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

WiSe 2015/2016
Blatt 11
15.01.2016

Angewandte Numerik 2

Abgabetermin: Freitag, 22.01.2016, **vor der Übung**

Für dieses Übungsblatt gibt es 12 Theorie- und 14 Matlab-Punkte, sowie 15 Theorie- und 7 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 50-Prozent-Grenzen liegen aktuell (inklusive Blatt 11) bei 85,5 Theoriepunkten und 101 Matlabpunkten.

Aufgabe 39 *(Mehrschrittverfahren (Milne-Simpson), Konsistenzordnung, Konvergenz)* (7T+3T Punkte)

Bestimmen Sie die Konsistenzordnung des Milne-Simpson-Verfahrens mit Schrittzahl $k = 2$:

$$y_{j+1} = y_{j-1} + \frac{h}{3}(f_{j+1} + 4f_j + f_{j-1})$$

und prüfen Sie, ob es konvergent ist.

Hinweise:

i) Zeigen Sie zunächst die Konsistenz und die Konsistenzordnung ($p = 4$) des linearen Mehrschrittverfahrens.

ii) Dazu können Sie Lemma 3.7.11 und Satz 3.7.12 aus dem Skript verwenden. Gehen Sie davon aus, dass die Startwerte („Anlaufstück“) geeignet gewählt wurden.

iii) Zeigen Sie anschließend die Konvergenz.

Aufgabe 40 *(Konstruktion eines expliziten Zweischrittverfahrens)* (4T*+2T*+2T*+1T* Punkte)

Konstruieren Sie ein explizites Zweischrittverfahren der Form

$$y_{j+2} + \alpha_1 y_{j+1} + \alpha_0 y_j = h(\beta_0 f(t_j, y_j) + \beta_1 f(t_{j+1}, y_{j+1})).$$

a) Bestimmen Sie dazu zunächst α_0, β_0 und β_1 in Abhängikeit von α_1 so, dass das Verfahren mindestens die Ordnung 2 hat.

b) Für welche Werte von α_1 ist dieses Verfahren dann konvergent?

c) Lässt sich α_1 so wählen, dass sich ein Verfahren der Konsistenzordnung 3 ergibt?

d) Ist dieses Verfahren konvergent?

Aufgabe 41 *(Programmieraufgabe: Konvergenz von expliziten Mehrschrittverfahren)*
(9M+5M+2T+3M*+2T*+4M*+4T* Punkte)

In dieser Aufgabe sollen die Ergebnisse der vorigen Aufgabe 40 numerisch überprüft werden. Dazu betrachten wir zunächst das konstruierte Zweischrittverfahren mit der Konsistenzordnung 3:

$$y_{j+1} + 4y_j - 5y_{j-1} = h(4f_j + 2f_{j-1}). \tag{1}$$

a) Schreiben Sie eine Matlabfunktion `[yk, tk] = explLinMehrSchritt(f, y0, t0, tEnd, N, a, b, anlauf)`, die für einen gegebenen Startwert $y^0 \in \mathbb{R}^n$ die 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 expliziten linearen Mehrschritt-Verfahren näherungsweise berechnet.

Die Parameter sind weitgehend analog zu den bisher programmierten Einschritt-Verfahren: `f` ist die Funktion f als *function handle*, `y0` ist der Startwert $y^0 \in \mathbb{R}^n$, `t0` ist der Anfangszeitpunkt, `tEnd` ist der Zeitpunkt, bis zu dem eine Näherungslösung berechnet werden soll, `N` ist die Anzahl der Iterationen, die zwischen `t0` und `tEnd` durchgeführt werden sollen, `a` ist der Spaltenvektor $a = (a_0, \dots, a_k)$ der Koeffizienten des Mehrschritt- (k-Schritt-)Verfahrens, `b` ist der Spaltenvektor $b = (b_0, \dots, b_{k-1})$ der Koeffizienten der (expliziten) linearen Verfahrensfunktion F und `anlauf` ist eine Funktion (als *function handle*) zur Berechnung der Näherungswerte $y^1, \dots, y^{k-1}$ im Anlaufstück. Der Rückgabewert `yk` ist der Vektor der einzelnen Näherungswerte y^k und `tk` ist das Gitter mit den diskreten Zeitpunkten t_k.

Die Funktion `anlauf` soll die aus den Einschritt-Verfahren bekannten Parameter und Rückgabewerte haben, also beispielsweise durch `yk = anlauf(f, y0, tk)` aufgerufen werden können. `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 aus dem Anlaufstück. Der Rückgabewert `yk` ist der Vektor der einzelnen Näherungswerte y^k des Anlaufstücks.

b) Schreiben Sie ein Matlabskript `mainA41`, das die Anfangswertaufgabe

$$y' = y, \quad y(0) = 1 \tag{2}$$

mit obigem Zweischrittverfahren (1) löst. Verwenden Sie für die Werte im Anlaufstück die exakten Anfangsdaten $y_0 = 1$ und $y_1 = e^h$. Plotten Sie Ihre Näherungslösung für die Schrittweite $h = 1/10$ und die exakte Lösung auf dem Intervall $[t_0, t_{End}] = [0, 1]$ in ein Schaubild.

c) Was stellen Sie fest?

d) Plotten Sie nun in drei weitere Schaubilder die numerischen Lösungen für die Schrittweiten $h = 1/20$, $h = 1/40$ und $h = 1/80$ sowie jeweils die exakte Lösung auf dem Intervall $[t_0, t_{End}] = [0, 1]$. Begrenzen Sie den Bereich der dargestellten Werte mit `xlim ([0,1])` und `ylim ([0,5])`.

e) Interpretieren Sie das erhaltene Ergebnis.

f) Lösen Sie nun die Anfangswertaufgabe (2) auch mit einem konvergenten Zweischrittverfahren der Konsistenzordnung 2. Wählen Sie dazu die Koeffizientenvektoren a und b entsprechend Aufgabe 40, Teil b). Berechnen Sie auch die numerischen Konvergenzordnungen entsprechend der Formel aus Aufgabe 38, Teil d) (mit den Schrittweiten 1/20, 1/40, 1/80 bis 1/1280).

g) Variieren Sie die Koeffizienten des Mehrschrittverfahrens, und zwar sowohl innerhalb als auch außerhalb des in Aufgabe 40, Teil b) berechneten Konvergenzbereichs. Was können Sie nun beobachten?

Hinweise:
Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Matlab-Files in einer E-mail mit dem Betreff **Loesung-Blatt11** 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.