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

WiSe 2014/2015
Blatt 7
28.11.2014

Angewandte Numerik 2

Abgabetermin: Freitag, 05.12.2014, **vor der Übung**

Für dieses Übungsblatt gibt es 5 Theorie- und 28 Matlab-Punkte, sowie 3 Theorie- und 6 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 50-Prozent-Grenzen liegen aktuell (inklusive Blatt 7) bei 58,5 Theoriepunkten und 71 Matlabpunkten.

Aufgabe 27 *(Programmieraufgabe: Verschiedene Einschrittverfahren)*
(3M+2M+2M+3M+2M+3M+2M+3M*+2M*+5T Punkte)

a) Schreiben Sie eine Matlabfunktion `yk = eulerExplizit(f, y0, tk)`, 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 dem expliziten Euler-Verfahren

$$y^{k+1} = y^k + h_k\, f(t_k, y^k) \quad \text{mit} \quad h_k = (t_{k+1} - t_k)$$

berechnet. Der Parameter `f` ist dabei 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.

b) Schreiben Sie ein Matlabskript `vergleichESV`, das mit Ihrer Matlabfunktion `eulerExplizit` für das Anfangswertproblem

$$y'(t) = y(t) + e^t, \quad y(0) = 1$$

auf dem Intervall $[0, 1]$ eine Näherungslösung nach dem expliziten Euler-Verfahren berechnet. Wählen Sie als Schrittweite zunächst $h = \frac{1}{20}$ und lösen Sie ebenfalls für $h = \frac{1}{40}$. Vergleichen Sie die Ergebnisse mit der exakten Lösung $y(t) = (t + 1)e^t$. Zeichnen Sie dazu Ihre beiden Näherungslösungen (für die zwei verschiedenen Schrittweiten) und die exakte Lösung in ein Schaubild.

c) Zeichnen Sie in ein weiteres Schaubild die beiden Gitterfehler (für die beiden Schrittweiten) ein.

d) Schreiben Sie eine Matlabfunktion `yk = eulerImplizit(f, y0, tk, tol)`, die analog zu `eulerExplizit` die Anfangswertaufgabe mit dem impliziten Euler-Verfahren löst. Berechnen Sie die Näherung für y^{k+1} mit einer Fixpunktiteration. Die Fixpunktiteration soll abbrechen, wenn die Genauigkeit `tol` erreicht ist.

e) Bestimmen Sie mit `eulerImplizit` zwei weitere Näherungslösungen der Anfangswertaufgabe mit den Schrittweiten aus Aufgabenteil b). Zeichnen Sie diese Näherungslösungen und die exakte Lösung in ein neues Schaubild ein. Ergänzen Sie Ihr Schaubild mit den Gitterfehlern um die Gitterfehler der beiden neuen Lösungen.

f) Schreiben Sie eine Matlabfunktion `yk = trapezMethode(f, y0, tk, tol)`, die analog zu `eulerImplizit` die Anfangswertaufgabe mit dem Verfahren nach Beispiel 3.2.3 (c) aus dem Skript löst. Berechnen Sie die Näherung für y^{k+1} analog zu `eulerImplizit` mit einer Fixpunktiteration.

g) Bestimmen Sie mit `trapezMethode` wiederum zwei weitere Näherungslösungen der Anfangswertaufgabe und den Schrittweiten aus Aufgabenteil b), die Sie zusammen mit der exakten Lösung in ein neues Schaubild eintragen. Ergänzen Sie Ihr Schaubild mit den Gitterfehlern um die Gitterfehler der beiden neuen Lösungen.

h) Als letztes Verfahren sollen Sie in einer Matlabfunktion `yk = heun(f, y0, tk)` das Verfahren von Heun implementieren. Dieses Verfahren berechnet die Näherungslösungen ähnlich der Trapezmethode, ist aber ein explizites Verfahren und benötigt daher keine Fixpunktiteration. In einem ersten Schritt wird der Wert y^{k+1} mit einem Schritt des expliziten Euler-Verfahrens geschätzt. In einem zweiten Schritt wird diese Schätzung in die Trapezregel eingesetzt. Die Verfahrensfunktion lautet also $F(f, t_k, h_k, y^k) = \frac{1}{2}\left(f(t_k, y^k) + f(t_k + h_k,\, y_k + h_k f(t_k, y^k))\right)$.

i) Bestimmen Sie auch mit `heun` zwei Näherungslösungen der Anfangswertaufgabe und den Schrittweiten aus Aufgabenteil b), die Sie zusammen mit der exakten Lösung in ein neues Schaubild eintragen, und ergänzen Sie Ihr Schaubild mit den Gitterfehlern um die Gitterfehler der beiden neuen Lösungen.

j) Erläutern Sie die Schaubilder. Vergleichen Sie alle vier Verfahren.

Aufgabe 28 *(Programmieraufgabe: Konvergenzordnung verschiedener ESV)* (4M+(2M+1M*)+3T* Pkte)

In dieser Aufgabe wollen wir den Diskretisierungsfehler der Verfahren aus der letzten Aufgabe für verschiedene Schrittweiten untersuchen.

a) Schreiben Sie ein Matlabskript `konvergenzESV`, das für das Anfangswertproblem aus der vorigen Aufgabe und für die Schrittweiten $0.5^4, \ldots, 0.5^{14}$ den Diskretisierungsfehler des expliziten Euler-Verfahrens `eulerExplizit` berechnet und doppelt logarithmisch über den Schrittweiten plottet (Matlabbefehl `loglog`).

b) Ergänzen Sie Ihr Matlabskript `konvergenzESV` und Ihr Schaubild um die Diskretisierungsfehler der anderen Verfahren aus der vorigen Aufgabe (`eulerImplizit`, `trapezMethode` und `heun`).

c) Interpretieren Sie das Schaubild.

Aufgabe 29 *(Programmieraufgabe: Zeeman'sches Herzschlagmodell)* (5M 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.
Berechnen Sie in einem Matlabskript `zeemann` mit Ihrer Matlabfunktion `eulerExplizit` eine Näherungslösung im Zeitintervall $[0, 100]$, verwenden Sie die Parameterwerte $\alpha = 3$ und $\beta = 0.1$ und die Anfangswerte $l(0) = 1$ und $p(0) = 0$. Stellen Sie Ihre Näherungslösung grafisch dar.

Hinweise:
Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Files in einer E-mail mit dem Betreff **Loesung-Blatt07** an **angewandte.numerik@uni-ulm.de** (Abgabetermin jeweils wie beim Theorieteil). Drucken Sie zusätzlich allen Programmcode sowie die Ergebnisse aus und geben Sie diese vor der Übung ab. Der Source Code sollte strukturiert und, wenn nötig, dokumentiert sein.