, wobei $R : \mathbb{C} \to \mathbb{C} \cup \infty$, $z \mapsto R(z) = \frac{P(z)}{Q(z)}$ eine rationale Funktion von Polynomen P und Q ist.

iii) Berechnen Sie das Stabilitätsgebiet.

Aufgabe 34 *(*Programmieraufgabe: Implizite Runge-Kutta-Verfahren) (6M+2M+6M+3M+5M* Punkte)

a) Schreiben Sie analog zu Aufgabe 28 von Blatt 07 eine Matlabfunktion `yk = rungeKuttaImplizit(f, y0, tk, bt, anzIt)`, die für einen gegebenen Startwert $y^0 \in \mathbb{R}^n$ 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 impliziten Runge-Kutta-Verfahren berechnet.

Sie dürfen dazu Ihre Funktion `yk = rungeKutta(f, y0, tk, bt)` aus Aufgabe 28 erweitern. Die Parameter sind wie in Aufgabe 28: `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.

Verwenden Sie zur Lösung des impliziten Gleichungssystems, mit dem die Funktionen $k_1(t,h,y),\ldots,k_m(t,h,y)$ berechnet werden, eine Fixpunktiteration mit einer festen Anzahl Schritten und Startwerten $k_j(t,h,y)=0$ $(j=1,\ldots,m)$. Der zusätzliche Parameter `anzIt` gibt dann die Anzahl der durchzuführenden Iterationen an.

Ihre Matlabfunktion `yk = rungeKuttaImplizit(f, y0, tk, bt, anzIt)` soll (völlig analog zu Aufgabe 28) den Algorithmus eines impliziten 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 = rungeKuttaImplizit(f, y0, tk, bt, anzIt)` 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 kann jetzt, anders als beim expliziten Runge-Kutta-Verfahren aus Aufgabe 28, auf und oberhalb der Diagonalen auch Werte $\beta_{i,j}\neq 0$ enthalten.

b) Schreiben Sie eine Matlabfunktion `bt = rK3Implizit`, die das 3-stufige Butcher-Tableau

$$\begin{array}{c|ccc} 0 & \frac{1}{6} & -\frac{1}{6} & 0 \\ \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 \\ 1 & \frac{1}{6} & \frac{5}{6} & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

eines impliziten Runge-Kutta-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) Testen Sie Ihre Matlabfunktionen mit der Anfangswertaufgabe

$$y'(t)=-2ty(t)^2,\quad y(0)=1,$$

deren exakte Lösung durch $y(t)=\frac{1}{t^2+1}$ gegeben ist.

Wählen Sie `anzIt` $=1,2,3,4,5$ und berechnen Sie jeweils mit den Schrittweiten $h:=\frac{1}{N}$, $N=10,20,40,80,160,320,640$ die Fehler $e_N=|y_N-y(1)|$ an der Stelle $t=1$. Plotten Sie jeweils den Fehler e_N über N in einem doppelt logarithmischen Plot.

Was beobachten Sie?

d) Berechnen Sie auch die Konvergenzordnungen

$$p_N=\frac{\ln\left(\frac{e_{N/2}}{e_N}\right)}{\ln(2)}\quad (N=20,40,80,160,320,640),$$

die sich aus Ihren numerischen Werten ergeben.

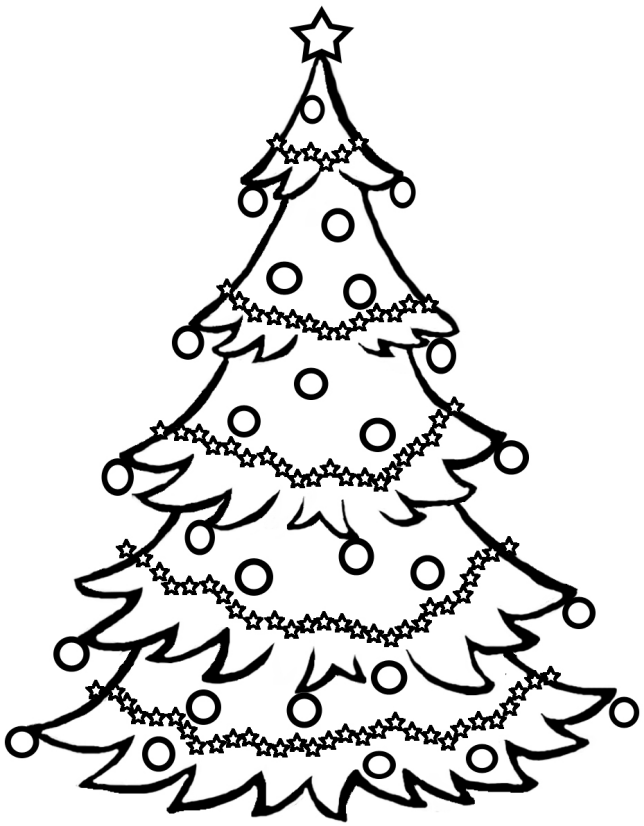
e) Testen Sie Ihre Routinen auch für das Anfangswertproblem

$$\begin{array}{rcrcrcll} y_1'(t) &=& & & y_2(t) &-& y_3(t), & y_1(0)=1, \\ y_2'(t) &=& -2y_1(t) &+& 3y_2(t) &-& y_3(t), & y_2(0)=-1, \\ y_3'(t) &=& -y_1(t) &+& y_2(t) &+& y_3(t), & y_3(0)=2, \end{array}$$

dessen exakte Lösung durch

$$y(t)=e^t\begin{pmatrix}1-4t\\ 1-4t-2e^t\\ 4-2e^t\end{pmatrix}$$

gegeben ist.



Frohe Weihnachten und einen guten Rutsch!

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-Blatt09** an **angewandte.numerik@uni-ulm.de**.